
AWS DevOps Monitoring Dashboard Implementation Guide



AWS DevOps Monitoring Dashboard: Implementation Guide

Copyright © Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

Welcome	1
Cost	2
Architecture overview	3
Solution components	4
Amazon EventBridge Events rule	4
Amazon Kinesis Data Firehose	4
AWS Lambda	4
Amazon Simple Storage Service (Amazon S3)	5
AWS Glue and Amazon Athena	5
Amazon QuickSight	5
Amazon CloudWatch synthetics canary and Amazon CloudWatch alarm	5
Security	6
AWS IAM roles	6
Amazon S3	6
Deployment considerations	7
Regional deployment	7
AWS CI/CD pipeline deployment	7
Amazon QuickSight deployment	7
Amazon CloudWatch synthetics canary and Amazon CloudWatch alarm deployment	7
AWS CloudFormation template	8
Automated deployment	9
Prerequisites	9
Deployment overview	9
Step 1: Launch the stack	9
Step 2: Configure Amazon QuickSight	11
Resources	13
Update the stack	14
DevOps metrics list	15
Code change volume metrics	15
Mean time to recover metrics	15
Change failure rate metrics	15
Deployment metrics	15
Build metrics	15
Pipeline metrics	16
Retrieve the Amazon QuickSight Principal ARN	17
Set up Amazon CloudWatch synthetics canary and Amazon CloudWatch alarm	18
Database schema information	21
Amazon QuickSight dashboards visuals	24
Code change volume dashboards	24
Mean time to recover dashboards	24
Change failure rate dashboards	25
Deployment dashboards	26
Build dashboards	27
Build activity dashboards	27
Build resource utilization dashboards	30
Pipeline dashboards	33
Build visualizations with Amazon Athena and Tableau	36
Running queries and work with query results and output files in Amazon Athena	37
Uninstall the solution	38
Using the AWS Management Console	38
Using AWS Command Line Interface	38
Collection of operational metrics	39
Source code	40
Revisions	41

Contributors	42
Notices	43

Ingesting, analyzing, and visualizing metrics with AWS DevOps Monitoring Dashboard

AWS Implementation Guide

Solutions Builder Team

Publication date: March 2021 ([last update \(p. 41\): June 2021](#))

Collecting performance and operational metrics in your continuous integration/continuous delivery (CI/CD) pipeline is important in order to measure your return on investment in DevOps automation. These metrics also inform you about how to improve your software delivery process. However, the process of aggregating, analyzing, and visualizing metrics from various components through the pipeline can be complex and time-consuming.

The AWS DevOps Monitoring Dashboard solution automates the process for monitoring and visualizing CI/CD metric following AWS best practices. This solution allows organizations of all sizes to track and measure the activities of their development teams. This helps DevOps leaders measure the impact of their DevOps initiatives and make data-driven decisions to drive continuous improvement in their development teams.

This solution supports ingestion, analysis and visualization of data from [AWS Developer Tools](#) to calculate key DevOps metrics, such as Mean Time To Recover, Change Failure Rate, Deployment and Code Change Volume. For a complete list of metrics, refer to [DevOps metrics list \(p. 15\)](#). These metrics are presented in [Amazon QuickSight](#) dashboards for visualization. For more information about data visualization, refer to [Amazon QuickSight dashboards visuals \(p. 24\)](#).

You can also use other visualization tools, such as Tableau, to build visualizations from the Amazon Athena database. For more information, refer to [Build visualizations with Amazon Athena and Tableau \(p. 36\)](#). To directly work with query results and output files in Amazon Athena, refer to [Running queries and work with query results and output files in Amazon Athena \(p. 37\)](#).

This implementation guide describes architectural considerations and configuration steps for deploying this solution in the Amazon Web Services (AWS) Cloud. This solution's [AWS CloudFormation](#) template launches and configures the AWS services required to deploy the solution using AWS best practices for security, availability, performance efficiency, and cost optimization.

This solution is intended for deployment in an enterprise by IT infrastructure architects, administrators, developers, and DevOps professionals who have practical experience with the AWS Cloud.

Cost

You are responsible for the cost of the AWS services used while running this solution. The total cost for running this solution depends on the amount of data ingested, stored, and processed, the amount of data scanned by Amazon Athena queries, and the number of Amazon QuickSight readers and authors, along with their access time to dashboards. We recommend creating a [budget](#) through [AWS Cost Explorer](#) to help manage costs.

As of March 2021, the cost for running this solution with the default settings in the US East (N. Virginia) AWS Region is approximately **\$34.20 per month**, assuming that 100GB data per month is generated from your development teams' CI/CD activities, such as code changes and code deployments, and you have one QuickSight author and 10 readers with each accessing dashboards twice a month.

The following table is an example cost breakdown for running this solution in the US East (N. Virginia) Region (excludes free tier). Prices are subject to change. For full details, refer to the pricing page for each AWS service used in this solution.

AWS Service	Dimensions/Month	Cost/Month
Amazon Athena	100 queries, 10 GB data scanned per query	~\$5.00
Amazon Kinesis Data Firehose	100 GB	~\$2.90
Amazon Simple Storage Service (Amazon S3)	100 GB	~\$2.30
Amazon QuickSight	1 author, 10 readers, access 2 times per month for each reader	~\$24.00
Total		~\$34.20

Note

This solution implements data partition and parquet data storage for performance optimization and cost reduction. When running your own queries, we recommend that you use the `created_at(timestamp)` partition key. For more information, refer to [Performance tuning in Athena](#) in the *Amazon Athena User Guide*.

Architecture overview

Deploying this solution with the default parameters builds the following environment in the AWS Cloud.

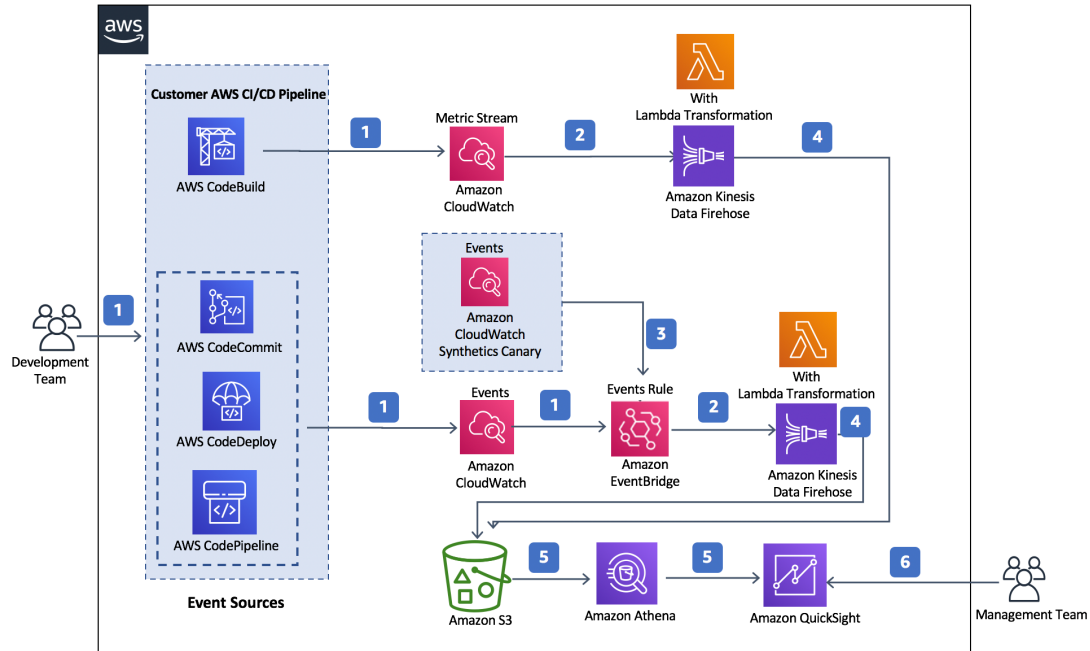


Figure 1: AWS DevOps Monitoring Dashboard architecture on AWS

This solution runs the following workflow:

1. A developer initiates an activity in an AWS CI/CD pipeline, such as pushing a code change to [AWS CodeCommit](#) or deploying an application using [AWS CodeDeploy](#). These activities create events.
2. An Amazon EventBridge events rule detects the events based on predefined event patterns and then sends the event data to an [Amazon Kinesis Data Firehose](#) delivery stream. One event rule is created per event source. For activities in AWS CodeBuild, a CloudWatch metric stream is set up to capture CloudWatch metrics and deliver them to a Kinesis Data Firehose delivery stream.
3. An Amazon EventBridge events rule is also created to capture events from an Amazon CloudWatch alarm that monitors the status of an Amazon CloudWatch synthetic canary, if you have set up the canary and alarm in your account. This alarm is needed to gather data for calculating Mean Time to Recover (MTTR) metrics.
4. Amazon Kinesis Data Firehose uses an AWS Lambda function for data transformation. The Lambda function extracts relevant data to each metric and sends it to an Amazon S3 bucket for downstream processing.
5. The data in Amazon S3 is linked to an Amazon Athena database, which runs queries against this data and returns query results to Amazon QuickSight.
6. Amazon QuickSight obtains the query results and builds dashboard visualizations for your management team.

Solution components

Amazon EventBridge Events rule

This solution creates one [Amazon EventBridge](#) events rule for each data source as follows:

- CodeCommit events rule: This rule is invoked by [AWS CodeCommit](#) events that match a predefined event pattern for code pushes. It routes the events to a target [Amazon Kinesis Data Firehose](#) delivery stream for processing.
- CodeDeploy events rule: This rule is invoked by [AWS CodeDeploy](#) events that match a predefined event pattern for code deployment state changes. It routes the events to a target [Amazon Kinesis Data Firehose](#) delivery stream for processing.
- CodePipeline events rule: This rule is invoked by [AWS CodePipeline](#) events that match a predefined event pattern for changes in CodePipeline action execution states. This rule routes the events to a target [Kinesis Data Firehose](#) delivery stream for processing.
- Canary events rule: This rule is invoked by [Amazon CloudWatch Alarm](#) events that match a predefined event pattern for alarm linked to an [Amazon CloudWatch synthetic canary](#). It routes the events to a target [Amazon Kinesis Data Firehose](#) delivery stream for processing.
- Athena partitions events rule: This rule runs on a daily schedule to invoke a Lambda function to add a new daily partition to an Amazon Athena table.

Amazon Kinesis Data Firehose

This solution creates two [Amazon Kinesis Data Firehose](#) delivery streams to process raw data from data sources. The Firehose delivery streams call an AWS Lambda function to transform source records before delivering it to an [Amazon Simple Storage Service \(Amazon S3\)](#) bucket. The output records from Kinesis Data Firehose delivery stream are converted into parquet format for performance optimization and cost reduction. Server-side encryption for source records is turned on to protect data in transit and Amazon S3 encryption is turned on to protect data in destination.

AWS Lambda

This solution creates the following [AWS Lambda](#) functions:

- Event Parser Lambdas: These functions perform Lambda transformation within Amazon Kinesis Data Firehose. They parse raw data from data sources, extracts relevant data, and returns it back to Firehose delivery stream for downstream operation.
- Query Runner Lambda: This function runs Amazon Athena queries to add Athena partitions and create views at solution deployment.
- Athena Partition Lambda: This function runs on a daily schedule to add a new daily partition to the Amazon Athena table.
- QuickSight Custom Resource Lambda: This function creates Amazon QuickSight resources at solution deployment.
- Solution Helper Custom Resource Lambda: This function generates UUID for each solution deployment.

Amazon Simple Storage Service (Amazon S3)

This solution creates the Amazon S3 `aws-devops-metrics-<random-ID>` metrics bucket to store metrics output from Amazon Kinesis Firehose delivery stream. The data is stored in a partitioned folder structure (`s3://aws-devops-metrics-<random-ID>/DevopsEvents/created_at=yyyy-mm-dd/` and `s3://aws-devops-metrics-<random-ID>/CodeBuildEvents/created_at=yyyy-mm-dd/`) where `created_at` is the partition key. This solution also creates an S3 `aws-devops-metrics-logging-<random-ID>` logging bucket to store access logs for the metrics bucket.

AWS Glue and Amazon Athena

This solution creates an [AWS Glue](#) and [Amazon Athena](#) database, which consists of two primary tables as the entry point to data in the Amazon S3 metrics bucket and a few views with each containing a subset of the data from the primary table. There is one view for each metric. This solution also creates a custom Athena workgroup for all query executions and cost management. For more information, refer to [Database schema information \(p. 21\)](#).

Amazon QuickSight

This solution uses Amazon QuickSight for data visualization. You must create an Amazon QuickSight enterprise admin user account if you don't already have one. To create a user account, refer to [Managing users in Amazon QuickSight enterprise edition](#) in the *Amazon QuickSight User Guide*. This solution deploys all required Amazon QuickSight resources, such as data source, datasets, analysis, and dashboards into your account. If you don't provide an Amazon QuickSight enterprise admin user account, this solution will not deploy Amazon QuickSight resources in your account. For more information, refer to [Amazon QuickSight dashboards visuals \(p. 24\)](#).

Amazon CloudWatch synthetics canary and Amazon CloudWatch alarm

This solution uses Amazon CloudWatch synthetics canary and Amazon CloudWatch Alarm to collect data needed for calculating Mean Time to Recover (MTTR) metrics. Synthetics canaries are configurable scripts that run on a schedule to monitor your endpoints and APIs. The CloudWatch alarm is invoked when a canary job state changes (failure or success). When the canary job recovers to its success state from a previously failed state, an [Amazon EventBridge](#) events rule is invoked, which in turn routes events to an Amazon Kinesis Data Firehose delivery stream for downstream processing. You can create your own canary and alarm or use the `canary-alarm.template` included in this solution. For more information, refer to [Set up Amazon CloudWatch synthetics canary and Amazon CloudWatch Alarm \(p. 18\)](#).

Security

When you build systems on AWS infrastructure, security responsibilities are shared between you and AWS. This [shared model](#) reduces your operational burden because AWS operates, manages, and controls the components including the host operating system, the virtualization layer, and the physical security of the facilities in which the services operate. For more information, refer to [AWS Cloud Security](#).

AWS IAM roles

AWS Identity and Access Management (IAM) roles allow you to assign granular access policies and permissions to services and users in the AWS Cloud. This solution creates IAM roles that grant the solution's AWS Lambda functions access to create Regional resources.

Amazon S3

All Amazon S3 buckets are encrypted with [SSE-S3 managed encryption](#). None of the Amazon S3 buckets are available publicly. The Amazon S3 buckets are configured with the retention policy set to **Retain**.

Deployment considerations

Regional deployment

This solution uses services such as Amazon Athena, Amazon Kinesis Data Firehose and Amazon QuickSight (optional), which are not currently available in all AWS Regions. You must launch this solution in an AWS Region where these services are available. For the most current availability by Region, refer to the [AWS Regional Services List](#).

AWS CI/CD pipeline deployment

This solution must be launched in the same Region and account where your AWS CI/CD pipeline is deployed. Refer to [Set Up a CI/CD Pipeline on AWS](#) if you do not currently have a pipeline set up on AWS.

Amazon QuickSight deployment

This solution requires Amazon QuickSight resources to be deployed in an Amazon QuickSight Enterprise edition account in the same Region. If you plan to use the Amazon QuickSight dashboard feature, you must subscribe to Amazon QuickSight Enterprise edition in the account where you deploy the solution. Refer to [Signing Up for An Amazon QuickSight Subscription](#) if you do not have an Amazon QuickSight Enterprise account set up. Ensure that you have the QuickSight Principal ARN, as you will need it later when you deploy the solution. For information, refer to [Retrieve the Amazon QuickSight Principal ARN \(p. 17\)](#).

Amazon CloudWatch synthetics canary and Amazon CloudWatch alarm deployment

The solution uses Amazon CloudWatch synthetics canary and Amazon CloudWatch alarm to collect data needed for calculating Mean Time to Recover (MTTR) metrics. For more information, refer to [Set up Amazon CloudWatch synthetics canary and Amazon CloudWatch alarm \(p. 18\)](#).

AWS CloudFormation template

This solution uses AWS CloudFormation to automate the deployment of the AWS DevOps Monitoring Dashboard solution in the AWS Cloud. It includes the following CloudFormation template, which you can download before deployment.

[View
Template](#)

aws-devops-monitoring-dashboard.template: Use this template to launch the solution. The default configuration deploys Amazon EventBridge events rules, AWS Lambda functions, Amazon Simple Storage Service (Amazon S3) buckets, Amazon Kinesis Data Firehose, AWS Glue and Amazon Athena databases, and Amazon QuickSight resources (optional). You can also customize the template to meet your specific needs.

Automated deployment

Before you launch the solution, review the considerations discussed in this guide. Follow the step-by-step instructions in this section to configure and deploy the solution into your account.

Time to deploy the template: Approximately 10 mins

Prerequisites

1. You must have AWS CI/CD pipeline installed in your account. This consists of AWS CodeCommit, AWS CodeBuild, AWS CodeDeploy, and AWS CodePipeline. Refer to [Set Up a CI/CD Pipeline on AWS](#) if you do not currently have a pipeline set up on AWS.
2. If you plan to use the Amazon QuickSight dashboard feature, you must subscribe to Amazon QuickSight Enterprise edition in the account where you deploy the solution. Refer to [Signing Up for An Amazon QuickSight Subscription](#) if you do not have a QuickSight Enterprise account set up. Ensure that you have the QuickSight Principal ARN, as you will need it later when deploy the solution. For more information, refer to [Retrieve the Amazon QuickSight Principal ARN \(p. 17\)](#). Also, ensure that your QuickSight account has permission to access Amazon Athena. You can choose to skip the Amazon S3 bucket configuration when you set up the Amazon Athena permission.

Deployment overview

[Step 1. Launch the Stack \(p. 9\)](#)

- Launch the AWS CloudFormation template into your AWS account.
- Review the template parameters, and enter the values as needed: **Athena Query Data Duration (Days)**, **AWS CodeCommit Repository List**, **S3 Transition Days**, **Amazon QuickSight Principal ARN**.

[Step 2. Configure Amazon QuickSight \(p. 11\)](#)

- After the stack is successfully deployed, set up Amazon QuickSight for data visualization.

Step 1: Launch the stack

This automated AWS CloudFormation template deploys the AWS DevOps Monitoring Dashboard solution in the AWS Cloud. Before you launch the stack, you must complete the [prerequisites \(p. 9\)](#).

Note

You are responsible for the cost of the AWS services used while running this solution. For more details, refer to the [Cost \(p. 2\)](#) section in this guide, and refer to the pricing webpage for each AWS service you used in this solution.

1. Sign in to the AWS Management Console and select the button to launch the `aws-devops-monitoring-dashboard` AWS CloudFormation template.



You can also [download the template](#) as a starting point for your own implementation.

2. The template launches in the US East (N. Virginia) Region by default. To launch the solution in a different AWS Region, use the Region selector in the console navigation bar.

Note

This solution uses services such as Amazon Athena, Amazon Kinesis Data Firehose and Amazon QuickSight (optional), which are not currently available in all AWS Regions. You must launch this solution in an AWS Region where these services are available. For the most current availability by Region, refer to the [AWS Regional Services List](#).

3. On the **Create stack** page, verify that the correct template URL is in the **Amazon S3 URL** text box and choose **Next**.
4. On the **Specify stack details** page, assign a name to your solution stack. For information about naming character limitations, refer to [IAM and STS quotas](#) in the *AWS Identity and Access Management User Guide*.
5. Under **Parameters**, review the parameters for this solution template and modify them as necessary. This solution uses the following default values:

Parameter	Default	Description
Athena Query Data Duration (Days)	90	Enter a duration (days) that Athena query uses to retrieve data. By default Athena query retrieves data within the last 90 days. We recommend that you to limit the duration for performance optimization and cost reduction.
AWS CodeCommit Repository List	'ALL'	List of the names of AWS CodeCommit repositories that will be monitored. Must be single-quoted and comma separated. For example: 'MyRepository1', 'MyRepository2' To monitor all the repositories, leave default 'ALL' value.
S3 Transition Days	365	Enter the number of days after which you would like to transition Amazon S3 objects to Amazon S3 Glacier storage class. By default objects are transitioned to Amazon S3

Parameter	Default	Description
		Glacier 365 days (one year) after creation.
Amazon QuickSight Principal ARN	<Optional Input>	Provide an Amazon QuickSight admin user ARN to automatically create QuickSight resources. Amazon QuickSight Enterprise edition must be activated for the account. For example: <code>arn:aws:quicksight:AWSRegion:AWSAccountID:default/QuickSightUserName</code> . To deactivate QuickSight dashboards creation, do not enter a value. For more information, refer to Prerequisites Step 2 (p. 9) .

6. Choose **Next**.
7. On the **Configure stack options** page, choose **Next**.
8. On the **Review** page, review and confirm the settings. Check the box acknowledging that the template will create AWS Identity and Access Management (IAM) resources.
9. Choose **Create stack** to deploy the stack.

You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should receive a CREATE_COMPLETE status in approximately 10 minutes.

Note

If you provided an Amazon QuickSight Principal ARN, this solution launches a nested stack to create QuickSight resources into the account you provided.

Step 2: Configure Amazon QuickSight

This solution uses Amazon QuickSight for data visualization. Follow these instructions to configure permissions, and view datasets, analysis, and dashboards in Amazon QuickSight.

Note

You can also set up your own visualization tools, such as Tableau. For more information, refer to [Build visualizations with Amazon Athena and Tableau \(p. 36\)](#).

1. After the stack successfully deploys, go to the **Outputs** tab of the stack and make a note of the values for **QSAnalysisURL**, **QSDashboardURL**, and **DevOpsMetricsS3Bucket**.
2. Sign in to the AWS Management Console and navigate to Amazon QuickSight.
3. Change the Region in the URL to match the Region where you deployed the solution. For example, if the solution was deployed in the us-east-1 Region, the QuickSight URL will mirror the following path: `https://us-east-1.quicksight.aws.amazon.com/sn`.
4. Select your username on the upper right corner, then choose **Manage QuickSight**.
5. From the left navigation menu, select **Security & permissions**.
6. Under **QuickSight access to AWS Services**, choose **Add or remove**.
7. Select **IAM**, **Amazon S3**, and **Amazon Athena**. If these options are already selected, uncheck and recheck the options.

8. Choose **Amazon S3**, choose the **Details** link.
9. Choose **Select S3 buckets**.
10. Select the bucket name for **DevOpsMetricsS3Bucket**, and check the check box under **Write permission for Athena Workgroup** for the bucket.
11. Select **Finish**, then choose **Update**.
12. From the **Output** tab of the stack, select **QSAnalysisURL** and **QSDashboardURL** to open dashboards and analyses. You can also navigate to them in the Amazon QuickSight console. This solution creates one analysis, one dashboard, and multiple datasets. This solution creates Amazon QuickSight resources that are prefixed with the stack name. For example, `<stack-name>-analysis`. Refer to [Amazon QuickSight dashboards visuals \(p. 24\)](#) for sample dashboards visuals. You might receive a **No Data** message if the Amazon S3 metrics bucket is empty after launching this solution. You can refresh the pages to view data and visuals after this solution finishes processing data and sends metrics to Amazon S3.

Note

This solution creates Amazon QuickSight datasets that use Direct Query to query Amazon S3 buckets for data. For better performance, use SPICE. Refer to [Using SPICE Data in an Analysis](#) in the *Amazon QuickSight User Guide* for more information about configuring and using SPICE.

Resources

AWS services

AWS CloudFormation	Amazon Kinesis Data Firehose
AWS Glue	Amazon Simple Storage Service
Amazon EventBridge	Amazon QuickSight
AWS Identity and Access Management	Amazon Athena
AWS Lambda	Amazon CloudWatch

Update the stack

If you have previously deployed the solution, follow this procedure to update the `aws-devops-monitoring-dashboard` CloudFormation stack to get the latest version of the solution's framework.

1. Sign in to the AWS CloudFormation Console, select the existing AWS DevOps Monitoring Dashboard AWS CloudFormation stack, and select **Update**.
2. Select **Replace current template**.
3. Under **Specify template**:
 - a. Select **Amazon S3 URL**.
 - b. Copy the link of the latest template for the stack.
 - c. Paste the link in the **Amazon S3 URL** box.
 - d. Verify that the correct template URL shows in the **Amazon S3 URL** text box, and choose **Next**. Choose **Next** again.
4. Under **Parameters**, review the parameters for the template and modify them as necessary. Refer to [Step 1. Launch the Stack \(p. 9\)](#) for details about the parameters.
5. Choose **Next**.
6. On the **Configure stack options** page, choose **Next**.
7. On the **Review** page, review and confirm the settings. Check the box acknowledging that the template might create AWS Identity and Access Management (IAM) resources.
8. Choose **View change set** and verify the changes.
9. Choose **Update stack** to deploy the stack.

You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should see a status of **UPDATE_COMPLETE** in approximately 10 minutes depending on the options chosen.

DevOps metrics list

Code change volume metrics

The code change volume metrics indicate the code change frequency of developers in a source control, such as AWS CodeCommit. These metrics give DevOps leaders better visibility into the coding activities of their development teams. They can answer questions, such as who makes the most number of code changes and which repositories are the most active over time. The underlying data for these metrics are AWS CodeCommit events. To view an example dashboard for these metrics, refer to [Code change volume dashboards \(p. 24\)](#).

Mean time to recover metrics

Mean time to recover (MTTR) metrics present outage minutes and the average time it takes to restore a service from a failed state. These metrics help DevOps leaders correlate change activity to system stability, track problematic applications and identify opportunities to improve the stability of applications. The underlying data for these metrics are AWS CloudWatch alarm events. This alarm monitors the state of Amazon CloudWatch Synthetics canary. To view an example dashboard for these metrics, refer to [Mean time to recover dashboards \(p. 24\)](#).

Change failure rate metrics

The change failure rate metrics indicate how often deployment failures occur for an application. These metrics help DevOps leaders track the code quality of their development teams and drive improvements to reduce change failure rate over time. The underlying data for this metric are AWS CodeDeploy events. To view an example dashboard for these metrics, refer to [Change failure rate dashboards \(p. 25\)](#).

Deployment metrics

Deployment metrics present data about application deployment, such as deployment state (failure or success) and frequency. These metrics help DevOps leaders track the frequency and quality of their continuous software release to end-users. The underlying data for this metric are AWS CodeDeploy events. To view an example dashboard for these metrics, refer to [Deployment dashboards \(p. 26\)](#).

Build metrics

Build metrics present data about CodeBuild activities, such as build duration, build state (failure or success) and frequency, along with resource utilization metrics for CPU, memory, and storage utilization. These metrics help DevOps leaders track the frequency and quality of their code build process. These metrics can indicate which build projects or phases take the longest time to run, which build projects are the most active over time, and which build projects fail the most. The underlying data for these metrics are AWS CodeBuild metrics. To view an example dashboard for these metrics, refer to [Build dashboards \(p. 27\)](#).

Note

Resource utilization metrics are not available for builds shorter than one minute and they are not supported in all of the Regions where AWS CodeBuild is supported. For a complete list of the supported Regions, refer to [Monitoring CodeBuild resource utilization metrics](#) in the *AWS CodeBuild User Guide*.

Pipeline metrics

The Pipeline metrics present data about CodePipeline, such as pipeline execution state (failure, success, or other), execution duration and frequency along with state at stage and action level. These metrics give DevOps leaders better visibility into the pipeline activities of their development teams. These metrics can indicate which pipelines fail the most, which pipelines take the longest time to run, and which pipelines are the most active over time. The underlying data for these metrics are AWS CodePipeline events. To view an example dashboard for these metrics, refer to [Pipeline dashboards \(p. 33\)](#).

Retrieve the Amazon QuickSight Principal ARN

If you want to deploy Amazon QuickSight resources, you must retrieve the Amazon QuickSight Principal ARN before deploying this solution. To retrieve the Amazon QuickSight User Principal ARN, you must have access to a shell or terminal with the AWS CLI installed. For installation instructions, refer to [What Is the AWS Command Line Interface](#) in the *AWS CLI User Guide*. Optionally, you can use the [AWS CloudShell](#) service to run AWS CLI commands.

Running the following list-users command returns the list of users with their corresponding QuickSight User ARNs.

```
aws quicksight list-users --region <aws-region> --aws-account-id <account-id> --  
namespace <namespace-name>
```

The following example shows a valid ARN:

```
arn:aws:quicksight:<aws-region>:<account-id>:user/<namespace-name>/quicksight-user-name
```

The default namespace-name is default. For example, `arn:aws:quicksight:us-east-1:111111111111:user/default/myquicksightuser`.

Choose a user who has permissions to create Amazon QuickSight resources in that account and AWS Region, such as a QuickSight Admin user.

If you do not have an Amazon QuickSight Enterprise account, refer to [Signing Up for An Amazon QuickSight Subscription](#) to set up your account and then retrieve the Principal account ARN.

Set up Amazon CloudWatch synthetics canary and Amazon CloudWatch alarm

You can choose one of the following two ways to set up [Amazon CloudWatch synthetics canary](#) and its [Amazon CloudWatch Alarm](#). The goal is to create an Amazon CloudWatch alarm that monitors the state (success or failure) of a new or existing canary job. Whenever the canary job state changes, it invokes the alarm. This will generate events that are needed for calculating Mean time to recover (MTTR) metrics.

1. (Recommended) Automated setup

- This solution provides a [canary-alarm.template](#) that you can deploy to create an Amazon CloudWatch alarm and/or canary into your account.

2. Manual setup

- If you don't have a canary, sign in to the AWS Management Console and create a canary. To create a canary, refer to [Creating a canary](#) in the *Amazon CloudWatch User guide* to create one. If you have already created one, skip to the next step.
- To create an alarm, refer to [Create a CloudWatch Alarm Based on Static Threshold](#) in the *Amazon CloudWatch User Guide*. When you reach the select metrics step, make sure you select CloudWatchSynthetics metrics, your canary and SuccessPercent metric as shown below.
 - Select **CloudWatchSynthetics** metrics for the alarm.

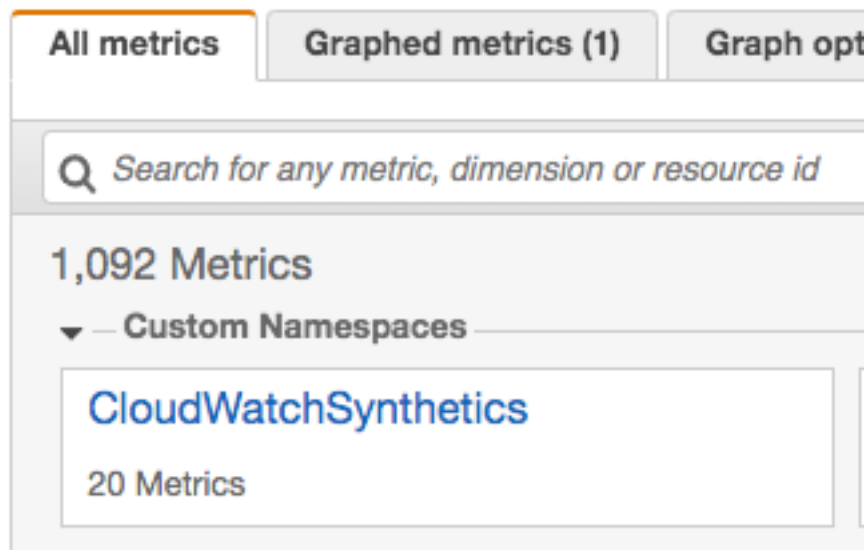


Figure 2: Select CloudWatchSynthetics metrics

- From the **All metrics** tab, select **By Canary** and then select your canary and **SuccessPercent** metric name.

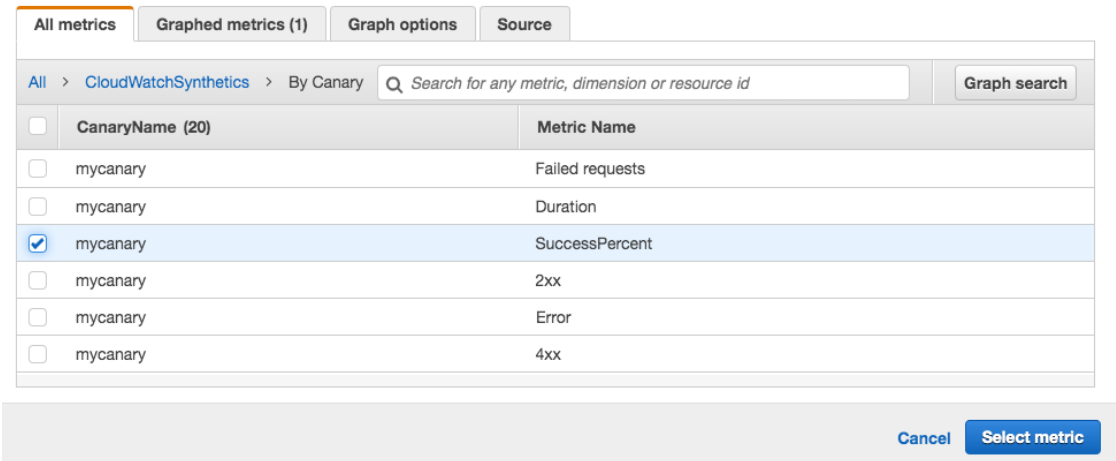


Figure 3: Select your canary and SuccessPercent metric

- Name the alarm `SO0143-[my-application-name]-[my-repository-name]-MTTR`. For example, `SO0143-[MyDemoApplication]-[MyDemoRepo]-MTTR`. `SO0143` is the solution ID. Application name is the name of the application that your canary monitors and repository name is the name of the repository where the source code for your application resides. This solution uses alarm name to determine if an alarm is used for MTTR metrics and what application and repository are associated with the metrics.
- Under **Conditions** of the alarm, leave threshold type as **Static** and choose whenever **SuccessPercent** is **Lower** than **100** or enter a threshold value that fits your use case.
- For an example of the alarm, refer to figure 4:

Metric

Edit

Graph
This alarm will trigger when the blue line goes below the red line for 1 datapoints within 5 minutes.

Namespace
CloudWatchSynthetics

Metric name

CanaryName

Statistic

Period

Conditions

Threshold type

Static
Use a value as a threshold

Anomaly detection
Use a band as a threshold

Whenever SuccessPercent is...

Define the alarm condition.

Greater
> threshold

Greater/Equal
>= threshold

Lower/Equal
<= threshold

Lower
< threshold

than...

Define the threshold value.

Must be a number

Figure 4: Alarm Example

Database schema information

The following diagrams displays a high-level database schema structure for the tables and views created in AWS Glue and Amazon Athena database. The `aws_devops_metrics_table` is one of the two primary tables that acts as an entry point to the data (not including AWS CodeBuild) in the Amazon S3 metrics bucket. The **detail** column in the table uses a struct data type and contains data for different metrics. The views (not including AWS CodeBuild) are built based on this table. Each view contains only a subset of the base table's data for a specific metrics, such as code change activity and code deployment. A view's data is mainly extracted from the **detail** column in the table for that metrics. The `aws_codebuild_metrics_table` is another primary table that points to AWS CodeBuild data in the Amazon S3 metrics bucket. It is the base table for the code build view.

The data model is not normalized and includes redundant attributes for reporting performance.



Figure 5: DevOps metrics database schema structure

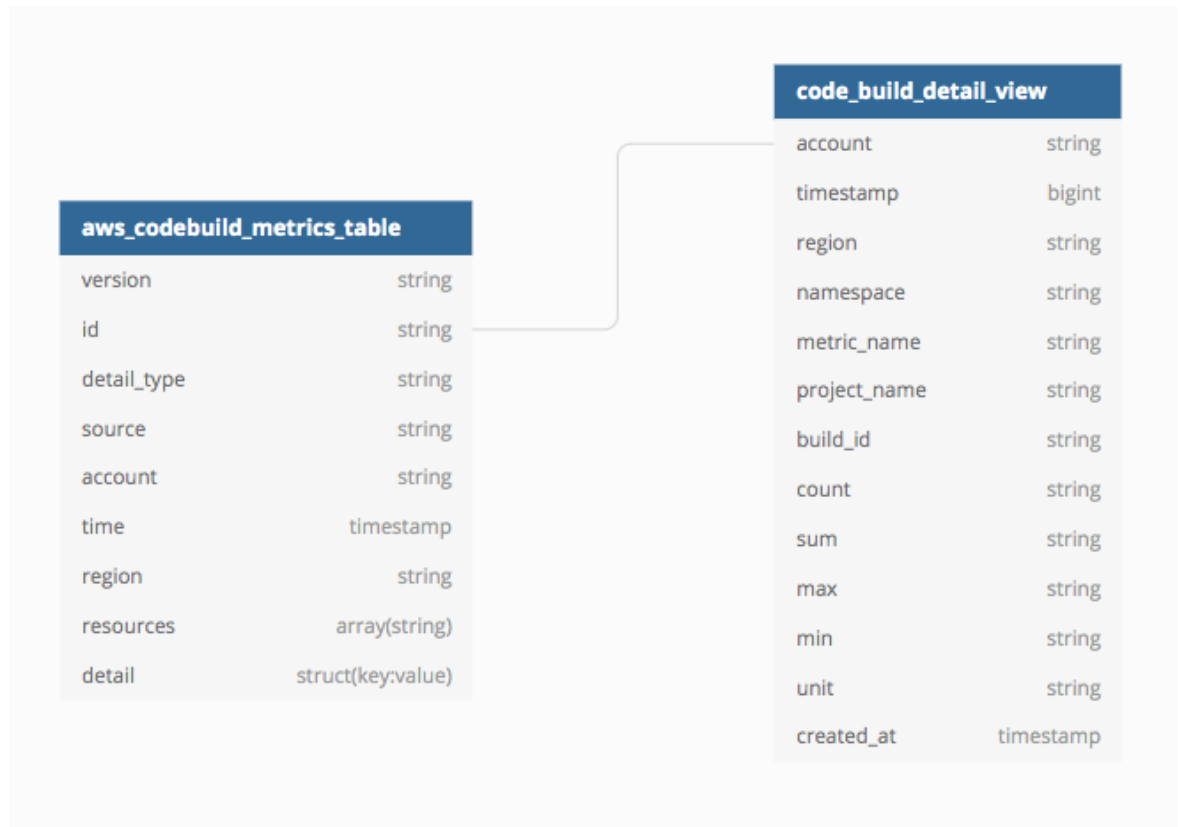


Figure 6: CodeBuild metrics database schema structure

Amazon QuickSight dashboards visuals

The following dashboards are examples of dashboard visuals that this solution deploys with Amazon QuickSight.

Code change volume dashboards

These dashboards display the number of code changes made by author and repository. They provide a weekly, monthly and aggregated view of the metrics by author and repository. You can filter data by author, repository or time period (default to last 90 days) using the custom filter as needed. For more information about the metrics, refer to [Code change volume metrics \(p. 15\)](#).

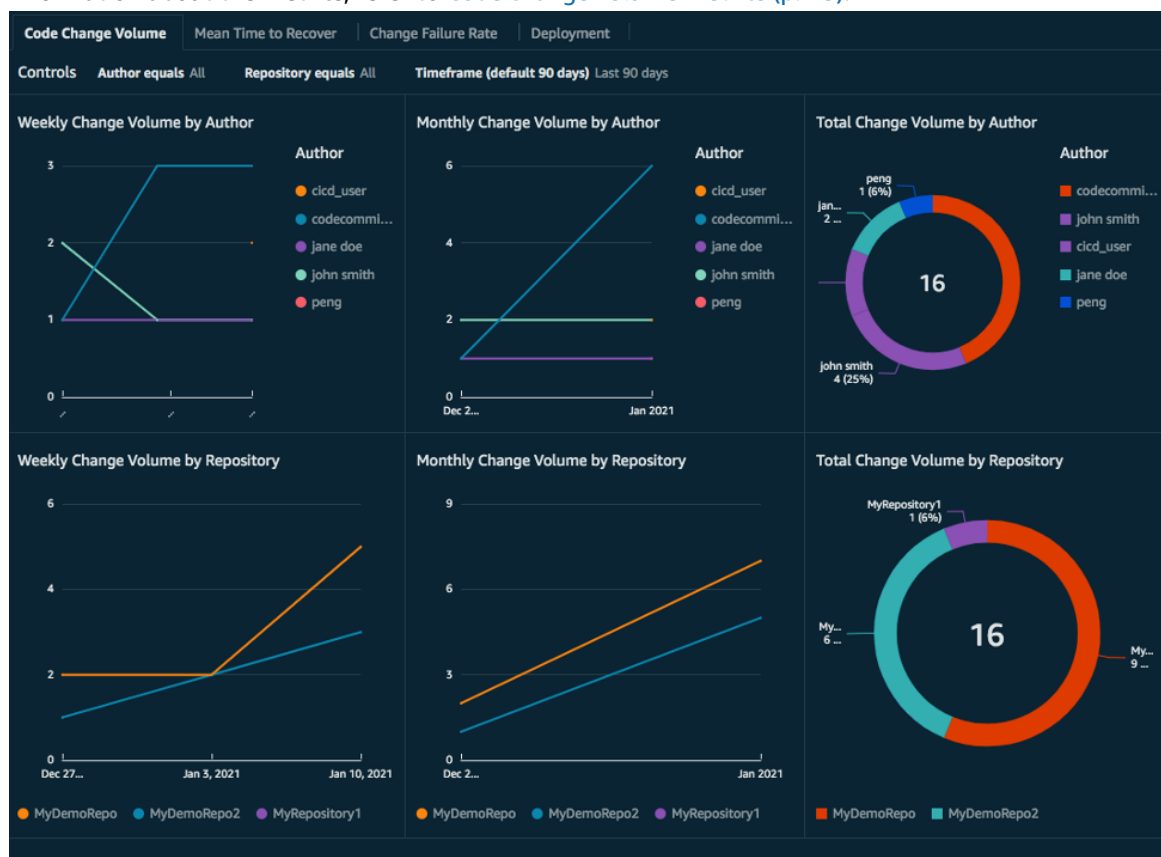


Figure 7: Code change volume dashboard

Mean time to recover dashboards

These dashboards display outage minutes by application and the average time it takes to restore an application from a failure to success state. They provide a weekly, monthly and aggregated view of the

metrics by application. You can filter data by application or time period (default to last 90 days) using the custom filter as needed. For more information about the metrics, refer to [Mean time to recover metrics \(p. 15\)](#).

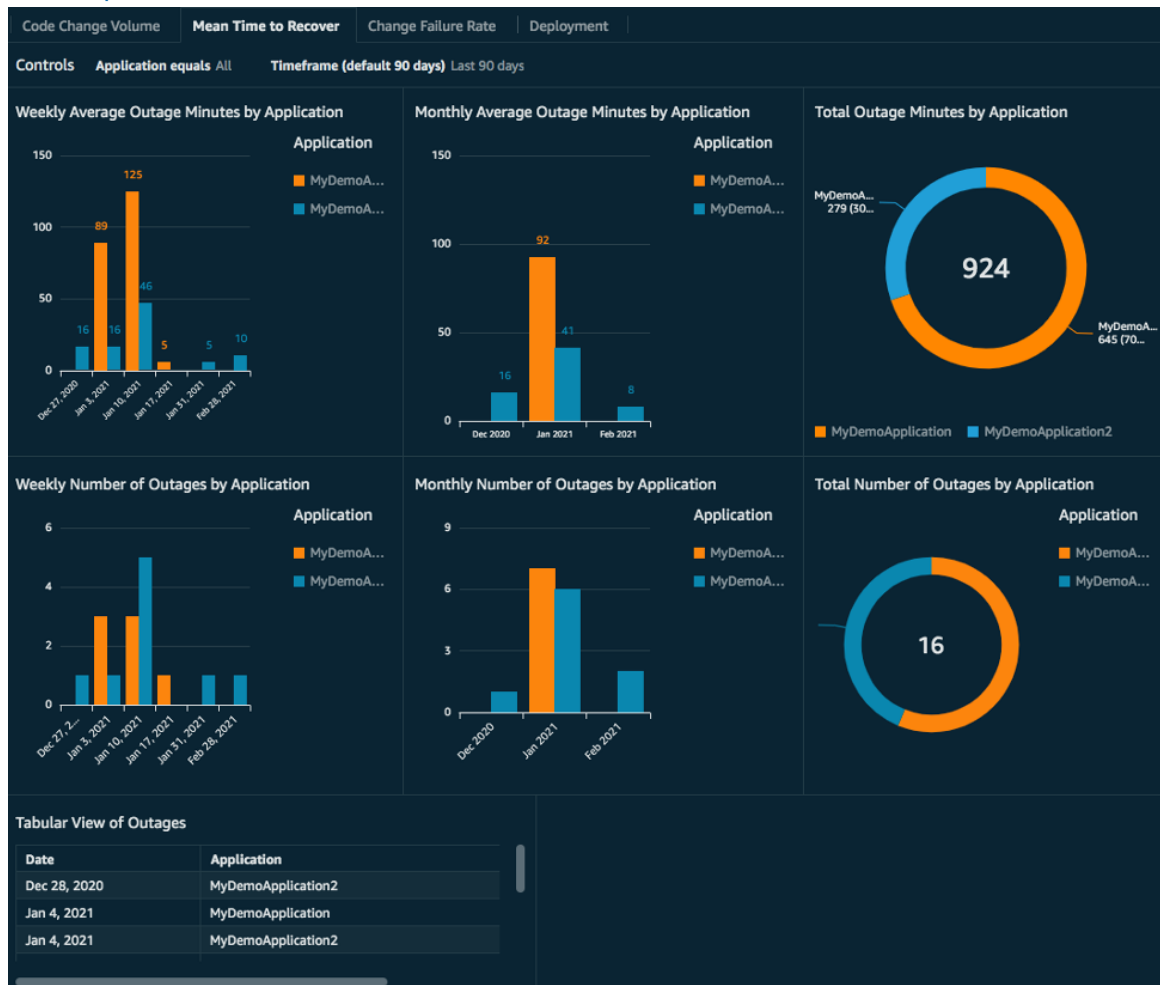


Figure 8: Mean time to recover dashboards

Change failure rate dashboards

These dashboards display the frequency of deployment failures per application by measuring the ratio of unsuccessful to total deployments. They provide a weekly, monthly and aggregated view of the metrics by application. You can filter metrics by application or time period (default to last 90 days) using the custom filter. For more information about the metrics, refer to [Change failure rate metrics \(p. 15\)](#).

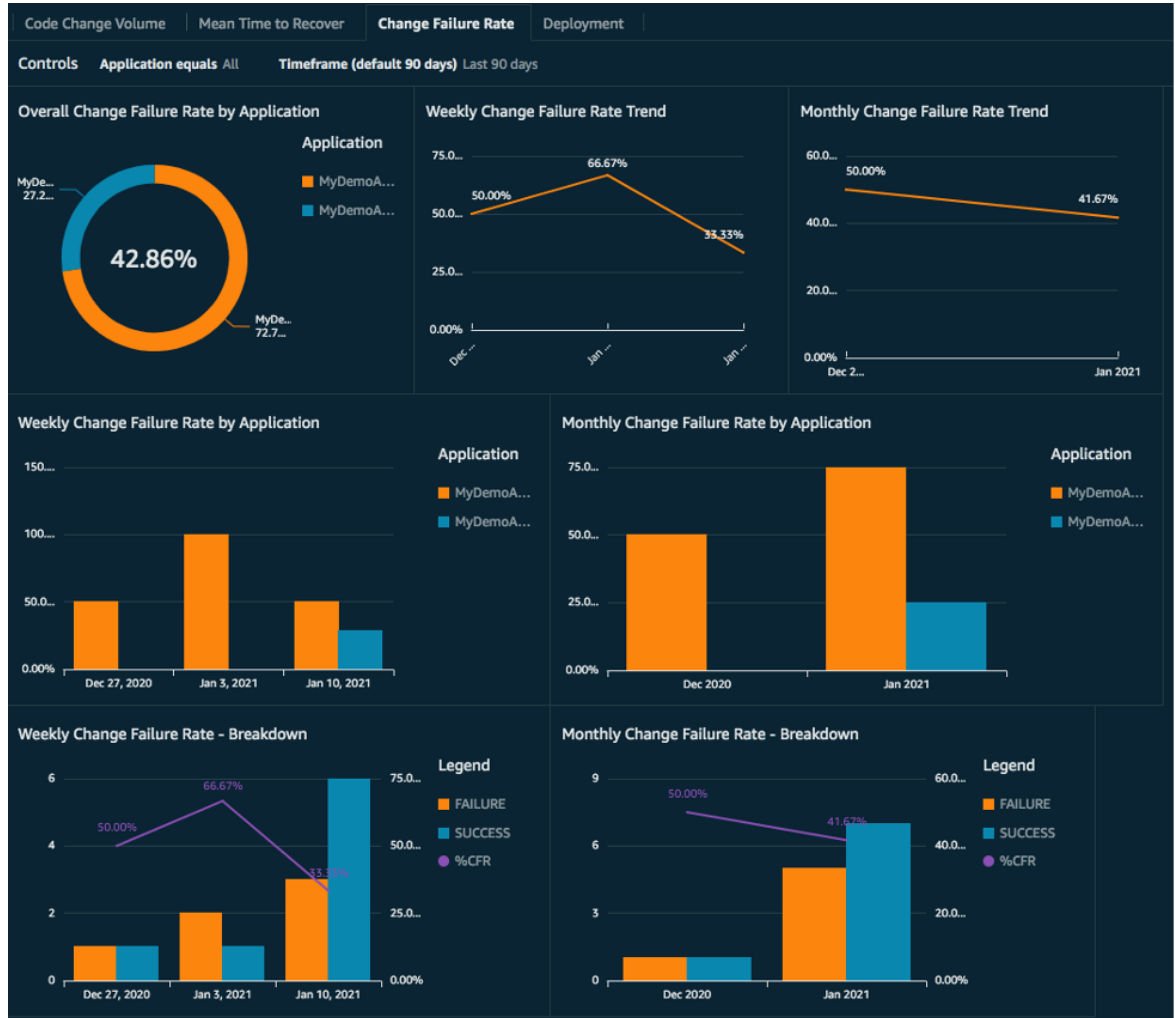


Figure 9: Change failure rate dashboards

Deployment dashboards

These dashboards display the deployment frequency and state (success/failure) by application. They provide a weekly, monthly and aggregated view of the metrics by application. You can filter metrics by application or time period (default to last 90 days) using the custom filter. For more information about the metrics, refer to [Deployment metrics \(p. 15\)](#).

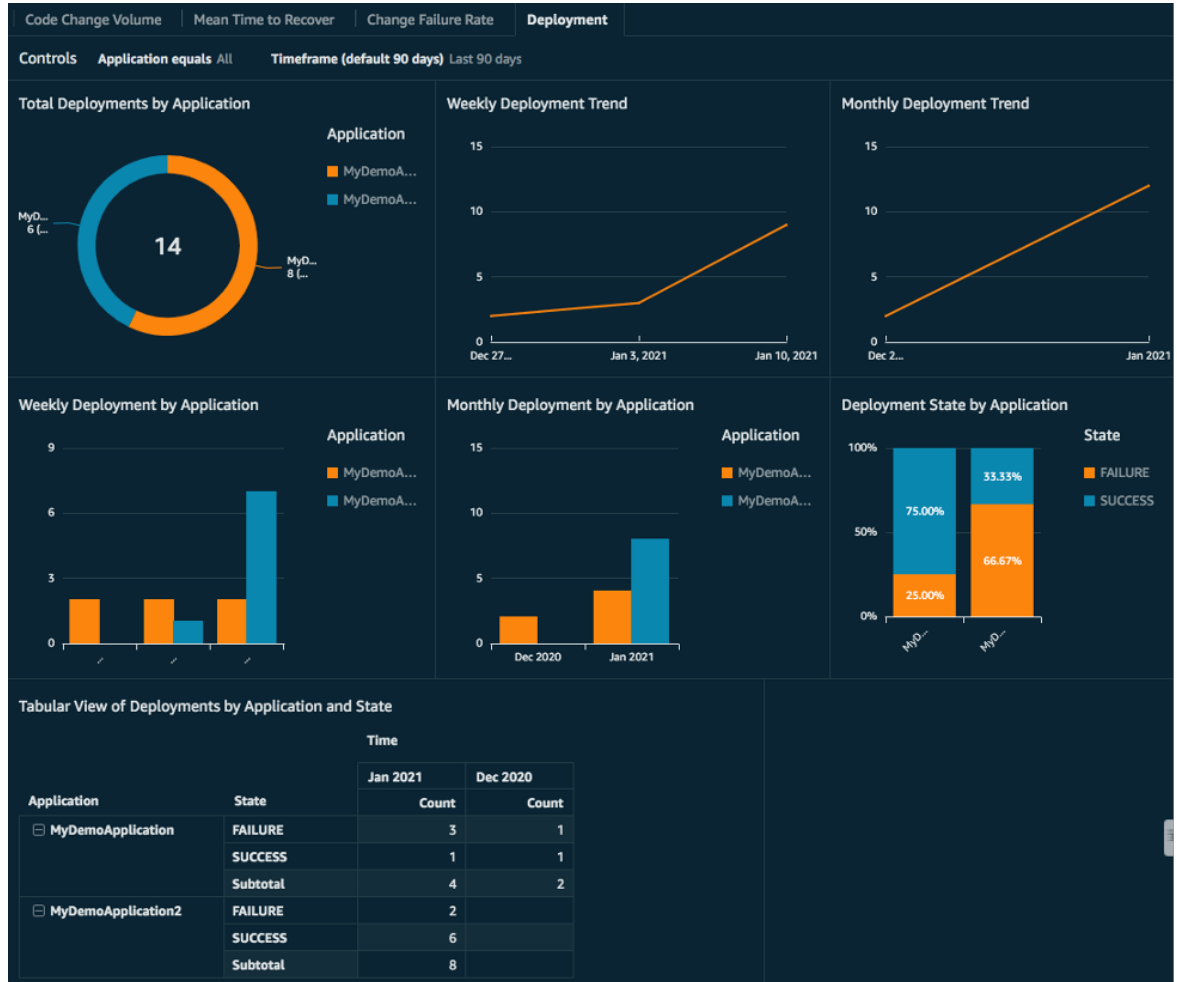


Figure 10: Deployment dashboards

Build dashboards

Build activity dashboards

These dashboards display the code build frequency, duration and state (success/failure) by project. They provide a weekly, monthly, and aggregated view of the metrics by project. You can filter metrics by project, metric name (FailedBuilds, SucceededBuilds, and BuildDuration), or time period (default to last 90 days) using the custom filter. For more information about the metrics, refer to [Build metrics \(p. 15\)](#).

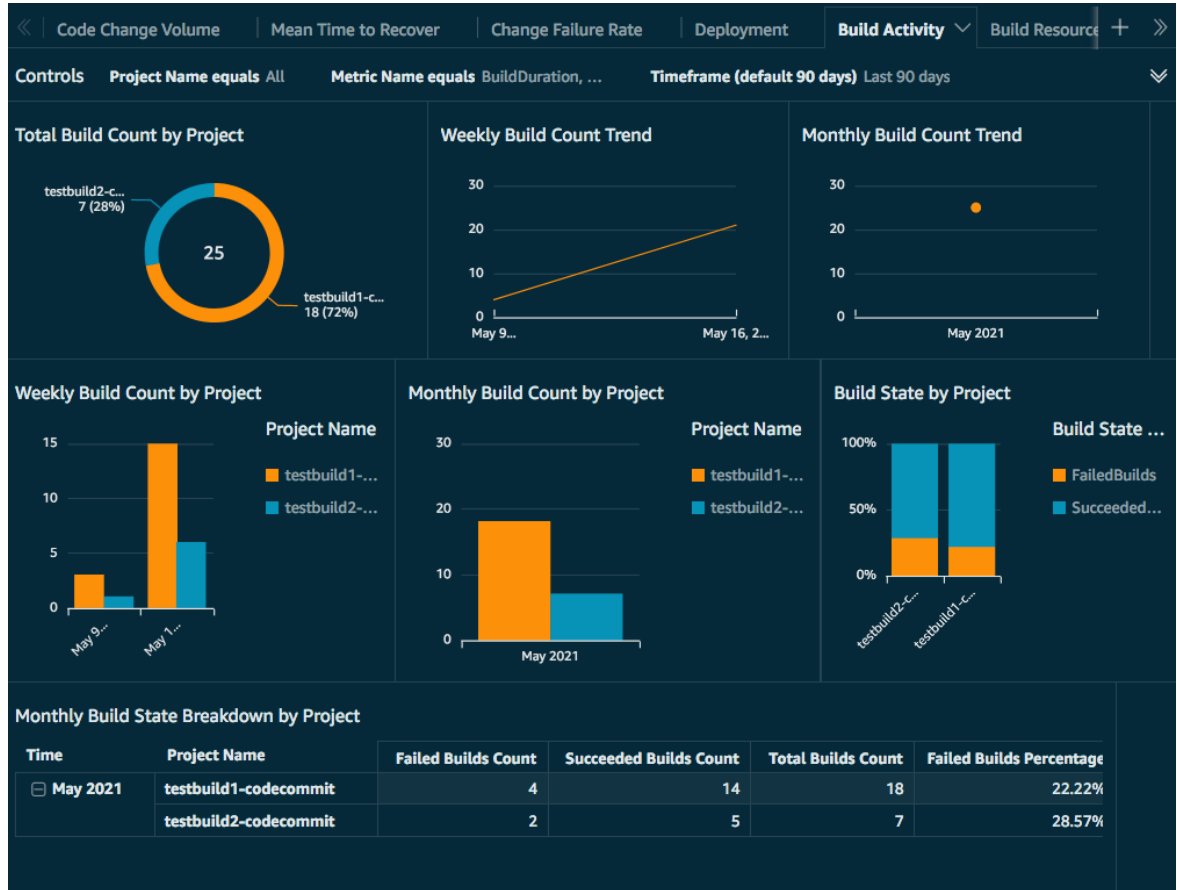


Figure 11: Build activity dashboards - 1

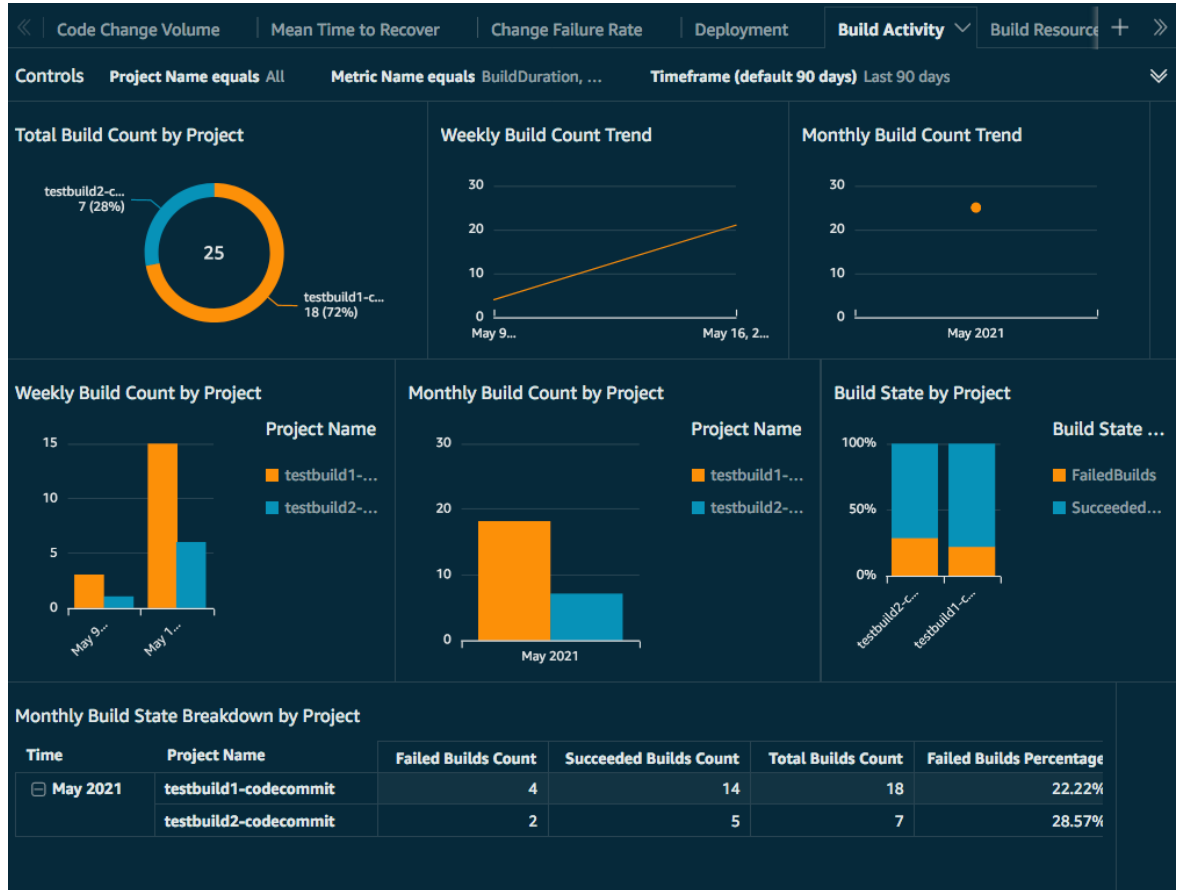


Figure 12: Build activity dashboards - 2

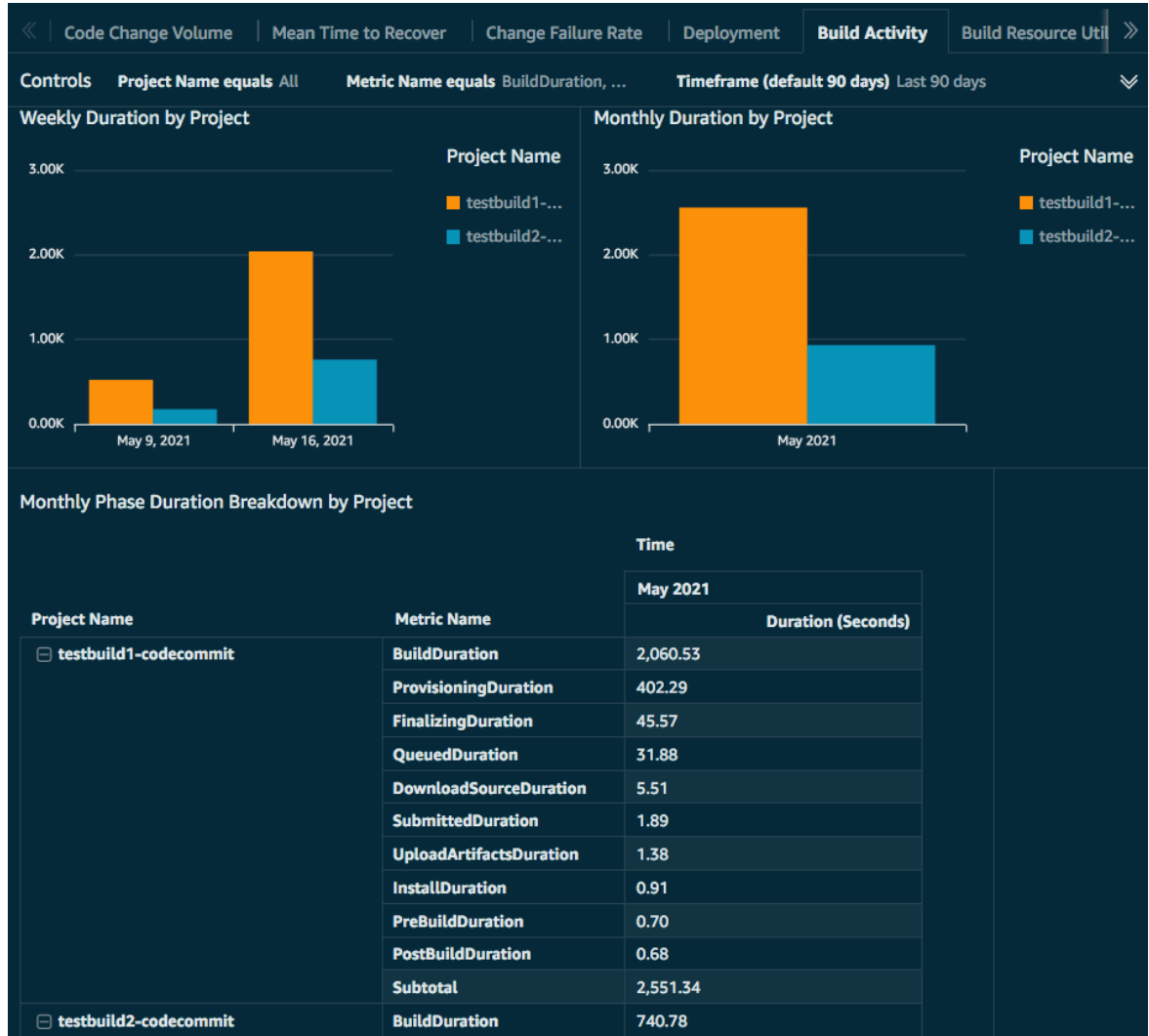


Figure 13: Build activity dashboards - 3

Build resource utilization dashboards

These dashboards display code build resource utilization metrics for CPU, memory, and storage utilization by project and build. They provide a weekly, monthly, and aggregated view of the metrics by projects and build. You can filter metrics by project, metric name (CPUUtilized, MemoryUtilized), or time period (default to last 90 days) using the custom filter. Resource utilization metrics are not available for builds shorter than one minute and they are not supported in all the AWS Regions where AWS CodeBuild is supported. For a complete list of the supported Regions, refer to Monitoring CodeBuild resource utilization metrics in the AWS CodeBuild User Guide. For more information about the metrics, refer to [Build metrics \(p. 15\)](#).

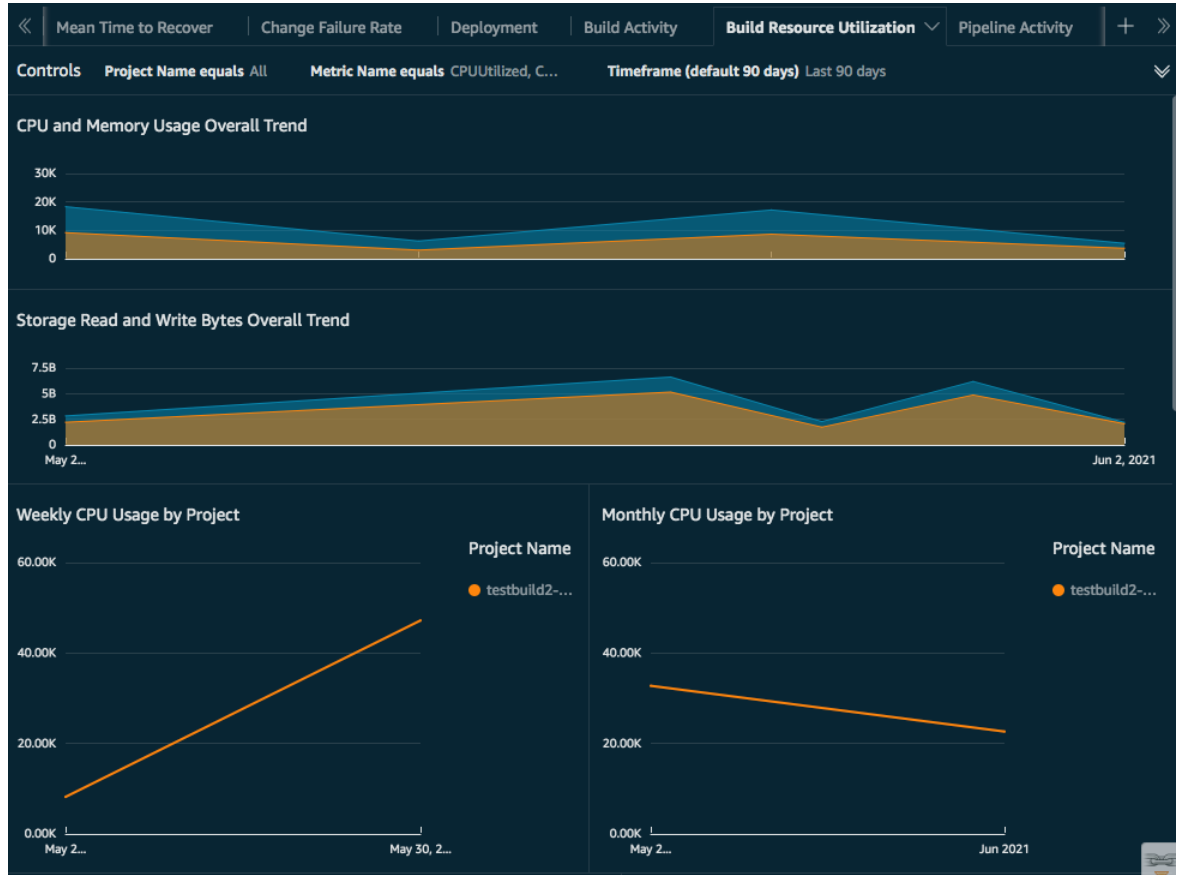


Figure 14: Build resource utilization dashboards - 1

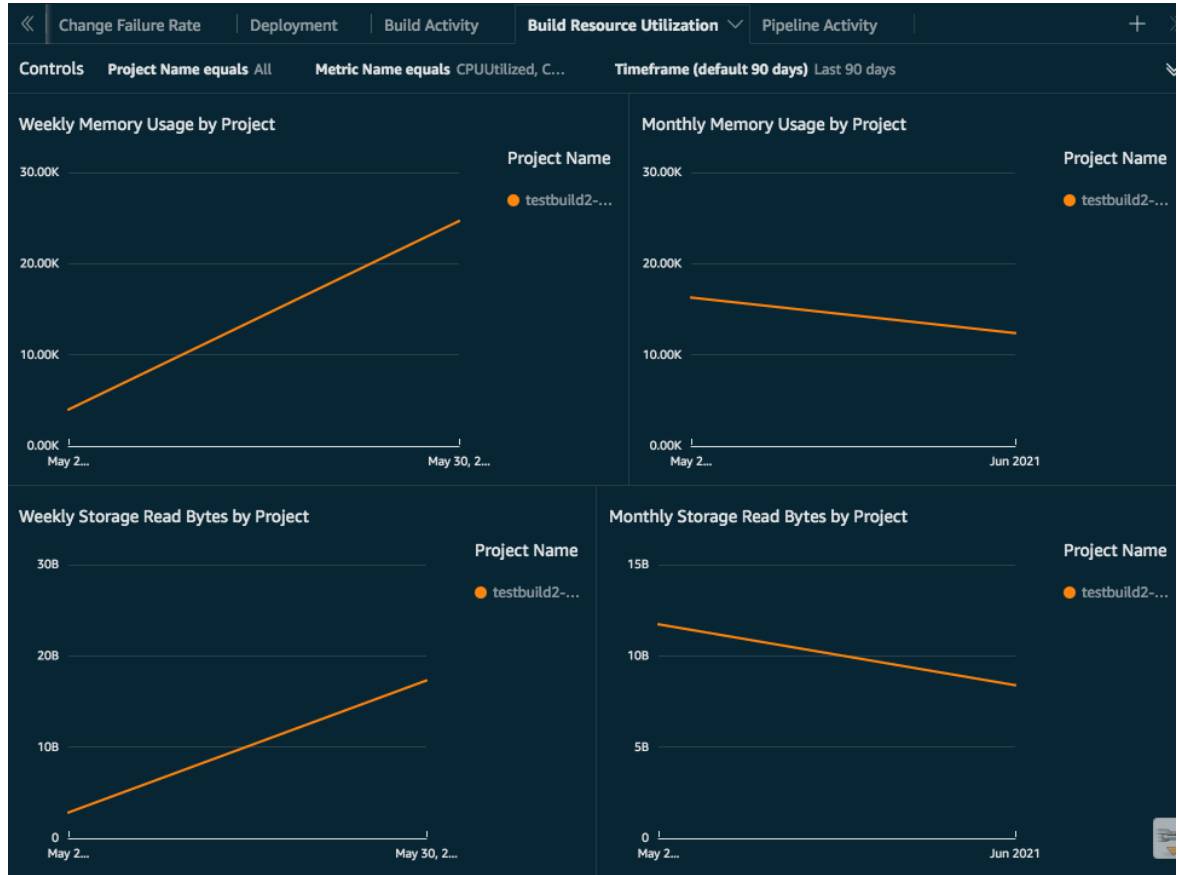


Figure 15: Build resource utilization dashboards - 2

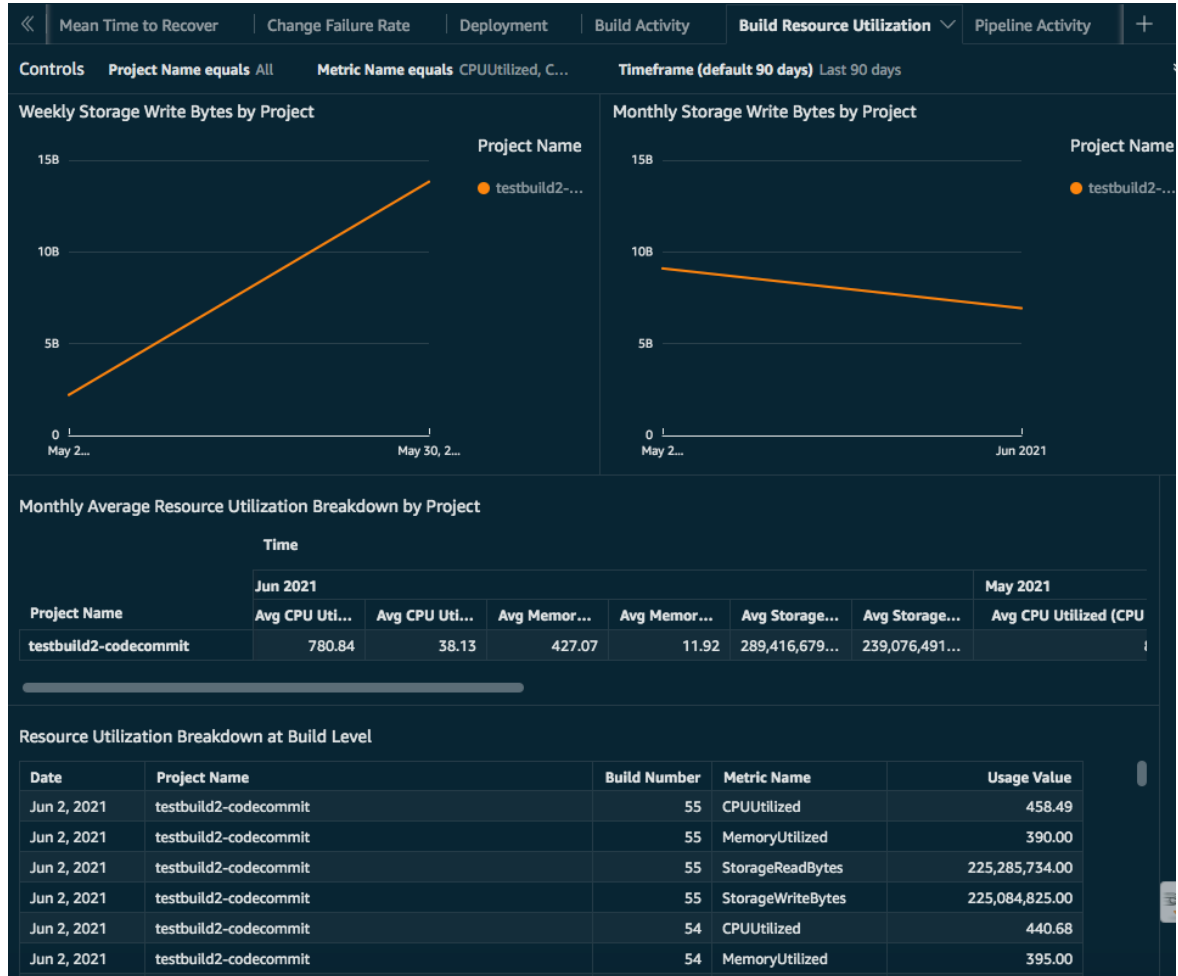


Figure 16: Build resource utilization dashboards - 3

Pipeline dashboards

These dashboards display pipeline execution state (failure, success, and others), execution duration and frequency in addition to the state at stage and action level. They provide a weekly, monthly, and aggregated view of the metrics by pipeline. You can filter metrics by pipeline, or time period (default to last 90 days) using the custom filter. For more information about the metrics, refer to [Pipeline metrics \(p. 16\)](#).



Figure 17: Pipeline dashboards - 1

AWS DevOps Monitoring Dashboard Implementation Guide
Pipeline dashboards

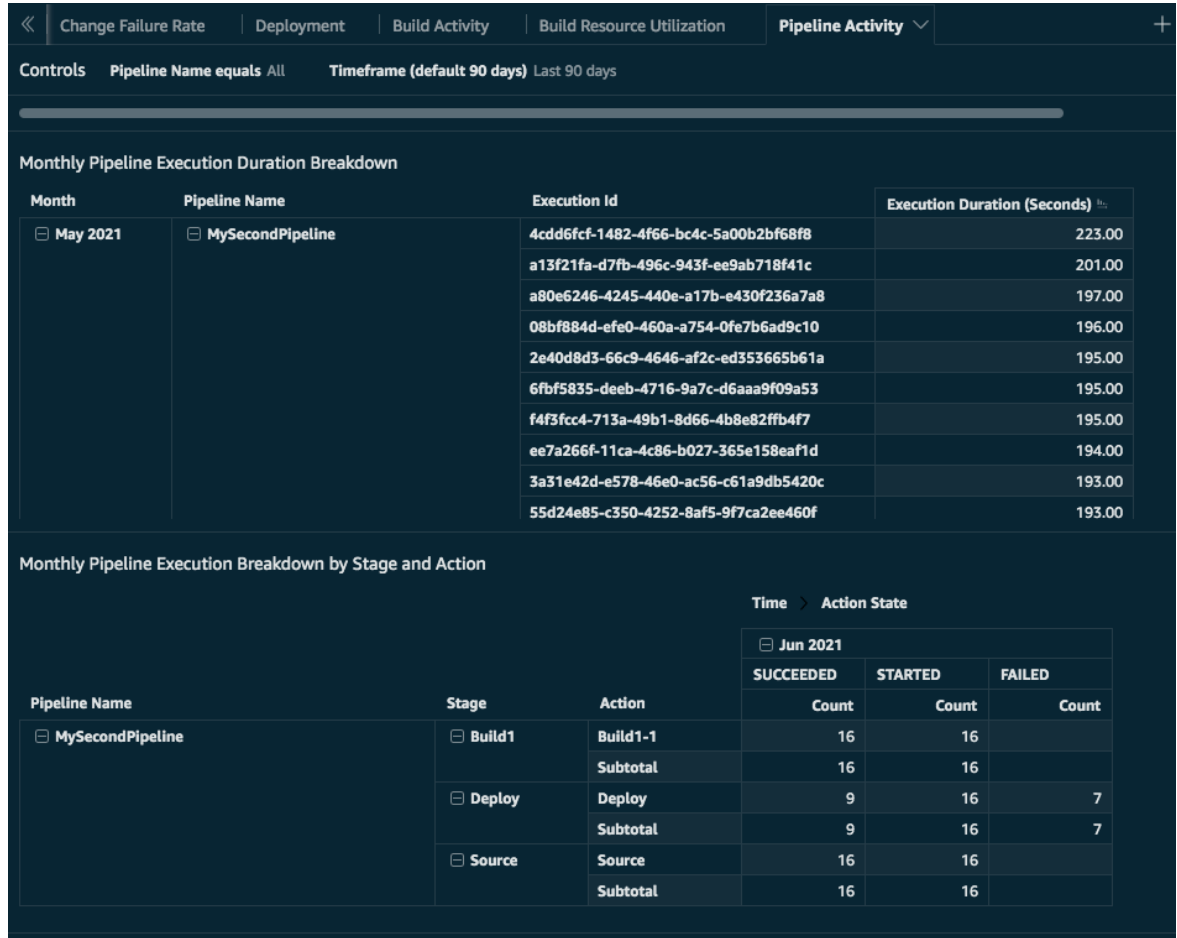


Figure 18: Pipeline dashboards - 2

Build visualizations with Amazon Athena and Tableau

You can build visualizations using Tableau and Amazon Athena for the views created by this solution. For more information, refer to [Building AWS Data Lake visualizations with Amazon Athena and Tableau](#). The following database information can be used to build database connection:

- Athena database name: `aws_devops_metrics_db_so0143`
- You can build visualizations for the following views:
 - `code_change_activity_view`: This view contains data related to code pushes to AWS CodeCommit.
 - `code_deployment_detail_view`: This view contains data related to code deployments using AWS CodeDeploy.
 - `code_build_detail_view`: This view contains data related to code builds generated by AWS CodeBuild.
 - `code_pipeline_detail_view`: This view contains data related to code builds generated by AWS CodePipeline.
 - `recovery_time_detail_view`: This view contains Amazon CloudWatch Alarm data related to Mean Time To Recover (MMTR) metrics. The **duration_minutes** column shows how long it takes to restore a service from its failure to success state at one time.
- Table:
 - `aws_devops_metrics_table` Table: This table is the entry point to most of data in the Amazon S3 metrics bucket (`s3://YourS3MetricsBucket/DevopsEvents/`). It is the base table for all the views except for `code_build_detail_view`. Do not directly build visualizations for this table. You should build visualizations for the views.
 - `aws_codebuild_metrics_table` Table: This table is the entry point to CodeBuild data in the Amazon S3 metrics bucket (`s3://YourS3MetricsBucket/CodeBuildEvents/`). It is the base table for `code_build_detail_view`. Do not directly build visualizations for this table. You should build visualizations for the view.

For more information about the database schema, refer to [Database schema information \(p. 21\)](#).

Running queries and work with query results and output files in Amazon Athena

You can [run SQL queries using Amazon Athena](#) to directly query the table and views created by the solution. For details about these table and views, refer to [Database schema information \(p. 21\)](#) and [Build visualizations with Amazon Athena and Tableau \(p. 36\)](#). Query results are stored as CSV files in the metrics S3 bucket under the prefix, `athena_query_output`. You can also download query result files directly from the Amazon Athena console.

Note

This solution implements data partition and parquet data storage for performance optimization and cost reduction. When running your own queries, we recommend that you use the `created_at` (timestamp) partition key. For more information, refer to [Performance tuning in Athena](#) in the *Amazon Athena User Guide*.

Uninstall the solution

You can uninstall the AWS DevOps Monitoring Dashboard solution from the AWS Management Console or by using the AWS Command Line Interface (AWS CLI). You must manually delete the Amazon Simple Storage Service (Amazon S3) buckets created by this solution. To protect customer data, AWS Solutions Implementations do not automatically delete these resources in case you need to retain stored data.

Note: The Amazon S3 buckets are configured with the retention policy set to **Retain**. You must manually delete them.

Using the AWS Management Console

1. Sign in to the [AWS CloudFormation console](#).
2. Select this solution's installation stack.
3. Choose **Delete**.

Using AWS Command Line Interface

Determine whether the AWS Command Line Interface (AWS CLI) is available in your environment. For installation instructions, refer to [What Is the AWS Command Line Interface](#) in the *AWS CLI User Guide*. Optionally, you can use the [AWS CloudShell](#) service to run AWS CLI commands. After confirming that the AWS CLI is available, run the following command.

```
$ aws cloudformation delete-stack --stack-name <installation-stack-name>
```

Note

The Amazon S3 buckets are configured with the retention policy set to **Retain**. You must manually delete them.

Collection of operational metrics

This solution includes an option to send anonymous operational metrics to AWS. We use this data to better understand how customers use this solution and related services and products. When activated, the following information is collected and sent to AWS:

- **Solution ID:** The AWS solution identifier
- **Unique ID (UUID):** Randomly generated, unique identifier for each deployment of the solution template
- **Timestamp:** Data-collection timestamp
- **Data:** Nested structure containing the following information:
 - **Region:** The AWS Region in which the solution is deployed
 - **Version:** The version of the deployed solution
 - **RequestType:** Stack action - Create, Update, or Delete
 - **DataType:** Sender of the metrics such as Lambda function
 - **AthenaQueryExecutionCount:** Number of successful Athena queries run by the solution (mainly QuickSight)
 - **QuickSightDeployed:** Yes or no. Customer configuration at stack deployment
 - **AthenaQueryDataRetrievalDuration:** The duration in which Athena query retrieves data. By default Athena fetches data within the past 90 days. Customer configuration at stack deployment
 - **Repository:** All or a list entered by customer. Customer configuration at stack deployment
 - **S3TransitionDays:** The number of days after which Amazon S3 objects are transitioned to Amazon S3 Glacier storage class. Customer configuration at stack deployment

AWS owns the data gathered through this survey. Data collection is subject to the [AWS Privacy Policy](#). To opt out of this feature, complete the following task.

Modify the AWS CloudFormation template mapping section from:

```
"AnonymousData" : {  
  " SendAnonymousUsageData" : { "Data" : "Yes" }  
},
```

to

```
"AnonymousData" : {  
  " SendAnonymousUsageData" : { "Data" : "No" }  
},
```

Source code

Visit our [GitHub Repository](#) to download the templates and scripts for this solution, and to share your customizations with others. The CloudFormation templates are generated using the AWS Cloud Development Kit (CDK). Refer to the [README.md file](#) for additional information.

Revisions

Date	Change
March 2021	Initial release
June 2021	Release version 1.1: Added support for AWS CodeBuild and AWS CodePipeline metrics. For additional information, refer to the CHANGELOG.md file

Contributors

- Aijun Peng
- Mike O'Brien
- Nikhil Reddy

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents AWS current product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. AWS responsibilities and liabilities to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

AWS DevOps Monitoring Dashboard is licensed under the terms of the of the Apache License Version 2.0 available at [The Apache Software Foundation](#).