SAP Lens

AWS Well-Architected Framework

SAP Lens: AWS Well-Architected Framework

Table of Contents

SAP Lens - AWS Well-Architected Framework

Publication date: **October 4, 2022** (*Document revisions* (p. 123))

This paper describes the SAP Lens for the AWS Well-Architected Framework. It is a collection of customer-proven design principles and best practices for ensuring SAP workloads on AWS are well-architected. Use the SAP Lens as a supplement to the AWS Well-Architected Framework, which provides the foundations for building secure, high-performing, resilient, and efficient applications and workloads on AWS.

The SAP Lens is based on insights that AWS has gathered from customers, AWS Partners, and our SAP specialist community. The lens has been designed to help you adopt a cloud native approach to running SAP. It highlights some of the most common areas for improvement, aligned to the six pillars of the AWS Well-Architected Framework — operational excellence, security, reliability, performance efficiency, cost optimization, and sustainability.

In this content, we refer to SAP as the software running on AWS provided by the company SAP, best known for its enterprise resource planning (ERP) applications. The guidance is intended to cover all of the SAP software that can run on AWS, including SAP Business Suite, SAP S/4HANA, and supporting products. Where a recommendation is specific to an SAP application or database, this has been highlighted (for example, SAP HANA databases).

The intended readers of this document are SAP technology architects, cloud architects, and team members who build, operate, and maintain SAP systems on AWS.

How to use the lens

Use this lens to evaluate SAP on AWS workloads — before, during, and after implementation. This lens provides additional content to the AWS Well-Architected Framework and clarifies how to interpret and adopt those foundational best practices into SAP workload designs.

We recommend using this Lens and the Framework in tandem, working closely with your enterprise teams to address your SAP and enterprise requirements. To avoid duplication, we have provided links to the AWS Well-Architected Framework where the guidance is more comprehensive or has no specific SAP context.

To use this lens, follow these steps:

1. Familiarize yourself with this document and the broader AWS Well-Architected Framework and pillar whitepapers.
2. Gather your SAP-specific design documentation, operational procedures, and monitoring history (where available).
3. Compare your SAP workload implementation and operations to the best practices described in this document.
4. For each best practice, record whether it has been followed and prioritize evaluating those that are required.
5. Use the provided suggestions as solutions to address the areas where your workload is not well-architected.

If you require additional expert guidance, contact your AWS account team to engage an SAP specialist solution architect.

After reviewing your workload, you will have a list of best practices that shows where the workload is well-architected, and where it needs improvement:

- For the well-architected architectural components: Share your knowledge among your teams to amplify them across your organization.
- For the best practices that your workload does not follow yet: Treat them as technical debt and risks to your business. Follow your internal risk management process to continuously monitor and improve these risks.
- For areas that require further in-depth analysis or assistance with remediation: Contact AWS Professional Services or consult with AWS Partners on the AWS SAP Certified Partner List.

For more details, see the following links and information:

- AWS Documentation: The Review Process - AWS Well-Architected Framework

Run an SAP Lens review in your AWS account

A common request from our customers has been to enable them to run a *self service* SAP Lens review in the AWS Well-Architected Tool (AWS WA Tool).

The SAP Lens is now available as a custom lens for the AWS Well-Architected Tool in the AWS Management Console. Custom lenses, such as the SAP Lens, are defined in a JSON file and allow you to tailor your workload reviews to particular technologies, help you meet governance needs, and extend the guidance already provided by the Well-Architected Framework and the AWS lenses.

To add the SAP Lens to the AWS Well-Architected Tool:

1. Download the SAP Lens JSON file. This file is used in Step 5.
2. Sign in to the AWS Management Console and open the AWS Well-Architected Tool console at https://console.aws.amazon.com/wellarchitected/.
3. In the left navigation pane, choose **Custom lenses**.
4. Choose **Create custom lens**.
5. Choose **Choose file** and select the JSON file you downloaded in Step 1.
6. (Optional) In the **Tags** section, add any tags you want to associate with the SAP Lens.
7. Choose **Submit & Preview** to preview the SAP Lens, or **Submit** to create the lens without previewing.

   If you choose to **Submit & Preview**, you can select **Next** to navigate through the SAP Lens preview, or select **Exit Preview** to go back to **Custom lenses**.
8. Select the SAP Lens and choose **Publish lens**.
9. In the **Version name** box, enter a unique identifier for the version change. This value can be up to 32 characters and must only contain alphanumeric characters and periods (".").
10. Choose **Publish custom lens**.

    After the SAP Lens has been published, it's in **PUBLISHED** status.

The SAP Lens can now be applied to workloads in your AWS account, and shared with other AWS accounts and IAM users. If your account is managed by AWS Organizations, you can share the lens with all accounts in the organization or in an OU without having to enumerate each account.

As you work through the SAP Lens checklist, risks can be identified and comments can be captured. A workload report is available in PDF format for sharing with stakeholders to document risks and future recommendations. Open risks can be managed and assigned in the tool and periodic milestone reviews can be performed.

For more information on using the AWS WA Tool, custom lenses, reports, and the risk dashboard, see the AWS Well-Architected Tool User Guide.

Providing feedback

We want to hear about your experience with the SAP Lens for the AWS Well-Architected Framework. Our goal is to make the SAP Lens the best guiding mechanism for implementing and running a Well-Architected SAP workload on AWS.

Upon completion of an SAP Lens Well-Architected review, consider filling out this brief, five minute survey to assist us in improving this service.

- SAP Lens Customer Feedback Survey

Join the SAP on AWS discussion

AWS has launched re:Post – A Reimagined Q&A Experience for the AWS Community, this is in addition to your regular customer account team and AWS Support channels. The SAP on AWS team regularly monitors the re:Post SAP on AWS topic for discussion and questions that could be answered to assist our customers and partners running SAP on AWS. If you have a question about the SAP Lens for Well-Architected Framework, consider joining the discussion over at re:Post and adding to the community knowledge base.

Definitions

Term	Description	Examples (if applicable)
SAP workload	A workload is a collection of SAP resources that delivers business value. In the SAP context, this includes customer-facing components of SAP applications as well as SAP backend processes. A workload might consist of a subset of resources in a single AWS account or be a collection of resources spanning multiple AWS accounts.	
SAP product	A product of SAP, the enterprise software company that provides solutions for business processes across all industries. Sometimes referred to as *SAP solutions*.	SAP S/4HANA On-premise edition, Concur, Qualtrics
SAP system	A logical grouping of architecture (a set of things that work together as part of a greater mechanism) that is typically characterized by an SAP System Identifier (SID).	Production ERP system
SAP System Identifier (SAP SID / DB SID)	A combination of letters and numbers used to uniquely identify an SAP system.	PRD, HDB, PR1
SAP environment	Integrated grouping of one or more SAP products or technology components. Are linked to form a *path to production.*	Sandbox, Development, QA, Training, Test, Pre-Production, and Production.
SAP instance or host	An instance is a copy of an Amazon Machine Image (AMI) running as a virtual server in the AWS Cloud. In the SAP context, usually a compute instance in Amazon EC2 service.	
SAP technical component	Administrative units that group together components of an SAP system running on an instance or host. These are technical architecture building blocks of SAP applications.	Application Server (PAS or AAS), SAP HANA Database, Web Dispatcher
Services (AWS services)	Over 200 cloud services that are used in combinations tailored to business or organizational needs. For information about many AWS services, see the Overview of Amazon Web Services whitepaper.	Amazon EC2, Amazon S3, Amazon EFS
SAP deployment / Deployment pattern	Used to described how SAP is deployed in reference to the options in SAP Provisioning Tools (SUM, SWPM).	Highly Available (HA),

Term	Description	Examples (if applicable)
		Distributed, Standalone
SAPS rating (referred to as *SAPS* in multiple locations)	SAP Application Performance Standard (SAPS) – is a hardware-independent unit of measurement that describes the performance of a system configuration in the SAP environment. It is derived from the Sales and Distribution (SD) benchmark, where 100 SAPS is defined as 2,000 fully business processed order line items per hour. For more information, see SAP Standard Application Benchmarks .	*The Amazon EC2 instance type `c5.large` provides 3,650 SAPS.*

AWS Documentation: AWS Glossary

Workload context checklist

To better understand your business's context, you need to gather the following information.

ID	Priority	Workload Context
☐ C1	Required	Name of the workload
☐ C2	Required	Description that contains the business purposes, key performance indicators (KPIs), and the intended users of the workload.
☐ C3	Required	Review owner who leads the lens review
☐ C4	Required	Workload owner who is responsible for maintaining the workload
☐ C5	Required	Business stakeholders who sponsor the workload
☐ C6	Required	Business partners who have a stake in the workload, such as information security, finance, and legal
☐ C7	Recommended	Architecture design document that describes the workload
☐ C8	Recommended	AWS account IDs associated with the workload
☐ C9	Recommended	Regulatory compliance requirements relevant to the workload (if any)

Well-Architected design principles

This section describes the design principles, best practices, and improvement suggestions that are relevant when designing and operating your SAP workload.

We recommend that you also read and apply the guidance found in each Well-Architected pillar, which includes foundational best practices for operational excellence, security, reliability, performance efficiency, and cost optimization that are relevant to all workloads.

Pillars

For a complete list of design principles, refer to Design principles arranged by pillar (p. 124).

Operational excellence

The operational excellence pillar focuses on the ability to develop and run workloads effectively, gain insight into their operations, and to continually improve supporting processes to deliver business value.

This section provides a set of design principles and recommendations specifically tailored to provide guidance for SAP workloads. The Operational Excellence Pillar whitepaper contains broader design principles and recommendations which we highly recommend you read in conjunction with the SAP guidance that follows. The design principles and best practices included here are a subset of those in the Operational Excellence Pillar, and have been enhanced with SAP-specific implementation guidance and suggestions.

1 - Design SAP workload to allow understanding and reaction to its state

How do you design your SAP workload so that you can understand its state? Design your SAP workload so that it provides the information necessary across all components for you to understand its internal and external state. Consider infrastructure, SAP technology/basis, front end, and network components. Design monitoring and logging approaches which capture metrics to allow real-time monitoring and also historical logging to allow remediation and post-event analysis.

ID	Priority	Best Practice
☐ BP 1.1	Required	Implement prerequisites for monitoring SAP on AWS
☐ BP 1.2	Required	Implement infrastructure monitoring for SAP
☐ BP 1.3	Required	Implement application and database monitoring for SAP

ID	Priority	Best Practice
☐ BP 1.4	Highly Recommended	Implement workload configuration monitoring
☐ BP 1.5	Highly Recommended	Implement user activity monitoring
☐ BP 1.6	Highly Recommended	Implement dependency monitoring
☐ BP 1.7	Recommended	Implement single pane of glass health monitoring across your SAP workloads
☐ BP 1.8	Recommended	Use automated response and recovery techniques to react to monitoring alerts

For more details, see the following links and information:

- AWS Documentation: AWS Data Provider for SAP
- AWS Service: Amazon CloudWatch
- SAP on AWS Blog: Set up observability for SAP HANA databases with Amazon CloudWatch Application Insights
- SAP on AWS Blog: Serverless Monitoring for SAP
- AWS Marketplace: Products and Tools for SAP Monitoring
- SAP Note: 1656250 - SAP on AWS: Support Prerequisites [Requires SAP Portal Access]
- SAP Documentation: SAP Solution Manager 7.2 - Application Operations

Best Practice 1.1 – Implement prerequisites for monitoring SAP on AWS

SAP certification requirements for SAP on AWS are outlined in SAP Note 1656250. This note includes instructions for setting up the AWS Data Provider for SAP, enabling Amazon CloudWatch detailed monitoring, and using SAP enhanced monitoring for SAP NetWeaver solutions. Enabling these prerequisites helps ensure that your SAP workload state is able to be fully understood and investigated by AWS and SAP. These prerequisites should feed into your overall SAP monitoring strategy.

Suggestion 1.1.1 - Check SAP support prerequisites

Check SAP Note 1656250 on the SAP support portal for the most up-to-date support requirements for SAP on AWS workloads. Follow the detailed instructions in this note.

- SAP Note: 1656250 - SAP on AWS: Support Prerequisites [Requires SAP Portal Access]

Suggestion 1.1.2 - Install AWS Data Provider for SAP NetWeaver workloads

The AWS Data Provider for SAP is a required installation on each of your EC2 instances supporting SAP NetWeaver workloads. The AWS Data Provider for SAP is an agent which collects performance-related metrics from AWS services and provides them to the SAP internal application monitoring system. SAP tools, such as transaction code ST06n and Solution Manager monitoring that use external metrics usually collected from the SAPOSCOL service, require the AWS Data Provider for SAP to access AWS metrics.

There are indirect costs associated with running the AWS Data Provider for SAP because of the detailed monitoring and increased API calls required for SAP to receive monitoring data at specific intervals. See AWS Data Provider for SAP - Introduction - Pricing for details.

- AWS Documentation: AWS Data Provider for SAP

Suggestion 1.1.3 - Create a monitoring strategy for your SAP workloads

Decide how you will observe the current and historical health of your SAP application from both an inside-out and outside-in perspective. Consider all components which work together to provide the end-user experience. Consider how you will capture metrics from underlying AWS compute, storage, and network services in addition to internal SAP application metrics and external user performance and reliability monitoring. Evaluate different tools for each component and decide how you can bring these together in a single place (for example, log aggregation) to perform root cause analysis when needed. Determine how you will use this information to design alert thresholds and remediation actions to be taken when thresholds are breached.

Understand the capabilities of SAP Solution Manager monitoring, third-party monitoring tools, and CloudWatch dashboards that can ingest custom SAP monitoring metrics as a starting point for your design.

- AWS Documentation: SAP NetWeaver on AWS: Monitoring Guide
- SAP on AWS Blog: Serverless Monitoring for SAP NetWeaver
- SAP on AWS Blog: Serverless Monitoring for SAP HANA
- SAP on AWS Blog: Set up observability for SAP HANA databases with Amazon CloudWatch Application Insights
- AWS Service Video: Gaining Better Observability of Your VMs with Amazon CloudWatch
- AWS Marketplace: Products and Tools for SAP Monitoring
- SAP Documentation: SAP Solution Manager 7.2 - Application Operations
- SAP Documentation: SAP NetWeaver Alert Monitor

Best Practice 1.2 – Implement infrastructure monitoring for SAP

Set up your infrastructure monitoring to provide information about supporting services that are used to keep your SAP application running and supporting your users. Some examples include CPU and memory utilization, storage and filesystem usage and performance (IOPS and throughput), and network throughput. Include any dependent foundational services used by SAP, such as on-premises Active Directory services, DNS, and third-party tools, such as high availability (HA) and backup software. Evaluate AWS tools and SAP-specific tools from the AWS Marketplace that can help correlate and visualize this information, such as DataDog, Splunk, DynaTrace, and Avantra. Use this information to identify trends and determine when a corrective action is required.

Suggestion 1.2.1 - Implement CloudWatch metrics and alarms for services supporting SAP

Implement Amazon CloudWatch detailed monitoring metrics and thresholds with alarms for all of your SAP systems. These metrics and alarms should include monitoring for common problems which can affect SAP system availability and performance. Common infrastructure monitoring areas focus on Amazon Elastic Compute Cloud (EC2) instances, Amazon Elastic Block Storage (Amazon EBS) volumes, and Elastic Load Balancing (ELB).

Common monitoring items include the following:

- Amazon EC2 high CPU utilization
- Amazon EC2 high memory utilization
- Amazon EBS storage paging
- Amazon EBS storage throughput
- Amazon EBS storage IOPS
- Amazon EBS storage space free and volumes full %
- Amazon EC2 network saturation

- ELB/ALB health and target group health

Base your alarm thresholds on healthy patterns of historical production usage of your system. Continually review and tweak your alarm thresholds to prevent problems.

Review the following resources to get started:

- SAP on AWS Blog: Serverless Monitoring for SAP NetWeaver
- SAP on AWS Blog: Serverless Monitoring for SAP HANA
- AWS Documentation: Create a CloudWatch Custom Metric
- AWS Documentation: Create a CloudWatch Dashboard
- AWS Documentation: Using CloudWatch Alarms

Suggestion 1.2.2 - Implement AWS service quota monitoring for SAP services

Implement a monitoring tool, such as Amazon CloudWatch, or other process to keep track of your AWS service quotas for required SAP resources in your landscape. Amazon EBS and Amazon EC2 instance quotas are the most common quotas to affect a SAP workload.

For EC2 instance quotas, consider that SAP landscapes can often use a mix of Amazon EC2 instance types and that some types have a different On-Demand service quota. For example, the `x*` and `u*` (High Memory) EC2 instance types have a different service quota that is separate from the combined quota for standard types like `c6`, `m6`, and `r6` instance families.

When planning new or scaling existing workloads, verify that your service quotas will support this and engage AWS Support if a quota increase is required.

- AWS Blog: AWS Systems Manager Automation now enables monitoring of service usage quota in Amazon CloudWatch
- AWS Documentation: Service Quotas - AWS General Reference
- AWS Documentation: On-Demand Instances - Amazon Elastic Compute Cloud Service Quotas
- AWS Documentation: Requesting a quota increase - Service Quotas

Best Practice 1.3 – Implement application and database monitoring for SAP

Set up your application and database monitoring to provide information about its internal state, status, and achievement of business outcomes. Some examples include transaction response time, available work processes, queue depth, error and dump messages, stalled batch jobs, and transaction throughput. Use this information to determine when a corrective action is required.

Suggestion 1.3.1 - Implement monitoring for databases supporting SAP applications

Continually monitor your SAP databases and establish alerts for common problems that can affect SAP system availability and performance. Common monitoring items include the following:

- Free space in data area
- Free space in logging area
- Excessive locking activity
- Cache utilization rates
- Average query response time
- Required security patches and hot fixes
- Top table sizes and growth

Base alerting thresholds on healthy patterns of historical productive usage of your system. Continually review and adjust your alarm thresholds to prevent problems and to react to workload changes or growth.

For details on how to enable monitoring for your specific database, see your database software provider installation and operational guides.

Consider Amazon CloudWatch Application Insights for SAP HANA databases to analyze metric patterns using historical data to detect anomalies, and continuously track errors and exceptions from HANA, operating system, and infrastructure logs.

- SAP on AWS Blog: Set up observability for SAP HANA databases with Amazon CloudWatch Application Insights

Suggestion 1.3.2 - Use SAP transactions and tools to understand the SAP application

Configure your SAP applications to provide information about their internal state, status, and the achievement of business outcomes. Use this information to determine when a response is required. Common monitoring items include the following:

- Availability of application (ASCS, PAS, AAS) and database services
- Number of active and concurrent users
- Availability of work processes for users
- Response time of user transactions
- Response time of batch and non-interactive transactions
- Error messages and dumps
- Failed jobs
- Full and slow queues

Set up the SAP EarlyWatch Alert reporting system in SAP Solution Manager to create regular reports on the status of your SAP systems. Regularly review and remediate issues found in these reports to prevent problems and avoid interruptions to workload service.

- SAP Note: 2729186 - General Process of EWA Generation [Requires SAP Portal Access]
- SAP Documentation: SAP Solution Manager 7.2 - Application Operations
- SAP Lens [Performance efficiency]: Best Practice 16.1 – Have data to evaluate performance (p. 83)

Suggestion 1.3.3 - Implement monitoring for your data recovery and protection mechanisms

Implement monitoring for mechanisms that safeguard your SAP data in the case of a failure or disaster. Common monitoring items include:

- Alerts for regular database backups, for example, to Amazon S3 with the AWS Backint Agent
- Alerts for database replication, for example, HANA system replication failure or delays across Availability Zones
- Alerts for file storage backups, for example, an EBS snapshot, an Amazon EFS backup, or an Amazon FSx backup
- Alerts for recovery mechanisms which provide data resilience across Regions, for example, AAmazon S3 buckets with cross-Region replication, Amazon S3 sync or CloudEndure Disaster Recovery
- Alerts for any recovery mechanisms which provide data resilience across accounts, for example, Amazon S3 buckets with same-Region replication to a WORM S3 bucket or logging account

See the following links for further information:

- AWS Blog: Monitor, Evaluate, and Demonstrate Backup Compliance with AWS Backup Audit Manager
- SAP Documentation: SAP HANA System Replication Verification and Monitoring

Suggestion 1.3.4 - Expose SAP monitoring data outside of SAP tools for independent observability

SAP monitoring tools are limited to application and operating system level monitoring and do not cover the wide range of supporting services that give an end-to-end view of SAP service availability and health. Configure your SAP applications to provide metrics to a more holistic, external monitoring and visualization tool of your choice.

Use the metrics collected in the previous best practices and externalize these results such that you have an independent tool which can monitor, alert, and report on trends. An independent tool allows observability, root cause analysis, historical and trend reporting without being linked to the SAP system's availability (that is, when SAP is in a disaster or fault mode).

- SAP on AWS Blog: Serverless Monitoring for SAP NetWeaver
- SAP on AWS Blog: Serverless Monitoring for SAP HANA
- SAP on AWS Blog: Set up observability for SAP HANA databases with Amazon CloudWatch Application Insights
- AWS Documentation: Create a CloudWatch Custom Metric
- AWS Marketplace: Products and Tools for SAP Monitoring

Best Practice 1.4 – Implement workload configuration monitoring

Design and configure your workload to provide information about its current configuration and changes to this configuration. Some examples are new or removed EC2 instances, scaling events, code change, patch levels, security group configuration, and resource deletion. Use this information to determine when a response is required and to decide whether a change was expected or permitted. Monitor the cost implications of configuration changes and adjust or analyze budgets if required.

Suggestion 1.4.1 - Implement workload configuration monitoring

Set up and configure AWS CloudTrail to monitor high priority and critical events, particularly in your SAP production accounts. Example events include new Amazon EC2 instances, Amazon EC2 decommissioning or changes, security group changes, and AWS KMS and IAM security change events. Use these events to configure CloudWatch Log Alarms (if required) and take action in the event of an unexpected change.

- AWS Documentation: What Is AWS CloudTrail?
- AWS Service: AWS CloudTrail
- AWS Documentation: Monitoring CloudTrail Log Files with Amazon CloudWatch Logs
- AWS Documentation: AWS CloudTrail Security Best Practices

Suggestion 1.4.2 - Implement workload configuration enforcement and remediation

Set up and configure AWS Config to track, evaluate, and enforce configuration policy of your AWS resources supporting your SAP production applications. Common examples include enforcing read-only protection on S3 buckets containing SAP backups, mandatory Amazon EBS encryption, blocking common network ports, and checking that all resources have required tags. Use AWS Config Managed Rules to improve the security and change control posture of your AWS environment supporting SAP. Use AWS tags to enforce configuration rules and apply automated remediation where possible.

- AWS Service: AWS Config

- AWS Documentation: Getting started with AWS Config
- AWS Documentation: Using AWS Config Rules
- SAP on AWS Blog: Audit your SAP systems with AWS Config – Part I
- SAP on AWS Blog: Audit your SAP systems with AWS Config – Part II
- SAP on AWS Blog: Tagging Recommendations for SAP on AWS

Suggestion 1.4.3 - Implement workload cost monitoring

Set up and configure AWS Budgets with custom budgets that alert you when you exceed (or are forecasted to exceed) your billing thresholds. Align budgets with your projected SAP environment spend and monitor for any anomalies to prevent cost overruns. Monitor your use and coverage of Reserved Instances and Savings Plans by using budget reports. Use AWS tags to assist in understanding cost allocation and usage across your SAP workload.

- AWS Blog: Getting Started with AWS Budgets
- AWS Blog: AWS Budgets Reports
- AWS Documentation: AWS Cost Explorer
- AWS Documentation: AWS Cost Anomaly Detection
- SAP on AWS Blog: Tagging Recommendations for SAP on AWS

Best Practice 1.5 – Implement user activity monitoring

Configure your SAP applications to provide information about user activity, for example, response time, number of active users, transaction abandonment rates, and order processing time. Consider both inside-out approaches (monitoring SAP internal dialogue response time) and outside-in approaches (deploying agents or robots at end-user locations geographically) to understand how connectivity plays a role in the experience. Use this information to help understand how the application is used, patterns of usage, and to determine when a response is required due to poor performance.

Suggestion 1.5.1 - Implement user experience monitoring from end-user locations

Consider outside-in monitoring approaches by deploying user agents or robots at end-user locations geographically to understand how network and connectivity play a role in SAP user experience. Often this type of end-user location-based monitoring can provide insight and early warning of problems not detectable in the central infrastructure and applications.

Implement Amazon CloudWatch RUM, SAP, or third-party tools which provide end-user experience reporting to measure the responsiveness of your SAP application from end-user locations. For example, SAP provides End-User Experience Monitoring in Solution Manager, and Amazon CloudWatch RUM allows the deployment of monitoring scripts to measure front-end user experience.

- SAP on AWS Blog: Monitor and Optimize SAP Fiori User Experience on AWS using CloudWatch RUM
- SAP Documentation: SAP User Experience Monitoring
- AWS Marketplace: Products and Tools for SAP Monitoring

Best Practice 1.6 – Implement dependency monitoring

Configure your workload to provide information about the status (for example, reachability or response time) of resources it depends on. Examples of external dependencies can include interfaces (for example, through SAP PI/PO), external data stores, DNS, on premises components, Active Directory controllers and network devices. Use this information to determine when a response is required. Consider third party monitoring tools that can provide cross-technology metrics to monitor the health of end-to-end dependencies.

Suggestion 1.6.1 - Implement health tracking for your key SAP interfaces and cross system business processes

Identify and monitor your key interfaces which your SAP workload is dependent on. Monitor the health of these interfaces endpoints, errors, queue length and success rates. Use in-built mechanisms in SAP or third-party integration tools to set up alerts on interface failure or delay and feed these into your monitoring tools. Consider all interface pathways:

- Between different AWS hosted SAP systems (direct via RFC or web service/HTTPS)
- Between AWS hosted SAP systems and on-premises systems (HTTPS/SFTP - through SAP PI or third-party integration platform)
- Between AWS hosted SAP systems and SAP Business Technology Platform (via SAP Cloud Connector)
- Between AWS hosted SAP systems and external party systems (typically via HTTPS over the internet/VPN)

Consider Solution Manager Business Process Monitoring for cross-system dependency monitoring throughout your SAP and non-SAP landscape.

- SAP Documentation: SAP Business Process and Interface Monitoring
- AWS Marketplace: Products and Tools for SAP Monitoring

Suggestion 1.6.2 - Implement health tracking for your enterprise services which SAP is dependent on

An SAP workload is typically dependent on several foundational enterprise services to be healthy for business users. Consider these foundation services in your monitoring approach and tools. Example foundational services include Direct Connect for on-premises system connectivity, Active Directory for authentication/SSO, Network Time Protocol (NTP) for time synchronization, antivirus services and connectivity to an operating system patch repository (for example, Microsoft Windows Update or SUSE patching).

- AWS Documentation: Collect metrics and logs from Amazon EC2 instances and on-premises servers with the CloudWatch Agent
- AWS Documentation: Enhanced monitoring capabilities for AWS Direct Connect

Best Practice 1.7 – Implement single pane of glass health monitoring across your SAP workloads

Configure your SAP applications, AWS services, and any dependent components to provide information about the flow of transactions across the workload. Combine metrics from multiple sources to create a single pane of glass visualization for the health of your SAP workload and make this dashboard accessible to your key users. Use this information to determine when a response is required and to assist you in quickly identifying the factors contributing to an issue impacting your business.

Suggestion 1.7.1 - Combine application metrics, workload configuration, user metrics, and dependency health in a single location

Combine application monitoring metrics, workload configuration data, user metrics and dependency health in a single location or tool to allow end-to-end monitoring of your SAP workload and its health for end-user business processes. This can be achieved through the use of SAP Solution Manager, custom CloudWatch dashboards and metrics, or third-party monitoring tools.

Best practice is to create business facing health dashboards with traffic light health and trends, which allow a drill-down view of workload availability. Drill down capabilities allow users and operators to assess the specific component of the technology stack which may be causing a problem or underperforming.

- AWS Documentation: Create a CloudWatch Dashboard
- SAP on AWS Blog: Serverless Monitoring for SAP NetWeaver
- SAP on AWS Blog: Serverless Monitoring for SAP HANA
- AWS Marketplace: Products and Tools for SAP Monitoring
- SAP Documentation: SAP Solution Manager 7.2 - Application Operations

Best Practice 1.8 – Use automated response and recovery techniques to react to monitoring alerts

Automate responses to events to reduce errors caused by manual processes, and to ensure prompt and consistent responses.

Suggestion 1.8.1 - Use automation services to automate your responses to events

There are multiple ways to automate the running of remediation activities for events triggered from your monitoring tools. Generally, you should seek to funnel all of your SAP application and database events into a single channel which can provide event-based automation in response.

- To respond to an event from a state change in your AWS resources, or from your own custom events from SAP, you could create EventBridge rules to invoke actions in Event targets (for example, Lambda functions, Amazon Simple Notification Service (Amazon SNS) topics, Amazon ECS tasks, and AWS Systems Manager Automation). AWS Systems Manager automation can be used to call the `sapcontrol` command and perform SAP system tasks automatically.
- To respond to a metric that crosses a threshold for a resource (for example, wait time), you should create CloudWatch alarms to perform one or more actions using Amazon EC2 actions, Auto Scaling actions, or to send a notification to an Amazon SNS topic.
- If you need to perform custom actions in response to an alarm, invoke Lambda through Amazon SNS notification or an AWS Systems Manager Automation (for example, using Action `aws:runCommand`) see AWS Blog: Automate Start or Stop of Distributed SAP HANA systems using AWS Systems Manager.
- Use Amazon SNS to publish Event Notifications and escalation messages to keep people informed.
- AWS also supports third-party systems through the AWS service APIs and SDKs. There are a number of monitoring tools provided by AWS Partners and third parties that allow for monitoring, notifications, and responses. Some of these tools include Avantra, New Relic, Splunk, Loggly, SumoLogic, and Datadog.
- Consider pushing events and interactions into third-party ITIL tools where applicable for your organization - such as AWS to ServiceNow integration.

You should keep critical manual procedures available for use when automated procedures fail.

2 – Reduce defects, ease remediation, and improve workflow of SAP change

How do you reduce defects, ease remediation, and improve flow into production? Adopt approaches that improve flow of changes into production, which allow refactoring, fast feedback on quality, and bug fixing. These accelerate beneficial changes entering production, limit issues deployed, and enable rapid identification and remediation of issues introduced through deployment activities.

ID	Priority	Best Practice
☐ BP 2.1	Required	Use version control and configuration management

ID	Priority	Best Practice
☐ BP 2.2	Required	Implement practices to improve code quality
☐ BP 2.3	Required	Use build and deployment management systems
☐ BP 2.4	Required	Use multiple environments
☐ BP 2.5	Required	Test and validate changes
☐ BP 2.6	Highly Recommended	Make frequent, small, and reversible changes
☐ BP 2.7	Recommended	Automate testing, integration, and deployment of changes

For more details, see the following links and information:

- AWS Video: Design with Ops in Mind
- AWS Documentation: AWS Developer Tools
- AWS Documentation: AWS Launch Wizard for SAP
- SAP on AWS Blog: DevOps for SAP – Driving Innovation and Lowering Costs

Best Practice 2.1 – Use version control and configuration management

Configuration Management systems reduce errors caused by manual processes and reduce the level of effort to deploy changes. Doing so supports tracking changes, deploying new versions, detecting changes to existing versions, and reverting to prior versions (for example, rolling back to a known good state in the event of a failure). Integrate the version control capabilities of your configuration management systems into all your procedures across SAP – the infrastructure, the database, the application, and SAP custom code and developments (for example, ABAP, Java, and UI5/JavaScript).

Consider different version control systems for each type of configuration, but consolidate metrics into a central release planning tool. Consider how non-transportable configuration and binary versioning is managed across your environments (for example - how do you know that your SAP Kernel versions are aligned across your landscape?).

Suggestion 2.1.1 - Implement SAP change control or other third-party tools for managing your SAP development code and versioning

Ensure you implement change control for all development approaches and custom code that support your SAP applications - ABAP, Java, UI5/JavaScript, and any other extensions or scripting areas. Consider all your SAP applications and how you will orchestrate code deployment across multiple SAP deployment patterns (for example, how will you simultaneously release related developments hosted on AWS and SAP Business Technology Platform).

- AWS Service: AWS CodeCommit
- AWS Video: Introduction to AWS CodeCommit
- SAP on AWS Blog: AWS DevOps tools for SAP, Part 1: Cloud Foundry
- SAP on AWS Blog: AWS DevOps tools for SAP, Part 2: SAP Fiori Apps
- SAP Documentation: SAP Change Control Management
- SAP Documentation: Best Practices for SAP BTP - Lifecycle Management

Suggestion 2.1.2 - Implement configuration management systems for your SAP applications

Implement configuration management tools for ABAP, Java, and other SAP technologies and consider how non-transportable configuration and binary versioning is managed across your landscape (for example - how do you know that your SAP Kernel versions are aligned across your environment?). Use SAP Solution Manager to plan and implement configuration and version changes to your SAP applications.

- SAP on AWS Blog: Maintain an SAP landscape inventory with AWS Systems Manager and Amazon Athena
- SAP Documentation: Enhanced Change & Transport System (CTS+)
- SAP Documentation: SAP Solution Manager: Planning Landscape Changes

Suggestion 2.1.3 - Implement configuration management systems for operating systems

Use AMI baking or in-place configuration management software such as Ansible, Chef or Puppet to align configuration management across your SAP workload operating systems. Consider security focused configuration management tools which will alert you to vulnerabilities and prompt you to keep your operating systems patched and hardened.

- AWS Documentation: AWS Systems Manager - State Manager
- AWS Documentation: Configuration management in Amazon EC2
- AWS Documentation: What is AWS OpsWorks?
- AWS Documentation: What is Amazon Inspector?

Suggestion 2.1.4 - Implement configuration management systems for databases

Work with your database software vendor to understand configuration management approaches for your database.

- SAP Documentation: SAP HANA Platform Lifecycle Management

Suggestion 2.1.5 - Implement configuration management systems for infrastructure

Use infrastructure as code (IaC) approaches to provision and manage AWS resources supporting your SAP workloads. AWS CloudFormation and AWS Cloud Development Kit (AWS CDK) are tools you can use to provision and manage configuration in AWS resources programmatically.

Consider configuration audit and control tools such as AWS Config: Conformance Packs that allow you to deploy rules and policies to evaluate your infrastructure periodically to assess compliance and resolve any problems with applicable best practices and standards.

- AWS Documentation: AWS Launch Wizard for SAP
- AWS Documentation: AWS Systems Manager Inventory
- AWS Documentation: AWS Systems Manager Change Manager
- SAP on AWS Blog: Infrastructure as Code Example: Terraform and SAP on AWS
- SAP Lens [Reliability]: Best Practice 11.3 - Define an approach to restore service availability (p. 62)

Best Practice 2.2 – Implement practices to improve code quality

Implement practices to improve code quality and minimize defects. For example, test-driven development, code reviews, and standards adoption. Use SAP Code Inspector tools at a minimum.

Suggestion 2.2.1 - Implement practices to improve code quality

For example, test-driven development, pair programming, code reviews, and standards adoption.

Suggestion 2.2.2 - Use Code Amazon Inspector tools for SAP development and integrate this process into your CI/CD pipeline

Consider the following tools for automated code inspection and linting in your SAP workloads:

- AWS Documentation: Amazon CodeGuru - for AWS Java and Python development
- SAP Documentation: SAP Code Inspector for ABAP and SAP-specific development

Best Practice 2.3 – Use build and deployment management systems

Use build and deployment management systems. Ensure you are using SAP certified build and deployment systems such as the ABAP Change and Transport System (CTS), Web IDE or SAP tools. These systems reduce errors caused by manual processes and reduce the level of effort to deploy changes.

Suggestion 2.3.1 - Implement SAP build and deployment systems

Implement SAP certified build and deployment systems such as the ABAP Change and Transport System (CTS), Web IDE, SAP BTP Continuous Delivery service or other SAP tools.

- AWS Whitepaper: Practicing Continuous Integration and Continuous Delivery on AWS
- SAP on AWS Blog: AWS DevOps tools for SAP, Part 2: SAP Fiori Apps
- SAP Documentation: Software Logistics Toolset - Change and Transport Tools
- SAP Documentation: Deploying Applications to BTP

Best Practice 2.4 – Use multiple environments

Use multiple SAP environments to experiment, develop, and test your workload. Use increasing levels of controls as environments approach production to gain confidence your workload will operate as intended when deployed. Generally, a three-tier environment for development, test, and production is minimum for SAP landscapes.

Suggestion 2.4.1 - Use temporary environments for experimentation

Provide technology testing and developer teams with sandbox or temporary environments with minimized controls to enable experimentation and risk mitigation.

- AWS Documentation: AWS Launch Wizard for SAP
- SAP on AWS Blog: Infrastructure as Code Example: Terraform and SAP on AWS
- SAP on AWS Blog: Automate Start or Stop of Distributed SAP HANA systems using AWS Systems Manager

Suggestion 2.4.2 - Provide development environments to allow work in parallel and improved agility

Provide non-production environments to allow work in parallel, increasing development and test agility. Implement more rigorous controls in the environments approaching production to allow developers the necessary means for innovation. Generally, a three-tier environment for Development, Test and Production is minimum for SAP environments.

Suggestion 2.4.3 - Provide a consolidated test environment that replicates production as closely as possible to improve release quality

Test and staging environments should mirror as closely as possible the interfaces, security, resilience, and performance characteristics of your production environment to identify architectural and code interaction problems before being released. Consider shutting down secondary resources in clusters or

scaling down (both horizontally and vertically) application server performance of this environment when not in use to improve landscape cost efficiency.

Suggestion 2.4.4 - Use infrastructure as code (IaC) and configuration management systems to deploy environments consistently

Use infrastructure as code (IaC) and configuration management systems to deploy environments that are configured consistent with the controls present in production to ensure systems operate as expected when deployed. Use tagging and resource groups to label and enhance environment metadata such that it can be used for automation and compliance purposes.

- AWS Documentation: AWS Launch Wizard for SAP
- SAP on AWS Blog: Infrastructure as Code Example: Terraform and SAP on AWS
- AWS Documentation: What are AWS Resource Groups?
- SAP on AWS Blog: Tagging Recommendations for SAP on AWS

Suggestion 2.4.5 - Turn off non-production environments when not in use

When environments are not in use, turn them off to avoid costs associated with idle resources (for example, development systems on evenings and weekends).

- SAP on AWS Blog: Automate Start or Stop of Distributed SAP HANA systems using AWS Systems Manager

Best Practice 2.5 – Test and validate changes

Changes should be tested and the results validated at all lifecycle stages (for example, development, test, and production). Use testing results to confirm new features and mitigate the risk and impact of failed deployments. Automate testing and validation to ensure consistency of review, to reduce errors caused by manual processes, and reduce the level of effort.

Suggestion 2.5.1 - Changes should be tested and the results validated at all lifecycle stages (for example, development, test, and production)

Suggestion 2.5.2 - Maintain a baseline of testing results across functional testing, performance and resiliency to compare to when releasing change and major projects.

Suggestion 2.5.3 - Understand what level of testing is required for differing levels of change. For example, a full suite of testing vs targeted regression testing for minor changes. Agree on test definitions and scope of change testing required to release to production.

Suggestion 2.5.4. - Automate testing where possible with third-party tools and test harnesses. Focus on regular change types and frequent releases first.

Best Practice 2.6 – Make frequent, small, and reversible changes

Frequent, small, and reversible changes reduce the scope and impact of a change. Although many SAP NetWeaver solutions only support a “patch forward” approach, consider using feature toggles in custom development to allow rollback. This eases troubleshooting, enables faster remediation, and provides the option to roll back a change.

Suggestion 2.6.1 - Divide development and releases into frequent and smaller changes where possible

Suggestion 2.6.2 - Because many SAP solutions only support a “patch forward” approach (and do not allow reversible transports), consider using feature toggles in custom development to allow disablement of features rather than rollback/withdraw

Suggestion 2.6.3 - For non-reversible SAP changes, consider additional rollback options, such as whole system snapshots, database backup, and restore options

- AWS Blog: Amazon EBS crash-consistent snapshots
- AWS Documentation: Restoring to a specified time using Point-In-Time Recovery (PITR)
- AWS Documentation: AWS Backint for SAP HANA

Best Practice 2.7 – Automate testing, integration, and deployment of changes

Automate build, deployment, and testing of the workload. This reduces errors caused by manual processes and reduces the effort to deploy changes.

Suggestion 2.7.1 - Fully automate the integration and deployment pipeline from code check-in through build, testing, deployment, and validation

Suggestion 2.7.2 - Implement SAP Solution Manager ChaRM, Focused Build or third-party change and release management tools to orchestrate end-to-end build to deployment pipelines for application changes

- SAP Documentation: SAP Solution Manager Change Request Management
- SAP Documentation: SAP Focused Build
- AWS Marketplace: Products and Tools for DevOps
- AWS Marketplace: Products and Tools for Testing
- SAP on AWS Blog: AWS DevOps tools for SAP, Part 1: Cloud Foundry

3 – Understand how you will operate the workload

How do you know that you are ready to support and operate a workload? Evaluate the operational readiness of your workload, processes and procedures, and personnel to understand the operational risks related to your workload. Create runbooks for common operations, playbooks for issues and automate as many operations as possible to improve resilience and reduce errors.

ID	Priority	Best Practice
☐ BP 3.1	Required	Ensure personnel capability
☐ BP 3.2	Required	Ensure your cloud operating model matches your operational aims
☐ BP 3.3	Required	Share design standards and educate new support personnel in procedures
☐ BP 3.4	Required	Use runbooks to perform SAP landscape operations
☐ BP 3.5	Required	Use playbooks to investigate issues
☐ BP 3.6	Highly Recommended	Use automation to perform SAP landscape operations

For more details, see the following links and information:

- AWS Whitepaper: AWS Cloud Operating Model

- AWS Service: AWS Cloud Adoption Framework (AWS CAF)
- AWS Service: AWS Config
- AWS Service: AWS Systems Manager
- AWS Documentation: AWS Systems Manager Features
- SAP on AWS Blog: DevOps for SAP – Driving Innovation and Lowering Costs

Best Practice 3.1 – Ensure personnel capability

Have a mechanism to validate that you have the appropriate number of trained personnel to provide hands-on support for operational needs and that they have the appropriate SAP, AWS, or third-party certifications. Train personnel and adjust personnel capacity as necessary to maintain effective support.

Suggestion 3.1.1 - Assess the learning and certification needs of your SAP operations team

According to your environment and dependencies, different certifications might apply. Assess the certification needs of your team to be able to support your technology stack:

- AWS Documentation: AWS Training
- AWS Documentation: AWS Certifications
- SAP Documentation: SAP Certifications
- Operating System Certifications
  - SUSE Documentation: SUSE Enterprise Linux Certifications
  - Red Hat Documentation: Red Hat Enterprise Linux Certifications
  - Microsoft Documentation: Microsoft Windows Certifications

Best Practice 3.2 – Ensure your cloud operating model matches your operational aims

Identify the appropriate cloud operating model for your SAP workloads such that it aligns with your identified business requirements for speed to deployment, security, operations, and responsibility of cloud platform support. An appropriate cloud operating model is critical for successful adoption of cloud and delivering greater business agility.

Suggestion 3.2.1 - Adopt the appropriate cloud operating model for your business aims

According to your IT and business requirements, ensure that the appropriate cloud operating model is adopted. Decide which teams will build and operate your workload. Plan to move towards a model of shared ownership where the SAP Basis/Technology team and development team both build and run your SAP workload in a DevOps model.

- AWS Guidance: AWS Cloud Adoption Framework (AWS CAF)
- AWS Well-Architected Framework [Operational Excellence]: Operating Models 2x2
- AWS Well-Architected Framework [Operational Excellence]: Organizational Culture

Best Practice 3.3 – Share design standards and educate new support personnel in procedures

Share existing best practices, design standards, checklists, operating procedures, and governance requirements across teams. Ensure all teams are aware of support procedures across all components of your SAP workload.

Suggestion 3.3.1 - Share existing best practices, design standards, checklists, operating procedures, and guidance and governance requirements across teams to reduce complexity and maximize the benefits from development efforts

Suggestion 3.3.2 - Ensure that procedures exist to request changes, additions, and exceptions to design standards to support continual improvement and innovation

Suggestion 3.3.3 - Ensure that teams are aware of published content so that they can limit rework and wasted effort

Suggestion 3.3.4 - Ensure that teams know how to log support calls for different components of your SAP workload

Who provides support for your operating system, database, and SAP application? For example, understand whether AWS or your operating system vendor would provide support directly for clustering or patching issues. In the case of EC2-inclusive operating system licenses, AWS provides this support directly.

- AWS Documentation: How to log a case with AWS Support
- AWS Documentation: AWS Support
- SAP Note: 1656250 - SAP on AWS: Support prerequisites [Requires SAP Portal Access]

Best Practice 3.4 – Use runbooks to perform SAP landscape operations

Runbooks are documented procedures to achieve specific outcomes. Enable consistent and prompt responses to well-understood events by documenting procedures in runbooks. Understand common SAP operations that are run and create specific, versioned documentation with a review cycle.

- AWS Well-Architected Framework [Operational Excellence]: Operational Readiness
- AWS Documentation: Runbooks and automation using AWS Incident Manager

Suggestion 3.4.1 - Create specific runbooks for SAP security operations

Consider creating runbooks for common SAP security operations:

- User provisioning and identity management
- Firefighter access
- Authorization changes
- Security and authorization audits
- Encryption key rotation
- TLS certificate management

Suggestion 3.4.2 - Create specific runbooks for SAP scaling and performance operations

Consider creating runbooks for common scaling and performance operations:

- Disk volume re-sizing
- Horizontal and vertical scaling of SAP application servers
- Re-sizing of database server
- Addition or removal of servers from load balancing

Suggestion 3.4.3 - Create specific runbooks for SAP operations during faults

Consider creating runbooks for operations during faults:

- System restarts and order of restarting systems
- SAP backups and restores
- Cluster failover
- Storage failure
- Critical interface restarts and replays
- DNS and network routing changes
- Ransomware recovery

- SAP Lens [Reliability]: Best Practice 10.3 – Define an approach to help ensure the availability of critical SAP data (p. 55)

Suggestion 3.4.4 - Create specific runbooks for SAP maintenance operations

Consider creating runbooks for maintenance operations:

- Starting and stopping SAP
- Refreshing / System Copy of SAP
- Daily health checks
- Error management / ABAP dumps
- Patching SAP application, operating system, and database
- Log rotation, clean up, and archival

Consider database and application log and trace files cleanups for your SAP environment, for example, SAP Note: 2399996 - Automating SAP HANA Cleanup [Requires SAP Portal Access]

Best Practice 3.5 – Use playbooks to investigate issues

Enable consistent and prompt responses to issues that are not well understood, by documenting the investigation process in playbooks. Validate and evolve these playbooks by using them regularly in operations but also in non-production environments and designated practice sessions like game days.

Suggestion 3.5.1 - Create problem playbooks for use in incident response

Understand the frequently occurring problems and troubleshooting steps used for each of the identified problems and create specific, versioned documentation with a review cycle. Suggested playbooks should include:

- Performance Issue Investigation
- Capacity Issue Investigation
- Authentication and Sign On Issue Investigation
- Security Incident Investigation
- Connectivity and Networking Investigation
- Ransomware and Virus Investigation
- Interface Error Investigation
- Batch Job Error Investigation
- Deployment or Transport Error investigation

Ensure that your playbooks include integration and communication steps with related support functions and teams. Common communications steps include notification and progress updates to a critical incident desk, a security incident team and/or a change management team.

Suggestion 3.5.2 - Run regular SAP game days to test operational procedures and validate playbooks

Consider running SAP game days regularly for your operational team. A game day simulates a failure or event to test systems, processes, and team responses. The purpose is to actually perform the actions the team would perform as if an exceptional event happened. These should be conducted regularly so that your team builds "muscle memory" on how to respond. Your game days should cover the areas of operations, security, reliability, performance, and cost. Using a dedicated experimentation environment, simulate real world scenarios in order to validate and practice operational procedures and recovery processes.

Best Practice 3.6 – Use automation to perform SAP landscape operations

Create automation pipelines for your SAP environment builds and landscape operations. Automation using Infrastructure as Code techniques (for example, CloudFormation, Launch Wizard for SAP) allows repeatable and agile environment creation or extension. Automated pipelines and landscape operations reduce errors caused by manual processes, reduce the effort to deploy changes and improves speed to react to your business needs.

Create automated SAP landscape operational pipelines that allow you to perform common environment tasks in an automated fashion (for example, System Copy, Start SAP, Stop SAP, Scale SAP). Invoke these pipelines in response to operational events such as time-based system shutdown or automatic scaling due to user load.

Suggestion 3.6.1- Implement infrastructure as code techniques to create repeatable and code-driven build pipelines for your SAP landscape

Use tools such as AWS CloudFormation, AWS Cloud Development Kit (AWS CDK) or AWS Launch Wizard for SAP to create repeatable, controlled and quick environment deployments.

- SAP on AWS Blog: Infrastructure as Code Example: Terraform and SAP on AWS
- AWS Documentation: AWS Launch Wizard for SAP

Suggestion 3.6.2 - Implement common SAP landscape operations with automation

Use orchestration and infrastructure as code (IaC) tools in combination to perform your common SAP landscape operations in an automated fashion. Tools such as AWS CloudFormation, AWS Systems Manager – Run Automations, SAP Landscape Management (LaMa) and AWS Lambda can be orchestrated to perform common SAP landscape operations in deployment pipelines.

Consider third-party automation tools where complex or deep integration between tools is required (For example: Terraform, Ansible, Chef).

Consider using automated operations as responses to SAP workload events to allow a self-healing and self-maintaining landscape.

- SAP Note: 2574820 - SAP Landscape Management Cloud Manager for Amazon Web Services (AWS) [Requires SAP Portal Access]
- AWS Documentation: AWS Launch Wizard for SAP
- AWS Documentation: AWS Systems Manager Automation
- AWS Marketplace: Products and Tools for DevOps

4 – Validate and improve your SAP workload regularly

How will you validate your SAP workloads continue to operate efficiently? Aim to improve your SAP workload regularly and take advantage of new service releases from AWS. Dedicate time and resources to maintain your SAP workload. Aim for continual incremental improvement to evolve the efficiency of your SAP workload. Plan to patch, make small changes and re-evaluate previous design decisions with corrective actions which improve performance, resilience or cost effectiveness.

ID	Priority	Best Practice
☐ BP 4.1	Required	Understand and plan for lifecycle events of your SAP workload
☐ BP 4.2	Required	Regularly perform patch management for software currency
☐ BP 4.3	Highly Recommended	Regularly test business continuity plans and fault recovery
☐ BP 4.4	Highly Recommended	Perform regular workload reviews to optimize for resiliency, performance, agility, and cost

Best Practice 4.1 – Understand and plan for lifecycle events of your SAP workload

SAP workloads are highly reliant on SAP to provide new software and vulnerability patching, operating system and database kernels, and escalation for support. SAP regularly publishes information about SAP software releases: Release types, maintenance durations, planned availability, and upgrade paths in their Product Availability Matrix (PAM) and SAP Notes. You should obtain specific details each of your SAP applications and track these locally to understand if your SAP software is current, supported and when it will be end of life from a maintenance perspective.

The PAM also offers information about platform availability and compatibility: including database platform and operating systems supported which should guide you in patching and upgrading these underlying components of your SAP workload. Operating System vendors also have their own patching and support lifecycle which should be taken into account when planning SAP maintenance and lifecycle events such as upgrades.

Suggestion 4.1.1 - Create an operational roadmap for your SAP applications taking into account key support and lifecycle dates

List all of your SAP software applications, kernel versions. operating systems, and database versions in a central register and consolidate with PAM information on supported versions and maintenance windows. Use this list as a consolidated view to plan patching, upgrades and platform changes in all components required to keep SAP current and within support.

- SAP Documentation: SAP Release & Maintenance Strategy: Product Availability Matrix [Requires SAP Portal Access]

Suggestion 4.1.2 - Maintain a calendar for expiring of credentials, certificates and licenses

Alongside the major SAP lifecycle events and patching mentioned previously, ensure you have an operational calendar which plans minor system events. Examples of these maintenance events could be expiry of system credentials, expiry of certificates (for example, for STRUST integration between systems) and any license renewal work or updates required (for example, temporary SAP or database licenses for migration, development or POC purposes).

- AWS Documentation: AWS Certificate Manager

Suggestion 4.1.3 - Plan for upgrades or alternatives before SAP software becomes end of life

Create an SAP landscape roadmap visualizing your key SAP lifecycle events and operational upkeep - patching, software upgrades, migrations and re-platforming if required. Communicate this lifecycle calendar to business and technical stakeholders. Plan investment to fund these SAP lifecycle activities/ projects. Plan in advance with your business stakeholders where maintenance windows can occur and downtime or restarts will be required.

- SAP Documentation: SAP Roadmap Explorer

Suggestion 4.1.4 - Stay up to date and subscribe to key SAP notes for support advice

Subscribe to key SAP notes and Knowledge Base Articles (KBAs) for your SAP workload such that you will be notified upon any changes or updates to supportability and advice. Use “Favorite” SAP notes functionality to keep a list of frequently accessed and important notes for your SAP workload to make them easily accessible and comparable.

- SAP Support Portal - Favorite SAP Notes [Requires SAP Portal Access]

Best Practice 4.2 – Regularly perform patch management for software currency

Perform regular patch management to gain features, address issues, and remain compliant with governance. Consider patches at the operating system, database and SAP application layer. Understand whether your patching process will be to patch your existing servers, or provision and patch a new server. Automate patch management to reduce errors caused by manual processes, reduce the level of effort to patch and reduce the application downtime required for major SAP, database, and kernel patching.

Suggestion 4.2.1 - Implement SAP patch management procedures to regularly review SAP Security Notes and newly released patches

Consider patches at the operating system, database and SAP application layer.

- AWS Documentation: AWS Security Bulletins
- SAP Documentation: SAP EarlyWatch Alert
- SAP Documentation: SAP Security News

Operating System	Guidance
SUSE Linux Enterprise Server	SUSE Update Advisories
Red Hat Enterprise Linux	Red Hat Security AdvisoriesRed Hat Customer Portal (Sign in with AWS)
Microsoft Windows	Microsoft Security Alerts
Oracle Enterprise Linux	Oracle Security Alerts

For further discussion on this item see [Security]: Best Practice 6.2 - Build and protect the operating system (p. 36).

Suggestion 4.2.2 - Consider automated tools to align and automate patches across your SAP landscape

Tools such as AWS Systems Manager and AWS OpsWorks can assist you to align, plan, test, and deploy patching across your SAP workload. Consider an automated approach to patching to minimize effort and maintenance windows.

- AWS Documentation: AWS Systems Manager Patch Manager
- AWS Documentation: AWS OpsWorks
- AWS Documentation: What is AWS OpsWorks?
- SAP Lens [Security]: Best Practice 6.2 - Build and protect the operating system. (p. 36)

Best Practice 4.3 – Regularly test business continuity plans and fault recovery

SAP systems are generally business critical and depended upon for major customer facing transactions. Enabling the quick resumption of IT operations and minimizing data loss during a fault or disaster situation is critical for operational excellence. Business continuity plans (BCP) and fault recovery procedures are required to ensure that your operations team and systems know what to do, when to do it, and workload service can be resumed promptly in case of a fault.

Critical to the successful resumption of services is that your BCP procedures and fault recovery plans are regularly tested, improved upon and refined as your systems and support team evolves. Testing your BCP and recovery plans outside of real crisis situations ensures that when a real system fails or disaster does occur, you can be confident in your ability to successfully resume service and that you will meet your recovery time objective (RTO) and recovery point objective (RPO).

Suggestion 4.3.1. - Create a BCP and fault recovery testing calendar

Create a calendar which schedules regular (at least annually) BCP and fault scenario recovery testing for your SAP workload. Involve technology operational teams, support personnel and business stakeholders in this test so that procedures are understood and expectations are aligned. Aim to test your systems in as real a situation as possible.

Consider testing the following scenarios and validating recovery metrics for each of them:

- SAP application service failure

  (for example, SAP application service fails to start due to a configuration change)
- Single instance host failure

  (for example, SAP application server EC2 instance becomes unreachable)
- Single storage volume failure

  (for example, a single EBS volume becomes unreachable)
- Network failure and switch over to redundant connection

  (for example, your on-premises Direct Connect connection is unreachable)
- Automated failover between primary and secondary clustered components

  (for example, SUSE HAE cluster forces primary HANA database to move to the secondary database in an alternate Availability Zone)
- Manual fail over between primary and secondary components

  (for example, manual invocation of Oracle DataGuard switch over to secondary database in an alternate Availability Zone)

- Load balancing between multiply redundant components

  (for example, primary web dispatcher fails in a high availability pair across Availability Zones)
- Recovery of your SAP application in an alternate AWS Region (if required)
- Recovery from backup in event of ransomware

  (for example, recovering your entire SAP ERP system from Amazon S3 WORM backup)

Suggestion 4.3.2 - Regularly review and update BCP and fault recovery procedures as part of workload changes

As your workload evolves and changes over time, ensure that BCP and recovery procedures are considered in these changes. When a code or infrastructure change might affect your RTO or RPO, ensure that documentation and configuration is updated, and the new BCP and recovery process is tested as part of the release process or regular test calendar.

- AWS Documentation: Business Continuity Plan (BCP) Definition
- AWS Documentation: Architecture Guidance for Availability and Reliability of SAP on AWS
- SAP Lens [Reliability]: Best Practice 11.4 - Conduct periodic tests of resilience (p. 64)

Best Practice 4.4 – Perform regular workload reviews to optimize for resiliency, performance, agility, and cost

When running SAP on AWS, plan and dedicate time and resources for continual incremental improvement to evolve the effectiveness and efficiency of your workload. AWS regularly releases new services, approaches, improved SLAs, and price reductions that you can take advantage of to optimize your SAP workload. Understand and validate whether new service releases are applicable to your SAP workload and, where appropriate, implement them in your production environment to evolve your workload.

Suggestion 4.4.1 - Plan regular reviews of your SAP workload

Work with your AWS team, AWS Partner, or internal experts to periodically review your SAP workload using the Well-Architected Framework SAP Lens (this document). Plan to review your workload at least every once a year. Identify, validate, and prioritize improvement activities and issue remediation and incorporate this into your backlog.

Consider harvesting your learnings from operational incidents into curated questions with best practices guidance. Create Operational Readiness Review (ORR) runbooks to assist when deploying new SAP landscapes, applications, or workloads.

- AWS Whitepaper (this document): SAP Lens for Well-Architected
- AWS Whitepaper: Operational Readiness Reviews (ORR)

Suggestion 4.4.2 - Review Amazon EC2 instance sizing and performance

Review the CPU usage and memory utilization of your SAP workload by validating historical CloudWatch metrics. Review each SAP component for low CPU or memory utilization and consider right sizing EC2 instances to better match workload requirements. Consider newly released and SAP-certified EC2 instance types for performance fit and cost optimization. Plan to take advantage of new improvements in your operational backlog.

See Cost Optimization (p. 87) for Amazon EC2 usage in SAP workloads.

- AWS Documentation: Amazon EC2 instance types for SAP

Suggestion 4.4.3 - Review Amazon EBS sizing and performance

Review storage usage across your SAP workload by validating volume consumption, throughput and IOPS usage from CloudWatch historical metrics. Review each SAP component for oversized storage or low throughput/IOPS utilization and consider right sizing Amazon EBS storage sizes and types in order to better match workload requirements. Consider newly released and SAP certified Amazon EBS types for performance fit and cost optimization. Plan to take advantage of new improvements in your operational backlog

- AWS Documentation: Right Sizing
- SAP Lens [Cost Optimization]: Best Practice 18.4 - Evaluate the cost impact of storage options based on the required characteristics (p. 101)

Suggestion 4.4.4 - Review new services that improve agility or improve efficiency in your SAP workload operations

Review new supporting service releases that could improve operations in your SAP workload. If you have a technical account manager (TAM) as part of an AWS Support agreement, they can assist you in a new service briefing and optimization discussion.

Consider new releases such as shared file storage, interface services (for example, AWS Transfer, API Gateway), security services (for example, Amazon GuardDuty, AWS Firewall), backup tools (for example, AWS Backint) and automation tools (for example, Launch Wizard for SAP).

Plan to take advantage of new improvements in your operational backlog.

- AWS Documentation: "What's New" Feed
- AWS Documentation: Proactive Services from Support

Suggestion 4.4.5 - Monitor SAP on AWS blogs and announcements

Consider subscribing to the SAP on AWS Blog feed and AWS "What's New" feed to stay up to date with newly released service announcements, innovation approaches and price reductions.

- SAP on AWS Blog Feed

Suggestion 4.4.6 - Plan periodic enhancement work to take advantage of new and improved AWS services

Ensure that your operational budget allows for planned support team effort for implementation and testing of new AWS services and workload evolution on a periodic basis.

Security

The security pillar encompasses the ability to protect data, systems, and assets to take advantage of cloud technologies to improve your security. This lens highlights some core principles and resources for SAP. As many of these practices are not unique to SAP, we suggest you consider the core principles for your enterprise and focus on establishing controls across your entire landscape. The Security Pillar whitepaper contains broader design principles and recommendations which we highly recommend you read in conjunction with the SAP guidance that follows.

Regardless of your deployment strategy—whether AWS based, on premises, or hybrid—be certain to follow the guidelines recommended by SAP Security Notes and News and keep abreast of the most up-to-date security recommendations specific to SAP workloads.

5 – Understand security standards and how they apply to your SAP workload

How do you define the security standards and controls to align with the criticality of your SAP workload? Standards are published documents that define the policies and procedures required to secure your systems following best practices for a product, organization, industry, or jurisdiction. They provide a framework against which your SAP workload can be evaluated. Some standards are mandatory to ensure compliance with regulatory requirements, while others are optional but help with establishing roles and responsibilities.

ID	Priority	Best Practice
☐ BP 5.1	Required	Define security roles and responsibilities
☐ BP 5.2	Highly Recommended	Classify the data within your SAP workloads
☐ BP 5.3	Highly Recommended	Determine the required security controls based on application and data classification
☐ BP 5.4	Highly Recommended	Create a strategy for managing security controls

Best Practice 5.1 – Define security roles and responsibilities

By defining the requirements to secure your SAP workloads, you can identify risks that must be addressed and ensure that security-related roles and responsibilities are appropriately assigned. In the suggestions, we discuss standards for AWS, SAP, and any service providers to form a baseline on which you can build your security strategy.

Suggestion 5.1.1 - Understand the AWS shared responsibility model

AWS is responsible for security of the cloud and you, as the customer, are responsible for security in the cloud. Be aware of and understand the following resources:

- AWS Documentation: AWS Shared Responsibility Model
- AWS Documentation:AWS Response to Abuse and Compromise
- AWS Documentation: AWS Acceptable Use Policy

Understand the division of responsibilities between you and your partners in the context of the AWS shared responsibility model

Suggestion 5.1.2 - Understand the security foundations across SAP and AWS including compliance certificates, reports, and attestations

Understand the security standards and compliance certifications that SAP and AWS support. Determine which are relevant to your industry and country (for example, PCI-DSS, GDPR, HIPAA). These controls can help strengthen your own compliance and certification programs, and reduce the effort required to meet your security standards.

Refer to the SAP and AWS documentation for more details:

- AWS Documentation: AWS Compliance
- AWS Documentation: AWS Compliance Center
- AWS Documentation: Compliance Programs

- AWS Documentation: Compliance Services in Scope
- SAP Documentation: Trust Center

Suggestion 5.1.3 - Assess the security foundation of the service providers that support your SAP workload

If you are dependent on third-party organizations to manage all or part of your SAP workload, assess the ability of the third party to meet the required security controls. This includes the legal and regulatory requirements mandated by your enterprise.

Best Practice 5.2 – Classify the data within your SAP workloads

Data sensitivity can impact the controls required to mitigate risk. AWS suggests referring to standard frameworks within your industry or organization and adopting these to classify your SAP workloads and the data contained within them.

Suggestion 5.2.1 - Determine data classification and handling requirements

Identify any data classification frameworks already in place in your organization. These frameworks can help you to categorize data based on the sensitivity of information, such as data that must be safeguarded for confidentiality, integrity, and availability. Standard classification models exist, for example, the US Information Categorization Scheme, that may be customizable based on your industry, business, or IT requirements.

Understand how data should be handled according to the guidelines appropriate for the classification. This includes specific security controls related to standards or regulatory requirements, such as PCI-DSS or GDPR, and common privacy considerations, such as handling personal identifiable information (PII). The following documents provide additional information:

- AWS Documentation: Data Classification: Secure Cloud Adoption Whitepaper
- AWS Documentation: General Data Protection Regulation (GDPR) Center
- NIST Security and Privacy Controls for Information Systems and Organizations
- ISO/IEC 27001:2013 FAQs
- Well-Architected Framework [Security]: Data Protection

Suggestion 5.2.2 - Identify SAP data types with specific handling rules

Based on the business processes supported by your SAP system, there may be requirements for the handling and storage of data. Familiarize yourself with the guidance for your location and industry. SAP examples may include:

- Assess whether a digital payments add-on is necessary to protect stored cardholder data and ensure PCI compliance.
- Assess HR data for data residency requirements, for example, some countries and jurisdictions might require data to be stored within a specific geographical location.
- Consider which data may need to be scrambled in non-production systems to obscure sensitive data but maintain data integrity.

Suggestion 5.2.3 - Classify all your workloads according to the defined framework

Classify your SAP systems according to their business usage and the existence of critical data types. Transactional systems such as SAP ERP are more likely to contain sensitive data than analytical systems such as SAP BW or management systems such as Solution Manager, although this should be validated by functional and security experts.

Additionally, assess whether the same controls apply to non-production workloads. For example, do non-production workloads include production data and therefore must they adhere to the same security controls?

Best Practice 5.3 – Assess the need for specific security controls for your SAP workloads

Based on the data classification, evaluate any controls that can help you to meet the standards and requirements established in the previous best practices. These include location, AWS account strategy and scrambling requirements for non-production SAP workloads.

Suggestion 5.3.1 - Assess any geographical location requirements

Your SAP workloads might be deployed in one or many AWS Regions and Availability Zones (AZs). Each AWS Region consists of multiple, isolated, and physically separate AZs within a geographic area. In addition to evaluating the Region for latency, resilience, and sustainability specifications, you should consider whether security and compliance requirements can be met. Examples of isolated Regions with specific operating jurisdictions include:

- AWS GovCloud (US) - designed to host sensitive data, regulated workloads, and address the most stringent US government security and compliance requirements
- Amazon Web Services in China - AWS has collaborated with local partners to ensure China's legal and regulatory requirements are met

Some industries and countries will have data residency requirements that all customer content processed and stored in an IT system must remain within a specific country's borders.

- AWS Documentation: AWS Security blogs for data residency

Before deciding on a location, review the availability of services for that AWS Region to ensure that the services that you require are available.

- AWS Documentation: AWS Regional Services

Suggestion 5.3.2 - Determine how your SAP workloads align with your AWS account strategy and landing zone

An important consideration when running SAP workloads in AWS is the AWS account strategy and landing zone approach that you adopt to meet your organization's security controls. You should consider separating SAP from non-SAP workloads and having production workloads in a separate account from non-production workloads.

Understand your organization's existing AWS account management strategy, including the use of the AWS Organizations and AWS Control Tower. Consider isolating security and log capabilities into an isolated account. Refer to the following for additional details:

- Well-Architected Framework [Security]: AWS Account Management and Separation
- AWS Documentation: Establishing your best practice AWS environment
- AWS Documentation: Organizing Your AWS Environment Using Multiple Accounts
- AWS Documentation:AWS multi-account strategy for your AWS Control Tower landing zone

The account strategy you adopt will also affect the network configuration within AWS. As part of determining the appropriate AWS account strategy for your SAP workloads you should consider the following:

- Requirements for cross-account access, such as the need for setting up VPC Peering or Transit Gateway to allow communication between non-production and production systems. For example, the movement of SAP transports through your landscape.
- Dependencies on shared services (such as directory management resources) and network management components that are deployed in different AWS accounts from your SAP workloads.
- In addition to the core security services, such as IAM and network controls, consider how AWS managed security services can help achieve security goals or uplift your security posture. AWS provides security services to assist with web application firewalls, traffic auditing, DDOS protection, CVE management, configuration auditing, and virus and threat detection.
- AWS Documentation: AWS Foundational Security Best Practices Controls

Suggestion 5.3.3 - Review the controls for data scrambling (if applicable)

Many SAP customers rely on copies of production data for testing purposes, including regression and performance testing. If creating a copy of production data, decide which controls you must add to ensure that your production data is protected from unintended access and modifications.

Consider the following options:

- Traditional data scrambling mechanisms provided by SAP or third-party providers
- The use of additional accounts or network controls to limit access during a copy of production data
- Use of a non-production account with the same controls as production

Best Practice 5.4 – Create a strategy for managing security controls

Having evaluated business requirements based on data classification, create a strategy that balances the security controls of your broader organization with the application guides and open standards available. Take into consideration the implementation effort and acknowledge risk.

Suggestion 5.4.1 - Identify a matrix to assess risk

A range of risk management frameworks are available for specific industries and geographies. Understand the risk framework adopted by your organization and how this applies to managing risks related to your SAP workloads.

- AWS Documentation: Example Risk Matrix
- AWS Blog: Scaling a governance, risk, and compliance program for the cloud
- NIST Risk Management Framework

Suggestion 5.4.2 - Evaluate security and compliance requirements mandated by your organization

Consult with your cloud center of excellence, legal team, compliance teams, and managed service provider to understand their security baseline and how controls are enforced. Evaluate whether all of these controls can easily be applied to your SAP workload and identify areas that might require an exception, for example allow and deny lists for AWS services, inbound and outbound traffic flow and access restrictions.

Suggestion 5.4.3 - Identify and agree on a process for exceptions

In some situations, software, business, or support requirements for SAP might require you to deviate from the standard security patterns. Identify a process to agree and document any exceptions with a change advisory board or security design authority and reassess the process on a regular basis.

AWS Documentation: Change Management in the Cloud

6 – Use infrastructure and software controls to reduce security misconfigurations

How do you protect your SAP application and the underlying database, operating system, storage, and networks? We recommend that SAP software solutions and the associated underlying configurations—such as operating system and database patches, parameters, cloud services, and infrastructure —be hardened. Hardening helps ensure the safety of all SAP environments, both production and non-production, at the appropriate level determined by your organization.

Use the AWS Shared Responsibility Model to guide your activities regarding the security of your SAP environment. For example, firmware updates for your EC2 instances are "security of the cloud" activities for which AWS is responsible, while operating system and application management for those same EC2 instances are "security in the cloud" activities for which you are responsible.

ID	Priority	Best Practice
☐ BP 6.1	Required	Ensure that security and auditing are built into the SAP network design
☐ BP 6.2	Required	Build and protect the operating system
☐ BP 6.3	Required	Protect the database and the application
☐ BP 6.4	Required	Establish a plan for upgrading and patching all applicable software

For more details, refer to the following information:

- AWS Documentation: Best practices for Security, Identity, & Compliance
- SAP Note: 2191528 - Third-party report showing security vulnerabilities [Requires SAP Portal Access]
- SAP Documentation: ABAP Platform Security Guide

Best Practice 6.1 – Ensure that security and auditing are built into the SAP network design

Protecting access to the network that hosts your SAP workloads is the first line of defense against malicious activity. Evaluate your business requirements and the specific SAP solution to determine the ports, protocols, and traffic patterns that need to be enabled. Consider the security standards of your organization and the tools and patterns available to simplify network design. Audit on a regular basis or as changes occur.

Suggestion 6.1.1 – Understand network traffic flows for SAP

Start by understanding your traffic flows. Network traffic patterns for SAP workloads can be categorized as inbound traffic, outbound traffic, and internal traffic. You should identify whether the source and destination fall within your trusted network boundary to assist with defining your rule sets.

In addition to known inbound traffic and outbound traffic flows such as user access and interface connections, consider SAP-specific requirements, including connections to SAP Support (via SAProuter) and SAP SaaS offerings that restrict access based on source IP addresses.

For internal traffic, consider traffic between components and systems, as well as AWS and shared services. Tools such as VPC Flow Logs and VPC Reachability Analyzer can help you understand traffic flows into and out of your Amazon VPC.

For more details, refer to the following information:

- AWS Documentation: Attack surface reduction
- SAP Documentation: TCP/IP Ports for All SAP Products

Suggestion 6.1.2 – Evaluate options to permit and restrict traffic flows

First, understand how you connect users and systems in your on-premises network to the AWS account in which your SAP systems are running. This is covered in Network-to-Amazon VPC connectivity options.

Two primary methods for controlling the flow of network traffic into and out of your VPC include the use of security groups and network access control lists (network ACL). A security group acts as a virtual firewall at the EC2 instance level to control inbound and outbound traffic and is stateful. A network ACL is an optional layer of security for your VPC that acts as a firewall for controlling traffic in and out of one or more subnets, and — unlike security groups — a network ACL is stateless.

Also consider the dependencies of network components outside of your VPC. This can include external network components provided by AWS such as CloudWatch endpoints. This also can include internet hosted services such as software repositories for operating system patches.

In addition to the standard options in AWS, SAP itself provides additional network security options, including the use of the SAProuter, the SAP Web Dispatcher, and SAP Gateway network-based access control lists. These work in tandem with AWS services and configurations to permit or restrict network access to SAP systems.

For more details, refer to the following information:

- SAP on AWS Blog: VPC Subnet Zoning Patterns for SAP on AWS
- Well-Architected Framework [Security]: Infrastructure Protection – Protecting Networks
- Well-Architected Framework [Management and Governance Cloud Environment Guide]: Network Connectivity
- SAP Documentation: Network and Communication Security

Suggestion 6.1.3 – Use design guidelines and AWS tooling to simplify network security

SAP systems often have complex integration requirements, and the cloud offers additional ways to simplify network security management. Consider the following approaches:

- Avoid referring to individual IP addresses or IP ranges where possible to simplify management.
- Use a standard set of SAP system numbers across all your SAP workloads to reduce the range of network ports required.
- AWS PrivateLink removes the requirement for outbound internet access from your VPC to access AWS services such as Amazon S3 and CloudWatch. Where possible and not mandated by business requirements, you can prevent SAP traffic to and from these services from traversing the internet, routing all traffic through AWS managed network components.
- Simplify security groups by the use of VPC Prefix Lists and/or security group rules that reference other security groups rather than IP address ranges.
- Use automation to create, update, and manage security groups to avoid configuration drift.
- Consider the use of AWS Firewall Manager to provide centralized management of security groups across VPCs and AWS accounts.

- Consider the use of SAProuter, SAP Web Dispatcher, and Elastic Load Balancing to obfuscate the entry points to backend systems.
- Consider the use of multiple SAP Internet Communication Manager (ICM) entry points to provide finer grain access control.
- Consider AWS Shield, a managed Distributed Denial of Service (DDoS) protection service, to safeguard applications running on AWS. Use to protect public-facing SAP Fiori or API endpoints.
- Consider AWS WAF, a web application firewall that helps protect your web applications or APIs against common web exploits and bots that may affect availability, compromise security, or consume excessive resources. Use to protect public-facing user interfaces and APIs, for example, SAP Fiori applications.

For more details, refer to the following information:

- SAP Documentation: Network-based Access Control Lists
- SAP Documentation: TCP/IP Ports for All SAP Products

Best Practice 6.2 – Build and protect the operating system

Protecting the operating system underlying your SAP software reduces the possibility that a malicious actor could gain unauthorized access to data within the SAP application, impact software availability, or otherwise destabilize your mission-critical implementation. Follow recommendations from SAP, the operating system vendor, the database vendor, and AWS to help secure the operating system. Depending on your chosen SAP solution and operating system, you may need to enable/disable services, set specific kernel parameters, and apply different combinations of security patches. Consider how SAP requirements align with those of your organization, and identify any conflicts.

Suggestion 6.2.1 – Determine an approach for provisioning a secure operating system

An Amazon Machine Image (AMI) provides the information required to launch an EC2 instance. You should be confident that your AMIs are secure at the operating system level; otherwise, security holes could be propagated to any number of instances as AMIs are reused and updated over time.

AMIs can be either standard images from the operating system vendor or custom images that you build yourself. In both cases, you need to have a consistent approach for ensuring the operating system is secure at launch and maintained in an on-going basis. Using infrastructure as code (IaC) tools such as AWS CloudFormation can assist with achieving image security consistency. For HANA-based SAP solutions, the AWS Launch Wizard for SAP simplifies the installation process, including pre- and post-installation scripts that can be customized to automate the installation of security components.

Refer to the AWS Well-Architected Framework [Security Pillar] guidance on protecting compute resources, specifically the information on performing vulnerability management and reducing the attack surface, for additional details.

- Well-Architected Framework [Security]: Protecting Compute

Suggestion 6.2.2 – Determine an approach for building and patching a secure operating system

As mentioned in the Well-Architected Framework [Security Pillar] discussion on protecting compute, if your chosen operating system is supported by the EC2 Image Builder, it can simplify the building, testing, and deployment of your SAP-specific AMIs and their ongoing patch management. AWS Systems Manager Patch Manager should also be investigated for maintaining the security posture of your operating system by automating security patch application.

- Well-Architected Framework [Security]: Protecting Compute
- AWS Documentation: EC2 Image Builder
- AWS Documentation: AWS Systems Manager Patch Manager

Suggestion 6.2.3 – Review additional security recommendations applicable to your operating system

Determine the complete list of items that are required to harden the operating system underlying the SAP software. For example, file system permissions on Linux-based systems should be set according to SAP guidelines, while limiting Administrator group access is a best practice on Windows-based systems.

The following SAP-specific recommendations might be relevant to your environment:

- SAP Documentation: SAP NetWeaver Security Guide - Operating System Security
- SAP Note: 2808515 - Installing security software on SAP servers running on Linux

Operating System	Guidance
All Supported UNIX/ Linux Operating Systems	• SAP Documentation: SAP System Security Under UNIX/LINUX
SUSE Linux Enterprise Server	• SAP Note: 2684254 - SAP HANA DB: Recommended OS settings for SLES 15 / SLES for SAP Applications 15 [Requires SAP Portal Access] • SAP Note: 2578899 - SUSE Linux Enterprise Server 15: Installation Note [Requires SAP Portal Access] • Operating system-specific Documentation: SUSE Hardening Guide
Red Hat Enterprise Linux	• SAP Note: 2777782 - SAP HANA DB: Recommended OS Settings for RHEL 8 [Requires SAP Portal Access] • SAP Note: 2772999 - Red Hat Enterprise Linux 8.x: Installation and Configuration (with particular mention of SELinux support) [Requires SAP Portal Access] • Red Hat Documentation: Red Hat Enterprise Linux Security Hardening Guide for SAP HANA 2.0 • Red Hat Blog: Security recommendations for SAP HANA on RHEL
Microsoft Windows	• SAP Documentation: SAP System Security on Windows • SAP Note: 1837765 - Security policies for <SID>adm and SAPService<SID> on Windows [Requires SAP Portal Access]
Oracle Enterprise Linux	• (Consult SAP or Vendor documentation for guidance)

Suggestion 6.2.4 – Validate the security posture of the operating system

After the operating system has been securely deployed and patched, validating the operating system security posture ensures that the operating system maintains an ongoing high level of security without violation. Consider automating this validation using third-party host intrusion protection, intrusion detection, antivirus, and operating system firewall software.

Consider the following services:

- Amazon Inspector is an automated vulnerability management service that continually scans AWS workloads for software vulnerabilities and unintended network exposure.
- Amazon GuardDuty Malware Protection is a continuous security monitoring service to analyze and process threats from multiple data sources. Use it to highlight activity that may indicate an instance compromise, such as cryptocurrency mining, denial of service activity, EC2 credential compromise, or data exfiltration using DNS.
- AWS Security Hub and AWS Config can be used for aggregation and assessment of operating system based alerts and configuration, along with other AWS services.

For more details, refer to the following information:

- Well-Architected Framework [Security]: Secure Operation
- Well-Architected Framework [Security]: Detection
- Well-Architected Framework [Security]: Protecting Compute

Best Practice 6.3 – Protect the database and the application

Security vigilance is imperative at the database and application layers, as a malicious actor gaining access at even a read-only level could compromise the security of critical business data. In all cases, follow the standard SAP best practices for database access protection and application security. These apply to both on-premises and cloud-based installations, and there are guidelines for each supported underlying database for SAP systems.

Suggestion 6.3.1 Follow SAP guidance on database security for your chosen database

Refer to the following for appropriate guidelines:

Database	Documentation
SAP HANA	• AWS Documentation: AWS SAP HANA Security • SAP Documentation: SAP HANA Security Guide • SAP Documentation: SAP HANA Administration Guide • SAP Note: 2159014 - FAQ: SAP HANA Security [Requires SAP Portal Access]
SAP ASE	SAP Documentation: Security Administration in SAP ASE
IBM Db2	(Consult SAP or Vendor documentation for guidance)
Oracle	SAP Documentation: SAP Database Guide - Oracle
Microsoft SQL Server	SAP Note: 3019299 - Security Audit Questions or Security Customization in NetWeaver and SQL Server systems [Requires SAP Portal Access]
SAP MaxDB	SAP Documentation: SAP MaxDB Security Guide

Suggestion 6.3.2 – Follow SAP guidance on application security

For SAP NetWeaver-based solutions, prescriptive guidance can be found in the SAP NetWeaver Security Guide.

- SAP Documentation: ABAP Platform Security Guide

Best Practice 6.4 – Establish a plan for upgrading and patching all applicable software

SAP and the vendors of the underlying operating systems and databases release standard security updates on a fixed schedule as well as provide emergency updates to fix vulnerabilities. Be aware of the latest security information from each vendor. We recommend that you keep your SAP application and all underlying components updated with the latest security fixes on a scheduled basis to avoid introducing security holes. We also recommend that you put a plan in place for applying emergency fixes when critical security patches are released.

Suggestion 6.4.1 - Subscribe to alerts from the vendors of operating system, database, and software solutions

Subscribing to your various vendor portals for security updates can help you become aware of new security issues and remediations as they are released. This can help you plan for required changes.

- AWS Documentation: AWS Security Bulletins
- SAP Documentation: SAP EarlyWatch Alert
- SAP Documentation: SAP Security News

Operating System	Guidance
SUSE Linux Enterprise Server	SUSE Update Advisories
Red Hat Enterprise Linux	Red Hat Security Advisories
Microsoft Windows	Microsoft Security Alerts
Oracle Enterprise Linux	Oracle Security Alerts

Suggestion 6.4.2 – Review the recommended changes and risk to your business and implementation effort

SAP teams must learn to balance the need for system uptime with the criticality of system changes that have been recommended to improve SAP security. Failure to do so can introduce unnecessary risks such as service interruptions, financial impact, or lost productivity. Review the recommended changes and implementation steps to fix vulnerabilities from your vendors and plan to implement them promptly. This directly relates to the Operational Excellence best practices discussed in this Lens, particularly the creation of runbooks for security.

- SAP Lens [Operational Excellence]: Suggestion 3.4.1 - Create specific runbooks for SAP security operations (p. 22)

Suggestion 6.4.3 – Establish a plan to address vulnerabilities in a timely manner

Applying new SAP security recommendations and security-related patches as quickly as possible is paramount both for AWS based SAP solutions and those installed elsewhere. Regularly review the SAP Security Notes and News, and create a process to remediate security issues quickly with the patches, notes, and recommendations found there. In some cases, SAP administrators may also have to put in temporary mitigation or control measures until the underlying vulnerability can be addressed. Also follow the Security Pillar recommendations around incident response.

- Well-Architected Framework [Security]: Incident Response
- SAP Documentation: SAP Security Notes and News

7 – Control access to your SAP workload through identity and permissions

How do you control access to your SAP workload? Use mechanisms provided by AWS, SAP, and other third parties to ensure that end users and interfacing systems are properly identified and authenticated. How are permissions controlled to ensure least privilege? How is access audited and reported on? Start by identifying your user categories and then systematically work through the controls and your identity management approach to limit access to your SAP workload.

ID	Priority	Best Practice
☐ BP 7.1	Required	Understand your SAP user categories and access mechanisms
☐ BP 7.2	Required	Manage privileged access for your SAP workload
☐ BP 7.3	Required	Understand your organization's identity management approach, and its application to SAP
☐ BP 7.4	Highly Recommended	Implement logging and reporting for user access and authorization changes and events

Best Practice 7.1 – Understand your SAP user categories and access mechanisms

The types of users accessing your SAP system will determine the security controls you need to apply. By examining each use case, you can develop a strategy. This should include how you manage identities, authentication, tooling and mechanisms to support those requirements.

Suggestion 7.1.1 Understand data access permissions and permitted actions

SAP systems often contain highly sensitive business data. As you define your user types, understand the data access permissions. (For example, an administrative database user does not have the fine-grained controls of an application user, and therefore may be more critical.) Also refer to [Security]: Best Practice 5.2 - Classify the data within your SAP workloads (p. 31).

Consider the following questions in relation to your SAP system access:

- Do the actions taken by an administrative or service user need to be traceable to a uniquely identifiable individual?
- At which layer of the application will the access be granted?
- Can you restrict access to a subset of functionality via permissions?
- Can you restrict access to a subset of functionality via other controls, for example exposing only certain services?
- Is there a requirement to audit the actions taken?

Suggestion 7.1.2 – Understand the network and/or location from which users will access the SAP systems

Network and/or location often contributes to the security risk profile and may determine whether the user is considered trusted or untrusted. Typically, this is coupled with the controls to prevent unauthorized access (refer to Best Practice 6.1 - Ensure that security and auditing are built into the SAP network design (p. 34)).

This can influence your design. For example, an untrusted internet user or device may require additional factors of authentication to access your SAP workload, when compared with a trusted user from your corporate network.

Best Practice 7.2 – Manage privileged access to your SAP workload

Adopt an approach of least privilege where possible. Only grant the minimum access required to perform a particular role to a minimum set of users, while managing usability and efficiency. There are

administrative accounts (for example, `<sid>adm`), which by default, have access to significantly impact the reliability or data security of your SAP workload. Consider how you can limit this risk.

Suggestion 7.2.1 – Manage AWS credentials and authentication

AWS Identity and Access Management (IAM) enables you to manage access to AWS services and resources securely. Using IAM, you can create and manage AWS users and groups for different SAP and cloud administration tasks. Use IAM permissions to allow and deny users access to AWS resources. Standard guidance should be followed, in particular restricting and securing root access.

- AWS Documentation: Security best practices in IAM

For access that is not assigned to a user but is required for the operation of the SAP application, pay particular attention to ensuring least privilege.

- AWS Documentation: Using an IAM role to grant permissions to applications running on Amazon EC2 instances

IAM Access Analyzer helps identify security risks associated with resources that are shared with an external entity, validates policies against IAM policy grammar and best practice, and can generate an IAM policy from the analysis of AWS CloudTrail logs. Consider its use as a mechanism for continuously reducing permissions based on user and role access patterns.

Suggestion 7.2.2 – Manage SAP Administrative credentials and authentication

Implement a process for approving and granting elevated permissions only when required, for a limited time-period. Use auditing functionality that addresses who and why the access was granted.

Restrict the use of username/password for privileged accounts. Disable direct access where possible. Store credentials securely, for example, in a privileged access management solution or password vault.

Evaluate how Systems Manager could be used to restrict direct operating system access for specific tasks using runbooks, RunCommand, and AWS Secrets Manager.

- AWS Documentation: Restricting access to root-level commands through SSM Agent
- AWS Documentation: Referencing AWS Secrets Manager secrets from Parameter Store

Best Practice 7.3 – Understand your organization's identity management approach, and its application to SAP

Typical SAP workloads will consist of multiple systems and therefore multiple identities. A centralized approach for managing these users can reduce the security risk and operational complexity.
Your focus should be on how to use AWS services and third-party tools in your approach to SAP security, considering such options as centralized user management, single sign-on, and multi-factor authentication.

Suggestion 7.3.1 – Determine an Identity Provider for named users

Users will be associated with an identity store, for example Active Directory. This acts as a central repository for managing identity information, such as roles, permissions, and identifiers. For each set of identities, determine if this can be associated with an Identity Provider. An identity provider enables you to off-load the authentication of users. It facilitates single sign-on (SSO) and also manages the user identity lifecycle (for example joiners, movers, leavers).

Consider exceptions for named users that are not associated with a human. This may include batch, job scheduling, integration, and monitoring users.

- AWS Documentation: AWS Directory Service | Amazon Web Services (AWS)
- AWS Documentation: AWS Identity Services

Suggestion 7.3.2 – Determine the authentication mechanisms

Understand the supported authentication mechanisms (for example, SAML, Kerberos, X.509, SAP Single Sign-On tickets) at each of the layers for your SAP workload. Evaluate the requirements to integrate with your application. Where possible use single sign-on to avoid the administrative and security impact of managing multiple user credentials.

- SAP Documentation: User Authentication and single sign-on
- AWS Documentation: Cloud applications - AWS IAM Identity Center (successor to AWS Single Sign-On)
- SAP on AWS Blog: Enable SAP Single Sign On with AWS IAM Identity Center (successor to AWS Single Sign-On) Part 1: Integrate SAP NetWeaver ABAP with IAM Identity Center
- SAP on AWS Blog: Enable SAP Single Sign On with AWS IAM Identity Center (successor to AWS Single Sign-On) Part 2: Integrate SAP NetWeaver Java
- SAP on AWS Blog: Enable Single Sign On for SAP Cloud Platform Foundry and SAP Cloud Platform Neo with IAM Identity Center

Suggestion 7.3.3 – Consider multi-factor authentication

Multi-Factor Authentication (MFA) is a best practice that adds an extra layer of protection on top of your logon credentials. These multiple factors provide increased security for your SAP application. Use cases include: access to SAP from an untrusted device; access to the AWS Management Console; and privileged activities such as deletion of backups or termination of EC2 instances.

- SAP on AWS Blog: Securing SAP Fiori with MFA
- AWS Documentation: Using MFA devices with your IAM sign-in page - AWS Identity and Access
- AWS Documentation: Configuring MFA delete -Amazon Simple Storage Service
- AWS Documentation: Amazon EC2: Requires MFA (GetSessionToken) for specific EC2 operations

Suggestion 7.3.4 – Determine the approach to certificate management

Client-based certificates can be used for authentication without the need for credentials. Determine an approach which includes time-based expiration for session management and certificate rotation for system to system communication. AWS provides a Certificate Authority (CA) that is trusted by SAP. Certificates can be issued and managed using AWS Certificate Manager (ACM).

- SAP Note: 2801396 - SAP Global Trust List [Requires SAP Portal Access]
- SAP Note: 3040959 - How to get a CA signed server certificate in ABAP [Requires SAP Portal Access]
- SAP Lens [Operational Excellence]: Suggestion 3.4.1 - Create specific runbooks for SAP security operations (p. 22)
- SAP Lens [Operational Excellence]: Suggestion 4.1.2 - Maintain a calendar for expiring of credentials, certificates and licenses (p. 25)

Best Practice 7.4 – Implement logging and reporting for user access and authorization changes and events

User access and authorization events in your SAP systems should be logged, analyzed, and audited regularly. Consolidate and correlate security events from your SAP applications and database with other components of your architecture. This can allow for end-to-end tracing in the event of a critical security problem or breach. Automate analysis of events in a central Security Information and Event Management

(SIEM) system. This can allow your operations team to understand if any unexpected or suspicious activity occurs outside of the bounds of normal system controls. They can then remediate as needed.

Suggestion 7.4.1 – Log AWS Identity and Access Management (IAM) events

Consider keeping a historical log of AWS IAM events. This can be used in detection or audit of user and authorization changes within AWS accounts. Determine your log retention period and types of events to log based on your organizations required security policies.

Enable your operations team to answer audit questions at the infrastructure level for your SAP system:

- When and by whom was the new AWS console/CLI user created?
- When and by whom was the AWS IAM role modified?
- When did the AWS user last successfully sign in?
- Is there a suspicious number of failed sign-in attempts to the AWS account?

For further information, consider the following:

- AWS Documentation: IAM Best Practices: Monitor activity in your AWS account
- AWS Documentation: Logging IAM and AWS STS API calls with AWS CloudTrail
- AWS Well-Architected Framework [Security]: Detection
- AWS Security Blog: Visualizing Amazon GuardDuty findings
- AWS Security Blog:Amazon GuardDuty Enhances Detection of EC2 Instance Credential Exfiltration

Suggestion 7.4.2 – Log user and authorization changes in your operating system

Consider keeping a historical log of operating system (OS) user and authorization events such that they can be used in detection or audit. Determine your log retention period and types of events to log based on your organizations required security policies.

Enable your operations team to answer audit questions at the operating system level for your SAP system such as:

- When and by whom was the new superuser OS account created?
- When and by whom was the OS account permissions modified?
- When did the OS user last successfully sign in?
- Is there a suspicious number of failed sign-in attempts for the OS account?
- When did your OS user last use elevated permissions?

For further information on auditing at the operating system consider:

Operating System	Guidance
SUSE Linux Enterprise Server	Setting Up the Linux Audit Framework | Security Guide
Red Hat Enterprise Linux	Chapter 14. Auditing the system Red Hat Enterprise Linux 8 | Security Guide
Microsoft Windows	Windows Audit Policy Recommendations
Oracle Enterprise Linux	Oracle Linux 8 Enhancing System Security - Using System Auditing and Monitoring

Suggestion 7.4.3 – Log SAP application and database user and authorization events

Consider keeping a historical log of SAP user and authorization events such that they can be used in detection or audit. Consider both the application stack (for example, ABAP authorizations) and your database (for example, SAP HANA). Determine your log retention period and types of events to log based on your organizations required security policies.

Enable your operations team to answer audit questions at the SAP application and database level for events such as:

- When and by whom was the new SAP or database account created?
- When and by whom was the SAP or database account permissions modified?
- When did the SAP or database user last successfully sign in?
- Is there a suspicious number of failed sign-in attempts for the account?
- What sensitive transaction codes or tools did the account last use?

For further information consider the following:

- SAP Documentation: SAP Access Control and Governance | User Access
- SAP Documentation: SAP NetWeaver ABAP: The Security Audit Log
- SAP Documentation: SAP NetWeaver JAVA: The Security Audit Log
- SAP Documentation: SAP HANA: Auditing Activity in SAP HANA

Suggestion 7.4.4 – Consolidate user and authorization events in a Security Information and Event Management (SIEM) system for analysis

Consider sending all your user and authorization events from across your SAP workload components into a central SIEM tool to allow correlation and analysis. Use tools like SAP Enterprise Threat Detection, third-party add-ons or directly ship your SAP audit logs from your application and database servers to an ingestion and analysis tool.

Establish baseline behaviors for your workload and monitor for abnormalities to improve detection of security incidents.

Consider AWS Marketplace SIEM solutions to monitor your workload in real-time, identify security issues, and expedite root-cause analysis and remediation.

For further information, consider the following resources:

- AWS Marketplace: SIEM Solutions
- AWS Documentation: AWS Security Hub
- SAP Documentation: SAP Enterprise Threat Detection
- Well-Architected Framework [Security]: Security Incident Response
- AWS Documentation: AWS Security Incident Response - Technical Whitepaper

8 – Protect your SAP data at rest and in transit

How do you protect your SAP data? SAP systems often run the core functions within a business and store sensitive enterprise data. Best practice is to encrypt data at rest and in transit using at least one encryption mechanism to meet internal or external security requirements and controls. In addition to the controls listed in the AWS Shared Responsibility Model, AWS provides multiple encryption solutions. Many AWS services have features which allow you to enable encryption with minimal effort

and performance impact. There are encryption options available for the database and SAP application layer that you can consider.

ID	Priority	Best Practice
☐ BP 8.1	Highly Recommended	Encrypt data at rest
☐ BP 8.2	Highly Recommended	Encrypt data in transit
☐ BP 8.3	Highly Recommended	Secure your data recovery mechanisms to protect against threats

Best Practice 8.1 – Encrypt data at rest

Data at rest refers to any data stored digitally. We use encryption to ensure that this data is only visible to authorized users and remains protected when access to the storage or database is compromised independently of the application.

Suggestion 8.1.1 – Define at which levels encryption will be applied

In general, the further up the stack you deploy encryption, the more secure your data is. This increased security is accompanied by additional complexity for deployment and management. AWS recommends using the encryption at rest options available within its services. Consider additional operating system or database encryption when required, as defined in [Security]: Best Practice 5.3 - Assess the need for specific security controls for your SAP workloads (p. 32).

Suggestion 8.1.2 – Understand AWS encryption options for SAP services and solutions

Many AWS services used by SAP support the encryption of data at rest. Refer to the following documentation for further details.

- AWS Documentation: Use encryption with EBS-backed AMIs
- AWS Documentation: Amazon EBS Encryption
- AWS Documentation: Amazon EFS encryption
- AWS Documentation: Amazon FSx encryption
- AWS Documentation: FSx for ONTAP encryption
- AWS Documentation: Amazon S3 Encryption

Data stored in these services can be encrypted at rest using either AWS or customer managed keys from AWS KMS.

Operating system encryption options include BitLocker, DM-crypt and SuSE Remote Disk.

The following links may assist with finding information about encryption options for your database:

Database	Guidance
SAP HANA	• SAP Documentation: Server-Side Data Encryption Services • SAP Documentation:HANA Local Secure Store (LSS)
SAP ASE	SAP Documentation: SAP ASE Overview of Encryption
IBM Db2	IBM Documentation: Db2 Encryption Overview

Database	Guidance
Oracle	SAP Note: 2591575 - Using Oracle Transparent Data Encryption (TDE) with SAP NetWeaver [Requires SAP Portal Access]
Microsoft SQL Server	SAP Note: 1380493 - SQL Server Transparent Data Encryption (TDE) [Requires SAP Portal Access]
SAP MaxDB	SAP Documentation: SAP MaxDB Database Administration - Encryption

Suggestion 8.1.3 – Define encryption methods and key management stores

Typically, key management is defined at the enterprise level and this will determine which key management options are permitted for use with your SAP workloads. AWS KMS is a secure and resilient service to simplify the management of encryption keys for AWS services. If you have a requirement to manage your own hardware security modules (HSMs), you can use AWS CloudHSM.

- AWS Documentation: AWS encryption tool and service options
- AWS Documentation: AWS Key Management Service (AWS KMS)
- AWS Documentation: AWS CloudHSM

Also consider mechanisms to protect master keys. How do you restrict access, manage rotation, and ensure recoverability of the keys?

Be aware that HANA data at rest encryption root keys can only be stored securely in the instance secure store in the file system (Instance SSFS) or within the SAP Data Custodian SaaS Solution. If using instance store the master key could be stored in AWS Secrets Manager with a rotation policy.

- SAP Note: 2154997 - Migration of hdbuserstore entries to ABAP SSFS [Requires SAP Portal Access]
- SAP Note: 2755815 - How to Ensure Recoverability of Hana's Data-At-Rest Encryption [Requires SAP Portal Access]

Best Practice 8.2 – Encrypt data in transit

Using encryption of data in transit makes it harder for your data to be intercepted, accessed, or tampered with while it's moving from one point to another. Ensure that there are secure protocols and network-level encryption in place to minimize potential threats and provide the level of protection aligned with your requirements.

Well-Architected Framework [Security]: Protecting Data in Transit

Suggestion 8.2.1 – Encrypt application traffic based on SAP and database protocols

For application traffic using SAP Protocols (SAPGUI Dialog, RFC, and CPIC) use SAP SNC to enforce Transport Layer Security.

- SAP Documentation: SNC-Protected Communication Paths in SAP Systems

For database traffic, use a secure connection between the client and database, where available.

Database	Guidance
SAP HANA	SAP Documentation: SAP HANA: Securing Data Communication

Database	Guidance
SAP ASE	SAP Documentation: SSL in SAP ASE
IBM Db2	SAP Note: 2385640 - DB6: database connection using SSL encryption [Requires SAP Portal Access]
Oracle	SAP Note: 973450 - Oracle Database network encryption and data integrity [Requires SAP Portal Access]
Microsoft SQL Server	SAP Note: 1570930 - SQL Server network encryption with SAP [Requires SAP Portal Access]
SAP MaxDB	SAP Documentation: MaxDB Network and Communication

Suggestion 8.2.2 – Encrypt SAP application traffic based on internet protocols

For application traffic based on internet protocols (HTTP, P4 (RMI), LDAP) use SSL/TLS to enforce Transport Layer Security.

- SAP Documentation: Transport Layer Security

Suggestion 8.2.3 – Encrypt data exchange based on file transfer or message transfer protocols

For file-based transfers, AWS provides AWS Transfer Family for secure file exchange over SFTP or FTPS. AWS Transfer Family supports the transfer of data to and from Amazon S3 and Amazon EFS.

- AWS Documentation: AWS Transfer Family

Using message-level data integrity checks helps ensure that data is not being tampered with while being transferred. Consider the use of one or more of the message level security standards supported by SAP to sign and verify the integrity of the data in messages.

- SAP Documentation: SAP ABAP Web Services Message-Level Security
- SAP Documentation: SAP NetWeaver Process Integration Security Guide
- SAP Documentation: SAP Cloud Integration Message-Level Security

For IDOC based messages use SNC to secure the RFC connection used by ALE.

- SAP Documentation: Handling Sensitive Data in IDocs

Suggestion 8.2.4 – Encrypt administrative access

It is common to use both Windows and SSH-based tools for the administration of SAP. In addition to security controls such as Bastian Hosts consider if it is possible to Encrypt this traffic.

Alternatively, AWS Systems Manager Session Manager provides a secure mechanism to access the operating system via the AWS Management Console using TLS for encryption.

- AWS Documentation: Amazon EC2 Windows Guide - Encryption in Transit
- AWS Documentation: Amazon EC2 Linux Guide - Encryption in Transit
- AWS Documentation: Data protection in AWS Systems Manager – Data Encryption

Suggestion 8.2.5 – Evaluate the features of AWS services that enable encryption in transit

In addition to application-based encryption, many AWS services provide encryption in transit capabilities. Evaluate your corporate standards, the implementation effort and associated benefits for each service. The following are some examples that are relevant for SAP workloads.

- AWS Documentation: Amazon S3 - Encryption in Transit - On by default and recommended for backups to Amazon S3.
- AWS Documentation: Amazon EFS - Encryption in Transit / Amazon FSx - May be required for shared filesystems.
- AWS Documentation: Elastic Load Balancing - Review your encryption requirements and whether end-to-end TLS with pass-through is required as this feature may not be available for all Load Balancer types.
- AWS Documentation: Amazon EC2 - Encryption in Transit - Only later generation instance types have this feature.

Suggestion 8.2.6 – Implement network level encryption

SAP customers will typically use either Direct Connect or a combination of Direct Connect and VPN, to provide reliable connectivity to their resources on AWS.

AWS Direct Connect does not encrypt your traffic in transit. If encryption is required, transport level encryption should be implemented, for example, using a VPN over Direct Connect.

AWS provides Site-to-Site VPN that can be used for network channel encryption. You can also choose to deploy third-party VPN solutions like OpenVPN from AWS Marketplace or with a bring your own license.

Alternatively, consider AWS PrivateLink for supported AWS services and solutions, including AWS Partners offering SaaS services. AWS PrivateLink provides private connectivity without exposing your traffic to the internet.

- AWS Documentation: AWS Managed VPN
- AWS Documentation: AWS Client VPN
- AWS Documentation: AWS Direct Connect + VPN
- AWS Documentation: Software Site-to-Site VPN
- AWS Documentation: AWS PrivateLink

Best Practice 8.3 – Secure your data recovery mechanisms to protect against threats

To help protect against malicious activities, follow the guidelines set out within your organization's security framework. Protecting against ransomware provides an overview of the key items to address before an incident and as part of an incident response including network controls, patching, and least privilege permissions. For SAP systems, the threat is similar to other applications, but the impact is potentially greater. If SAP is a system of record, or required for mission critical transactions, consider the following suggestions to secure a backup against a malicious attack.

- SAP Note: 2663467 - Tips to avoid a Ransomware situation [Requires SAP Portal Access]
- SAP Note: 2496239 - Ransomware / malware on Windows [Requires SAP Portal Access]

Suggestion 8.3.1 – Secure backups in a separate account with additional controls

By securing backups in an account that is isolated from the primary copy of your data, either directly or using replication, it's possible to minimize the risk of a compromised system also impacting your data recovery mechanisms.

The secondary account can be viewed as a "data bunker" with access requirements aligned to the use case.

For backups using Amazon S3, additional controls might include S3 Object Lock to store objects using a write-once-read-many (WORM) model or multi-factor authentication delete.

If using replication, understand the different options available, including delete marker replication (by default deletion markers are not replicated) and S3 Replication Time Control. To optimize costs, ensure that housekeeping is performed on both the primary and secondary buckets.

Consider AWS Backup Audit Manager to monitor and prove compliance for immutable backups across Regions and accounts.

Suggestion 8.3.2 – Validate your ability to recover

Backups are the last line of defense when protecting your data from malicious activities, but might prove worthless if recovery is not possible due to incomplete backups or backups that are not valid. Recovery might not be possible if you are unable to access or decrypt backups. Consider how you protect encryption keys and credentials.

Perform recovery tests aligned with a malicious scenario, including a rebuild in an alternate account.

- SAP Lens [Operational Excellence]: Best Practice 4.3 - Regularly test business continuity plans and fault recovery (p. 27)

9 – Implement a security strategy for logging, testing, and responding to security events

Do you have a strategic security plan that is supported by the appropriate logging, testing, and documented response methodology? Having a strategic security plan helps shape the proactive and reactive tasks that must be accomplished to ensure that all security challenges are met successfully. The procedures for logging, detection, and additional protection to help identify and remediate security incidents for SAP on AWS workloads are identical to those detailed in the Well-Architected Framework Security Pillar. Review the best practices regarding detection and incident response within the Security Pillar in addition to the guidance in this section.

ID	Priority	Best Practice
☐ BP 9.1	Required	Understand your security strategy for SAP application and database security event analysis
☐ BP 9.2	Highly Recommended	Perform periodic tests for security bugs
☐ BP 9.3	Highly Recommended	Have a documented plan for responding to security events

- Well-Architected Framework [Security]: Detection
- Well-Architected Framework [Security]: Incident Response

Best Practice 9.1 – Understand your strategy for SAP application and database security event analysis

Without keeping security logs at the appropriate levels of granularity, vital data required for incident response, forensic security analysis, and threat modeling can be lost. SAP security staff must be able

to evaluate potential security incidents affecting SAP systems in alignment with your business security requirements. For SAP workloads running on AWS, the AWS services described in the Well-Architected Framework Security Pillar are a helpful starting point in conjunction with the following suggestions.

- Well-Architected Framework [Security]: Detection – Configure

Suggestion 9.1.1 – Determine which logs are required to detect security events

For individual SAP software and supported databases refer to the SAP NetWeaver Guide Finder as well as the SAP NetWeaver Security Guide for what logs might be applicable (for example, read access logging). In addition, review the SAP advisory on security logging and related topics surrounding best practices for your development activities.

- SAP Documentation: SAP NetWeaver Guide Finder
- SAP Documentation: ABAP Platform Security Guide
- SAP Documentation: Security Logging

Suggestion 9.1.2 – Develop mechanisms for storing and analyzing logs

Having relevant data regarding potential security events is necessary for any secure SAP installation, but it is equally important to store that data securely and have the necessary tools for searching and analyzing the data in an efficient and timely manner. One option within AWS includes using the CloudWatch Agent to store instance logs and SAP application logs relevant to security in an Amazon CloudWatch log group. Such logs could also be exported to Amazon S3 for holistic log analysis and for integration with third-party log analytics solutions.

Refer to the following for help with assembling, combining, and analyzing your SAP on AWS security logs:

- SAP Lens [Security]: Suggestion 7.4.4 - Consolidate user and authorization events in a Security Information and Event Management (SIEM) system for analysis (p. 42)
- SAP on AWS Blog: Set up observability for SAP HANA databases with Amazon CloudWatch Application Insights
- SAP on AWS Blog: SAP HANA monitoring: A serverless approach using Amazon CloudWatch
- SAP on AWS Blog: SAP Monitoring: A serverless approach using Amazon CloudWatch

Suggestion 9.1.3 – Use machine learning to analyse and determine events of importance

Consider applying pattern recognition, anomaly detection, or both to security logs to assist in determining potential threats and events of importance to your SAP workload. AWS managed services, such as AWS Security Hub and Amazon GuardDuty, can help, combined with third-party security solutions from the AWS Marketplace.

- AWS Video: An Overview of AWS Security Hub
- AWS Documentation: Getting started with GuardDuty

Best Practice 9.2 – Perform periodic tests for security bugs

As described in the Well-Architected Framework Security Pillar incident response sections on simulations, assembling a runbook and conducting game days are recommended for all workloads, including those for SAP on AWS. This type of periodic testing can identify new attack vectors and vulnerabilities as well as prepare your SAP security resources for a rapid and effective response in the event of a security incident.

Well-Architected Framework [Security]: Incident Response – Simulation

Suggestion 9.2.1 – Include SAP applications as targets in addition to standard security and penetration testing

Probative security testing is an important part of maintaining a secure environment. In addition to conducting standard penetration testing in AWS, make sure to include your SAP solution as an additional potential target for malicious activities. Keep in mind SAP-specific software solutions that often are publicly exposed in your architecture such as SAProuter, Web Dispatcher, Cloud Connector, and SAP Fiori.

- AWS Documentation: Penetration Testing

Best Practice 9.3 – Have a documented plan for responding to security events

Without a documented plan for addressing a security event involving your SAP applications, the security team's response may be delayed, less comprehensive, and less effective both in mitigating the event and understanding its cause. Document security response patterns thoroughly for your SAP applications.

Suggestion 9.3.1 - Prepare for security events by having a documented incident management plan

This directly aligns with the AWS Well-Architected Framework Security Pillar guidance on incident response preparation. Refer to this documentation and be sure to include your SAP applications accordingly:

- Well-Architected Framework [Security]: Incident Response – Prepare

Reliability

The reliability pillar encompasses the ability of a workload to perform its intended function correctly and consistently when it's expected to. This includes the ability to operate and test the workload through its total lifecycle. For many businesses, being well-architected for reliability is a key requirement for SAP workloads. This is due to the mission critical nature of many SAP workloads and a need to understand the SAP architecture and the restrictions this imposes.

As with other pillars, we recommend reviewing the AWS Well-Architected Framework, particularly for the best practices of foundations, change management, and failure management. When considering reliability with the SAP Lens, focus first on having a clear and balanced understanding of your non-functional requirements across individual systems, including the business priorities that drive these requirements.

You should differentiate between how you achieve availability and reliability, and define your approach to disaster recovery. Availability can be defined as how you design the system so that users can continue to access it by providing resilience for single points of failure. Disaster recovery focuses on one-time recovery objectives where a decision to invoke a planned set of actions is made by the system owner.

10 – Design to withstand failure

How do you design your SAP workload to withstand failure? Work backwards from your business requirements to define an approach for meeting the availability goals of your SAP infrastructure and data. For each failure scenario, the resiliency requirements, acceptable data loss, and mean time to recover (MTTR) need to be proportionate to the criticality of the component and the business applications it supports. Select from one of the AWS architecture patterns provided for SAP availability, and evaluate it based on the criteria you define. These criteria should include the risk and impact for

each failure as well as taking into consideration cost and performance. In all cases, use initial and periodic testing to validate your decisions.

ID	Priority	Best Practice
☐ BP 10.1	Required	Agree on SAP workload availability goals that align with your business requirements
☐ BP 10.2	Highly Recommended	Select an SAP deployment pattern suitable for your availability and capacity requirements
☐ BP 10.3	Highly Recommended	Define an approach to help ensure availability of critical SAP data
☐ BP 10.4	Recommended	Validate the design against a set of criteria based on the business requirements

For more details, see the following information:

- AWS Documentation: Architecture Guidance for Availability and Reliability of SAP on AWS including Failure Scenarios and Architecture Patterns
- AWS Documentation: The Amazon Builders' Library: Static stability using Availability Zones
- AWS Documentation: AWS Direct Connect Resiliency Recommendations
- AWS Documentation: Disaster Recovery of Workloads on AWS: Recovery in the Cloud
- SAP Documentation: SAP HANA System Architecture Overview

Best Practice 10.1 – Agree on SAP workload availability goals that align with your business requirements

Understanding your availability goals is the first step to help ensure that you focus on the factors important to your organization. This helps you to define criteria that can be used to evaluate your architectural patterns.

Suggestion 10.1.1 – Identify SAP applications in scope and their interdependencies

Identify the SAP applications that you have deployed or will deploy in AWS. Understand any dependencies these applications have regardless of their location.

Suggestion 10.1.2 – Classify systems based on the impact of failure

There is no open standard for system classification aligned with planned availability and risk of failure. Defining systems using terms such as Mission Critical or Highly Important can help with defining patterns, identifying application grouping, and justifying costs. Production applications might be impacted differently by an outage. Factors to consider might include:

- Revenue generating or revenue reporting
- External or internal facing
- Core business vs. technical support
- Closely coupled vs. loosely coupled with other systems

Non-production environments can also play an important role in indirectly supporting the business. They should also be classified according to project phase and scale, taking into account transport paths (such

as business as usual and projects) and supporting role (such as development, unit test, production copy, and training).

Suggestion 10.1.3 – Assess the business impact of an outage

The impact should be measurable and take into consideration the duration of the outage. Examples of areas of impact include health and safety, financial, legal, regulatory, or brand.

Suggestion 10.1.4 – Understand compliance and regulatory requirements

Understand compliance or regulatory requirements for data residency and distance between locations to help ensure business continuity.

Suggestion 10.1.5 – Define minimum acceptable percentage uptime

For each system, or group of systems, agree and document an acceptable availability percentage which matches with business requirements. The following terms are used in this context

- MTTR – Mean Time to Recovery
- RTO – Recovery Time Objective
- RPO – Recovery Point Objective

A full explanation of the terms can be found in the Well-Architected Framework [Reliability]: Availability. Additional information on reliability in SAP can be found in the whitepaper:

- AWS Documentation: Architecture Guidance for Availability and Reliability of SAP on AWS

Best Practice 10.2 – Select an architecture suitable for your availability and capacity requirements

There are standard architectural patterns for SAP availability to suit the requirements of most customers deploying SAP on AWS. Use the following suggestions to determine what patterns best meet your availability and capacity requirements. Evaluate the risk and impact of each failure scenario against your business requirements.

Additional information on availability in SAP can be found in the whitepaper Architecture Guidance for Availability and Reliability of SAP on AWS.

Suggestion 10.2.1 – Identify all components and AWS services that are required for your SAP system

Identify all the required technical components of your SAP system, starting with the core (database, application servers, central services, global file systems) and extending to optional components (for example, Web Dispatchers, SAProuter, Cloud Connector). Determine the required AWS services to support these components and review the shared responsibility model for resiliency.

- AWS Documentation: Shared Responsibility Model for Resiliency

Suggestion 10.2.2 – Use SLAs, durability, availability, and historical data as a guide to the likelihood of failure

Likelihood of failure is subjective. Published service level agreements (SLAs) and past performance can only be used to guide the risk of potential future failure. However, the assumed frequency of various scenarios remains a useful data point. Something which is statistically likely to happen once a year might have a greater impact on design decisions than a failure that is yet to occur.

The following information can be used to better understand the services:

- AWS Health Dashboard provides alerts and remediation guidance when AWS is experiencing events that might impact you
- AWS Post-Event Summaries are provided for all major service events which impact AWS service availability
- Amazon Compute Service Level Agreement lists service level agreements (SLAs) for compute services
- AWS Documentation: Amazon EBS Durability and Availability
- AWS Documentation: Amazon EFS Data Protection and availability
- AWS Documentation: AWS Direct Connect Resiliency Recommendations

The likelihood of failure of other supporting services should also be evaluated including, but not limited to, domain name services, load balancers, and serverless functions.

More information can be found in the Architecture Guidance for Availability and Reliability of SAP on AWS whitepaper.

Suggestion 10.2.3 – Assess options for clustering, resilience, and load balancing

An SAP system can be distributed across multiple hosts, with differing mechanisms for ensuring availability. For example, a clustering solution can be used to protect single points of failure (for example, the SAP database and SAP Central Services). The SAP application tier can be scaled horizontally and load balancing can be used to make the web dispatcher highly available.

- AWS Documentation: SAP NetWeaver Deployment and Operations Guide for Windows - High Availability System Deployment
- AWS Documentation: SAP on AWS – IBM Db2 HADR with Pacemaker
- AWS Documentation: SQL Server Deployment for High Availability
- SAP Documentation: High Availability Partners

Suggestion 10.2.4 - Determine the availability of EC2 instance families within Availability Zones

Some Amazon EC2 instance families (for example, X and U) are not available across all AZs. Check with your AWS account team or AWS Support to confirm that the EC2 instance families you want to use are available in the intended Availability Zones. Note that the logical AZ identifiers might be different across different accounts. See the AWS documentation for more information.

- AWS Documentation: AZ IDs for your AWS resources

Suggestion 10.2.5 – Investigate strategies for ensuring capacity

The best way to help ensure capacity is to have a similarly sized instance available in case of failure. Other strategies include cloud native options (for example, On-Demand Instances, EC2 instance recovery) or re-allocating shared capacity.

We recommend that you make a capacity commitment in at least two AZs for Amazon EC2 instances that support SAP single points of failure so that the capacity is available at the time you need it.

Amazon EC2 capacity can be reserved using Zonal Reserved Instances or On-Demand Capacity Reservations. Both Zonal Reserved Instances and On-Demand Capacity Reservations can be shared between AWS accounts within the same AWS organization, which allows for the approach of using sacrificial capacity from another environment in the event of significant failure (for example, a complete AZ failure).

Further guidance on capacity reservations is available in:

- AWS Documentation: Architecture Guidance for Availability and Reliability of SAP on AWS

Suggestion 10.2.6 – Design your VPC across multiple Availability Zones

Design your VPC and subnets to ensure that instances can be provisioned in multiple Availability Zones, even if your initial design only relies on one or two AZs. This builds resilience into your design and helps ensure that connectivity and access to services can be confirmed in advance.

Best Practice 10.3 – Define an approach to help ensure the availability of critical SAP data

The business data for an SAP application is primarily stored within the database, but may also include file-based data or binaries (for example, executables, libraries, scripts, configuration, and interface files).

Suggestion 10.3.1 – Evaluate MTTR requirements and identify how they can be met

In [Reliability] Suggestion 10.1.5 – Define minimum acceptable percentage uptime (p. 52), you will have defined the MTTR requirements for each of your applications. Having assessed the risk of failures and the mechanisms for protecting system availability, confirm your requirements can be met, and document the expectations for MTTR against each failure scenario. If compromises need to be made for cost, complexity, or consistency, consult with the business owners to reach an agreement.

Suggestion 10.3.2 – Determine in which failure scenarios a recovery from backup would be necessary

Backup is often a secondary mechanism for ensuring or recovering availability, but most architectures will have some reliance on backups. The following are examples of failure scenarios that could be used to guide your analysis. The granularity of the scenarios, classification, and impact will vary depending on your requirements and architecture.

	Comparative Risk of Occurrence	Backup Required	Potential Data Loss	Estimated Recovery Time
Planned / Controlled Maintenance	Planned			
Resource exhausted or compromised (High CPU utilization / File system full / Out of memory / Storage issues)	Medium			
Distributed stateless component failure (for example, Web Dispatchers)	Medium			
Distributed stateful component failure (for example, application servers)	Medium			
Single Point of Failure (Database / SAP Central Services)	Medium			
AZ / Network Failure	Low			
Core service failure (DNS / Amazon EFS / API calls)	Low / Medium			
Corruption / Accidental deletion / Malicious activities / Faulty code deployment	Low			
Region failure	Very low			

Suggestion 10.3.3 – Determine where data replication is required

Data replication is used to improve reliability by having copies of the same data in multiple locations and is often a requirement for systems with a low RPO. When determining whether replication is required for availability or recovery, consider whether the service is Zonal (for example, Amazon EC2 and Amazon EBS and the databases they support) or Regional (for example, shared storage and Amazon S3).

Database replication

Database	Replication Technology	Guidance
SAP HANA	HANA System Replication	SAP Documentation: HANA System Replication
SAP ASE	SAP Replication Server	SAP Documentation: SAP Replication Server
Oracle	Oracle Data Guard	SAP Note: 105047 - Support for Oracle functions in the SAP environment [Requires SAP Portal Access]
Microsoft SQL Server	SQL Server Always ON	• SAP Documentation: Database High-Availability with SQL Server AlwaysOn • AWS Documentation: SQL Server Deployment for High Availability
SAP MaxDB	MaxDb Standby Database	SAP Note: 952783 - FAQ: SAP MaxDB high availability [Requires SAP Portal Access]
IBM Db2	HADR	SAP Note: 1612105 - DB6: FAQ on Db2 High Availability Disaster Recovery (HADR) [Requires SAP Portal Access]

AWS service replication options

Service	Operating level	Replication options available	Guidance
Amazon EFS	File system	Continuous asynchronous replication within a Region and cross Region	Amazon EFS Replication
Amazon FSx for Windows File Server	File system	Scheduled asynchronous replication within a Region and cross Region using AWS DataSync	Scheduled replication using AWS DataSync
Amazon FSx for NetApp ONTAP	File system	Scheduled asynchronous replication within a region and cross region via NetApp SnapMirror	Scheduled replication using NetApp SnapMirror
Amazon S3	S3 bucket	Continuous asynchronous replication within a Region and cross Region	Amazon S3 Replicating objects

Service	Operating level	Replication options available	Guidance
AWS Elastic Disaster Recovery	EC2 instance	Continuous asynchronous replication within a Region and cross Region	AWS Elastic Disaster Recovery

Suggestion 10.3.4 – Build a strategy to ensure consistent configuration data and binaries

It is important to have consistent configuration data and binaries to help ensure predictable behavior and a tested setup following a failure. This can include operating system packages, application parameters, and cluster configuration. Determine how you could ensure alignment across all instances for an application, including those which are there for resilience (for example, additional application servers, secondary database nodes).

Amazon EFS, Amazon FSx, and Amazon S3 provide a durable location for shared binaries or configuration that can be managed centrally.

Refer to [Operational Excellence] Best Practice 2.1 - Use version control and configuration management (p. 16) pillar for mechanisms to control versions and manage configuration.

Suggestion 10.3.5 – Have a holistic approach to data consistency

The approach to ensuring the consistency of critical SAP data should not only focus on a single set of data but also should consider the dependencies within and between datasets and systems. For example, if you need to recover an SAP BW system, but not the source systems it pulls from, what would be the impact on change pointers and what mechanisms are in place to ensure a consistent recovery?

Suggestion 10.3.6 – Build a strategy for interfaces that permits data to be replayed or re-sent

For data exchange between systems, determine whether the integration is loosely coupled and if data can be replayed or re-sent, either at the source or target. Review if there are queuing capabilities to allow the scenario to be suspended or cached during an outage.

Suggestion 10.3.7 – Evaluate the use of a data bunker

Failure scenarios that result in the online data becoming unusable or unavailable due to situations such as accidental deletion or a malicious act might require a different approach to help ensure that data is protected or recoverable.

Although prevention is the best defense through a security framework covering network isolation and access control, the impact should be considered in the context of recovery and resilience.

Using a *write only* backup account with a reduced retention period is a common approach for this rare but potentially high impact scenario.

- SAP Lens [Security]: Best Practice 8.3 - Secure your data recovery mechanisms to protect against threats (p. 48)

Best Practice 10.4 – Validate the design against a set of criteria based on your business requirements

Establish a set of criteria based on your business requirements, balancing the risk of failure, impact on the business, and acceptable trade-offs. Use these criteria to validate the design and make adjustments where necessary.

Suggestion 10.4.1 – Assess the cost to your business of an outage

Failures, of either AWS services or SAP components, will impact your SAP system differently depending on the resilience and recovery strategies. The type of failure will determine the duration of the outage (RTO) and the potential data loss (RPO).

For each failure, assess the risk of an outage and the cost to your business. For example, are there revenue generating processes that will be impacted and what is the hourly cost associated with the system not being available?

Suggestion 10.4.2 – Assess the cost of your architecture

In SAP Landscapes, the largest elements of the AWS monthly bill typically are for Amazon EC2 and storage-related services. Understand the cost implications so that you select the best architecture to meet your reliability requirements. Key contributors include:

- Deployment patterns that don't maximize hardware utilization
- Redundant copies of data
- Operating system license costs
- Clustering software license costs
- Costs associated with maintenance, testing, and skilled resources

Refer to [Cost Optimization]: Cost optimization Best Practices (p. 87) for further details.

Suggestion 10.4.3 – Evaluate your design against other pillars in the framework

Reliability cannot be designed in isolation, but should be assessed against the rest of the pillars of the AWS Well-Architected Framework. Example questions you might ask to evaluate this include:

- Operational excellence — Do you have the experience and skills to manage the solution?
- Security — Is your data protected during replication, recovery, etc.
- Performance — Does replication or the backup activity impact user performance?
- Cost optimization — Does the cost of the solution align with the assumed risk?
- Sustainability — Does the solution align with your sustainability and environmental impact initiatives?

11 – Detect and react to failures

How do you detect and react to failures impacting your SAP workload? Design how software or operating procedures can help ensure the health and resilience of your SAP workload. Monitor for potential and actual failures, focusing on prevention where possible. Consider whether a component is distributed or is a single point of failure and design a resiliency solution that minimizes the impact to your workload. In addition to testing periodically to understand your risk profile, examine how automation could improve your resilience.

ID	Priority	Best Practice
☐ BP 11.1	Required	Monitor failures of SAP applications, AWS resources, and connectivity
☐ BP 11.2	Highly Recommended	Define an approach to maintain availability
☐ BP 11.3	Highly Recommended	Define an approach to restore service availability
☐ BP 11.4	Highly Recommended	Conduct periodic tests of resilience

ID	Priority	Best Practice
☐ BP 11.5	Recommended	Automate reaction to failures

For more details, refer to the following:

- AWS Documentation: Architecture Guidance for Availability and Reliability of SAP on AWS including Failure Scenarios and Architecture Patterns

Best Practice 11.1 – Monitor failures of the SAP application, AWS resources, and connectivity

Monitoring for failures of the SAP application, AWS resources, and connectivity helps you to react to failures or potential failures in a timely manner.

Suggestion 11.1.1 – Use AWS Personal Health Dashboard and notifications

The AWS Health Dashboard gives you a personalized view of the status of the AWS services that power your applications, enabling you to quickly see when there are issues impacting your SAP workload. For example, in the event of a lost Amazon Elastic Block Store (Amazon EBS) volume associated with one of your Amazon EC2 instances.

The dashboard also provides forward looking notifications, and you can set up alerts across multiple channels, including email, so that you receive timely and relevant information to help plan for scheduled changes. For example, in the event of AWS hardware maintenance activities that impact one of your Amazon EC2 instances, you would receive a notification with information to help you plan for and proactively address any issues associated with the upcoming change.

Suggestion 11.1.2 – Evaluate AWS services to understand the health of your SAP system

AWS provides a number of management and governance services that you should evaluate, including Amazon CloudWatch and Amazon CloudWatch Application Insights for SAP. Focus on the metrics that indicate a failure or potential failure, such as EC2 instance failure, high CPU utilization, and file system utilization.

Refer to the Operational Excellence pillar for more details:

- SAP Lens [Operational Excellence]: Best Practice 1.1 - Implement prerequisites for monitoring SAP on AWS (p. 8)
- SAP Lens [Operational Excellence]: Best Practice 1.4 - Implement workload configuration monitoring (p. 12)

Suggestion 11.1.3 – Evaluate the capability of SAP tools to monitor failures

Tools from SAP, such as Solution Manager and Landscape Manager, allow you to view any monitoring data in the context of the application. The following monitoring solutions are available from SAP. Review any additional licensing costs as part of the evaluation of these tools.

- SAP Documentation: SAP Focused run
- SAP Documentation: SAP Solution Manager
- SAP Documentation: SAP Landscape Manager (LaMa)
- SAP Note: 2574820 - SAP Landscape Management Cloud Manager for Amazon Web Services (AWS) [Requires SAP Portal Access]

Suggestion 11.1.4 – Evaluate third-party tools for AWS and SAP monitoring

The following monitoring solutions are available from the AWS Marketplace. You should evaluate these and other third-party tools.

- AWS Documentation: Monitoring Solutions in AWS Marketplace

Best Practice 11.2 – Define an approach to maintain availability

Maintain availability by having a resilient architecture that can sustain the failure of a single technical component or AWS service. Implement mechanisms, which could include redundant capacity, load balancing, and software clusters.

Suggestion 11.2.1 – Avoid failures due to exhausted resources or service deterioration

Investigate over-provisioning of resources, proactive monitoring of growth, and throttling usage by setting limits.

The operational excellence pillar covers the different ways in which you can understand the state of your SAP application and ensure that the appropriate actions are taken, see [Operational Excellence]: 1 - Design SAP workload to allow understanding and reaction to its state (p. 7).

The performance pillar can assist with guidance on right-sizing and scaling capacity [Performance]: 16 - Understand ongoing performance and optimization options (p. 83).

Suggestion 11.2.2 – Have a strategy for scheduled maintenance

If your business has a requirement to minimize scheduled outages, you should develop a strategy for maintenance at all levels – SAP application, database, operating system, and AWS. Consider the following:

- Use of replication and cluster solutions to alternate the primary and secondary node.
- Excess capacity and mechanisms to scale up and down to facilitate rolling outages.
- Use of a live patching approach for the operating system, if possible.
  - SUSE Linux Enterprise Live Patching
  - Red Hat Reducing downtime for SAP HANA Whitepaper
- AWS Documentation: AWS Systems Manager Patch Manager Patch Groups
- SAP Note: 1913302 - HANA: Suspend DB connections for short maintenance tasks [Requires SAP Portal Access]
- SAP Note: 2077934 - Rolling kernel switch in HA environments [Requires SAP Portal Access]
- SAP Note: 953653 - Rolling Kernel Switch [Requires SAP Portal Access]
- SAP Note: 2254173 - Linux: Rolling Kernel Switch in Pacemaker-based NetWeaver HA environments [Requires SAP Portal Access]

You should also evaluate the elastic capabilities of AWS services to reduce the overall downtime of scheduled maintenance by temporarily increasing performance. For example, scaling up the size of the Amazon EC2 instance running your database to provide more CPU and storage throughput for upgrade activities, or switching your EBS volumes type from `gp2` to `io2` to improve storage throughput during a database reorganization.

Suggestion 11.2.3 – Protect SAP single points of failure with software clusters or other mechanisms

You can use a high availability (HA) clustering solution for autonomous failover of SAP single points of failure (SAP Central Services and database) across Availability Zones.

There are multiple SAP-certified clustering solutions listed on the SAP website. SAP clustering solutions are supported by the cluster software vendors themselves, not by SAP. SAP only certifies the solution. Any custom-built solution is not certified and will need to be supported by the solution builder.

If you choose not to use a clustering solution for your single points of failure, consider scripting or runbooks to minimize the errors associated with restoring services.

Suggestion 11.2.4 – Consider redundant capacity or automatic scaling for components that support it

Evaluate static, dynamic, or scheduled capacity changes to match your usage. Examine the minimum capacity requirements and how they would be impacted by failures and maintenance. Overprovision where appropriate to allow time to recover from failure.

If you need to maintain 100% capacity in the event of an AZ failure, then you should consider deploying the application tier across three AZs, each with 50% of the total required capacity.

In addition to deploying the SAP Application Server Layer across multiple AZs, you could consider scaling solutions such as the one described in the following SAP on AWS Blog post that leverages the capabilities of Amazon EC2 Auto Scaling.

- SAP on AWS Blog: Using AWS to enable SAP Application Auto Scaling
- AWS Documentation: Amazon EC2 Instance Types for SAP
- SAP Note: 1656099 - SAP Applications on AWS: Supported DB/OS and Amazon EC2 products [Requires SAP Portal Access]

Suggestion 11.2.5 – Ensure the availability of capacity for all identified failure scenarios

The following are examples of failure scenarios that could be used to guide your analysis. Granularity and coverage of the scenarios, classification, and impact will vary depending on your requirements and architecture.

Failure scenario examples	Comparative Risk of Occurrence
Planned / Controlled Maintenance	Planned
Resource exhausted or compromised (High CPU utilization / File system full / Out of memory / Storage issues)	Medium
Distributed stateless component failure (for example, web dispatchers)	Medium
Distributed stateful component failure (for example, application servers)	Medium
Single point of failure (Database / SAP Central Services)	Medium
AZ / Network failure	Low
Core service failure (DNS / Amazon EFS / API calls)	Low / Medium
Corruption / Accidental deletion / Malicious activities / Faulty code deployment	Low
Region failure	Very Low

Further guidance on capacity reservations is available in [Reliability]: Suggestion 10.2.5 - Investigate strategies for ensuring capacity (p. 53) and in the AWS whitepaper: Architecture Guidance for Availability and Reliability of SAP on AWS.

You can review what Reserved Instances you have available within your AWS account using the Reserved Instances section of the Amazon EC2 console. You can review what On-Demand Capacity Reservations you have available using the Capacity Reservations section of the Amazon EC2 console.

Suggestion 11.2.6 – Use AWS services that have inherent availability where applicable

Several AWS services have inherent availability as part of their design and run across multiple Availability Zones for high availability. Some of the relevant services used in an SAP context include:

- AWS Service: Amazon EFS
- AWS Service: Elastic Load Balancing
- AWS Service: Route 53
- AWS Service: AWS Transit Gateway
- AWS Service: Amazon S3
- AWS Service: Amazon FSx

In addition, components that use stateless services, such as bastian hosts or SAProuter, can use Auto Scaling Groups to achieve high availability.

Suggestion 11.2.7 -– Follow AWS best practices to ensure network connectivity

Evaluate one or more of the following AWS best practices to ensure the resilience of network connectivity to the AWS Region in use:

- AWS Documentation: AWS Direct Connect Resiliency Toolkit
- AWS Documentation: AWS VPN CloudHub
- AWS Documentation: AWS Cloud WAN

If your cluster solution relies on an overlay IP consider the following to enable access from outside of the VPC:

- AWS Documentation: SAP on AWS High Availability with Overlay IP Address Routing

Best Practice 11.3 – Define an approach to restore service availability

Restoring availability assumes that for a particular failure scenario, some loss of service will occur. The restore approach should examine the amount of time needed to restore service, and the actions required to meet the availability goal.

Suggestion 11.3.1 – Enable instance recovery for EC2 instances

AWS provides two modes of instance recovery: simplified (on by default) and Amazon CloudWatch action-based (configurable). Both modes monitor an Amazon EC2 instance and automatically recover the instance if it becomes impaired due to an underlying hardware failure. This feature can remove the need for manual intervention, but startup, application restart, and load times should be factored into the recovery time objective (RTO).

CloudWatch action-based alarms are customizable, which can help you to control the recovery time of an instance for standalone instances.

If you intend to use a clustering solution to protect against hardware failure, you should evaluate if instance recovery is compatible with the cluster solution.

- AWS Documentation: Amazon EC2 Instance Recovery
- SAP on AWS Documentation: Technical requirements for high availability clusters

Suggestion 11.3.2 – Have a strategy to rebuild EC2 instances using AMIs and infrastructure as code

The benefit of infrastructure as code (IaC) is the ability to build and tear down entire environments programmatically. If architected for resiliency, an environment can be implemented in minutes using AWS CloudFormation templates or AWS Systems Manager automation. Automation is critical for maintaining high availability and fast recovery.

You should evaluate the following AWS services as part of your strategy:

- AWS Service: EC2 Image Builder
- AWS Service: AWS Launch Wizard for SAP
- AWS Service: AWS Cloud Development Kit (AWS CDK)
- SAP on AWS Blog: DevOps for SAP

Suggestion 11.3.3 – Understand Amazon EBS failures

Failure of one or more EBS volumes could impact the availability and durability of your SAP workload. Therefore, you should understand the Amazon EBS failure rates, notification mechanisms, and recovery options.

- AWS Documentation: Amazon EBS Durability
- AWS Documentation: Monitor the status of your volumes
- AWS Service: AWS Health Dashboard
- AWS Documentation: Volume recovery using Amazon EBS Snapshots

Suggestion 11.3.4 – Have a strategy for reacting to AWS Personal Health Dashboard notifications

You should have a strategy for receiving and actioning notifications from your AWS Personal Health Dashboard. This could include using CloudWatch to start Amazon SNS or integration with your ITSM tools via the AWS Health API.

Suggestion 11.3.5 – Ensure that you are protected against accidental or malicious events impacting availability

You should consider the following approaches for ensuring that you are protected against accidental or malicious events that could impact the availability of your SAP workload.

- Implement a principle of least privilege and enforce separation of duties within AWS Identity and Access Management.
- Follow the guidance in AWS Knowledge Center article: How do I protect my data against accidental EC2 instance termination?
- Follow the Best practices for Amazon EC2.
- You should also follow the security guidance in [Security]: Best Practice 8.3 - Secure your data recovery mechanisms to protect against threats. (p. 48)

Suggestion 11.3.6 – Identify dependencies beyond the SAP workload in AWS

Understand the underlying dependencies for your SAP business processes, including shared services and supporting components or systems. Some examples include Active Directory, DNS, identity providers, SaaS services, and on-premises systems. Assess the impact of failure and the required mitigations.

Best Practice 11.4 – Conduct periodic tests of resilience

Periodically test resilience against critical failure scenarios to prove that software and procedures result in a predictable outcome. Evaluate any changes to architecture, software, or support personnel to determine if additional testing is necessary.

Suggestion 11.4.1 – Define the in-scope critical failure scenarios based on your business requirements

You should define which critical failure scenarios you are able to test, aligned with your business requirements. The following are examples of failure scenarios which could be used to guide your analysis. Granularity and coverage of the scenarios, classification and impact will vary depending on your requirements and architecture.

Failure Scenario Examples	Comparative Risk of Occurrence
Planned / Controlled Maintenance	Planned
Resource exhausted or compromised (High CPU utilization / File system full / Out of memory / Storage issues)	Medium
Distributed stateless component failure (for example, web dispatchers)	Medium
Distributed stateful component failure (for example, application servers)	Medium
Single point of failure (Database / SAP Central Services)	Medium
AZ / Network failure	Low
Core service failure (DNS / Amazon EFS / API calls)	Low / Medium
Corruption / Accidental deletion / Malicious activities / Faulty code deployment	Low
Region failure	Very Low

Suggestion 11.4.2 – Define a set of test cases to simulate critical failures

You should have a complete set of tests defined to simulate the critical failure scenarios that would impact your SAP workload.

You should be aware that for some failure scenarios a simulation might not fully represent the actual failure that would occur. For example, to simulate a hardware issue, you cannot cause a failure of an EC2 instance, but for Nitro-based instances you can generate a kernel panic to cause the instance to reboot.

In addition, AWS Fault Injection Simulation is designed to help simulate failures within your AWS resources.

- AWS Documentation: High Availability Configuration Guide for SAP on HANA
- AWS Documentation: Send a diagnostic interrupt

Suggestion 11.4.3 – Define the expected behavior for each test case

You should have a documented set of expected outcomes to baseline your testing.

Suggestion 11.4.4 – Define an approach for evaluating the impact of a change and the subsequent testing required

You should have an approach defined to evaluate the impact of a change on your environment and the testing required as part of that change to help ensure that it does not invalidate your approach to availability and reliability. Examples of these types of changes include software upgrades, patches, and parameter changes.

Suggestion 11.4.5 – Define a test schedule

Ensure that you have a test schedule that covers the initial implementation, testing of changes, and periodic validation of your environment.

Suggestion 11.4.6 – Review the testing outcomes

Based on the test outcomes, identify any improvements to the test cases, configuration or architecture.

Suggestion 11.4.7 – Define the required activities to return to a pre-test state

As part of each test, you should define the required activities to return to the pre-test state. This is to ensure that each test case is isolated from other tests and that the testing does not impact the availability and reliability of a production system.

Best Practice 11.5 – Automate reaction to failure

You can minimize the impact to service by automating the response to failure. Design automation to respond to failure, impaired capacity, or loss of connectivity. Ensure clear arbitration criteria are defined to avoid false positives.

Suggestion 11.5.1 – Evaluate your automation for application awareness

For automation solutions that protect an application, evaluate the impact on state – for example, connected user sessions, logon targets, data replication consistency, and data corruption risk.

Suggestion 11.5.2 – Evaluate the health check mechanisms that initiate automation

Health checks should be designed with controls to help ensure that automations are not started because of false positives.

Where possible, rely on the data plane over the control plane for resilience. The control plane is used to configure resources, and the data plane delivers services. Data planes typically have higher availability design goals than control planes and are usually less complex.

- AWS Documentation: Static stability using Availability Zones

12 – Plan for data recovery

How do you plan for logical, data-related recovery for your SAP workload? Work backwards from the business requirements to define an approach to recover or reconstruct your business data. Depending on how you have architected for resilience, different scenarios might fit in this category. At a minimum, your backup or disaster recovery (DR) posture should protect you from accidental deletion, logical data corruption, and malware. Be deliberate about the decision to restore, taking into account the time to return to service and the dependencies between systems.

ID	Priority	Best Practice
☐ BP 12.1	Required	Establish a method for consistent recovery of business data

ID	Priority	Best Practice
☐ BP 12.2	Highly Recommended	Establish a method for recovering configuration data
☐ BP 12.3	Highly Recommended	Define a recovery approach for your complete SAP estate
☐ BP 12.4	Recommended	Conduct periodic tests to validate your recovery procedure

Best Practice 12.1 – Establish a method for consistent recovery of business data

Define data recovery plans that can help ensure business data consistency for an individual system in the event of data loss or corruption.

Suggestion 12.1.1 - Ensure that database backups are consistent by using backup mechanisms that are aware of database state

SAP provides mechanisms for integrating with a database vendor's backup capability (for example, brtools) and providing visibility within SAP transactions or management consoles. In addition, there are options to integrate with third-party backup providers or storage solutions including AWS Backint Agent for SAP HANA. These supported options have awareness of database state, continuously capturing changes or quiescing the database (pausing or reducing activity) while a consistent copy is taken, for example using storage snapshots.

Review the SAP guides for individual database vendors as well as AWS documentation:

- AWS Documentation: AWS Backint Agent for SAP HANA
- SAP Documentation: Guide Finder for SAP NetWeaver and ABAP Platform
- SAP on AWS Blog: How to back up Microsoft SQL Server databases for SAP with VSS Snapshots
- AWS Blog: Taking crash-consistent snapshots across multiple Amazon EBS volumes on an Amazon EC2 instance

Suggestion 12.1.2 – Evaluate the durability and recoverability of file-based data critical to your business

Business data that is not stored within a database might require a separate backup strategy.

In a standard SAP NetWeaver system, this often includes file-based interface files, SAP transport directory contents, and logs including batch logs, job logs, and work process directory logs. Non-SAP NetWeaver and supporting systems, such as document management solutions, might have other file-based business data which should be evaluated. Evaluate Amazon EFS or Amazon FSx to increase availability and durability of such file systems.

File system backups can be performed using snapshots, AWS Backup, or third-party backup solutions.

Business data should be evaluated independently from binaries and configuration data, which might be able to be re-provisioned via SAP download, re-install, or infrastructure as code.

- SAP Lens [Operational Excellence]: Suggestion 12.2.1 - Define infrastructure as code approach to the creation and change of configuration (p. 67)
- SAP Lens [Operational Excellence]: Suggestion 12.2.2 - Define an approach for backups of file system contents, including the root volume (p. 67)

Suggestion 12.1.3 – Evaluate the durability and location of database backups and logs

Backups and logs contain a record of your live data, but can themselves be susceptible to failure. Consider how you minimize the impact of a failure by evaluating the location of your backups in relation to your active data copies. It's important to consider the following:

- The time it takes to secure the backups – impacting your recovery point
- The time it takes to retrieve/recover the backups – impacting your recovery time

Additional information:

- AWS Documentation: AWS Backint Agent for HANA
- AWS Documentation: FSR (Fast Snapshot Restores)
- AWS Documentation: Amazon S3 Replication options

Suggestion 12.1.4 – Evaluate your requirements for a point-in-time recovery

If you have a requirement to recover to any particular point in time, does your backup design allow for this? Is the backup method database-aware and can you roll your database forward to a consistent recovery point? Have you considered any file-based recovery required for consistency?

Consider the following:

- The log interval and how quickly logs are secured
- Incremental or differential backups to improve the recovery time
- If a backup catalog or other mechanism is required
- Is it possible to use database or storage options to go back in time?

Suggestion 12.1.5 – Review mechanisms for recovery caused by data loss

Determine the implications of recovering from a significant data loss situation, such as data corruption, deletion, or a faulty code deployment that is unable to be reverted. Evaluate the propagation of data loss when using database or storage-based replication, and the RTO and RPO impact of using a secondary restore mechanism, such as backups.

Suggestion 12.1.6 – Create a data bunker

Following the guidance in Suggestion 10.3.7 - Determine in which failure scenarios a recovery from backup would be necessary (p. 55), create a data bunker to secure your backups from accidental deletion or malicious activities.

Best Practice 12.2 – Establish a method for recovering configuration data

A number of different types of data, which are required to run an SAP workload, do not reside in the SAP database. This includes operating system configuration, metadata to recreate the required AWS resources, and data required by the SAP applications stored within a file system. Define a process for recovering or recreating this data in the event of data loss.

Suggestion 12.2.1 – Define infrastructure as code approach to the creation and change of configuration

Manual changes made directly to individual instances can quickly lead to inconsistencies in configuration between systems and a reliance on backups to recover state. By using infrastructure as code, you can deploy your SAP systems and implement changes in the same manner that you would manage

application code. DevOps mechanisms, such as a code pipeline, can provide additional control and testing to help ensure consistency and repeatability within your landscape.

You should evaluate the following AWS services as part of your approach:

- AWS Service: AWS Launch Wizard for SAP
- AWS Service: EC2 Image Builder
- AWS Service: AWS Cloud Development Kit (AWS CDK)
- SAP on AWS Blog: DevOps for SAP
- AWS Documentation: Introduction to DevOps on AWS
- AWS Documentation: Launch AWS Service Catalog products created with AWS Launch Wizard

Suggestion 12.2.2 – Define an approach for backups of file system contents, including the root volume

Operating system packages and configuration, application binaries, and file system contents are integral to running an SAP system, but are not part of the core SAP database backup. Evaluate mechanisms to secure and restore this data, including Amazon Machine Images (AMIs), EBS volume snapshots, and other backup options.

The frequency and alignment of AMIs, snapshots, and file system copies should be considered, as well as the granularity for recovery and the time taken.

In certain scenarios, using infrastructure as code might reduce backup requirements for non-business data by focusing on recreation versus restoration.

- SAP Documentation: Required File Systems and Directories
- AWS Documentation: Designing a backup and recovery solution

Suggestion 12.2.3 – Document any manual settings

Any manual activities which are not contained in the database, deployable by code, or able to be restored using volume backups, should be recorded to ensure an SAP system can be recreated in the worst case scenario.

Best Practice 12.3 - Define a recovery approach for your complete SAP estate

If your SAP estate consists of multiple SAP systems, you need to create a detailed approach that defines the order in which each system is recovered, based on business priorities. Evaluate how data loss might impact consistency across systems and business operations.

Suggestion 12.3.1 – Create a business continuity plan that includes restore priority and plans to ensure consistency

Have a business continuity plan (BCP) that determines the priority to restore each SAP system based on the classification of systems determined in [Reliability]: Suggestion 10.1.2 – Classify systems based on the impact of failure (p. 52). The plan should also consider the impact of cross system consistency requirements as well as the use of multi-tenant databases on the restore priority.

Suggestion 12.3.2 – Evaluate any dependencies on shared services

As you define your recovery approach, consider which shared services are either part of the foundation for running your SAP workload (for example, DNS, Active Directory) or required to perform the restore

itself (for example, backup tools). Evaluate risks and restore prerequisites associated with these dependencies.

Suggestion 12.3.3 – Create runbooks to be followed in a disaster

A predefined runbook ensures that a proven set of steps is followed in the event of a disaster, reducing the risk or critical activities being missed.

Best Practice 12.4 – Conduct periodic tests to validate your recovery procedure

Periodically test recovery from critical failure scenarios to prove that software and procedures result in a predictable outcome, and to validate the state and health of the backup files. You should evaluate any change to architecture, software, or support personnel to determine if additional testing is necessary.

Suggestion 12.4.1 – Identify the failure scenarios for recovery testing

You should define the failure scenarios in which a recovery would be required based on [Reliability]: Suggestion 10.3.2 – Determine in which failure scenarios a recovery from backup would be necessary (p. 55) and decide the appropriate level of testing required to validate the process and tooling.

Suggestion 12.4.2 – Determine the impact of a system change on your recovery approach

Define an approach for evaluating the impact of a change and the subsequent recovery testing required to ensure it does not invalidate your approach. Examples of the types of change that could impact your workload recovery include software upgrades, patches and parameter changes.

A recovery test should also be planned in the event of a significant change in the operating model used to support your SAP environments, for example, a change in Managed Service Partner or key personnel.

Suggestion 12.4.3 – Define a recovery test plan

You should have a complete set of tests defined to simulate the critical failure scenarios that would result in the need for a recovery. Recovery testing should be planned for during initial implementation and subsequently on a periodic basis or when required.

- SAP Lens [Operational Excellence]: Best Practice 4.3 - Regularly test business continuity plans and fault recovery (p. 27).

Performance efficiency

The performance efficiency pillar focuses on the allocation of AWS resources to meet the requirements of SAP workloads supporting your business. Performance optimization should be a data driven process of monitoring and measuring performance, and adjusting allocated resources to your requirements in order to maintain efficiency as demand changes and technologies evolve.

13 – Select the optimal compute solution

How do you select the optimal compute solution for your SAP workload? Evaluate and estimate the performance requirements using metrics from the SAP tools and existing workloads. Map the compute requirements to the SAP supported instances best suited to your workload. Consider any specific storage or network requirements for the instance types as well as the availability of the required instance types in your chosen AWS Region and Availability Zones.

ID	Priority	Best Practice
☐ BP 13.1	Required	Evaluate or estimate performance requirements
☐ BP 13.2	Required	Select EC2 instances suitable for SAP workloads
☐ BP 13.3	Highly Recommended	Select architectures which allow for independent scaling of systems or components
☐ BP 13.4	Highly Recommended	Choose instance location for performance considering network performance and latency

For more details, see the following information:

- AWS Documentation: Amazon EC2 Instance Types for SAP
- SAP Documentation: Certified and Supported SAP HANA Hardware
- SAP Note: 1656099 - SAP Applications on AWS: Supported DB/OS and Amazon EC2 products [Requires SAP Portal Access]
- SAP Note: 1656250 - SAP on AWS: Support prerequisites [Requires SAP Portal Access]

Best Practice 13.1 – Evaluate or estimate performance requirements

Future hardware requirements can be estimated by examining the capacity and usage patterns of existing SAP systems. SAP provides several tools for sizing hardware for new and existing systems. To further validate sizing estimates, proof of concept (POC) deployments and performance testing can be used.

Suggestion 13.1.1 – Reference SAPS performance metrics of source hardware

SAP benchmarks hardware using SAP Application Performance Standard (SAPS), which is a hardware-independent unit of measurement that describes the performance of a system configuration in the SAP environment. Consult your existing hardware vendor and the SAP benchmark directory to obtain SAPS values for your on-premises server hardware.

A sizing based on SAPS is appropriate for a migration that introduces minimal changes to the underlying capacity requirements, often referred to as a lift and shift migration.

Suggestion 13.1.2 – Reference SAP EarlyWatch Alert reports and monitoring tools for historical usage details

SAP EarlyWatch Alert reports provide utilization information of your SAP application, such as peak memory and CPU usage. A holistic analysis of these reports spanning several peak events, such as month-end closing and large batch loads, can provide valuable insights into system usage.

In addition to EarlyWatch alert reports, infrastructure level monitoring tools can provide more granularity and further insights.

Suggestion 13.1.3 – Use SAP HANA sizing reports to estimate compute requirements

When migrating to SAP HANA, use SAP provided tools for estimating the size of the target compute. The output generated by these tools details the hardware sizing requirements for your SAP HANA database.

- SAP Documentation: SAP HANA Administration Guide for HANA Platform

- AWS Documentation: SAP HANA Sizing
- SAP Note: 1793345 – Sizing for SAP Suite on HANA [Requires SAP Portal Access]
- SAP Note: 1872170 – ABAP on HANA sizing report (S/4HANA, Suite on HANA...) [Requires SAP Portal Access]
- SAP Note: 2296290 – New Sizing Report for SAP BW/4HANA [Requires SAP Portal Access]
- SAP Note: 1958910 - EarlyWatch Alert For HANA Database [Requires SAP Portal Access]

Suggestion 13.1.4 – Use SAP Quick Sizer for greenfield implementations and functional changes

SAP Quick Sizer can be used for sizing new SAP implementations or for those undergoing changes (for example, increased user base, new functionality or modules). The tool helps you to translate your application's requirements into hardware specifications. For best results, technical and functional teams should collaborate to input values into the Quick Sizer tool.

We recommend the use of SAP expert sizing to validate the sizing of complex implementations.

For more information on SAP tools and services, refer to the following:

- SAP Documentation: SAP: Sizing Benchmarks

Suggestion 13.1.5 – Use proof of concept deployments for sizing accuracy

You can take advantage of the flexibility of AWS services to right-size your SAP workloads and scale as business demands change. Use proofs of concept (POCs) to test migrations to cloud and analyze the performance requirements. This can help right-size the workloads for both cost and performance.

Best Practice 13.2 - Select EC2 instances suitable for SAP workloads

AWS works with SAP to ensure that AWS services are suitable to implement and operate SAP software across a wide selection of instance types. Use guidance from the relevant SAP notes and documentation to identify suitable instances. EC2 instance families offer different ratios of CPU and memory, as well as storage and network throughput characteristics suitable for running SAP workloads. Map your requirements to the appropriate instance type using performance metrics, SAPS figures, and compute estimates. Confirm availability of these instances in your selected Region and Availability Zone.

Suggestion 13.2.1 – Follow SAP guidance on supported databases, operating systems, and AWS services

AWS offers services that can be used for the deployment of SAP products. SAP Note: 1656099 - SAP Applications on AWS: Supported DB/OS and Amazon EC2 products describes which SAP products, database and operating system combinations and Amazon EC2 instance types are currently supported.

You can determine the availability of individual instance types within a specific AZ using the AWS CLI to describe instance type offerings.

- AWS Documentation: Amazon EC2 Instance Types for SAP
- SAP Documentation: SAP NetWeaver benchmarks

Suggestion 13.2.2 – Use hardware metrics and SAPS to guide selection

Each SAP supported Amazon EC2 instance family provides a specific vCPU to memory ratio. You should evaluate each instance family based on your requirements to understand the performance profile. The

current generation of Amazon EC2 instances (based on AWS Nitro) offers the best performance and should be used if available and certified for the deployment scenario.

SAP application servers can use either the general purpose (`m*`) or memory optimized (`r*`) instances. Where there is a requirement for a higher vCPU to memory ratio, consider using compute optimized (`c*`) instances. For AnyDB database servers, memory optimized (`r*`) instances are a good fit for the required core to memory ratio, but additional analysis should be done to validate the sizing, especially if your deployment is subject to per-CPU licensing. For SAP HANA databases that run in memory, memory optimized (`r*`, `x*`, `u*`) are your only options.

Suggestion 13.2.3 – Use SAP HANA hardware directory and memory requirement to select EC2 instances for SAP HANA

AWS has SAP HANA certification for a subset of Amazon EC2 instances to run SAP HANA workloads. Details of these instances, and the IaaS application types supported (OLAP, OLTP, SAP Business One, Scale-Out) can be found in Certified and Supported SAP HANA Hardware and Amazon EC2 Instance Types for SAP.

Database size and actual working memory usage will determine the memory requirement and instance selection.

For non-production workloads, additional options exist. Refer to the blog:

- SAP on AWS Blog: Smaller X1e instances for SAP HANA non-production workloads

Suggestion 13.2.4 – Be aware of EC2 instance features and throughput characteristics

Amazon EC2 instances have different features and throughput characteristics which should be evaluated based on your use case, particularly for workloads with high I/O and throughput requirements. These include enhanced networking capabilities through the Elastic Network Adapter (ENA), I/O performance, Amazon EBS optimization, and suitability for placement groups. For a full list of features, see:

- AWS Documentation: General Purpose Instances
- AWS Documentation: Memory Optimized Instances
- AWS Documentation: Compute Optimized Instances

Best Practice 13.3 – Select architectures which allow for independent scaling of systems or components

SAP systems and components should have the flexibility to scale without being constrained. This might be accomplished within the allocated hardware or by using horizontal scaling of some components. Consider which architectures allow for this scaling and evaluate any associated trade-offs.

Suggestion 13.3.1 – Consider cross-system or cross-component performance impact

Isolate individual systems or components (for example, Central Services, application servers, and database) to avoid negative performance impact between components. Deploying multiple smaller instance sizes can provide options for instance reuse, workload-based scaling, and capacity on-demand. There are exceptions when trying to optimize the use of resources for cost reasons. Refer to the cost pillar for more details.

Suggestion 13.3.2 – Consider capacity flexibility for peak performance

By selecting architectures which allow for scaling of components, such as the application servers, it will be possible to adapt your capacity to match with performance requirements. This allows your SAP systems to scale for exceptional demand including month end processing or seasonal peaks.

Best Practice 13.4 – Choose location to minimize latency

Deploy your SAP instances in Regions and Availability Zones that minimize latency for key business processes impacting end users, critical interfaces, and intra-system traffic.

Suggestion 13.4.1 – Select Region and cloud connectivity to optimize performance

Choose a Region based on proximity to your SAP end users and corporate data center. Select and size any cloud connectivity options (such as AWS Direct Connect and VPN) to accommodate your data transfer requirements.

Use SAP performance tools to understand the breakdown of user response time (such as network, GUI, application, and database) and evaluate the impact of any changes to the network round trip time as a result of increased latency. We recommend you focus on high frequency, low latency interfaces between systems in different locations.

If increased latency impacts certain end user groups, consider the use of end user compute services and accelerators.

- AWS Documentation: AWS Direct Connect
- AWS Documentation: What is AWS Global Accelerator? - AWS Global Accelerator
- SAP on AWS Blog: Deploying SAP GUI on Amazon AppStream 2.0

Suggestion 13.4.2 – Be aware of SAP guidelines for intra-system latency

SAP provides guidance for acceptable network latency for two traffic patterns: between SAP application servers and the database, and between SAP HANA database servers (primary and secondary) for the purposes of data replication. AWS regional networks generally meet or exceed these requirements.

Network Latency between SAP application servers and database (Pattern 1)

The following SAP guidance for database to application server connectivity is based on systems running in a single data center, which does not reflect the resiliency benefits of a Multi-AZ deployment. An Availability Zone is one or more discrete data centers with redundant power, networking, and connectivity in an AWS Region separated by a meaningful distance (at least 10km).

Resilient SAP architectures in AWS typically involve deploying infrastructure across multiple AZs, including SAP application server and database instances.

If you have SAP transactions or batch jobs with time-critical performance requirements and that make a significant number of database calls, we recommend that you run these jobs on SAP application servers located in the same AZ as the database. This can be achieved by using SAP Logon Groups (transaction SMLG) for end users and a batch server group (transaction SM61) for background processing jobs. This helps ensure that the latency sensitive parts of the SAP workload run on application servers with the lowest latency to the database. Use tools such as NIPING to measure latency.

- SAP Note: 1100926 - FAQ: Network performance [Requires SAP Portal Access]
- SAP Note: 2543171 - Latency issue between application server and database [Requires SAP Portal Access]

Network Latency between SAP HANA primary and secondary servers (Pattern 2)

The network latency between the primary and secondary instances will be a factor in the redo log shipping wait time. For synchronous replication, SAP recommends that this wait time should be in the low single millisecond range. This requirement can be achieved across AWS Availability Zones. Use tools such as NIPING to measure latency.

- SAP Documentation: SAP HANA system replication network requirements

Suggestion 13.4.3 – Use placement groups for SAP HANA scale out

To meet the SAP certification for internode communication in an SAP HANA scale-out deployment, it is necessary to use a cluster placement group.

- AWS Documentation: Placement groups - Amazon Elastic Compute Cloud

14 – Select the optimal storage solution

How do you select the optimal storage solutions for your SAP workloads? How you configure this storage will impact the performance of your system. AWS offers a wide range of services, including block, file, and object storage, to meet the storage needs of your SAP databases, applications, and backups. We recommend following the guidelines that have been benchmarked and certified by SAP. For SAP HANA, there are very specific guidelines. Other databases will require more analysis to match your workload.

ID	Priority	Best Practice
☐ BP 14.1	Required	Create mount points and volume associations to align with function
☐ BP 14.2	Required	Select and configure Amazon EBS types aligned with performance requirements
☐ BP 14.3	Recommended	Evaluate Amazon EFS and Amazon FSx performance suitability for your SAP use case
☐ BP 14.4	Recommended	Consider memory as an alternative to storage
☐ BP 14.5	Recommended	Choose appropriate backup solutions and schedule

Best Practice 14.1 – Create mount points and volume associations to align with function

SAP filesystems have unique performance and sharing requirements. For example, the performance profile of the database may require the data filesystem to support a high number of read I/O operations, while the log filesystem is more likely to be constrained by throughput. Filesystems such as `sapmnt` and `trans` need to be shared so that all application servers can access logs and transport files. Recognizing these differences, consider the mapping of filesystems to volumes to ensure there are no performance bottlenecks and that access requirements are met.

Suggestion 14.1.1 – Identify the SAP filesystems and directory requirements for each System

SAP filesystems include system directories (root, boot), executables, page or swap, and application-specific requirements. Each of these should be analyzed to consider:

- The impact when a file system is at capacity (100% used), particularly for the root directory
- Consistency of build, including whether it is included in an AMI, or deployment patterns
- Resilience requirements
- Sharing requirements
- Performance profile

The core SAP filesystem requirements are listed in the SAP documentation. Use these as a baseline and include other requirements specific to your organization.

- SAP Documentation: SAP Required Filesystems and Directories

Suggestion 14.1.2 – Map the appropriate AWS storage service to match with filesystem function

A filesystem can either be local or shared (NFS / SMB). For shared filesystems consider using AWS services, such as Amazon EFS and Amazon FSx, which provide reliability and availability benefits when compared with a hosted NFS server.

Amazon EC2 instance store is another filesystem option which provides temporary block-level storage for your instance. We do not recommend its use due to lack of persistence, availability across instance types, and because it prevents the use of instance recovery.

Suggestion 14.1.3 – Use supported filesystem types

The SAP-supported Linux distributions recommend a number of different file system types. Later versions are standardizing on XFS, but support should be reviewed to ensure there is no performance or functionality impact for your operating system and database version.

- SAP Note: 405827 - Linux: Recommended file systems [Requires SAP Portal Access]
- SAP Note: 2972496 - SAP HANA Filesystem Types [Requires SAP Portal Access]

Best Practice 14.2 – Select and configure EBS types aligned with performance requirements

For each filesystem function and storage service, evaluate storage layout guidelines and tuning options to ensure that IOPS and throughput performance are optimized.

Suggestion 14.2.1 – Evaluate storage characteristics and options for EBS volume types

AWS has a range of volume types with unique characteristics to suit different performance requirements of SAP workloads. Use historical data, or sizing to evaluate the IOPS and throughput requirements. Select your volume type by considering performance, durability, flexibility, and cost.

IOPS and throughput of the `gp3` , `io1`, and `io2` Block Express volume types are independent of the volume size.

IOPS and throughput of the `gp2` volume type is aligned to the volume size. Oversizing the volume may be required to ensure the required IOPS and throughput is available.

If choosing Amazon EBS io2 Block Express, ensure that it is available for the selected instance types.

- AWS documentation: Amazon EBS volume types

Suggestion 14.2.2 – Scale linearly using LVM striping mechanisms

When the performance requirement cannot be met by a single EBS volume, consider striping using Logical Volume Management (LVM). For example, if a single volume has 250 MiB/s throughput capacity, having a stripe set across four volumes can deliver 1000 MiB/s throughput.

Volumes should be of the same size and performance characteristics.

In SAP HANA benchmark testing, the best performance has been achieved using a 256 KB stripe size for data volumes and a 64 KB stripe size for log volumes.

Be aware of instance limits for throughput, I/O, and number of attached volumes.

- AWS Documentation: Create an LVM Logical Volume on an EBS Volume
- SAP Note: 2931808 - Usage of Logical Volume Manager (LVM) with SAP HANA [Requires SAP Portal Access]
- AWS Documentation: Operating System and Storage Configuration - SAP HANA on AWS

Suggestion 14.2.3 – To ensure SAP HANA performance, follow AWS provided storage guidelines

AWS works with SAP to certify storage for SAP HANA workloads according to defined performance benchmarks. The configuration provided by AWS balances performance, cost, and durability within the framework of the SAP TDI 5 Storage KPIs. A compliant storage layout is detailed in the documentation and used in Launch Wizard and quick start deployments.

- AWS Documentation: Storage Configuration for SAP HANA - SAP HANA on AWS

If you deviate from the AWS configuration, it is recommended that you run the SAP HANA Hardware and Cloud Measurement Tools.

- SAP Note: 2493172 - SAP HANA Hardware and Cloud Measurement Tools [Requires SAP Portal Access]

When deciding between general purpose and Provisioned IOPS EBS types, it should be noted that general purpose types meet the SAP HANA Storage KPIs. An in-memory database, such as SAP HANA, has to load data from disk to memory upon database startup. Using Provisioned IOPS EBS types can significantly improve startup times and also speed up tasks such as backups and restores, which are dependent on storage performance.

Suggestion 14.2.4 – For performant local backups at a low cost, use `st1` storage

When SAP solutions need local storage for storing backups, consider using a `st1` instance type for its low cost and high throughput. `st1` is a low-cost block storage type designed for frequently accessed, throughput-intensive workloads.

For SAP HANA, consider using AWS Backint Agent for SAP HANA to avoid the performance and cost impact of a two-stage backup.

Best Practice 14.3 - Evaluate Amazon EFS and Amazon FSx performance suitability for your SAP use case

Amazon EFS (Linux) and Amazon FSx (Windows) provide highly durable and available file systems that can span multiple Availability Zones. Both solutions are designed to deliver high performance, however, when choosing to use network file systems consider the access patterns. For example, many small files, highly parallel writes or high write/read ratios might not be suitable. For SAP workloads, this might apply to SAP HANA XSA, Java executables, or large numbers of job and spool logs.

Amazon FSx for NetApp ONTAP is also a SAP-certified storage type for workloads including S/4HANA, Business Suite on HANA, BW/4HANA, Business Warehouse on HANA, and Data Mart Solutions on HANA. FSx for ONTAP allows you to easily create application-consistent snapshots, space-efficient database clones in seconds and automatic replication of your database across AWS Regions.

Suggestion 14.3.1 – Evaluate scale and performance options

Amazon EFS has two modes for performance (general purpose and max I/O) and two different performance modes (bursting mode and provisioned). For SAP applications, general purpose

performance mode usually provides sufficient I/O. There may be scenarios in which provisioned throughput should be considered, such as when the amount of data in your file system is low relative to throughput demands.

- AWS Documentation: Amazon Elastic File System (EFS) | FAQs - Scale and Performance
- AWS Documentation: Amazon FSx for Windows File Server Features | Scale and Performance
- AWS Documentation: SAP HANA on AWS with Amazon FSx for NetApp ONTAP

Suggestion 14.3.2 - Consider temporary provisioning for short-term requirements

Use cases related to migrations or one-off activities might benefit from a temporary file system where performance characteristics can be adjusted for the duration of the event.

Best Practice 14.4 – Consider memory as an alternative to storage

Consider the performance advantages of using memory for supported scenarios in the database or application layer. SAP HANA uses memory by default, but might benefit from options to optimize load or offload static data. Relational databases should take advantage of caching, and application servers should consider if swap is a requirement.

Suggestion 14.4.1 – Optimize memory usage for SAP HANA

Seek to understand the correlation between SAP HANA memory requirements and operating system memory indicators to help ensure that memory bottlenecks do not impact performance.

- SAP Documentation: SAP HANA Memory Usage and the Operating System
- SAP Note: 1999997 - FAQ: SAP HANA Memory [Requires SAP Portal Access]

To improve database startup performance in scenarios that do not involve a host restart, consider the use of the SAP HANA Fast Restart option. The SAP HANA Fast Restart option dedicates a portion of RAM as a temporary file system (`tempfs`), which is treated by the operating system as persistent memory (until operating system restart) and allows placement of column store main portion in that `tempfs`, which remains there through an index server restart or crash. Thus, no reloading from storage (using I/O) is necessary.

- SAP Documentation: HANA fast restart documentation

Suggestion 14.4.2 – Use database caching for relational databases

For a relational database with high read IOPS requirements, database caching allows you to significantly increase throughput and lower the data retrieval latency. The cache acts as an adjacent data access layer to your database, to improve read performance.

The following documentation provides information about caching use-cases, but as most of this detail is relevant to AWS databases, consult SAP Notes for information specific to your relational database configuration.

- AWS Documentation: Caching (including database caching)

Suggestion 14.4.3 – Evaluate swap space requirements for SAP applications

When physical memory resources are exhausted, SAP uses swap to move inactive pages to a dedicated disk-based storage area. Although having swap can prevent the application from crashing due to

memory insufficient memory, we recommend applying configuration parameters and memory sizing so that swap is used infrequently.

If swap usage is expected, evaluate the characteristics of the allocated volume to avoid further performance issues. Swap can prevent out of memory situations for SAP applications, when the host runs out of physical memory.

- SAP Note: 153641 - Swap space requirement for R/3 64-bit kernel [Requires SAP Portal Access]
- SAP Note: 2999334 - SWAP Utilization (HANA related) [Requires SAP Portal Access]
- SAP Note: 2488097 - FAQ: Memory usage for the ABAP Server on Windows [Requires SAP Portal Access]

Best Practice 14.5 – Choose appropriate backup solutions and schedule

Depending on the backup method, there is the potential to dramatically increase both read and write operations on your storage, which can negatively impact the performance of your application. This is particularly true for database level backups which might be large in volume and lengthy in duration.

Suggestion 14.5.1 – Determine a suitable backup window

Define what is the most appropriate window for the running of backup operations aligned to your business requirements. Consider key dependencies such as the overnight batch schedule and the acceptable runtime.

Suggestion 14.5.2 – Consider options to minimize the performance impact of backups

Analyze any storage or network constraints and evaluate options to minimize the impact of the backup. This may include reducing the duration by using delta change backups either at a database or storage level. Refer to the Reliability Pillar to ensure this does not negatively impact the consistency of backups or the overall restoration time.

- SAP Lens [Reliability]: Best Practice 12.1 - Establish a method for consistent recovery of business data (p. 66)

15 – Evaluate tuning options for the operating system, database, and SAP application

How do you understand and weigh the effects of different tuning options on your SAP system performance? The great variance in performance recommendations for different combinations of SAP software offerings, supported operating systems and databases, and versions prohibit an exhaustive list of recommendations for performance excellence in a single document. With that in mind, the following guidance should be applicable to the majority of SAP use cases, and we will call out specific focus areas where applicable.

ID	Priority	Best Practice
☐ BP 15.1	Required	Follow operating system guidelines for SAP performance
☐ BP 15.2	Highly Recommended	Modify database parameters to align with hardware selection
☐ BP 15.3	Highly Recommended	Modify SAP parameters to align with hardware selection

ID	Priority	Best Practice
☐ BP 15.4	Recommended	Consider performance tuning for recovery and availability options

Best Practice 15.1 – Follow operating system guidelines for SAP performance

SAP provides specific guidance on how best to tune for optimal performance for each of the operating systems that are supported for the SAP software you are deploying. Be sure to read all of the relevant SAP documentation on the operating system on which you are deploying both to understand the relevant tuning parameters and to take advantage of any operating system-specific options to make performance tuning easier and more dynamic.

Suggestion 15.1.1 – Review operating system-related SAP notes prior to installation, version update, or infrastructure change

When building or updating your operating system (through automation or manually) confirm that the appropriate performance settings specific to your combination of SAP software and operating system version are applied.

Suggestion 15.1.2 – Evaluate operating system vendor-supplied SAP tuning

Red Hat and SUSE provide images and repositories which contain tools and configuration optimized for running SAP. These are available in the AWS Marketplace or in a bring-your-own-subscription (BYOS) model.

Vendors are invested in ensuring that their operating systems are optimised for the SAP application. Using vendor-supplied tuning tools such as `saptune` or the (Ansible) system roles for Red Hat Enterprise Linux can assist in defining a known baseline for performance tuning. While this does not preclude tuning the operating system to best accommodate your specific SAP workload, these tools can reduce the effort associated with researching, calculating and applying the most common requirements. Configuration associated with the `tuned` daemon can also adjust dynamically using information it gathers from the system, including CPU count and available memory.

Operating System	Guidance
SUSE Linux Enterprise Server	SAP Note: 1275776 - Linux: Preparing SLES for SAP environments [Requires SAP Portal Access]
Red Hat Enterprise Linux	SAP Note: 2777782 - SAP HANA DB: Recommended OS Settings for RHEL 8 [Requires SAP Portal Access]
Microsoft Windows	(Consult SAP or Vendor documentation for guidance)
Oracle Enterprise Linux	SAP Note: 2478541 - Operating System Requirements for Oracle Database [Requires SAP Portal Access]

Suggestion 15.1.3 – Apply relevant network parameters to the operating system

SAP system performance can be seriously impacted by network misconfiguration, particularly in SAP HANA scale-out database designs as well as in communication between different application server instances and the database instance in a system environment. While in many cases in AWS, the maximum network throughput of an instance is dictated by the instance family and size, tuning of the network settings at the operating system level and in the SAP software itself can have an impact.

Refer to the following AWS and SAP recommendations:

- AWS Documentation: Benchmarking Network Throughput between Amazon EC2 Linux instances in the same Amazon VPC
- AWS Documentation: Elastic Network Adapter – High Performance Network Interface for Amazon EC2
- AWS Documentation: Cluster Placement Groups
- SAP Note: 2198693 - Key Monitoring Metrics for SAP on Amazon Web Services (AWS) [Requires SAP Portal Access]
- SAP Note: 1612283 - Hardware Configuration Standards and Guidance [Requires SAP Portal Access]
- SAP Note: 2081065 - Troubleshooting SAP HANA Network [Requires SAP Portal Access]
- SAP Note: 1100926 - FAQ: Network performance [Requires SAP Portal Access]

Best Practice 15.2 – Modify database parameters to align with hardware selection

SAP provides specific guidance to optimize performance of an SAP system by modifying certain parameters of the underlying database. These parameters are specific to database type and can vary based on whether it's supporting an analytical or a transactional type application.

Suggestion 15.2.1 – Review SAP HANA-specific tuning parameters, if applicable

Operating System and SAP HANA Database parameters can significantly impact performance. Follow SAP on AWS recommendations for Operating system and storage configuration.

- AWS Documentation: SAP HANA on AWS – Operating System and Storage Configuration

Refer to SAP notes and documentation for guidance on SAP HANA parameters including memory allocation.

- SAP Note: 2000000 - FAQ: SAP HANA Performance Optimization [Requires SAP Portal Access]
- SAP Documentation: HANA Parameter: global_allocation_limit
- SAP Note: 1999997 - FAQ: SAP HANA Memory [Requires SAP Portal Access]
- SAP Note: 2926166 - How to limit the overall SAP HANA memory allocation [Requires SAP Portal Access]

Suggestion 15.2.2 – Review database tuning guidance for non-SAP HANA databases

Regardless of the underlying database for your SAP system, performance of the system is in part dependent on how the database is tuned. Each database has specific recommendations for tuning based on available compute, memory, and disk storage. Certain database parameters are dependent on your choice of underlying EC2 instance size; for example, the physical memory available will limit the `db_cache_size` for an Oracle database.

For information relevant to your database, refer to the following:

Database	Guidance
SAP ASE	SAP Note: 2473646 - Performance and Tuning information for ASE -SAP ASE [Requires SAP Portal Access]
IBM Db2	SAP Note: 2751102 – DB6: DB2 11.5 Standard Parameter Settings [Requires SAP Portal Access]

Database	Guidance
Oracle	SAP Note: 2470718 – Oracle Database Parameter 12.2 / 18c / 19c [Requires SAP Portal Access]
Microsoft SQL Server	SAP Note: 2779607 – Configuration Parameters for SQL Server 2019 [Requires SAP Portal Access] , SAP Note: 2729848 – SAP Installation Media and SQL4SAP for SQL Server 2019 [Requires SAP Portal Access]
SAP MaxDB	SAP Note: 819641 – FAQ: SAP MaxDB performance [Requires SAP Portal Access]

Best Practice 15.3 – Modify SAP parameters to align with hardware selection

Tuning SAP application parameters can help improve the performance of the application. These parameters are often dependent on the underlying hardware configuration and operating system type.

Suggestion 15.3.1 – Allow SAP to self-tune according to `PHYS_MEMSIZE`

For recent versions of SAP software, using kernel release 7.40 or higher, self-tuning of certain parameters is possible and recommended. For instance, many parameters are derived via formulas related to the main memory available on an instance (PHYS_MEMSIZE). This allows for automatic tuning of memory parameters when resizing an EC2 instance underlying the SAP software to meet changing performance requirements.

- SAP Documentation: SAP Memory Management: Parameter Reference
- SAP Note: 2085980 – New features in memory management as of Kernel Release 7.40 [Requires SAP Portal Access]

Suggestion 15.3.2 – Review SAP swap space and maximum memory use

When running SAP on AWS, overutilized swap space on disk can cause I/O credit exhaustion on Amazon EBS and lead to performance degradation. Evaluate the different EBS storage options available on AWS and configure swap space to meet your performance needs.

- SAP Note: 1597355 - Swap-space recommendation for Linux [Requires SAP Portal Access]
- SAP Documentation: Swap Space Requirements

Best Practice 15.4 – Consider performance tuning for recovery and availability options

In alignment with both the Well-Architected Reliability and Operational Excellence pillars, tuning of the SAP system given your chosen recovery and resiliency requirements should be evaluated to minimize any performance impact. Take into consideration items such as system performance during a backup, clustering options for the chosen database (for example, synchronous vs. asynchronous SAP HANA System Replication), and the distribution of load across multiple SAP application server instances.

Suggestion 15.4.1 – Review performance recommendations for backup and recovery solutions

Each supported database has different recommendations to optimize the performance of backups and recovery operations, and these often work in conjunction with your chosen software solution for managing backups and restores, including third-party offerings.

In general, following the guidelines for improving throughput between an EC2 instance and the storage target of your backup (such as, EBS volumes, S3 buckets, and EFS file systems) can improve the performance of your backup and recovery.

For example, when using Amazon S3 as a repository for backups and AWS Backint Agent for SAP HANA, you can enable performance improvements via changing configuration parameters, such as the maximum concurrency settings. Another common way to improve backup performance is to increase EBS volume performance characteristics, such as maximizing GP3 throughput configuration to 1,000 MB/s.

For more information, refer to the following:

- AWS Documentation: AWS Backint Agent for SAP HANA
- AWS Documentation: SAP NetWeaver on AWS – Backup and Recovery
- AWS Documentation: S3 Configuration Parameters
- SAP on AWS Blog: Build for availability and reliability

Database	Guidance
SAP HANA	• AWS Documentation: SAP HANA on AWS – Storage Configuration for SAP HANA • SAP Note: 1842096 - HANA Backup & Restore Performance [Requires SAP Portal Access] • SAP Note: 2945518 - Performance issues encountered on HANA when a data backup is running [Requires SAP Portal Access]
SAP ASE	(Consult SAP or Vendor documentation for guidance)
IBM Db2	(Consult SAP or Vendor documentation for guidance)
Oracle	SAP Note: 2084077 - How to plan backup cycle for Oracle database [Requires SAP Portal Access]
Microsoft SQL Server	SAP Note: 1420452 - FAQ: Restore and recovery with MS SQL Server [Requires SAP Portal Access]
SAP MaxDB	SAP Note: 1377148 - FAQ: SAP MaxDB backup / recovery [Requires SAP Portal Access]

Suggestion 15.4.2 – Review configuration of clustering parameters

Clustering options for SAP HANA and other databases often rely on a confirmed connection in a cluster (that is, a heartbeat) between the primary instance and the failover instance. SAP administrators must balance the speed with which an action can occur in the system with the potential for any failover side effects that may occur if there is a false negative interruption in communication. Follow recommendations for timeout parameters and related settings.

- AWS Documentation: SAP HANA on AWS: High Availability Configuration Guide for SLES and RHEL
- AWS Documentation: SAP HANA on AWS Operations Guide: Networking
- AWS Documentation: SAP on AWS – IBM Db2 HADR with Pacemaker
- SAP Note: 1612105 - DB6: FAQ on Db2 High Availability Disaster Recovery (HADR) [Requires SAP Portal Access]
- Operating System-specific Documentation: SUSE Linux SAP HSR Scale-up Performance Optimized Scenario
- Operating System-specific Documentation: Automated SAP HANA System Replication in Scale-Up in pacemaker cluster

16 – Understand ongoing performance and optimization options

What processes and procedures do you put in place to measure performance changes and opportunities for optimization? Baseline your applications performance requirement from its historical monitoring data and set relevant alerts to inform system administrators when deviations occur. Have procedures in place for system admins to remediate such issues with manual or automated actions.

ID	Priority	Best Practice
☐ BP 16.1	Required	Have data to evaluate performance
☐ BP 16.2	Required	Establish baseline performance requirements
☐ BP 16.3	Highly Recommended	Identify performance trends using data
☐ BP 16.4	Highly Recommended	Identify and triage performance issues
☐ BP 16.5	Highly Recommended	Scale to meet performance demands
☐ BP 16.6	Recommended	Develop mechanisms for simulating production load for analysis purposes
☐ BP 16.7	Recommended	Continually optimize sizing and configuration based on performance data

Best Practice 16.1 – Have data to evaluate performance

To evaluate the performance of an SAP system and take action in the event performance is suboptimal, monitoring data must be collected for compute, memory, storage, and networking as described in the Well-Architected Framework Performance Excellence guidelines concerning monitoring your resources. As stated in the Well-Architected Framework Operational Excellence pillar, understanding the current state of the system, putting in place key performance indicators, and collecting metrics in a timely manner for diagnosis are crucial for investigating performance issues.

- Well-Architected Framework [Performance Efficiency]: Monitor Your Resources to Ensure That They Are Performing as Expected
- Well-Architected Framework [Operational Excellence]: Understanding Workload Health

Suggestion 16.1.1 – Gather and store data relevant to performance metrics

To collect and view relevant SAP monitoring data, you should install and configure the AWS Data Provider for SAP and set up metrics in your chosen monitoring tools that support your SAP workload. Further details on monitoring and additional recommendations are available in the Operational Excellence pillar.

- AWS Documentation: AWS Data Provider for SAP
- SAP Lens [Operational Excellence]: Best Practice 1.1 - Implement prerequisites for monitoring SAP on AWS (p. 8)
- SAP Lens [Operational Excellence]: Best Practice 1.2 - Implement infrastructure monitoring for SAP (p. 9)
- SAP Lens [Operational Excellence]: Best Practice 1.3 - Implement application and database monitoring for SAP (p. 10)

Best Practice 16.2 – Establish baseline performance requirements

Every SAP application has unique performance requirements. Using historical monitoring data helps SAP administration teams understand the baseline performance of these applications, enabling them to identify and understand the extent of any performance changes. Relevant alerts can be put in place to detect anomalies, such as unintended CPU spikes, storage throughput deltas, memory consumption increases, and more complex performance decrements. This monitoring data can be used to further fine-tune performance.

Suggestion 16.2.1 – Collect and evaluate data that reflects SAP-specific KPIs

This suggestion aligns closely with additional suggestions in the Well-Architected Framework Performance Efficiency Pillar discussion regarding resource monitoring.

In addition to this general guidance, SAP-specific KPIs include dialog response time, buffer swaps, used memory. These KPIs might differ based on the type of SAP software and version you are running on. Further detail on KPI and monitoring recommendations is available in this document in the Operational Excellence pillar:

- SAP Lens [Operational Excellence]: Best Practice 1.2 - Implement infrastructure monitoring for SAP (p. 9)
- SAP Lens [Operational Excellence]: Best Practice 1.3 - Implement application and database monitoring for SAP (p. 10)

Best practice 16.3 – Identify performance trends using data

After baselines for performance are established, system administrators must monitor trends over time to see if KPIs remain stable within preferred norms. If the performance data indicates a trend toward unacceptable values of the KPI, system administrators can then take steps to avoid or mitigate performance impacts.

Suggestion 16.3.1 – Conduct regular reviews of SAP system performance

Periodic reviews of KPIs by system administrators can help identify trends in performance-related data as well as determine which alerts might be most beneficial. These alerts can then be used to automate notifications should the trend continue as well as put in place auto-remediation measures to address the potential performance issue (for example, dynamically changing SAP parameters in response to performance indicators). Examples of KPIs and related trends can be found in SAP EarlyWatch Alert reports, which in some cases can be customized with additional useful metrics. SAP service level reporting can also be useful if you have Service Level Agreements (SLAs) in place for your SAP workloads.

- SAP Documentation: Service Level Reporting
- SAP Note: 1040343 - SAP EarlyWatch Alert [Requires SAP Portal Access]
- SAP Note: 1829914 - Customize EWA Reports [Requires SAP Portal Access]

Suggestion 16.3.2 – Retain historical data to identify trends

You should retain performance data and associated logs for a predetermined period of time to understand trends in system behavior. Performance tuning of any SAP system will depend on the ability to look back over historical periods of days, weeks, and months to understand what constitutes a performance trend or cyclical performance event. Common events that require retention of data to observe performance impacts include:

- Month-end and year-end financial processing

- Increased reporting requirements around business milestones (for example, after a large semi-annual sales kick-off)
- On-boarding of a large new SAP user population within the business
- Technology changes, such as infrastructure sizing, database patches, operating system version updates, or SAP software upgrades

Best Practice 16.4 – Identify and triage performance issues

When key metrics indicate performance is degrading, have a process in place to remediate the underlying cause. Using automation (see the following best practice on dynamic scaling) can reduce the need for manual intervention, but when that is not possible, having an automated alerting process for administrators is vital.

Suggestion 16.4.1 – Configure performance alerts appropriately

Follow the guidelines as mentioned in the Well-Architected Framework Performance Efficiency pillar regarding monitoring and alerts, and make use of SAP alerting capabilities where they provide additional capabilities. Additional details are also available in [Operational Excellence] 1 - Design SAP workload to allow understanding and reaction to its state (p. 7).

- Well-Architected Framework [Performance Efficiency]: Monitoring
- SAP Documentation: SAP NetWeaver Alert Monitor

Suggestion 16.4.2 – Automatic remediation of performance incidents

While the management of performance incidents involves the best practices on operations detailed in the Well-Architected Framework Operational Excellence pillar, the proactive detection and automated remediation of potential performance impairment can prevent deepening a performance problem and can improve the end-user experience. When automated processes for mitigating a performance issue are not possible, having a detailed runbook in place on how the operational team should respond to a performance issue can accelerate the response to a performance incident.

- SAP Lens [Operational Excellence]: Best Practice 1.8 Use automated response and recovery techniques to react to monitoring alerts (p. 15)
- Well-Architected Framework [Operational Excellence]: Best Practices: Operate

Best Practice 16.5 – Scale to meet performance demands

One of the primary benefits of operating workloads in AWS is the ability to increase or decrease the compute capacity and change the storage performance characteristics to match the performance required for the use case. For SAP workloads, use dynamic scaling where applicable to avoid performance bottlenecks. Scenarios where dynamic scaling is not possible, such as scaling out an SAP HANA database cluster, use a manual deployment process.

Suggestion 16.5.1 – Reactively scale SAP workloads

In response to dynamic changes in workload performance requirements, scale your SAP resources accordingly. Where possible, use automation to scale in or out, but when that is not an option (such as scaling up a database instance), have a process in place to do so manually. Consider:

- Adding or removing application server capacity or changing instance sizes as required to meet demand
- Changing SAP parameters to redistribute virtual resources programmatically
- Modifying storage type (for example, Amazon EBS `gp3` to `io2` or vice versa in AWS), where applicable, to optimize storage performance

- AWS Documentation: Request modifications to your EBS volumes

Suggestion 16.5.2 – Schedule scaling for predictable SAP workloads

Whether in an automated or manual fashion, scaling an SAP workload up or down based on predictable performance patterns is advisable. For instance, when month-end financial processing on an SAP ECC system leads to a predictable 20% increase in processing requirements on application server instances, system administrators can proactively increase the number or size of application servers, then scale-in the number of instances when the usage predictably decreases.

Best Practice 16.6 – Develop mechanisms for simulating production load for analysis purposes

Having a clone of production data in a test system allows system administrators to simulate production SAP workloads and conduct vital performance tests, such as stress and volume testing. This type of testing can help identify potential performance bottlenecks and prevent performance issues from occurring in a live production environment.

Suggestion 16.6.1 – Define performance sensitive activities

Evaluate which transactions, reports, and operational activities could have an impact on your business if they do not meet peak load requirements or time-critical thresholds. For example, an overnight batch job which must complete in five hours, or a customer-facing transaction accessed concurrently by thousands of users during a quarterly business peak. Document and agree on the measurement approach, KPIs, and success criteria for these workload activities.

Suggestion 16.6.2 – Create an automated test approach for key activities

If required, develop a test strategy to confirm that your SAP workload performance benchmarks are met. Evaluate how test landscapes and tools can enable a repeatable suite of tests to measure the impact of operational activities, change releases and major patching on the performance of your workload.

Best Practice 16.7 – Continually optimize sizing and configuration based on performance data

Review performance metrics on a regular basis outside of your incident response process. By doing so, you can discover system components that are undersized, oversized, or no longer used at all. A regular performance optimization cadence should be established for your SAP workloads, focusing on right sizing your system components for real user load. This activity will improve user experience, eliminate unneeded aspects of your architecture, and help improve both the cost effectiveness and resilience of your workload.

Suggestion 16.7.1 – Perform regular architecture right sizing with historical performance metrics as a guide

Regularly review your SAP workload for opportunities to right-size its components. Consider whether storage, compute, network, and supporting services need to be increased or decreased to better fit your business performance requirements.

For further information, refer to the following resources:

- SAP Lens [Cost Optimization]: Best Practice 20.5 - Review usage for opportunities to optimize (p. 112)
- SAP Lens [Operational Excellence]: Best Practice 4.4 - Perform regular workload reviews to optimize for resiliency, performance, agility, and cost (p. 28)
- AWS Documentation: Right-Sizing

Cost optimization

The cost optimization pillar includes the continual process of refinement and improvement of a system over its entire lifecycle. This optimization should occur from the initial design of your first proof of concept to the ongoing operation of production workloads. Choose the right solution and pricing model. Build cost-aware systems that allow you to achieve business outcomes and minimize costs. To realize cost optimization over time, you need to identify data, infrastructure resources, or analytics jobs that can be deleted or downsized.

17 – Evaluate SAP architecture patterns for cost efficiency

How do you incorporate cost considerations into the evaluation of SAP architecture patterns? Ensure that when there is a decision to be made on architecture, that the cost implications are fully understood as part of design considerations.

ID	Priority	Best Practice
☐ BP 17.1	Required	Evaluate your use of SAP managed service offerings
☐ BP 17.2	Required	Evaluate the cost characteristics of your SAP application architecture pattern
☐ BP 17.3	Required	Understand business requirements to make cost optimized design decisions per environment
☐ BP 17.4	Highly Recommended	Review the size, granularity, and latest available EC2 instances for SAP components
☐ BP 17.5	Highly Recommended	Consider using on-demand capacity to improve cost efficiency
☐ BP 17.6	Recommended	Evaluate cost benefits and impact of shared services and solutions
☐ BP 17.7	Recommended	Evaluate the cost benefits of automation

Best Practice 17.1 – Evaluate your use of SAP managed service offerings

Per the AWS shared responsibility model, the customer has the responsibility of managing their SAP workloads on AWS. Optionally, a service provider can be used to manage your SAP workloads on AWS. When evaluating a service provider, the responsibility of both upfront and ongoing cost management should be delegated appropriately and should be treated as an ongoing process.

Suggestion 17.1.1 – Evaluate and understand available managed service offerings

A number of AWS and SAP partners provide services for the deployment and operation of your SAP landscape. The scope and maturity of services provided varies across partners. These types of services can provide efficiencies, for example, centralized support, bundled licenses, or automated deployment services. These can reduce your overall costs and should be evaluated based on your specific business requirements. Evaluate partners for AWS competencies including those with the AWS SAP Competency or belonging to the AWS Partner Network (APN).

SAP offers RISE with SAP, a single-tenant, SAP-managed S/4HANA solution that also includes SAP Business Technology Platform (BTP) and other SAP software in a single contract. AWS supports customer

choice and hosts many customers on the SAP RISE platform. SAP RISE can be hosted on AWS and combined with SAP BTP on AWS and other AWS workloads. When choosing AWS for RISE and BTP, you have options to simplify the architecture, improve connectivity, improve security, and reduce costs.

- AWS Blog: Your ERP environment, your choice: RISE with SAP presents another ERP modernization path for AWS customers
- AWS Blog: How to connect SAP solutions running on AWS with AWS accounts and services

Suggestion 17.1.2 – Understand the roles and responsibilities related to cost control

Different managed service offerings have different cost models to cover infrastructure, licensing, and services. Decide where the responsibility for cost control lies. The following questions can be asked as part of this process.

- Are the costs from the provider:
  - Based on a percentage of infrastructure spend?
  - Based on an agreed total cost of ownership (TCO)?
  - Variable (both up and down) according to changed business conditions?
- T-shirt sized (small, medium, large)? Is there an appropriate change control process in place to ensure that costs are controlled and understood?
- Is there sufficient visibility and transparency of the infrastructure costs?
- Does cost governance limit innovation and flexibility?

Suggestion 17.1.3 – Agree on an approach to cost management and optimization with all parties

When evaluating the different managed service offerings available, understand the managed services partner's approach to cost management. How can you work together to drive on-going cost optimization for your organization?

This evaluation should include a regular review process. It might also benefit from incentives, such as a shared reward model, that encourage the partner to take ownership so that both parties financially benefit from the cost savings achieved.

Best Practice 17.2 – Evaluate the cost characteristics of your SAP application architecture pattern

As you develop the architecture of your SAP landscape, consider the cost of the number of infrastructure components in addition to their size and location. By establishing the business requirements of the solution, acknowledging risk, and finding opportunities to optimize, you can realize significant cost savings

Suggestion 17.2.1 – Review your selected SAP installation patterns

For each SAP application, define a deployment pattern, such as standalone, distributed, or high availability (HA). Select the architectural pattern that best balances the cost and reliability characteristics to meet your business requirements. A useful approach is to quantify the cost of an outage to your business and work backwards from that. Balance the risk of an individual failure impacting availability against the cost of reducing that risk.

Additionally, consider whether your architecture has the flexibility for right sizing. There can be cost savings with operating system licensing, storage, and managing multiple application servers on a single host. For the application tier, instance sizes are available with fine granularity for CPU and memory, with near linear pricing in the supported instance families. Deploying multiple smaller instance sizes can provide more options for instance reuse and workload-based scaling.

Evaluate logical groupings and consider the effect of combining components, systems (SIDs), or landscapes. Would these activities increase operational complexity and decrease reliability?

- AWS Documentation: Architecture Guidance for Availability and Reliability of SAP on AWS
- SAP Lens [Reliability]: Reliability design principles (p. 51)
- AWS Well-Architected Framework [Reliability]: Reliability Pillar

Suggestion 17.2.2 – Evaluate exceptions for the use of multitenancy or hosting multiple databases in a single host

For most databases, size each system independently and take advantage of flexible instance sizing to match requirements for those systems. In some cases, it might make sense to deviate from that recommendation in the interest of cost. For example:

- When a HANA-based component requires less memory than the smallest EC2 instance size available, consider the use of SAP HANA Multitenant Database Containers. Hosting with other components allows for efficient use of the compute resources.
- Core-based database licensing models for relational databases, including Oracle and SQL Server
- Applications that are, or can be, tightly coupled for uptime requirements and version dependencies. This includes management tools (for example Solution Manager and SAP HANA Cockpit) and some SAP NetWeaver Gateway Deployment options (Fiori and ECC).

Suggestion 17.2.3 – Evaluate the use of the single host installation pattern for systems that do not require resiliency and scalability

For individual applications or environments, you should consider the advantages of a single host model. This can help save operating system costs, storage duplication, software license costs, and managed service costs. Common architectural options, particularly for non-production landscapes, include:

- Co-hosting database, application, and SAP central services
- Separate database (to minimize database licensing). Refer to [Cost Optimization]: Best Practice 18.3 - Evaluate licensing impact and optimization options (p. 99).
- Combined application and SAP Central Services

Suggestion 17.2.4 – Choose the most cost-effective Region that meets your requirements

The primary drivers for SAP Region selection are proximity, data residency, and service availability. For deployments where a choice exists, be aware that each AWS Region offers pricing based on local market conditions. AWS service pricing is therefore different in each Region. Review any price differences and their potential impact.

Suggestion 17.2.5 – Use architectures that can be scaled in the event of a failure

Recovery mechanisms and the elasticity of the cloud allow for a design where redundant resources do not need to be active and at 100% capacity. If your business requirements allow for a more flexible RTO or RPO, consider the following.

For the database:

- If your recovery point objectives allow, consider a secondary or standby database node that does not require equivalent compute capacity to apply changes from the primary. With an awareness on the recovery time impact, consider the cost advantages of deploying a smaller or shared instance for your secondary, and scale up only when required. Use of a smaller instance maintains the 1:1 relationship between primary and secondary system instances. A shared instance architecture pools the secondary database with a non-replicated system database onto a single instance. In the event of a failure, the non-replicated system must be stopped before a takeover can occur. This will increase the RTO.

- If using a smaller instance for the secondary SAP HANA database, turn off memory preload to accommodate a smaller memory footprint on the standby and reduce cost. SAP estimates the memory requirements in the help document for Secondary System Usage.
  - SUSE Documentation: SAP HANA System Replication Scale-Up - Cost Optimized Scenario | SUSE
- If your recovery time objective and resiliency requirements allow, consider data and log backup approaches that use Multi-AZ storage (such as Amazon FSx, Amazon EFS, or Amazon S3). These approaches allow for geographic protection of data without requiring redundant secondary resources. In the event of failure, secondary resources can be created on demand and quickly restored from cross-location backups and log storage.
  - SAP on AWS Blog: How to use snapshots to create an automated recovery procedure for SAP ASE databases

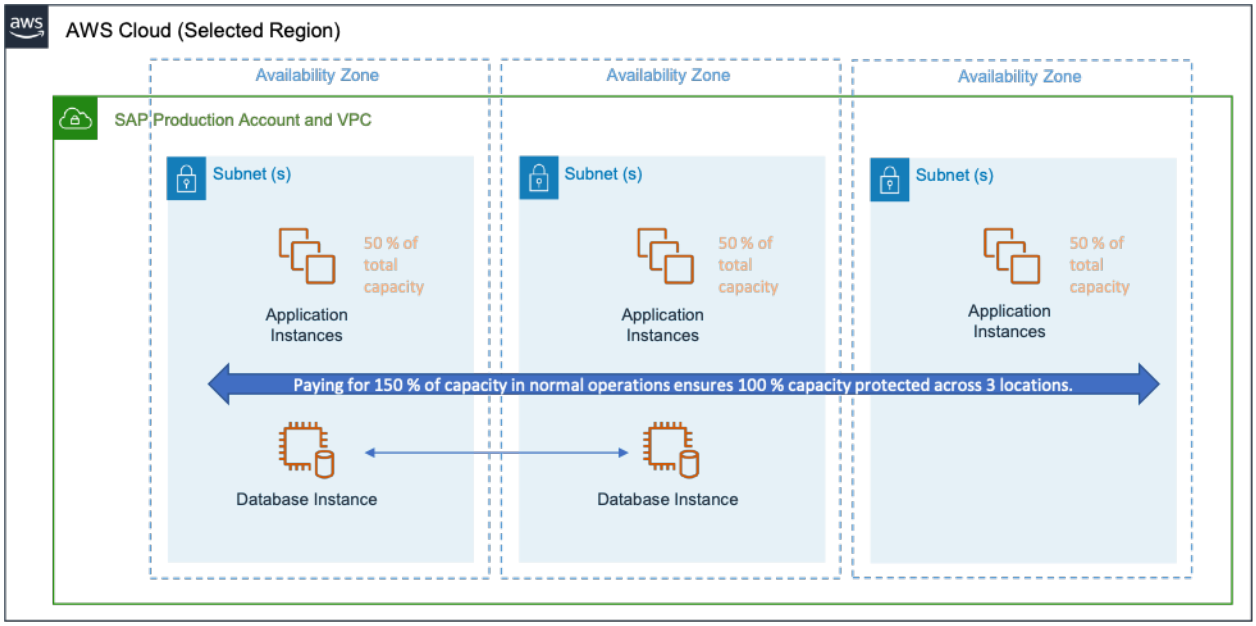
For the application:

- AWS instance recovery uses a CloudWatch alarm to monitor an Amazon EC2 instance. It automatically recovers the instance if it becomes impaired due to an underlying hardware failure. Evaluate if the failure scenarios covered provide sufficient protection.
- For scenarios in which an application server needs to be quickly recreated, options include EC2 instances that are provisioned but not running, templated AMIs, storage replication using common staging servers, or infrastructure as code (IaC).

Suggestion 17.2.6 – Consider the cost of minimum compute capacity during a failure

Distributing SAP components across Availability Zones can reduce the costs incurred for capacity reservations in the event of failure. By distributing components across Availability Zones, you avoid the need for excess capacity because you already have part of your workload geographically spread. This minimizes the scope of impact in the event of an AZ failure.

For example, if 100% capacity is an availability requirement for failure scenarios including the loss of an Availability Zone, instead of provisioning 200% capacity across two Availability Zones, provision 150% capacity across three.



Example of a three Availability Zone architecture with 150% of capacity in normal operations

Suggestion 17.2.7 – Evaluate the use of storage-only based recovery options

In general on AWS, we recommend database replication over storage replication to ensure protection from the broadest range of failure scenarios. For the application layer or for less critical instances, a DR solution that uses storage replication without the need for compute can reduce costs. It also minimizes the complexity associated with managing change.

- AWS Documentation: AWS Elastic Disaster Recovery - Amazon Web Services
- AWS Documentation: Creating backup copies across Regions - AWS Backup

Suggestion 17.2.8 – Understand networking-related costs

SAP customers often require a secure connection between their on-premises network and Amazon VPC. Using an appropriately sized Direct Connect, a VPN connection, or both, it is possible to meet performance and reliability requirements while minimizing cost.

Data transfer costs will be impacted by Region, VPC, and Availability Zone design. Evaluate how the distribution and replication of your SAP components can be optimized without compromising reliability.

For example, if two systems that transfer large amounts of data are in separate locations, consider the impact on data transfer costs.

- AWS Documentation: EC2 On-Demand Instance Pricing – Amazon Web Services
- AWS Documentation: Architecture Patterns - General SAP Guides

Further guidance can be found in the Cost Pillar of the Well-Architected Framework Review Plan for Data Transfer - Cost Optimization Pillar.

Best Practice 17.3 – Understand business requirements to make cost-optimized design decisions per environment

Optimize the cost of each system or environment individually based on its differing characteristics. Consider capacity, performance, reliability and operating hours to match business requirements. For environments or applications that are less critical for the experience of end users or business processes, minimize storage, compute, and operating hours to reduce cost. Balance the cost savings of a reduced configuration with operational requirements for testing or support.

Suggestion 17.3.1 – Evaluate if non-production environments need a full copy of production data

Having full copies of production data in non-production will greatly impact your storage and compute costs. Consider minimizing the number of copies of production data while still meeting your testing requirements. Options to minimize costs of non-production environment data storage include:

- Using less storage capacity for development and test systems.
- Using data slicing tools to carve out a smaller subset of test data in non-production systems.
- Consider the use of temporary production copies. These copies can be created on-demand and then quickly decommissioned, or archived, after the business need or test has passed.
- Evaluate if the SAP recommendation for SAP HANA databases of 50% working memory is required in non-production systems.

Suggestion 17.3.2 – Evaluate if non-production environments always need to have the same performance as production

Non-production systems and some support systems are likely to have a smaller set of users, handle significantly lower transaction volumes, or have flexible response time requirements. Consider the following:

- Reducing the SAP Application Performance Standard (SAPS) for your workload by using smaller EC2 instance types.
- Using fewer application servers.
- Using lower cost Amazon EBS storage types, for example, gp3 instead of `io2` .
- Using reduced performance characteristics for non-production systems volumes, for example, 3,000 IOPS instead of 10,000 IOPS.
- The elasticity of the cloud means that you can scale up your non-production testing resources that require production-like performance, such as load or scaling tests.

Suggestion 17.3.3 – Evaluate if non-production environments need the same operating hours as production

Non-production environments like test, training, and sandbox systems, might have reduced operating hours compared with production. Consider time zones and the working hours of your support teams to determine whether all systems are required 24 x 7. Use this information to select the lowest pricing model.

For example, running your SAP training system for 40 hours a week with an on-demand pricing model (~23% uptime) will be cheaper than running it always on at 100% with a 3-year Reserved Instance or Savings Plan.

Suggestion 17.3.4 – Evaluate if non-production environments consistently need the same reliability as production

Choose the most cost-effective architecture to match each system's individual reliability requirements. See [Reliability]: Best Practice 10.1 - Agree on SAP workload availability goals that align with your business requirements (p. 52). Further guidance can be found in the Reliability Pillar of the AWS Well-Architected Framework.

Where a production-like architecture exists solely for testing purposes, consider how often it needs to mirror production. If database high availability is needed in a non-production environment for reliability or performance testing, you can choose to shut down or scale down the secondary instance outside of these testing windows to save cost.

Cost benefits can be realized through the use of automation and by using on-demand pricing for environments that don't need production-like performance at all times.

Suggestion 17.3.5 – Evaluate the business requirements for non-core systems including support and legacy systems

If environments exist for reference purposes only, or have a less critical business role, evaluate the uptime, performance, and reliability requirements needed compared to your core production systems.

For example, a legacy ERP system might require being kept for reference purposes from a prior application conversion or business restructure. Costs for this system can be optimized by running the EC2 instances only when required, thus only paying for Amazon EBS storage. A more cost-effective solution could be archiving the system via backup to Amazon S3 and Amazon S3 Glacier.

Best Practice 17.4 – Review the size, granularity, and latest available EC2 instances for SAP components

Smaller EC2 instances provide greater cost flexibility in SAP workloads. They introduce options for horizontal scaling that allow for compute to be switched off when not in use or scaled up only during peak loads. Adopting a consistent EC2 instance size at the application tier will help you maximize the benefits of Reserved Instance and Savings Plans commitments across all workloads. Take into account the latest available AWS SAP certified instances. The operational impact, license costs, support, sharing and reusability for each component should also be evaluated.

Suggestion 17.4.1 – Evaluate the cost benefits of multiple smaller application servers to provide flexibility

For many SAP workloads, application servers can be designed to be immutable. Having a standard application server configuration, which is scaled horizontally by replicating the base unit, gives options for consistent repeatable units. The advantages are reusability, compute utilization, reservations, and automation. Per-unit requirements, such as operating system licensing, storage duplication, and management costs, should be factored into the evaluation.

Consider the following:

- SAP on AWS Blog: DevOps for SAP – Driving Innovation and Lowering Costs.
- SAP on AWS Blog: Using AWS to allow SAP Application Auto Scaling

Suggestion 17.4.2 – Evaluate the cost benefits of an SAP HANA scale-out configuration where supported

SAP OLAP workloads can be deployed in both scale-up and scale-out configurations. SAP recommends to scale-up before scaling out to reduce operational complexity. However, scale-out implementations might be applicable for larger analytical or native SAP HANA workloads, which require significant compute (SAPS).

In certain cases, S/4HANA also supports scale-out configuration but with restrictions. See SAP Note: 2408419 - SAP S/4HANA - Multi-Node Support [Requires SAP Portal Access].

When considering scale-up vs. scale-out consider the following:

- Certified EC2 instance sizes available for scale-up and scale-out
- The cost per GiB of EC2 memory for each instance family. Larger EC2 instances typically have a higher cost per GiB than smaller instances.
- The added complexity and operational overhead of managing data distribution in scale-out deployments. See SAP Note: 2081591 - FAQ: SAP HANA Table Distribution [Requires SAP Portal Access]

Best Practice 17.5 – Consider using on-demand capacity to improve cost efficiency

The on-demand pricing model is suitable for SAP workloads needing reduced operating hours, short-term projects, experimentation, or expanded capacity for small periods of time (for example, performance testing). Determine where you can use on-demand pricing in your SAP architecture.

Suggestion 17.5.1 – Evaluate the use of on-demand for SAP systems needing less than 24/7 operating hours

Based on the break-even point between the use of on-demand and other pricing models. (See [Reliability]: Best Practice 18.1 - Understand the payment and commitment options available for Amazon EC2 (p. 96)), evaluate if on-demand will provide the lowest cost. As part of this evaluation, consider the overall Savings Plan commitment.

Common use cases include non-production systems that are not needed outside of extended business hours or short-term business experiments, such as trial upgrades and proofs of concept (POCs).

- SAP on AWS Blog: Automate Start or Stop of Distributed SAP HANA systems using AWS Systems Manager

Suggestion 17.5.2 - Evaluate scheduled or dynamic scaling options for peak loads

On-demand capacity is commonly used in SAP workloads for peaks where capacity requirements spike for a short period of time. Consider the following:

- Use schedule-based SAP application server scaling for known usage pattern peaks such as period, month-end, year-end, or seasonal peaks.
- Use dynamic scaling of the application tier where peaks are less certain and need to be scaled based on real-time user load. Explore mechanisms which are SAP-aware and provide the required governance and controls.

Note: When evaluating dynamic scaling of the application tier, consider the impact on user connections and batch jobs if an SAP application server is shut down due to the stateful nature of SAP components. AWS, SAP, and APN partner-developed tools can help address this requirement.

- AWS Documentation: Systems Manager Automation actions reference
- SAP Documentation: SAP Landscape Management
- SAP on AWS Blog: Using AWS to enable SAP Application Auto Scaling

Best Practice 17.6 – Evaluate cost benefits and impact of shared services and solutions

Where the same function is required by multiple SAP systems, it can be a cost-effective option to centralize the management and costs by using existing solutions, sharing components, or both. Monitoring, backups, and connectivity are common functions that can be managed either within an AWS account boundary or in a dedicated account. Standardization, reducing duplication, and reducing complexity reduces cost.

Find appropriate ways to share resources for cost reduction while still maintaining appropriate isolation and without introducing dependencies that might impact operations.

Suggestion 17.6.1 – Evaluate the cost benefit of a 1-to-1 versus a 1-to-many setup for each shared service

A standard pattern for SAP landscapes is to isolate non-production and production workloads in separate accounts as part of a multi-account strategy. This can be a logical boundary for some services. Consider complexity and operational costs for each scenario including management boundaries which enforce segmentation, and any data transfer cost impact across Regions, AZs, VPCs, or accounts.

In a multi-account design, some AWS services can be hosted centrally and accessed by several applications and accounts in a hub and spoke design to save cost. Such services include:

- Dedicated VPC with NAT gateway for all outbound traffic from spoke VPCs
- Centralized model for load balancers and Web Dispatchers
- Shared Amazon EFS or Amazon FSx for transports and other file sharing needs

Suggestion 17.6.2 – Evaluate where reuse of existing services can reduce costs

This suggestion applies across a number of levels:

- Where AWS provides services, these often minimize overhead and work with consumption-based pricing. Some examples include Amazon EFS, AWS Backint Agent for SAP HANA, and AWS Backup.
- Your business might have an enterprise-wide standard for some functions (for example, enterprise backup) which should be used for operational consistency and economies of scale.
- AWS Partner solutions might be available through the AWS Marketplace or BYOL (bring your own license) based on your specific business requirements.

- License-included AWS Marketplace machine images might help reduce upfront costs. Licensing restrictions should be considered in this scenario as they could impact solution flexibility by restricting portability to different instance types.

Suggestion 17.6.3 – Understand the impacts of using build vs. buy vs. open source approaches

Whether this is AWS or APN partner solution, there are varying degrees of build it yourself vs. open source vs. off the shelf product. Examples include backup solutions, high availability (HA) solutions, and shared storage solutions.

When considering a build-it-yourself approach or the use of an open source solution, you should consider the following:

- Service level agreements
- Required skills to build and maintain
- Business impact of a service outage

You should also evaluate the available commercial models for any solutions you intend to buy based on your specific business requirements and functionality each solution provides. Consider the terms of any commercial model, for example, the right to use vs. pay per use charges and how any such charges are calculated.

Best Practice 17.7 – Evaluate the cost benefits of automation

The benefits of adopting automation in AWS can include improved efficiency and productivity, which can translate into lower costs for your organization.

Suggestion 17.7.1 – Evaluate build automation efficiencies

Automation of the build process by using infrastructure as code has cost efficiencies which can improve your time to market and productivity. The advantages of quality, consistency, repeatability, and recoverability that DevOps best practices can introduce need to be balanced against a higher upfront investment in the development of automation.

Working with AWS Professional Services or an AWS Partner can reduce the overall effort by leveraging their experience.

AWS Launch Wizard for SAP can accelerate SAP deployments with automation. It's a service that guides you through the sizing, configuration and deployment of SAP HANA applications on AWS following SAP best practice. The service is available at no additional cost, with support provided by AWS.

- AWS Documentation: Infrastructure as Code
- AWS Documentation: AWS CloudFormation
- AWS Documentation: AWS Launch Wizard for SAP
- SAP on AWS Blog: AWS for SAP DevOps

Suggestion 17.7.2 – Evaluate automation efficiencies for operations

Reduce the cost and manual effort of repeatable tasks by investigating how AWS and third-party tools could be used to automate the running and monitoring of operations. Consider the following:

- AWS Service: AWS Systems Manager

Further guidance can be found in [Operational Excellence] Best Practice 3.6 - Use automation to perform SAP landscape operations (p. 24).

18 – Evaluate SAP compute resources for cost efficiency

How do you assess compute and storage options for your SAP workloads? When implementing or migrating SAP to AWS, you should select cost-effective EC2 instances and storage solutions for the SAP workload to meet your cost targets.

ID	Priority	Best Practice
☐ BP 18.1	Required	Understand the payment and commitment options available for Amazon EC2
☐ BP 18.2	Required	Use cost as a key consideration for EC2 instance selection
☐ BP 18.3	Highly Recommended	Evaluate licensing impact and optimization options
☐ BP 18.4	Highly Recommended	Evaluate the cost impact of storage options based on the required characteristics

Best Practice 18.1 – Understand the payment and commitment options available for Amazon EC2

Consider the use of Reserved Instances and Savings Plans to provide a significant discount compared to on-demand pricing. They are available with 1-year and 3-year commitment terms with three payment options: All Upfront, Partial Upfront, and No Upfront.

Suggestion 18.1.1 – Understand the breakeven points between pricing models

Reserved Instances are categorized into Standard Reserved Instances (up to 72% discount off on-demand rates) and Convertible Reserved Instances (up to 54% discount off on-demand rates). Savings Plans are categorized into Compute Savings Plans (up to 66% discount off on-demand rates) and EC2 Instance Savings Plans (up to 72% discount off on-demand rates).

The discount off the Amazon EC2 on-demand hourly rate you can achieve will depend on the following factors:

- Commitment term selected
- Payment option selected
- Reserved Instance or Savings Plan type selected
- Instance family

Memory-optimized instance families, such as X2, X1, and X1e, provide higher savings for commitment. Therefore, understanding pricing options is important for SAP, particularly for SAP HANA workloads.

Use the advanced option within the AWS Pricing Calculator to determine the break-even point. You should be aware of the assumptions used by this calculator. To illustrate this, consider the example where we use the following formula to determine the point using a Reserved Instance or Savings Plan will provide a lower TCO than using on-demand for each instance family.

*(Effective Hourly rate of Commitment / Hourly rate of On-Demand) * 730 hours*

Reference the effective hourly rate for each RI commitment term and type and for each Savings Plan commitment period and type. Compare and contrast the following examples illustrating different break-even points:

Example 1: In North Virginia (us-east-1), for the M5 family, the breakeven where a 3 year no upfront Standard Reserved Instance or EC2 Savings Plan would offer a lower TCO is 315 hours per month (~16 hrs a day, Monday to Friday).

Example 2: In North Virginia (us-east-1), for the X1 instance family, the breakeven where a 3 year no upfront Standard Reserved Instance or EC2 Savings Plan would offer a lower TCO is 235 hours per month (~12 hrs a day, Monday to Friday).

Use comprehensive guidance on cost management and the Well-Architected Framework Cost Optimization Pillar. The following SAP on AWS Pricing Guide also provides guidance specific to SAP workloads running on AWS. When analyzing costs, be aware that all AWS pricing (with the exception of the AWS China Regions) is in US dollars (USD). However, it is possible to select an alternative currency for payment: currencies AWS currently supports.

- AWS Documentation: Savings Plans - Compute Savings Plans and Reserved Instances
- AWS Documentation: Savings Plans - Plan Types
- AWS Documentation: Types of Reserved Instances

Suggestion 18.1.2 – Understand the considerations of each pricing model relevant to SAP

In addition to the hourly rate discount, there are other benefits of Reserved Instances and Savings Plans you should consider. This AWS Documentation: Comparing Savings Plans to RIs table provides a comparison of Reserved Instances and Savings Plans.

Zonal Reserved Instances can be used to provide capacity reservations within a specific Availability Zone. Savings Plans do not provide a capacity reservation but you can combine with On-demand Capacity Reservations to provide the same features of a Zonal Reserved Instance. See [Reliability]: Best Practice 10.2 - Select an architecture suitable for your availability and capacity requirements (p. 53), for further information on capacity strategies.

Amazon EC2 Spot Instances let you take advantage of unused EC2 capacity in the AWS Cloud. Spot Instances are available at up to a 90% discount compared to On-Demand Instance prices. Spot Instances can be reclaimed by AWS with two-minutes notice when AWS requires the capacity. Therefore, Spot Instances are not generally suited for running SAP workloads.

When using on-demand instances, you should consider the additional operational impact of stopping and starting the SAP systems and underlying EC2 instances based on the required operating hours in addition to application performance impact each time the system is started.

Suggestion 18.1.3 – Evaluate your enterprise strategy for consolidated billing and sharing of Reserved Instance and Savings Plans commitment

With Consolidated Billing, Reserved Instances and Savings Plans are applied to usage across all accounts within an AWS Organization. The management account of an organization can turn off the Reserved Instance discount and Savings Plans discount sharing for any accounts in that organization, including the management account. This means that Reserved Instances and Savings Plans discounts aren't shared between any accounts that have sharing turned off. To share a Reserved Instances or Savings Plans discount with an account, both accounts must have sharing turned on. This preference isn't permanent, and you can change it at any time.

- AWS Documentation: Consolidated billing for AWS Organizations
- AWS Documentation: Turning off reserved instances and Savings Plans discount sharing

A key factor that will determine your strategy for sharing of commitment will be the overall AWS account strategy your organization has adopted. Whether your SAP workloads are running in their own dedicated AWS accounts or along with other workloads hosted in AWS should also be considered. To understand how discounts for Reserved Instances and Savings Plans are applied across your organization's consolidated bill refer to:

- AWS Documentation: Understanding Consolidated Bills

As detailed in SAP note: 1656250 - SAP on AWS: Support prerequisites [Requires SAP Portal Access], SAP on AWS is only supported if a fee-based AWS Support agreement (Business support or higher) is in place. Determine the appropriate support plan based on cost and requirements.

- AWS Documentation: Compare AWS Support Plans

Be aware that AWS calculates support fees independently for each member account within an organization.

Best Practice 18.2 – Use cost as a key consideration for EC2 instance selection

By selecting the appropriate SAP Certified EC2 instances for your workload, it is possible to optimize for cost. Perform a thorough analysis of each system, ensuring that decisions are data driven where possible. Generic guidance can be found in the Well-Architected Framework Cost Optimization Pillar - Cost-Effective Resources.

Suggestion 18.2.1 – Select the latest generation instances available within your Region

The latest generation of Amazon EC2 instances often offers the lowest cost with better performance and should be used if available and certified for the deployment scenario.

- AWS Documentation: Amazon EC2 Instance Types for SAP (includes how to check for instance type availability)

> **Note**
> Some Amazon EC2 instance families (for example X1 and High Memory) might not be available across all Availability Zones within a Region. During planning, confirm that the instance types you require for your SAP workload are available in your target Availability Zones.

Suggestion 18.2.2 – Balance cost with performance requirements

Each SAP supported Amazon EC2 instance family will provide specific performance measured in SAPS. You should evaluate each instance family based on your performance requirements. An understanding of the cost per SAPS and cost per GiB ratios is recommended.

Compute-optimized (C*)	General Purpose (M*)	Memory-optimized (R*)
1 vCPU: 2 GiB	1 vCPU: 4 GiB	1 vCPU: 8 GiB

If a workload component requires more memory over SAPS (CPU), you should select the instance family that provides the lowest cost per GiB memory. If the component requires more SAPS (CPU) over memory, you should select the instance family that provides the lowest cost per SAPS.

SAP Certified instance families powered by an AMD processor typically provide a 10% cost saving over the comparable Intel-based EC2. For example, the C5a is 10% lower cost than the C5 family for the same performance KPIs.

For non-production SAP HANA workloads consider using one of the instance families that meets the requirements detailed in SAP Note: 2271345 - Cost-Optimized SAP HANA Hardware for Non-Production Usage [Requires SAP Portal Access].

Suggestion 18.2.3 – Review the predictability of your growth profile and peak capacity requirements

An existing SAP landscape on AWS or a homogeneous migration is likely to have more predictable growth and usage patterns than a new greenfield implementation or heterogeneous migration.

For systems where you lack data on historical growth, you should consider the cost benefits of selecting an EC2 instance size sufficient for the short or medium term growth. Plan to scale the instance size as your requirement changes. You should ensure that your architecture design provides the flexibility to move between different EC2 instance families as your resource consumption changes.

Similarly, you should evaluate that changes to peak capacity have been accounted for.

When sizing an SAP HANA environment consider not only the database size but also the working memory requirement. Consult SAP HANA sizing reports and tools to estimate your size and usage.

Suggestion 18.2.4 – Consider instance commitment flexibility

When a component (for example, SAP HANA database) needs to scale up during the commitment period, evaluate if this will result in moving to a different instance family. This will impact your pricing model selection.

- AWS Documentation: Amazon EC2 Instance Types

Best Practice 18.3 – Evaluate licensing impact and optimization options

When moving SAP workloads to AWS, there might be commercial impacts with the software licenses your SAP workloads require. You should understand these impacts and the options available to you.

> **Disclaimer**
> Any discussion of Database licensing policies in this document is for informational purposes only and is based on the information available at the time of publication. For more specific information, users should consult their own license agreements with the specific Database Vendor.

Suggestion 18.3.1 – Understand the impact of CPU and memory on software license

Evaluate the different vCPU and memory ratios available with the supported Amazon EC2 Instance Types for SAP to optimize license costs.

- SAP Documentation: SAP HANA Allocated Memory Pools and Allocation Limits

For Oracle based environments, review:

- Oracle License Considerations, Licensing Oracle Software in the Cloud Computing Environment
- Oracle Premium Support requirements detailed in SAP Note: 2069760 - Oracle Linux 7.x SAP Installation and Upgrade [Requires SAP Portal Access]

For Microsoft Windows and SQL Server environments, review:

- AWS Documentation: Microsoft Licensing on AWS
- SAP Note: 2139358 - Effect of changes in licensing terms of SQL Server [Requires SAP Portal Access]

For IBM Db2 environments, review:

- Eligible Public Cloud BYOSL Policy

- AWS Documentation: Track IBM license usage with AWS License Manager

Understand the impact for ISV and third-party products licensed by CPU or memory:

- Consider the use of the Optimize CPU feature to optimize license costs
- Consider the use of AWS License Manager to manage your software licenses and associated costs
- AWS Documentation: Physical Cores by Amazon EC2 Instance Type

Suggestion 18.3.2 – Understand operating system purchasing options

For each of the SAP supported operating systems, there is a set of purchasing options available.

1. Amazon EC2 provided license
2. AWS Marketplace provided license
3. Bring your own licenses (BYOL)

Not all options are available for each operating system. You should evaluate your requirements and licensing agreements to determine which option is the most cost effective. You can include the costs of the following operating systems as part of the Amazon EC2 cost:

- Windows Server
- Red Hat Enterprise Linux
- SUSE Linux Enterprise Server

You can purchase the following operating systems via the AWS Marketplace:

- Red Hat Enterprise Linux for SAP (based on Red Hat Enterprise Linux base EC2 cost)
- SUSE Linux Enterprise Server for SAP (based on Amazon Linux base EC2 cost)

You use bring your own licenses (BYOL) for the following operating systems:

- Windows Server
- Red Hat Enterprise Linux[1]
- SUSE Linux Enterprise Server
- Red Hat Enterprise Linux for SAP[2]
- SUSE Linux Enterprise Server for SAP[2]
- Oracle Enterprise Linux (Oracle Premium Support requirements are detailed in SAP Note: 2069760 - Oracle Linux 7.x SAP Installation and Upgrade) [Requires SAP Portal Access]

[1] Consider SAP Note: 2871484 - SAP supported variants of Red Hat Enterprise Linux [Requires SAP Portal Access] as SAP no longer supports standard Red Hat Enterprise Linux for any SAP workloads as of RHEL 8.

[2] These products have a longer term support which might reduce your operational costs for upgrades – see SUSE Documentation: SUSE Enterprise Support Policy and Red Hat Documentation: Red Hat Enterprise Support Policy for more details.

Suggestion 18.3.3 – Consider the use of Amazon EC2 Dedicated Hosts to mitigate licensing restrictions

Amazon EC2 offers Dedicated Hosts, which allow you to access hardware that's fully dedicated for your use. You can use your own licensed software on dedicated infrastructure. Amazon EC2 Dedicated Hosts

integrate with AWS License Manager, a service which helps you manage your software licenses, including Windows Server and SQL Server licenses.

Suggestion 18.3.4 – Evaluate the cost benefits of moving away from a per gigabyte or per core licensing model

As part of your migration to cloud, consider use of the SAP Runtime database licensing model.

SAP provides the ability for customers to license SAP HANA, SAP ASE and third-party databases under their Runtime database license model. Runtime databases licensed from SAP are solely to support software and SAP named users licensed from SAP. Runtime databases from SAP are licensed as a percentage of the SAP software fee, commonly referred to as the SAP Application Value (SAV).

Runtime licenses are not based on number of gigabytes of memory or CPU cores and therefore can provide a cost benefit over per gigabyte or per core licensing models, particularly when you have multiple non-production systems, as the SAP Runtime database license applies to all environments covered under your SAP license agreement.

If you already have the right to use the SAP HANA Database Runtime license within your SAP license agreement, you should determine if you additionally have the right to use the SAP ASE Database Runtime license for SAP components that cannot use SAP HANA as the underlying database or to reduce the infrastructure costs associated with using SAP HANA for that component.

- Refer to the SAP Documentation: SAP Product Use and Support Guide, or consult with your SAP account team

Best Practice 18.4 – Evaluate the cost impact of storage options based on the required characteristics

Select from object storage, file storage, and block storage services to host, archive, and secure your SAP system. Design your storage to reduce cost and increase agility.

Suggestion 18.4.1 – Evaluate the most cost-effective way to design for the I/O and throughput requirements of your workload

For most SAP requirements, solid state drives (SSDs) are recommended for your EBS volumes. To ensure a flexible, cost-effective selection, we recommend starting with the General Purpose Amazon EBS type gp3, if supported by the instance family. Over time, review the usage using CloudWatch metrics and application/database monitoring. If higher durability or I/O rates greater than 16,000 per volume are required, consider the Provisioned IOPS Amazon EBS type.

- AWS Documentation: Amazon EBS volume types

To balance cost and performance considerations, the storage configuration used for SAP HANA data and log volumes should meet the SAP storage KPI. The storage layouts outlined in the following document have been tested for the SAP TDI guidelines: SAP HANA Tailored Data Center Integration

- AWS Documentation: Storage Configuration for SAP HANA

Suggestion 18.4.2 – Plan for dynamic changes to storage size and configuration

Optimize storage costs by right sizing storage according to data usage or IOPS requirements.

Extend volume size dynamically as required. Evaluate the option of changing volume types during activities that require increased performance such as application upgrades.

- AWS Documentation: Requesting Volume Modifications

Ensure all orphaned or unused volumes are reviewed regularly to ensure cost control.

- AWS Documentation: List Amazon EBS volume or snapshot information

Suggestion 18.4.3 – Evaluate the cost benefits for object storage

The core data for an SAP system is contained within the database and resides on Amazon EBS. Amazon S3 can provide low-cost object storage for auxiliary data, such as backups or archives and large objects such as images or documents. Cost can be further optimized by selecting the appropriate storage type for your retention and durability needs.

Suggestion 18.4.4 – Evaluate the cost benefits for shared file systems

Amazon Elastic File System (Amazon EFS) provides a serverless, set-and-forget, elastic file system that lets you share file data without provisioning or managing storage. Cost can be further optimized by selecting the appropriate storage class based on your performance and availability requirement.

Amazon FSx provides a fully managed highly available and durable file storage solution built on Windows Server. Data deduplication allows you to optimize costs even further by removing redundant data.

Common SAP use cases for Amazon EFS or Amazon FSx include `sapmnt`, transports, interface files, storing backups, and software. Use of Amazon EFS or Amazon FSx can provide cost benefits over deploying your own highly available NFS solution.

- AWS Documentation: Amazon EFS
- AWS Documentation: Amazon FSx

19 – Optimize SAP data usage for storage cost efficiency

How do you optimize your SAP data usage to minimize your storage and memory-related costs? Design your database storage, backup, and supporting file systems to consider cost, and regularly evaluate location, retention, and housekeeping strategies.

ID	Priority	Best Practice
☐ BP 19.1	Required	Understand access and retention requirements
☐ BP 19.2	Highly Recommended	Delete unnecessary data through regular housekeeping
☐ BP 19.3	Recommended	Use compression, reorganization, and reclaim strategies
☐ BP 19.4	Highly Recommended	Review backup strategies for improvements
☐ BP 19.5	Recommended	Consider tiering options for live data
☐ BP 19.6	Recommended	Evaluate archiving and offloading options

Best Practice 19.1 – Understand access and retention requirements

Understand the ways in which you access and retain data. Consider active data, document management systems, and backups.

Suggestion 19.1.1 – Categorize the different types of business data in the SAP system

By categorizing the different types of data and how frequently data is accessed (data temperature) from a business perspective, it is possible to identify opportunities to archive or offload data from your SAP system to optimize cost.

The following are some of the common data types found in an SAP system:

- **Reference** — Data for which the values change infrequently, for example, city, country, and exchange rates
- **SAP Master Data** — Data for which the values change rarely, for example, SAP Customer Master, product
- **Audit** — Data kept for audit purposes, for example, change logs
- **Transaction** — Data created as part of day-to-day business operations, for example, sales orders
- **Analytical** — Data created to support analysis and decision making, for example, monthly sales reporting

Classify the data temperature as follows:

- **Hot** — Data is accessed frequently
- **Warm** — Data is not accessed frequently
- **Cold** — Data is only accessed sporadically

Classify retention requirements as follows:

- Retain for disaster recovery (DR) purposes
- Retain for reference purposes
- Retain for compliance or audit purposes

Best Practice 19.2 – Delete unnecessary data through regular housekeeping

Reduce your data footprint to save costs by minimizing database size and other filesystem usage through regular housekeeping and reorganization activities.

Suggestion 19.2.1 – Review sizing and perform regular housekeeping on SAP technical tables

SAP provides comprehensive guidance on the data management of technical tables. By identifying and addressing the growth of these tables, it is possible to reduce storage and compute costs. This is especially relevant for SAP HANA instances because of the direct relationship between database size and memory requirements.

- SAP Note: 2388483 - How-To: Data Management for Technical Tables [Requires SAP Portal Access]

Use the "largest table" SQL Statements referenced to get comparative table sizes, in particular those marked as Basis tables. A frequent example in established SAP customers is the high number of completed SAP workflow items which could be deleted or archived. Housekeeping prior to a migration can also improve timelines and performance. If using SAP HANA, the report `/SDF/HDB_SIZING` can provide cleanup details and anticipated disk requirement.

Suggestion 19.2.2 – Control filesystem growth through automatic or regular cleanup of logs, traces, interface files, and backups

As storage costs are driven by usage, there are opportunities to optimize the baseline usage, in addition to the multiplier effect of copies and backups of files which are no longer useful for fault analysis.

- SAP Note: 2399996 - How-To: Configuring automatic SAP HANA Cleanup with SAP HANACleaner [Requires SAP Portal Access]

Best Practice 19.3 – Use compression, reorganization, and reclaim strategies

All databases supported by SAP provide mechanisms for reclaiming space. These mechanisms should be part of regular maintenance activities to minimize cost increases associated with extending memory or EBS volumes.

Suggestion 19.3.1 – Use database compression

Compression is a default characteristic in SAP HANA. Use of compression in other databases might require additional licenses but should be explored for cost and performance benefits. The following notes provide a starting point for the various databases, but refer to SAP and database documentation for additional information.

Database	SAP Documentation or SAP Notes
SAP HANA	SAP Note: 2112604 - FAQ: SAP HANA Compression [Requires SAP Portal Access]
SAP ASE	(Consult SAP or Vendor documentation for guidance)
IBM Db2	SAP Note: 1555903 - DB6: Supported IBM Db2 Database Features [Requires SAP Portal Access]
Oracle	SAP Note: 1289494 - FAQ: Oracle compression [Requires SAP Portal Access]
Microsoft SQL Server	SAP Note: 1488135 - Database compression for SQL Server [Requires SAP Portal Access]
SAP MaxDB	(Consult SAP or Vendor documentation for guidance)

Suggestion 19.3.2 – Use database reorganizations and reclaim operations

Space which is unused within the database, due to organic use or targeted archive and cleanup activities, might require a reorganization or reclaim operation to realize the space savings. By reclaiming space regularly, you will reduce the overall growth and requirement for additional storage or memory. The following notes provide a starting point for the various databases, but refer to SAP and database documentation:

Database	SAP Documentation or SAP Notes
SAP HANA	SAP Note: 2499913 - How to shrink SAP HANA Data Volume size [Requires SAP Portal Access]
SAP ASE	SAP Note: 2543407 - reorg rebuild with online - SAP ASE for Business Suite [Requires SAP Portal Access]

Database	SAP Documentation or SAP Notes
IBM Db2	SAP Note: 1942183 - DB6: When to consider a table or index reorganization [Requires SAP Portal Access]
Oracle	SAP Note: 541538 - FAQ: Reorganization [Requires SAP Portal Access]
Microsoft SQL Server	SAP Note: 1721843 - MSSQL: Post-steps after archiving, deleting or compression [Requires SAP Portal Access]
SAP MaxDB	(Consult SAP or Vendor documentation for guidance)

Best Practice 19.4 – Review backup strategy for improvements

When running SAP on AWS, you should evaluate your approach to backups and retention to optimize the costs associated with location, retention, and recovery.

Suggestion 19.4.1 – Evaluate the locations of your backups

Amazon S3 is the suggested long-term storage solution for your SAP system backups for its low cost, durability, and storage class options. To copy the data on your Amazon EBS volumes to Amazon S3, you can use point in time snapshots, integrated database tools, or direct API calls to transfer data.

Snapshots are *incremental* backups, which means that only the blocks on the device that have changed after your most recent snapshot are saved. This minimizes the time required to create the snapshot and saves storage costs by not duplicating data.

Database backup solutions require a knowledge of database state to ensure consistency. AWS provides an SAP HANA backup solution (AWS Backint Agent for SAP HANA) at no additional cost which integrates directly with Amazon S3. For other SAP supported databases, there are database vendors or third-party provided backup tools available which support backing up directly to Amazon S3.

- SAP Documentation: Featured backup solutions

For ad-hoc requirements or as a staging area, it might be necessary to first back up to Amazon EBS. For these use cases, an ST1 volume type is a low-cost HDD volume which provides throughput and performance characteristics suitable for backups. Selecting ST1 can reduce your overall storage costs when the need to back up the SAP database to disk is required.

- AWS Documentation: Amazon EBS volume types

If using Amazon EFS for backups, consider EFS-Infrequent Access. This storage class reduces storage costs for files that are not accessed every day. Amazon EFS One Zone-Infrequent Access is not recommended for backups due to the data only residing in one Availability Zone.

Suggestion 19.4.2 – Review and implement a retention policy for standard backups

To control costs, you need to implement a retention policy aligned with your business requirements that covers the storage services in use.

Amazon S3 offers a range of storage classes designed for different use cases with characteristics such as cost per GB, minimum storage duration charge, and retrieval fee (where applicable). Understanding

the retention and access requirements for your backups will help determine which storage class is most suitable to meet your requirements.

S3 Lifecycle policies can be used to automatically transfer to a different storage class without any changes to your application. For example, backups with shorter retention periods might be better suited to S3 Standard than S3-IA or Amazon S3 Glacier options due to the minimum storage duration charges and retrieval fees associated with these classes. Backups with longer retention periods such as monthly backups for audit purposes are better suited to S3-IA or S3 Glacier dependent on the required retention period.

Amazon Data Lifecycle Manager can be used to automate the creation, retention, and deletion of EBS snapshots and EBS-backed AMIs.

Amazon EFS lifecycle management automatically manages cost-effective file storage for your file systems.

- AWS Documentation: Amazon S3 Storage Classes
- AWS Documentation: Amazon S3 Storage Classes Infographic
- AWS Service: Amazon Data Lifecycle Manager for EBS snapshots and EBS-backed AMIs
- AWS Documentation: Amazon EFS Lifecycle Management
- AWS Service: AWS Backup

Suggestion 19.4.3 – Create a strategy for ad-hoc backups

Ad-hoc backups of a system or component might be required prior to a change or as a reference for system state at a particular point in time. When the required retention does not align with your standard lifecycle policy, you might need to adopt a separate schedule or process for ensuring that storage usage and deletion is cost effective for the individual requirements of that backup.

- AWS Documentation: Amazon S3 Storage Lifecycle Management
- AWS Documentation: Amazon EBS Snapshots Archive

Suggestion 19.4.4 – Review backup setup against recovery approach

Backups are used to revert a system to a previous point in time and to guard against failure scenarios. Ensuring cost efficiency through a robust, but not excessive, use of backup storage requires that you review the recovery approach. Challenge assumptions on requirements for older more granular backups. Determine if these earlier backups would be required in the event of a recovery.

For example, it is a valid strategy to use both database and file system backups. However, if the primary mechanism for recovery is using database restore tools, there might be opportunities to optimize costs by reducing the retention or deleting snapshot backups for some volumes.

- AWS Documentation: Amazon EBS snapshots
- AWS Documentation: AWS Trusted Advisor best practice checklist

Best Practice 19.5 – Consider tiering options for live data

The primary driver of compute cost with SAP HANA is the amount of memory required. Therefore, the use of data offload and tiering options can drive the compute costs down. Although other databases might have tiering options, these have not been highlighted here. Consult with your database provider to understand the available options.

Suggestion 19.5.1 – Evaluate dynamic tiering, extension nodes, and near-line storage (NLS) for SAP HANA OLAP-based workloads

SAP HANA dynamic tiering is an optional add-on to the SAP HANA database to manage historical data. The purpose of dynamic tiering is to extend SAP HANA memory with a disk-centric columnar store (as opposed to SAP HANA's in-memory store) for managing infrequently accessed data. Dynamic tiering can only be used for native SAP HANA use cases and not Business Warehouse (BW) on HANA or BW/4 HANA use cases

An SAP HANA extension node is a special purpose SAP HANA worker node that is specifically set up and reserved for storing warm data. An SAP HANA extension node allows you to store warm data for your SAP Business Warehouse (BW) or native SAP HANA analytics use cases. The total amount of data that can be stored on the SAP HANA extension node ranges from 1x to 2x of the total amount of memory of your extension node.

SAP BW Near-Line Storage (NLS) with SAP IQ allows you to store cold data outside of the BW on HANA or BW/4 HANA database. NLS moves the cold data from the HANA database to store on storage on the SAP IQ Server.

- AWS Documentation: SAP Data Tiering

Suggestion 19.5.2 – Evaluate data aging and SAP HANA Native Storage Extension (NSE) for OLTP-based workloads

Data aging helps free-up SAP HANA memory by storing less frequently accessed data in the disk area.

- AWS Documentation: SAP Data Tiering

Suggestion 19.5.3 – Consider the use of data lakes for large volumes of analytical data

When analyzing SAP and non-SAP data, S3-based data lakes provide a cost-effective option for data storage.

- AWS Documentation: SAP OData connector for Amazon AppFlow
- SAP on AWS Blog: Building data lakes with SAP on AWS

Best Practice 19.6 – Evaluate archiving and offloading options

By considering options to archive infrequently accessed data or offload large objects to near-line storage, you can reduce your infrastructure and backup costs.

Suggestion 19.6.1 – Implement archiving for large tables with infrequently accessed data

Specifically for SAP HANA databases, there are cost benefits of managing your database growth using archiving strategies.

- SAP Documentation: Data Archiving

Suggestion 19.6.2 – Evaluate the archiving tools that support Amazon S3 as a destination

Amazon S3 is designed to be highly available and durable and offers a wide range of cost-effective storage classes. This makes it ideal for storing SAP archive data with the lowest total cost of ownership (TCO).

- AWS Documentation: Amazon S3 Storage Classes
- SAP Documentation: SAP Certified Archiving Solutions

Suggestion 19.6.3 – Use a data management system for large objects

Understand the options and cost benefits for offloading and managing data outside of the SAP database for large objects, such as invoices and images. Consider the business requirements for accessing the data, the implementation effort and the ongoing management complexity.

Large objects will increase your database size, inflating resource and backup costs. Data management system options might provide a lower-cost storage solution.

- SAP Documentation: SAP Document Management
- SAP Documentation: Search for Certified ILM Solutions

20 – Manage costs with visibility, planning, and governance

How do you practice Cloud Financial Management (CFM) to ensure cost optimization and awareness? From inception to operation, how do you establish control over your SAP cloud infrastructure budget and continue to optimize its usage aligned with your business requirements.

ID	Priority	Best Practice
☐ BP 20.1	Required	Plan consumption model and environment usage during project phases
☐ BP 20.2	Required	Establish a multi-year planned cost model taking advantage of different pricing approaches
☐ BP 20.3	Highly Recommended	Establish a budget and mechanisms for cost allocation and tracking including anomaly detection
☐ BP 20.4	Recommended	Establish cost-related approval procedures and controls
☐ BP 20.5	Recommended	Review usage for opportunities to optimize

Best Practice 20.1 – Plan consumption model and environment usage during project phases

During projects, including but not limited to migration or implementation projects, there is often a phased approach to how you deploy systems. There is also a stabilization period where you establish the sizing and usage profile. Take advantage of the flexibility and On-Demand Instance capabilities to minimize your costs during this period.

Suggestion 20.1.1 – Plan to deploy systems only as required

Reduced lead times should give you options to deploy systems only as required. For short lived project systems, use On-Demand Instances to build project systems for the duration of the requirement.

Suggestion 20.1.2 – Evaluate pricing options according to expected duration and usage profile

Project duration and working hours influence the pricing model. An on-demand pricing model is often the default choice at the beginning of a project. Ensure that a budget is defined and evaluated to adapt to cheaper options when appropriate.

Suggestion 20.1.3 – Plan to suspend or decommission systems when not in use

When projects are no longer active or have achieved their objectives, consider the cost savings from shutting down instances, in addition to the storage savings from decommissioning. Typically, a project

will make multiple copies of a system during migration. Remember to shut down systems when they are not being used.

Best Practice 20.2 – Establish a multi-year planned cost model taking advantage of different pricing approaches

Establish a multi-year plan of your capacity requirements to ensure that you are taking full advantage of pricing models to maximize any discounts available from AWS. Baseline and track your costs. Cloud pricing models provide flexibility that allows you to elastically match your infrastructure to changing business requirements. Before making commitments to long-term Savings Plans or Reserved Instances, understand and plan your expected SAP system usage over at least a 3-year horizon. Use testing, SAP Quick Sizer outputs, and growth forecasts to inform a commitment plan and take advantage of the maximum discounts for your workload.

Suggestion 20.2.1 – Establish a capacity estimate with understanding of your key business events

SAP workloads are generally stable, with known usage patterns and hours of operation. Establish a well understood steady state capacity baseline for your SAP systems. This can be done through performance testing and monitoring your production environment during the initial weeks of your deployment.

Extend your steady state capacity estimate to at least a three-year horizon, considering the following:

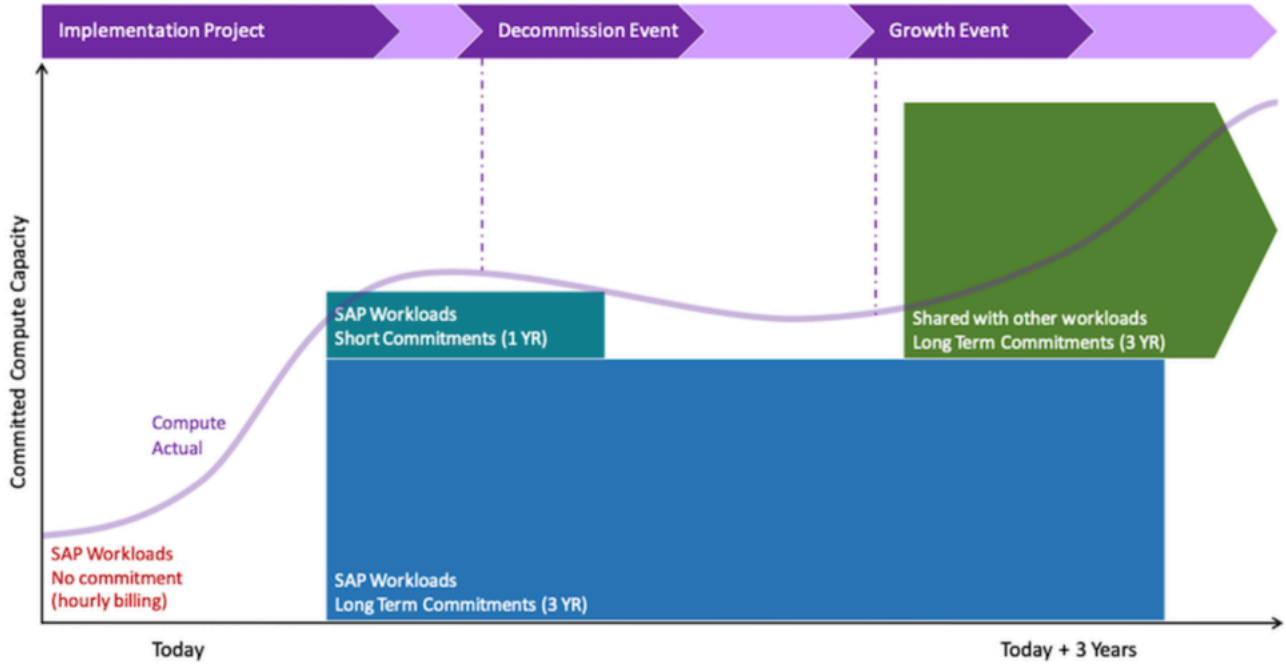
- Major business events, such as mergers, acquisitions, and divestment
- Regulation change that might affect data storage requirements or business process frequency
- Data growth due to normal business operations (particularly important for in-memory databases like SAP HANA as data affects compute sizing in addition to storage sizing)
- Major system upgrades, system replacement, or decommissioning

Suggestion 20.2.2 – Evaluate whether 1-year or 3-year commitments are appropriate

Evaluate how much of your SAP workload could take advantage of three-year committed compute and maximize available discounts by using your capacity estimate. Consider the following:

- Can you consider a three-year commitment for all compute needs of your SAP workload?
- Can you consider a three-year commitment for a subset of your needs which you are confident will not change? For example, SAP primary application servers or database primaries.
- Is your SAP workload part of a broader AWS Organization which could use excess compute commitments when changes in your SAP capacity requirements reduce the need for compute at a future point in time?
- Is your SAP workload part of a broader AWS Organization and could share compute commitments for non-production environments which do not need to be operating 24/7?
- For medium term capacity changes, does the benefit of committing to a 3-year compute plan out-weigh any excess or unused capacity wastage (for example, the break-even point versus a shorter-term commitment is in month 20)?
- Can you consider shorter term commitment (one year) for applications likely to be affected by major business changes or replacement in the short term?
- Are currency fluctuations a concern you need to consider? AWS pricing (with the exception of AWS China) is in US dollars (USD). If a fixed exchange rate is desirable, you might want to consider All Upfront pricing models, if possible.

Establish a plan to match workload capacity requirements with commitment duration to maximize discount.



Example timeline of planning SAP on AWS compute commitments

Suggestion 20.2.3 – Evaluate whether fixing compute types for greater discounts is appropriate

SAP workloads generally only use a limited set of AWS compute types and thus you should consider whether committing to specific compute families or specific instance types is appropriate for your workload to maximize discount. The two highest discount pricing approaches for compute are EC2 Instance Savings Plans and Standard Reserved Instances.

Consider the following:

- Identify frequently used compute types in your landscape and consider purchasing specific EC2 instance Savings Plans or Standard Reserved Instances for these. For example, if you are using `m5.xlarge` for your application servers across multiple SAP applications. This is a good candidate for an EC2 specific savings plan or Standard Reserved Instance as it is highly likely that you will always be using this commitment.
- Identify compute components which are highly likely to change EC2 families due to growth workloads or business events. Consider purchasing more generic compute savings plans or Convertible Reserved Instances for these items. For example, if you have an SAP HANA database which needs to move between an EC2 `r5` and `x1e` compute family due to a size increase after only 6 months. This is a good candidate for a short-term Convertible Reserved Instance or compute savings plan.
- Identify break-even points for general compute vs. specific compute pricing and take this into account when choosing your commitment type. For example, it might be cheaper to purchase a Standard Reserved Instance for three years rather than choose a 3-year Convertible Reserved Instance if your sizing change is in year 3. You might also consider selling the residual Reserved Instance value on the AWS Reserved Instance Marketplace.
- Before changing instance types, identify use of AWS Marketplace seller private offer or annual subscription software. This will avoid incurring additional software costs. Both plans offer savings by allowing you to commit to running software products on an Amazon EC2 Instance for specified duration. For example, the purchase of an annual subscription for software to run on an `r4.xlarge` instance. You decided to change the instance type to `r5.xlarge` . The annual subscription is no longer linked to the instance but is still active. This results in additional on-demand pricing for the

software on the `r5.xlarge` . Consider waiting for the annual subscription to expire before changing the instance size.

Suggestion 20.2.4 – Evaluate whether Savings Plans, Reserved Instances, or both are more appropriate

Choose a mixture of both Savings Plans and Reserved Instances, to obtain benefits from the different models, if appropriate for your SAP workload. Determine your commitment periods and compute requirements holistically and then select your pricing model.

Further information on the differences between Savings Plans and Reserved Instances can be found in [Cost Optimization]: Best Practice 18.1 - Understand the payment and commitment options available for Amazon EC2 (p. 96) and in Compute Savings Plans and Reserved Instances.

Consider [Reliability]: Best Practice 10.2 - Select an architecture suitable for your availability and capacity requirements (p. 53). It discusses the capacity reservation differences and tradeoffs between Savings Plans and Reserved Instances.

Suggestion 20.2.5 – Convert your capacity plan into a cost model for budgeting and tracking purposes

Convert your Savings Plans, Reserved Instance choices, and on-demand budget into a cost plan that estimates your AWS spend for your SAP landscape over at least three years. Combine your compute estimate with other AWS costs to finalize your SAP workload cost model for budgeting and tracking purposes.

When estimating your SAP costs, remember to include the following:

- Compute-attached storage costs (such as Amazon EBS volumes)
- Shared file storage costs (such as Amazon EFS, and Amazon FSx)
- Backup storage costs (such as Amazon S3, and Amazon S3 Glacier)
- Network and VPC costs (such as Elastic Load Balancers, NAT gateways, Transit Gateways, Network outbound costs, Direct Connect, and VPN)
- Management and governance service costs (such as CloudWatch detailed metrics, AWS CloudTrail, and AWS Config)
- Security service costs (such as AWS WAF, Amazon GuardDuty, and AWS Shield)
- AWS Support Costs (Business or higher)
- Consider enterprise discount programs or volume discounts that you might be eligible for (speak to your AWS account manager for further details)
- Currency: Be aware that AWS prices are in US dollars (USD). You can choose a billing currency and your bills will be computed in USD and converted to your preferred currency at a competitive exchange rate

Best Practice 20.3 – Establish a budget and mechanisms for cost allocation and tracking including anomaly detection

There are guidelines in the Well-Architected Framework for implementing financial management. Set expectations around cloud costs with annual, quarterly, monthly, or even daily budgets depending on your business needs. Adjust forecasts regularly to align with usage, and identify patterns and anomalies. Establish mechanisms for cost allocation using account and tagging strategies.

Suggestion 20.3.1 – Use cost and billing tools to gain spend visibility

SAP systems are often static in their usage patterns once established. If you use an on-demand pricing model, either on a permanent basis or during project phases, you might see a fluctuation in Amazon EC2

costs. If data volume management strategies are not put in place, Amazon EBS and Amazon S3 costs might be higher than expected.

AWS provide a suite of Cloud Financial Management Services including the following:

- • AWS Billing Conductor allows you to construct a cost allocation strategy that aligns with your business logic.
  - AWS Budgets can be used to set custom budget based on your expected usage and notify you when a threshold is exceeded.
  - AWS Cost Anomaly Detection uses advanced machine learning (ML) technologies to identify anomalous spend and root causes.
  - AWS Cost Explorer provide tools for visibility and analysis.

Further guidance can be found in the Well-Architected Framework [Cost Optimization]: Expenditure and Usage Awareness.

Suggestion 20.3.2 – Analyze and allocate spend using tags

You can create cost allocation tags that help identify pricing of AWS resources based on individual accounts, resources, business units, and SAP environments. These tags are then visible within the AWS billing reports and can be analyzed using Cost Explorer. You can use cost allocation tags to determine the costs associated with individual SAP environments. They help inform if action should be taken to reduce or remove costs associated with specific environments, such as temporary environments or project environments that are no longer required. You should have a process to identify resources that do not have cost allocation tags attached. Implement the actions required to add cost allocation tags to these resources.

- SAP on AWS Blog: Tagging recommendations for SAP on AWS

Best Practice 20.4 – Establish cost-related procedures and controls

It might be necessary to adapt traditional cost assessment processes to be cloud ready. Gain familiarity with how to implement the right financial practices and policies by reviewing the AWS Cloud Financial Management Guide.

Suggestion 20.4.1 – Educate administrators on cost implications

Introduce mechanisms to assign accountability and provide incentives for cost optimization.

Suggestion 20.4.2 – Only allow certain users the ability to provision instances using IAM controls

Use IAM policies aligned with resource type and job function within account boundaries to ensure cost control. For example, you might allow additional small-scale systems in a sandbox account to be controlled within a project team but have an additional approval process and restricted access for larger instances in a production account.

Best Practice 20.5 – Review usage for opportunities to optimize

Review your SAP workload periodically to identify opportunities to optimize cost. Regular reviews should focus on: minimizing the differences and anomalies found between your AWS bill and your SAP workload budget, checking that all your SAP cloud resources are appropriately sized and not over-provisioned, and understanding any new AWS service releases or cost reductions that could improve the cost effectiveness of your SAP workload.

Suggestion 20.5.1 – Minimize additional cost where your usage has been higher than initially planned

Your cloud usage might have grown outside of your estimated cost model due to unplanned business events or additional performance required. Analyze these changes with a view to optimizing the new cost. Consider additional Savings Plan commitments or Reserved Instances if this is a sustained change.

Where additional capacity is required for only short periods, consider horizontal scaling mechanisms (for example, additional SAP application servers) using automatic scaling or scheduled On-Demand Instance capacity to minimize cost further.

Suggestion 20.5.2 – Review SAP workload usage metrics and further right-size where possible

Regularly review the components supporting your SAP system to ensure they are right-sized. Use CloudWatch metrics to consider:

- Is the SAP EC2 compute the right size? Is CPU or memory utilization low? Could you move to a smaller EC2 instance size?
- Is SAP storage the right size? Is there excess space provisioned but unused? Could you reduce volume sizes?
- Is SAP storage appropriately performant? Is there excess IOPS or MBPS provisioned which could be reduced?
- Are backup and snapshots being managed appropriately? Do you have too many backup copies on S3 Standard which could be archived to Amazon S3 Infrequently Accessed or Amazon S3 Glacier?
- Use tools such as AWS Compute Optimizer and AWS Trusted Advisor to identify additional areas for optimization. Be aware of SAP specific compute and storage restrictions as per SAP note: 1656099 - SAP Applications on AWS: Supported DB/OS and Amazon EC2 products [Requires SAP Portal Access].

Use your findings to continually right-size your SAP workload components on a regular basis and maximize your use of Savings Plans and Reserved Instances.

Suggestion 20.5.3 – Understand new AWS services and plan to implement where further cost optimization can be achieved

AWS regularly releases new services and periodically reduces prices. Review new SAP on AWS service announcements and plan to take advantage of these in your architecture at a minimum every 12 months. If you have a technical account manager (TAM) as part of an Enterprise Support agreement with AWS, they can assist you in a regular new service briefing and optimization discussion.

Subscribe to the SAP on AWS blog and the What's New feed for the latest announcements and news.

See [Operational Excellence]: Best Practice 4.4 - Perform regular workload reviews to optimize for resiliency, performance, agility, and cost (p. 28) for further information on continued optimization of your SAP workload.

Sustainability

The sustainability pillar of the SAP Well-Architected Lens focuses on incorporating environmental sustainability best practices into the architectural designs and operations of your SAP systems. Specifically, this involves designing your SAP systems to minimize negative environmental consequences. This minimization is accomplished by architecting your SAP systems and performing your SAP development in a manner that reduces energy consumption by the underlying AWS resources. After these steps, monitor your sustainability posture and opportunities for iterative improvement over time. As always, review the overall AWS Well-Architected Framework Sustainability Pillar for recommendations that apply to all workloads, including SAP.

As a rule, the recommendations in the SAP Lens cost optimization pillar will result in a more sustainable architecture for SAP workloads, except as called out in the following sections. To avoid duplication of

recommendations, we advise that you review that pillar in addition to the best practices contained in this one.

- SAP Lens [Cost Optimization]: Cost optimization (p. 87)
- AWS Documentation: AWS Well-Architected Framework Sustainability Pillar

21 – Evaluate SAP architecture patterns to improve environmental sustainability

How do you design your SAP workload to minimize its environmental impact? SAP architectures, as a rule, become more environmentally sustainable as underlying infrastructure footprints and energy utilization are reduced. For most cases where the SAP workload resides in AWS, this aligns with a more cost-optimized cloud infrastructure. For instance, running fewer instances with a higher utilization, as described in the Well-Architected Framework Sustainability Pillar, is a standard method of achieving a lower-cost footprint. Business stakeholders and SAP architecture teams must keep in mind that this can require changes in business objectives (p. 91) to prioritize sustainability over performance goals. In addition, stakeholders should encourage the adoption of iterative code development practices that optimize for energy efficiency over time.

ID	Priority	Best Practice
☐ BP 21.1	Required	Understand business requirements to make sustainability-centric design decisions
☐ BP 21.2	Required	Implement sustainability improvements for infrastructure and SAP
☐ BP 21.3	Required	Implement sustainability monitoring for infrastructure and SAP

- SAP Lens [Cost Optimization]: Best Practice 17.3 – Understand business requirements to make cost-optimized design decisions per environment (p. 91)
- Well-Architected Framework [Sustainability]: Use the minimum amount of hardware to meet your needs
- Well-Architected Framework [Sustainability]: Development and deployment process

Best Practice 21.1 - Understand and influence business requirements to make sustainability-centric SAP design decisions

Business leaders and SAP architecture teams must make conscious decisions to design their SAP environments in a more sustainable fashion. They must understand and justify the rationale behind those choices. If business leaders cannot describe what tradeoffs they must make in the language of sustainability and energy efficiency, they will have difficulty making decisions based on those criteria.

Production SAP workloads are mission critical to most businesses who run SAP, with companies often prioritizing reliability and performance efficiency over infrastructure costs. While prioritizing sustainability almost always results in cloud infrastructure cost reductions, it might not create overall cost savings for an SAP workload. Running SAP workloads in a more sustainable fashion in AWS can increase the costs of software, personnel, or overall business functions. To meet sustainability goals, these tradeoffs along with other priorities must be clearly articulated, agreed upon, and measurable.

Suggestion 21.1.1 – Define criteria to measure and understand your sustainability impact

Understanding your sustainability goals is the first step to ensuring that you focus on the factors needed to meet those goals. Defining such criteria involves adopting metrics that can be used to measure and evaluate your current sustainability posture, report progress against goals, and accelerate improvements. By analyzing the current environmental impact of the underlying cloud-based SAP infrastructure, you can quantify the tradeoffs and changes required to meet your sustainability objectives.

For example, the sustainability pillar of the AWS Well-Architected Framework provides an introduction to emissions accounting, including examples of the scopes defined by the Greenhouse Gas Protocol. Keep in mind that emissions from your SAP workloads in AWS would count as a portion of your indirect Scope 3 emissions under these definitions. Scope 3 emissions can be measured by a sustainability metric known as CO2e (carbon dioxide equivalent). While not the only measure of sustainability, CO2e is commonly used to measure and compare emissions from greenhouse gases based on how severely they contribute to global warming, and shows how much a particular gas would contribute to global warming if it were carbon dioxide.

Given that the previously mentioned data is not directly obtainable in most SAP monitoring scenarios, you can use sustainability proxy metrics instead. Proxy metrics allow SAP architecture teams to evaluate correlated improvements made to a workload instead of real-time carbon metrics. Defining proxy metrics across compute, storage, and network infrastructure can help you understand how infrastructure changes can impact sustainability results. Example proxy metrics include vCPU minutes for compute, GBs provisioned for storage, and GBs transferred for network traffic. Proxy metrics combined with business metrics can define sustainability KPIs, which can be used to drive sustainability optimizations while keeping business needs in focus. One example would be to measure vCPU minutes per transaction and define an improvement goal to minimize this metric. Business stakeholders would have to weigh the cost, as reducing vCPUs could ultimately become detrimental to delivering on business needs. When running SAP workloads in AWS, the change in these measured resources correlates with a similar change in cost (except as noted in the following), making overall infrastructure spend a useful proxy metric.

By agreeing on a set of sustainability metrics, the SAP architect team can evaluate different technical approaches to reduce environmental impact. A small gap between the current and target sustainability goals and your chosen metrics might result in small architectural changes, while a large gap would require more drastic measures.

- Well-Architected Framework [Sustainability]: Cloud sustainability
- Well-Architected Framework [Sustainability]: Evaluate specific improvements

Suggestion 21.1.2 – Work with the business to establish more sustainable SLAs and only architect for the SLA negotiated

Many businesses define their SAP Service Level Agreements (SLAs) based on demands for minimal downtime, point-in-time recoverability, and response times that are as close to instantaneous as possible. SAP architecture teams can overestimate the risks and impact of downtime. As a result, they put in place SAP architectures that are likely to achieve a much higher performance objective or availability target than the agreed-upon business SLA. For example, a business might establish a warm Regional standby disaster recovery landscape when local cross-AZ (Availability Zone) high availability with backups in another Region would suffice to meet the RPO and RTO negotiated with the business. Similarly, retaining multiple backups of SAP test systems that are easily re-created results in the use of unneeded resources. This extra *insurance* increases the environmental impact (along with the cost) beyond what is acceptable for risk mitigation by the business.

SAP architecture teams can work with the business to negotiate SLAs with sustainability in mind. For example, rather than having a goal of instantaneous response for ad hoc queries on an S/4HANA system, a business can dictate that certain queries must be submitted in batch mode with a much lower SLA for processing. These reports could then run during periods of lower utilization, reducing the peak processing required and hence the necessary size of the EC2 instances. As another example, operational SLAs that prioritize response time over evaluating the environmental impact of the task, such as

overprovisioning storage to avoid having to do so again in the near future, should be reconsidered with sustainability in mind.

While these changes also align with cost optimization pillar of the SAP Lens, businesses often prefer higher infrastructure costs when faced with the potential for lower productivity or increased risk that less stringent SLAs may bring. When a business prioritizes sustainability, the higher cost of doing business may be acceptable.

- AWS Documentation: Multi-Region Architecture Patterns
- SAP Lens [Performance Efficiency]: Best Practice 13.1 – Evaluate or estimate performance requirements (p. 70)
- Well-Architected Framework [Sustainability]: Align SLAs with sustainability goals
- Well-Architected Framework [Sustainability]: Optimize software and architecture for asynchronous and scheduled jobs

Suggestion 21.1.3 – Understand how changes to SAP end user behavior can result in more sustainable SAP architectures

Businesses who want to run SAP more sustainably must understand how their end users are accessing the system regardless of negotiated SLAs. If all users arrive at 9:00 AM and run large reports simultaneously, this can create a CPU peak that can require resizing to larger EC2 instances. SAP architecture teams must understand the relationship between SAP user access patterns and the associated hardware footprint to design an SAP architecture that minimizes the correlated environmental impact of using more underlying infrastructure. SAP architecture teams can work with business stakeholders to encourage changes to user behavior (for example, staggered business start times and financial chargeback penalties to those with the highest utilization) to promote more sustainable SAP usage patterns and architectural designs.

Well-Architected Framework [Sustainability]: User behavior patterns

Best Practice 21.2 - Implement sustainability improvements for SAP software and underlying infrastructure

After the business requirements and criteria for sustainability improvements have been defined, the SAP architect team must implement them. Given that in most cases with standard SAP architectural patterns (except for those called out in the following sections), the recommendations in the SAP Lens cost optimization pillar directly align with sustainability best practices. We will not duplicate those suggestions here and refer you to the cost optimization pillar for guidance.

Suggestion 21.2.1 - Develop or redevelop your SAP code with sustainability in mind

Highly customized SAP systems require more resources to run as a general rule. Complex, bespoke programs can lead to statements, transactions, and reports that require additional CPU and memory beyond optimized native SAP transactions. Adopting sustainable development best practices, despite the additional development time, can notably improve the code performance and reduce infrastructure usage. This impact can be magnified for frequently run code.

Avoiding bespoke development entirely and staying with standard, well-tuned SAP transactions is one approach. Where development is required, optimize queries or use tools such as the ABAP Test Cockpit to improve the performance of your code. As mentioned in Suggestion 21.2.2, keep your SAP software version as current as is feasible. This helps reduce the need for specialized development, which otherwise would not be necessary due to newly introduced or improved functionality.

Well-Architected Framework [Sustainability]: Optimize areas of code that consume the most time or resources

SAP Documentation: ABAP Test Cockpit

Suggestion 21.2.2 – Perform regular patch management

Performing regular patch management ensures that your system will benefit from the latest SAP performance enhancements while avoiding unnecessary development (see Suggestion 21.2.1). By keeping your SAP systems and related software up-to-date, SAP operations teams can avoid more intensive patching and upgrade activities. These activities can require additional temporary hardware, which would add to measurable environmental impact. This suggestion is in direct alignment with the SAP Lens security and operational excellence pillars suggestions regarding SAP patching activities.

- SAP Lens [Operational Excellence]: Best Practice 4.2 – Regularly perform patch management for software currency
- SAP Lens [Security]: Best Practice 6.4 – Establish a plan for upgrading and patching all applicable software

Suggestion 21.2.3 - Implement data classification and tiering

As mentioned in the cost optimization section of this lens (p. 87), storage optimization should be a focus for reducing costs but can also be justified from a sustainability perspective. By default, data and objects in SAP systems are not archived nor moved to more energy efficient storage classes (or out of memory, in the case of HANA-based systems) as the data ages. Options such as HANA NSE Data Tiering, HANA extension nodes, Data Tiering Optimization, and data archiving or deletion should be evaluated. In addition, using lifecycle policies for database backups and snapshots can automate maintaining a more sustainable data footprint, as can evaluating whether the number of backups to retain is appropriate given the SLA. The use of tools such as SAP TDMS can also reduce the footprint of non-production systems by reducing the overall database footprint and associated storage.

- Well-Architected Framework [Sustainability]: Implement a data classification policy
- SAP Lens [Cost Optimization]: Best Practice 19.5 – Consider tiering options for live data (p. 106)
- Well-Architected Framework [Sustainability]: Use lifecycle policies to delete unnecessary data
- SAP Documentation: SAP Test Data Migration Server

Suggestion 21.2.4 - Favor scale-out architectures where possible

As described in the Cost optimization (p. 87) and Performance efficiency (p. 69) pillars of this lens, scale-out architectures can mitigate the risk of over-provisioning by adding smaller compute capacity whenever required. SAP application servers inherently are scale-out, so plan to use this capability most efficiently for the demand required. For databases, scale-out may or may not be possible. In situations where it is, the sustainability characteristics of incremental growth may outweigh the operational overhead of managing a scale-out environment.

- Well-Architected Framework [Sustainability]: Scale infrastructure with user load
- SAP Lens [Performance Efficiency]: Best Practice 13.3 – Select architectures that allow for independent scaling of systems or components (p. 72)
- SAP Lens [Cost Optimization]: Best Practice 17.4 – Review the size, granularity, and latest available EC2 instances for SAP components (p. 92)

Suggestion 21.2.5 - Shut down or terminate EC2 instances covered by Reserved Instances or Savings Plans even when there are no additional cost savings

Amazon EC2 Reserved Instances (RI) and Savings Plans provide lower prices compared to On-Demand pricing in exchange for specific usage commitments. The possibility exists that your compute usage could drop below this commitment at times, presenting the rare case for SAP systems where a more sustainable architecture is not directly tied to reducing your AWS spend. In such cases, your sustainability posture might still be improved despite no additional cost savings when following standard cost

optimization best practices. This guidance includes shutting down unused non-production systems, scaling in application servers during periods of low utilization, transitioning EC2 instances to the latest generation to improve CPU utilization, and minimizing the size of pilot-light HA/DR systems.

- SAP Lens [Cost Optimization]: Best Practice 18.1 – Understand the payment and commitment options available for Amazon EC2
- SAP Lens [Cost Optimization]: Best Practice 17.4 – Review the size, granularity, and latest available EC2 instances for SAP components
- Well-Architected Framework [Sustainability]: Use the minimum amount of hardware to meet your needs

Suggestion 21.2.6 – Consider sustainability when choosing AWS Regions

SAP landscapes are frequently deployed in AWS based on such factors as proximity to end users, data residency requirements, infrastructure cost, and compliance with specific governmental regulations. In a business that prioritizes sustainability, the environmental impact of deploying SAP workloads should also be considered when choosing the appropriate AWS Region. Regions that are near an Amazon renewable energy project may be more desirable in such cases.

- SAP Lens [Performance efficiency]: Best Practice 13.4 – Choose Regions and Availability Zones to minimize latency
- SAP Lens [Security]: Best Practice 5.2 – Classify the data within your SAP workloads
- Well-Architected Framework [Sustainability]: Region selection

Suggestion 21.2.7 - Leverage managed services whenever possible

A number of non-NetWeaver SAP products can be installed on AWS services that are managed above the infrastructure layer in a more sustainable fashion. One such example is the use of Amazon RDS for SAP BusinessObjects BI Platform. The use of managed services for common SAP-related functions is also advisable. Some examples include using AWS Backup for AMI and Amazon EBS snapshot management or using Amazon AppFlow for bidirectional data flow with your Amazon S3-based data lake.

Using managed services can reduce capacity guesswork and allow for AWS internal provisioning mechanisms to maximize underlying resource efficiency. As an example, where the option exists for your SAP systems, use Amazon Elastic File System (Amazon EFS) instead of Amazon EBS. Amazon EFS allows for the use of only the precise amount of storage required, while EBS uses allocated space that can leave a portion unused. As another example, Amazon EC2 instances may be removed from your environment and replaced by managed services, such as removing bastion hosts in public subnets and instead using AWS Systems Manager Session Manager for direct access.

- Well-Architected Framework [Sustainability]: Use managed services
- AWS Documentation: CMS and Audit Database Architecture Options
- AWS Documentation: SAP NetWeaver on AWS Backup and Recovery
- AWS Documentation: Amazon AppFlow Integrations

Best Practice 21.3 - Implement sustainability monitoring for infrastructure and SAP

Monitoring and reporting on the sustainability of SAP workloads in the AWS Cloud provides a crucial feedback mechanism. Such monitors indicate how suggestions you've implemented translate into quantifiable changes over time. This data also feeds into the sustainability reporting that will be delivered to shareholders, regulators, and sustainability-minded customers. Reports using the metrics discussed in BP 21.1 (p. 114) (for example, cost and usage as proxy metrics) can demonstrate

improvements made in the operational sustainability of your SAP landscape. This can demonstrate that you are successfully achieving the goals set by your overall corporate sustainability strategy.

- Well-Architected Framework [Sustainability]: Optimize areas of code that consume the most time or resources

Suggestion 21.3.1 - Develop sustainability-centric monitoring and reporting

Monitoring is vital to understanding the impact of changes applied to your SAP workloads to improve their overall sustainability. Establish a mechanism to monitor the sustainability of your AWS Cloud consumption based on common, standardized metrics. The AWS Customer Carbon Footprint Tool can be used to estimate the carbon emissions of AWS products and services that underlie your SAP systems. Greenhouse gas emissions are converted to the amount of carbon dioxide that would result in equivalent warming and are denoted by the services to which they are associated. Given that this reporting is AWS account-specific, separating your SAP systems into separate accounts from other workloads might be necessary to achieve the maximum benefits of the tool. That being said, a multi-account strategy is typically based on the security requirements of your organization, so refer to the SAP Lens security pillar guidance on this topic.

SAP also provides their SAP Sustainability Control Tower solution to expand reporting beyond the underlying SAP infrastructure costs to infrastructure costs and ultimately the overall business's sustainability posture.

- AWS Documentation: Understanding your customer carbon footprint tool
- Well-Architected Framework [Sustainability]: Evaluate specific improvements
- Well-Architected Framework [Security]: Assess the need for specific security controls for your SAP workloads (p. 32)
- SAP Documentation: SAP Sustainability Control Tower

Suggestion 21.3.2 - Periodically baseline and review reported results

As described in Suggestion 21.1.1, progress of an organization's sustainability initiative over time should be based on an initial reference position. As part of setting up the relevant sustainability monitoring tools, establish the baseline configuration data and reporting, from which progress will be tracked. Baselining should include the following considerations:

- Tagging of relevant SAP workloads to allow for more granular reporting.
- Current carbon dioxide-equivalent (CO2e) or other proxy metric for workloads running in AWS.

To ensure your organization's use of AWS services for SAP and their sustainability trajectory are monitored against established KPIs and overall business goals, establish a periodic reporting review as part of an improvement process. This review should include the following activities:

- Validate all relevant SAP workloads are being monitored appropriately, including new workloads added since data gathering was last baselined.
- Measure the results, look for gaps, and replicate areas of success.

Aligning these activities with the best practices from the SAP Lens operational excellence pillar can help you perform standardized sustainability reviews with other periodic operational activities. For example, discovering and removing unused resources, such as orphaned storage volumes or low-utilization instances, should be standard operational review tasks that also reduce environmental impact.

- Well-Architected Framework [Sustainability]: Improvement process
- Well-Architected Framework [Sustainability]: Measure results and replicate successes

- Well-Architected Framework [Operational Excellence]: Validate and improve your SAP workload regularly

Conclusion

This lens provided architectural guidance for designing, building, and operating reliable, secure, efficient, and cost-effective SAP workloads on AWS. We captured common patterns, lessons learned, and overarching SAP design recommendations.

Applying the framework to your workload helps you build robust, stable, and efficient systems, leaving you to focus on high value operational tasks in your SAP workloads and pushing the boundaries of the field to which you're committed.

The SAP ecosystem continues to evolve as the applications, tools, and processes grow and mature. As this evolution occurs, we will continue to update this paper to help you ensure that your SAP applications are well-architected.

Contributors

The following individuals and organizations contributed to this document:

- Peter Perbellini: ERP on AWS Specialist Solutions Architecture Lead, APJ, Amazon Web Services
- Adam Hill: Worldwide Tech Leader - SAP, Amazon Web Services
- Nerys Olver: Solutions Architect, SAP, Amazon Web Services
- Christopher Spruell: Principal Global Account SA, Amazon Web Services
- Eneko Bilbao: Principal SAP Specialist SA, Amazon Web Services
- Venkat Tatavarthy: Senior Partner SA, Amazon Web Services
- Bruce Ross: Senior Solution Architect: Well-Architected Lenses, Amazon Web Services
- Sabari Radhakrishnan: Principal SAP Specialist SA, Amazon Web Services
- Somckit Khemmanivanh: Senior System Development Engineer of EC2 Enterprise, Amazon Web Services
- Sander Bleijenbergh: Senior SA of ISV, Amazon Web Services
- Harpreet Singh: Senior Manager of Partner SA, Amazon Web Services
- Manoj Muthukrishnan: Senior SAP Specialist SA, Amazon Web Services
- John Studdert
- Wilson Puvvula
- Jongnam Lee
- Ben Potter

Document revisions

To be notified about updates to this whitepaper, subscribe to the RSS feed.

Change	Description	Date
Minor update (p. 123)	Corrected incorrect link.	October 28, 2022
Minor update (p. 123)	Updated text in Suggestion 1.1.2.	October 12, 2022
Whitepaper updated (p. 123)	Sustainability section added, custom lens link, and numerous changes throughout.	October 4, 2022
Minor update (p. 99)	Disclaimer added to Best Practice 18.3.	April 6, 2022
Minor update (p. 123)	Corrected broken links.	January 20, 2022
Initial publication (p. 123)	SAP Lens first published.	October 28, 2021

Note
To subscribe to RSS updates, you must have an RSS plug-in enabled for the browser you are using.

Design principles arranged by pillar

These are the design principles outlined in this paper organized by pillar of the AWS Well-Architected Framework.

Operational excellence (p. 7)

Security (p. 29)

Reliability (p. 51)

Performance efficiency (p. 69)

Cost optimization (p. 87)

Sustainability (p. 113)

Notices

AWS glossary

For the latest AWS terminology, see the AWS glossary in the *AWS General Reference*.