REFERENCE ARCHITECTURE

Enterprise Data Center Infrastructure
June 2019

Deploying IBM Cloud Private for Data* on VMware vSAN ReadyNodes*

Authors

Karol Brejna
Senior Architect,
Data Center Group (DCG), Intel

Lokendra Uppuluri
Senior Architect,
Data Center Group (DCG), Intel

Malgorzata Rembas
Cloud Solutions Architect,
Data Center Group (DCG), Intel

1. Executive Summary

IBM Cloud Private for Data* is a scalable cloud platform that runs on enterprise infrastructure and facilitates multi-cloud deployments. It extends the IBM Cloud Private* foundation with powerful tools for data management and analytics. It is built on open source frameworks and provides the following foundational services:

- Multi-cloud management and orchestration
- Microservices architecture that allows for analytics as a service (AaaS)
- Containerization for easy, consistent deployment of artificial intelligence (AI) and analytics algorithms
- Infrastructure and administration user accounts, security, logging, and monitoring
- Data-virtualization technology that helps access, govern, and analyze data irrespective of where it resides
- Effective operationalization of analytics and AI: data ingestion, data preparation, AI and analytics, and visualization

The Intel® Reference Solution for IBM Cloud Private for Data provides an integrated and extensible analytics platform with all the tools and services necessary to navigate multiple clouds, effectively harness data, and extract value using powerful analytics and AI tools. The combination of IBM Cloud Private for Data and VMware vSAN* on Intel® hardware can provide an economical, enterprise-grade way forward for digital transformation. In addition, the Intel Reference Solution for IBM Cloud Private for Data enables you to quickly deploy a performance-optimized infrastructure built on Intel® Xeon® Scalable processors, Intel® 3D NAND Solid State Drives (SSDs), Intel® Optane™ DC SSDs, and the Intel® Ethernet 700 Series.

This reference architecture (RA) will show you how to prepare, provision, deploy, and manage an IBM Cloud Private for Data solution. The intended audience for this RA includes system administrators and system architects. Some experience with Docker* and Kubernetes* is recommended.

2. IBM, Intel, VMware, Docker*, and Kubernetes*

Beyond IBM Cloud Private for Data software and Intel hardware, the Intel Reference Solution for IBM Cloud Private for Data harnesses the power of VMware vSAN ReadyNodes*, Docker containers, and Kubernetes container orchestration to provide a comprehensive, reliable cloud solution.

Contents

2.1 IBM Cloud Private for Data

IBM Cloud Private for Data is based on IBM Cloud Private, which is a scalable cloud platform that is built on a container architecture based on Kubernetes. IBM Cloud Private includes management services like logging, monitoring, access control, and event management, and it brings cloud-native container capabilities to enterprise IT for all container use cases. IBM Cloud Private for Data extends IBM Cloud Private to provide integrated data management and AI/analytics capabilities, including:

- **Multi-cloud-based compute**—rapid deployment, lower operating expenditures (OpEx), elasticity, scalability, and choice
- **Microservices architecture**—allows for Architecture-as-a-Service
- **Containerization**—easy, consistent deployment of AI and analytics algorithms
- **Infrastructure and administration**—user accounts, security, logging, and monitoring

2.2 Intel® Hardware

Intel and IBM chose Intel Xeon Scalable processors for the Intel Reference Solution for IBM Cloud Private for Data because these processors support the most demanding workloads. Beyond compute based on Intel Xeon Scalable processors, the Intel Reference Solution for IBM Cloud Private for Data uses the Intel SSD Data Center Family and Intel Optane DC SSDs for performant all-flash storage and Intel® Ethernet Network Adapters for low-latency performance across the storage, management, and virtualization networks that undergird the solution.

2.3 VMware vSAN ReadyNodes

The Intel Reference Solution for IBM Cloud Private for Data uses VMware vSAN ReadyNodes to deliver hyperconverged infrastructure and serve as the foundation for a transformed, software-defined data center. Verified Intel solutions—such as the Intel Reference Solution for IBM Cloud Private for Data—are built on hardware certified for VMware vSAN ReadyNode* and are tightly specified by Intel and VMware to deliver balanced and optimized performance.

2.4 Docker Containers

Docker containers provide lightweight, standalone, executable software packages that provide everything needed to run an application, regardless of the infrastructure or operating system.

2.5 Kubernetes

IBM Cloud Private provides an open container platform based on Kubernetes for orchestration that is extended to and made use of by IBM Cloud Private for Data. The platform enables automated deployment, scaling, and management of containerized applications.

3. Intel Reference Solution for IBM Cloud Private for Data System Architecture

The Intel Reference Solution for IBM Cloud Private for Data utilizes IBM Cloud Private as an underlying infrastructure. The architecture discussion in the following section introduces IBM Cloud Private architecture followed by IBM Cloud Private for Data architecture, with emphasis on the latter.

3.1 IBM Cloud Private* Cluster Overview

An IBM Cloud Private cluster has four main classes of nodes:

- **Boot Node**—Used for running installation, configuration, node scaling, and cluster updates
- **Master Node**—Provides management services and controls the worker nodes in a cluster; master nodes host processes that are responsible for resource allocation, state maintenance, scheduling, and monitoring
- **Worker Node**—Provides a containerized environment for running tasks; as demands increase, you can easily add more worker nodes to your cluster to improve performance and efficiency
- **Proxy Node**—Transmits external requests to the services created inside your cluster

IBM Cloud Private clusters can also have two optional types of nodes:

- **Management Node**—Hosts management services such as monitoring, metering, and logging
- **Vulnerability Advisor (VA) Node**—Used for running VA services, which can be resource intensive

3.1.1 IBM Cloud Private Components

IBM Cloud Private provides a container runtime (Docker) and a container orchestration platform (Kubernetes), along with an integrated layer of additional services.

Helm* is the open source component, which is essential to the IBM Cloud Private platform. It is the package-management system native to Kubernetes, which is used for application management inside an IBM Cloud Private cluster. These Kubernetes packages are called charts, and they contain the details about your application. Helm can create new charts from scratch, package them into archive TGZ files, interact with repositories where charts are stored, install and uninstall them into Kubernetes clusters, and manage their release cycles.

Other components of an IBM Cloud Private cluster work alongside these main components to provide services such as authentication, storage, networking, logging, and monitoring. A cluster-management console is also provided, which allows for centralized management of these services.

Figure 2 shows the IBM Cloud Private system architecture with nodes in virtual machines (VMs) on VMware ESXi* servers based on Intel hardware.

3.2 IBM Cloud Private for Data Architecture

IBM Cloud Private for Data is composed of pre-configured microservices that run on a multi-node IBM Cloud Private cluster. The microservices enable you to connect to your data sources so that you can catalog, govern, explore, profile, transform, and analyze your data from a single web application.

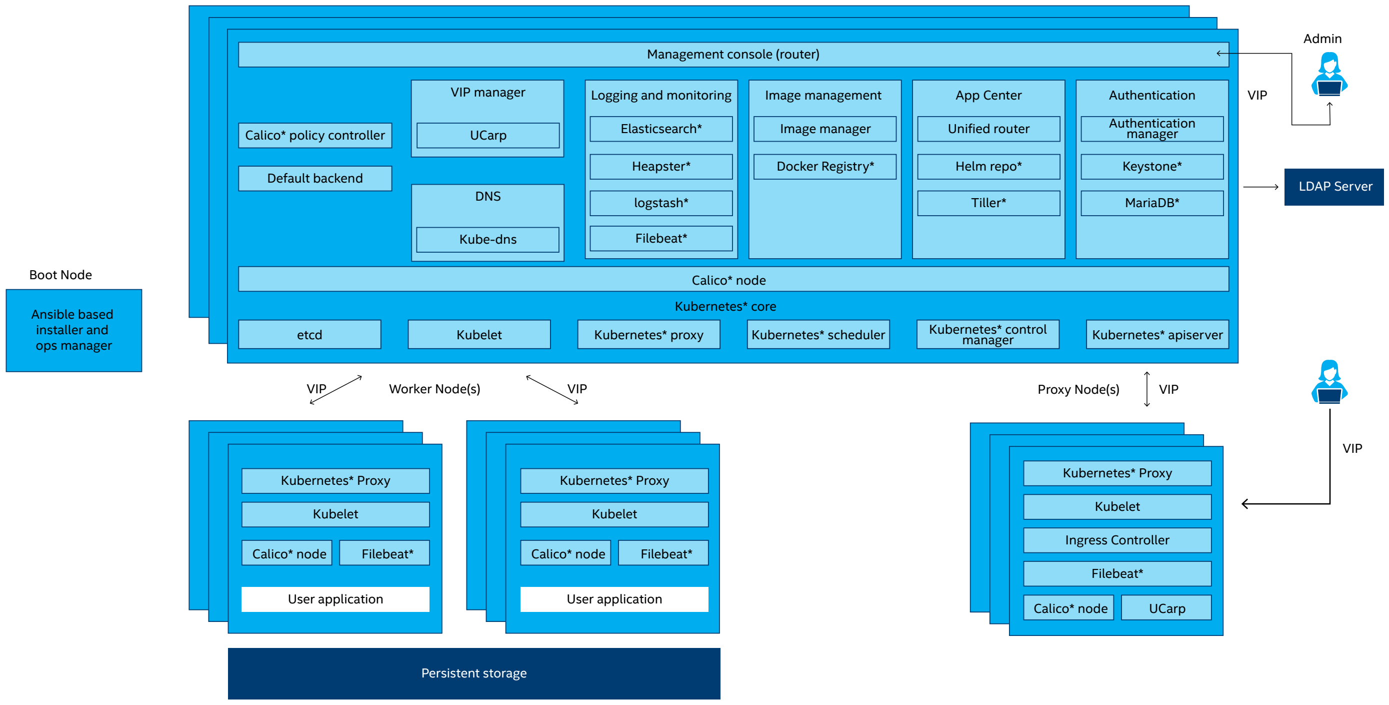


Figure 1. Interrelation of IBM Cloud Private* nodes

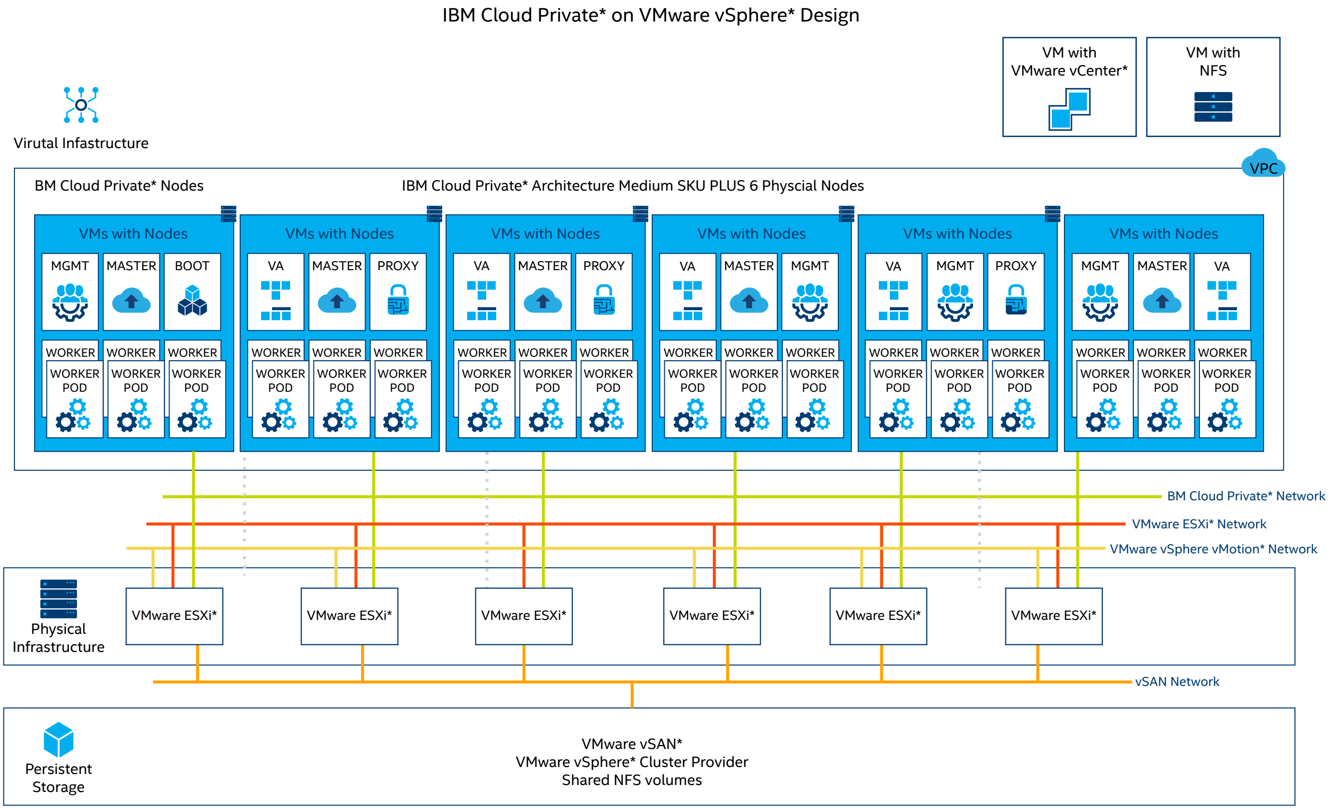


Figure 2. IBM Cloud Private* system architecture for a medium-sized deployment

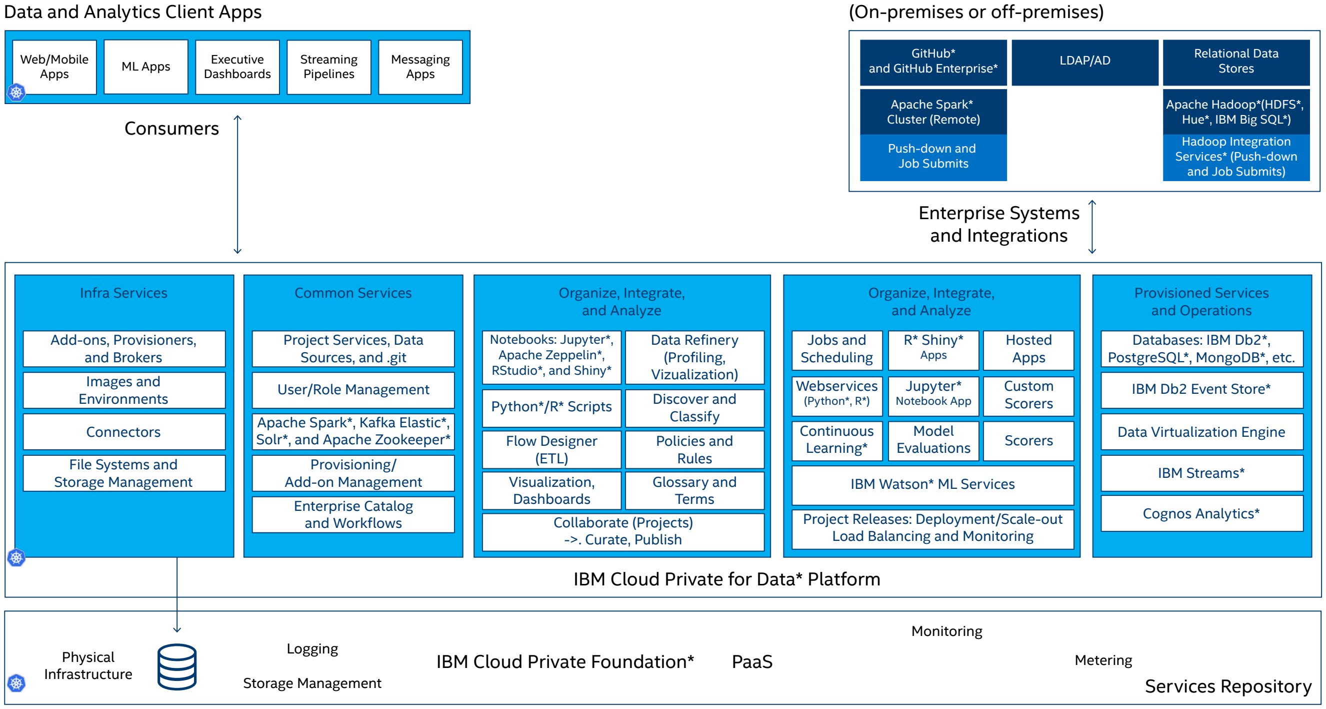

Data and Analytics Client Apps
Web/Mobile Apps
ML Apps
Executive Dashboards
Streaming Pipelines
Messaging Apps
Consumers
(On-premises or off-premises)
GitHub* and GitHub Enterprise*
LDAP/AD
Relational Data Stores
Apache Spark* Cluster (Remote)
Push-down and Job Submits
Apache Hadoop*(HDFS*, Hue*, IBM Big SQL*)
Hadoop Integration Services* (Push-down and Job Submits)
Enterprise Systems and Integrations
Infra Services
Add-ons, Provisioners, and Brokers
Images and Environments
Connectors
File Systems and Storage Management
Common Services
Project Services, Data Sources, and .git
User/Role Management
Apache Spark*, Kafka Elastic*, Solr*, and Apache Zookeeper*
Provisioning/ Add-on Management
Enterprise Catalog and Workflows
Organize, Integrate, and Analyze
Notebooks: Jupyter*, Apache Zeppelin*, RStudio*, and Shiny*
Data Refinery (Profiling, Vizualization)
Python*/R* Scripts
Discover and Classify
Flow Designer (ETL)
Policies and Rules
Visualization, Dashboards
Glossary and Terms
Collaborate (Projects) ->. Curate, Publish
Organize, Integrate, and Analyze
Jobs and Scheduling
R* Shiny* Apps
Hosted Apps
Webservices (Python*, R*)
Jupyter* Notebook App
Custom Scorers
Continuous Learning*
Model Evaluations
Scorers
IBM Watson* ML Services
Project Releases: Deployment/Scale-out Load Balancing and Monitoring
Provisioned Services and Operations
Databases: IBM Db2*, PostgreSQL*, MongoDB*, etc.
IBM Db2 Event Store*
Data Virtualization Engine
IBM Streams*
Cognos Analytics*
IBM Cloud Private for Data* Platform
Physical Infrastructure
Logging
Storage Management
IBM Cloud Private Foundation*
PaaS
Monitoring
Metering
Services Repository

Figure 3. IBM Cloud Private for Data* components overview

3.2.1 Software Components

Principal software components for Intel Reference Solution for IBM Cloud Private for Data deployment as an application on IBM Cloud Private in a VMware* infrastructure are shown in Table 1.

Table 1. Intel® Reference Solution for IBM Cloud Private for Data* component software and versions

SOFTWARE NAME	VERSION
VMware vSphere ESXi*	ESXi 6.7 U1
VMware vCenter*	VMware vCenter Server Appliance 6.7 U1*
VMware vSAN*	ESXi 6.7 U1
IBM Cloud Private* Enterprise Edition	3.1.2
IBM Cloud Private for Data* Enterprise Edition	1.2.1.1 (CC11ML)
IBM Db2 Warehouse*	3.3.0 (CC0EMEN)
Data-virtualization add-on	1.2.1.1 (CC11REN)
Red Hat* Enterprise Linux* (RHEL*)	v7.5

3.2.1.1 IBM Db2 Warehouse*

In general, IBM Db2 Warehouse* is a client-managed, preconfigured data warehouse that runs in private clouds, virtual private clouds, and other container-supported infrastructures. This data warehouse is designed to provide the ideal solution when you must maintain control of your data but want cloud-like flexibility. It includes in-memory processing to deliver fast answers to queries, in addition to massively parallel processing (MPP) to help you scale out and scale up capabilities as demand grows. For analytics, you can use Db2 Warehouse products to make use of familiar structured query language (SQL), integrated R* and Python*, or robust in-database analytics—including geospatial analytics.

This RA uses an integrated database based on IBM Db2 Warehouse (symmetric multiprocessing [SMP] version), which is designed for high performance and runs on a single node.

3.2.1.2 Data Virtualization

Data virtualization integrates data sources across multiple types and locations and turns them into one logical data view. This virtual data lake makes the job of getting value out of the data easy.

Creating connections to data sources provides a quick view across your organization's data. This virtual data platform enables real-time analytics without moving data, duplication, extract, transform, and load operations (ETLs), and additional storage requirements, so processing times are greatly accelerated. This brings real-time insightful results to decision-making applications or analysts more quickly and dependably than existing methods.

3.2.2 Hardware Components and Cluster Sizing

Hardware components and configuration details (such as networking components and BIOS settings) are consistent with the specification for the Intel Reference Solution for IBM Cloud Private for Data. However, Intel advises following the additional sizing guidelines in Table 2 to determine how many physical two-socket physical nodes in which hardware configuration ("Dev/Test," "Base," "Plus," or "Custom") are required to support a given number of VMs with workloads on the small, medium, or large IBM Cloud Private clusters. The number of workers should be considered when deciding how many physical nodes to deploy.

Use the following section as a starting point to help determine the ideal cluster size and configuration for your environment. The cluster configuration is based on recommendations from IBM: ibm.com/support/knowledgecenter/SSQNUZ_current/com.ibm.icpdata.doc/zen/install/reqs-ent.html.

3.2.2.1 Cluster Sizing

The configurations described in Table 2 include the capacity needed for boot, master, proxy, management, and VA node VMs, with 20 percent reserved for compute resources. Sizing of the VM nodes is shown in Table 3. The number of users specified assumes 25 percent of the users are generic analytics users, 25 percent are data scientists, 35 percent are data stewards, 14 percent are data engineers, and 1 percent are admins.

Table 2. Number of physical nodes required to support different ranges of VM-based workloads by cluster size and configuration for the Intel® Reference Solution for IBM Cloud Private for Data*

Configuration	SMALL—IBM CLOUD PRIVATE FOR DATA*	MEDIUM—IBM CLOUD PRIVATE FOR DATA	LARGE—IBM CLOUD PRIVATE FOR DATA
Node hardware configuration	6 worker-node VMs + 1 data-virtualization add-on worker-node VM	12 worker-node VMs + 1 data-virtualization add-on node VM + 1 IBM Db2* add-on worker-node VM	18 worker-node VMs + 1 data-virtualization add-on node VM + 2 Db2 add-on worker-node VMs
Users supported	40	80	120
Dev/Test SKU**	7 physical nodes	Not applicable (N/A)	N/A
Base SKU	5 physical nodes	8 physical nodes	10 physical nodes
Plus SKU	4 physical nodes	6 physical nodes	8 physical nodes
Custom SKU	N/A	5 physical nodes	7 physical nodes

** See SKU details in the following Dev/Test Configuration section

Table 3. VM sizing for the Intel® Reference Solution for IBM Cloud Private for Data*

	Small		Medium		Large	
	vCPU Cores	vMemory (GB)	vCPU Cores	vMemory (GB)	vCPU Cores	vMemory (GB)
Boot	2	8	2	8	4	8
Master	16	32	16	32	16	128
Management	8	16	16	32	16	128
Proxy	4	16	4	16	4	16
VA	N/A	N/A	6	24	6	48
Workers	16	32	16	32	16	32
Data virtualization	16	32	16	32	16	32
IBM Db2*	N/A	N/A	16	64	16	64

3.2.2.2 Intel Reference Solution for IBM Cloud Private for Data Configurations

Use Tables 4–7 to help determine the ideal Intel Reference Solution for IBM Cloud Private for Data configuration for your environment.

3.2.2.2.1 Dev/Test Configuration

Table 4. Intel® Reference Solution for IBM Cloud Private for Data* Dev/Test configuration component hardware

Role	Platform	Configuration
VMware vSphere ESXi*	Intel® Server Board S2600WF0	2 x Intel® Xeon® Silver 4210 processor (2.20 GHz, 10 cores, 20 threads), or a higher number Intel Xeon Scalable processor 12 x 16 GB 2,400Hz DDR4—192 GB RAM 1 x 480 GB Intel® SSD DC S3520 (M.2, SATA 3.0, 6 gigabit per second [Gb/s]) 1 x 375 GB Intel® Optane™ SSD DC P4800X (2.5-in. PCIe*) 3 x 2 TB Intel SSD DC P4510 (2.5-in. PCIe, NVM Express* [NVMe*] 3.0 x4) 1 x Intel® Ethernet Converged Network Adapter X710-DA2 1 x network interface controller (NIC)-integrated 1 gigabit Ethernet (GbE)

3.2.2.2.2 Base Configuration

Table 5. Intel® Reference Solution for IBM Cloud Private for Data* Base configuration component hardware

Role	Platform	Configuration
VMware vSphere ESXi*	Intel® Server Board S2600WF0	2 x Intel® Xeon® Gold 5218 processor (2.30 GHz, 16 cores, 32 threads), or a higher number Intel Xeon Scalable processor 12 x 32 GB 2,666MHz DDR4—384 GB RAM 2 x 480 GB Intel® SSD DC S3520 (M.2, SATA 3.0, 6 Gb/s) 2 x 375 GB Intel® Optane™ SSD DC P4800X (2.5-in. PCIe*) 6 x 2 TB Intel SSD DC P4510 (2.5-in. PCIe, NVMe* 3.0 x4) 2 x Intel® PCIe Extender AXXP3SWX08080 8-Port PCIe Gen3 x8 Switch AIC 1 x Intel® Ethernet Converged Network Adapter X710-DA2 1 x NIC-integrated 1 GbE

3.2.2.2.3 Plus Configuration

Table 6. Intel® Reference Solution for IBM Cloud Private for Data* Plus configuration component hardware

Role	Platform	Configuration
VMware vSphere ESXi*	Intel® Server Board S2600WF0	2 x Intel® Xeon® Gold 6248 processor (2.50 GHz, 20 cores, 40 threads), or a higher number Intel Xeon Scalable processor 24 x 32 GB 2,666MHz DDR4—768 GB RAM** 2 x 480 GB Intel® SSD DC S3520 (M.2, SATA 3.0, 6 Gb/s) 2 x 375 GB Intel® Optane™ SSD DC P4800X (2.5-in. PCIe*) 8 x 2 TB Intel SSD DC P4510 (2.5-in. PCIe, NVMe* 3.0 x4) 2 x Intel® PCIe Extender AXXP3SWX08080 8-Port PCIe Gen3 x8 Switch AIC 1 x Intel® Ethernet Converged Network Adapter X710-DA4 Or 2 x 25 GbE Intel® Ethernet Controller XXV710 1 x NIC-integrated 1 GbE

** A cost-optimized configuration can use 512 GB RAM (2 x CPU, 16 x 32 GB 2,666 MHz DDR4). However, this is an unbalanced configuration and will decrease memory-access performance.

3.2.2.2.4 Custom Configuration

Table 7. Intel® Reference Solution for IBM Cloud Private for Data* Custom configuration component hardware

Role	Platform	Configuration
VMware vSphere ESXi*	Intel® Server Board S2600WF0	2 x Intel® Xeon® Platinum 8268 processor (2.90 GHz, 24 cores, 48 threads), or a higher number Intel Xeon Scalable processor 24 x 32 GB 2,666MHz DDR4—768 GB RAM** 2 x 480 GB Intel® SSD DC S3520 (M.2, SATA 3.0, 6 Gb/s) 2 x 375 GB Intel® Optane™ SSD DC P4800X (2.5-in. PCIe*) 10 x 2 TB Intel SSD DC P4510 (2.5-in. PCIe, NVMe* 3.0 x4) 2 x Intel® PCIe Extender AXXP3SWX08080 8-Port PCIe Gen3 x8 Switch AIC 1 x Intel® Ethernet Converged Network Adapter X710-DA4 Or 2 x 25 GbE Intel® Ethernet Controller XXV710 1 x NIC-integrated 1 GbE

** A cost-optimized configuration can use 512 GB RAM (2 x CPU, 16 x 32 GB 2,666 MHz DDR4). However, this is an unbalanced configuration and will decrease memory-access performance.

3.2.2.3 BIOS Settings

Table 8. Intel® Reference Solution for IBM Cloud Private for Data* BIOS settings

Component	Setting
BIOS	SE5C620.86B.02.01.0008
Baseboard management controller (BMC)	1.93.870cf4f0
FRU	1.93/1.39/1.74
Intel® Management Engine (Intel® ME)	04.01.04.251

Table 9. VMware ESXi* node BIOS settings

Technologies	Setting
Trusted Platform Module (TPM) 2.0	Enabled
Intel® Trusted Execution Technology (Intel® TXT)	Enabled
Intel® Hyper-Threading Technology (Intel® HT Technology)	Enabled
Intel® Turbo Boost Technology	Enabled

Intel® Speed Shift Technology—hardware P-states (HWP) native	Enabled
Intel® Volume Management Device (Intel® VMD)	Disabled
Intel® Virtualization Technology for Directed I/O (Intel® VT-d), virtualization	Enabled
Power-management settings	Performance, workload input/output (I/O) intensive
PCIe*, Single-Root I/O Virtualization (SR-IOV) support	Enabled
Integrated I/O	Intel VT-d
Advanced	Set fan profiles to performance
	PCIe configuration, NIC configuration: disable 3, 4

3.2.3 Networking Components

Table 10. Network-infrastructure components

Role	Qty.	Platform	Description
High performance top-of-rack (ToR) switches	2	Arista DCS-7050SX2-72Q 40 GbE SFP+*	arista.com/en/products/7050x-series
BMC switch	1	Arista DCS-7010T 48 ports 1 GbE*	arista.com/en/products/7010-series

4. Intel Hardware Details

Intel processing, networking, and storage hardware provide the performance-optimized foundation for the Intel Reference Solution for IBM Cloud Private for Data.

4.1 2nd Generation Intel® Xeon® Scalable Processors

2nd Generation Intel Xeon Scalable processors provide a new foundation for more secure, agile, multi-cloud data centers. This platform provides businesses with breakthrough performance to handle system demands ranging from entry-level cloud servers to compute-hungry tasks including real-time analytics, virtualized infrastructure, and high-performance computing (HPC). This processor family includes technologies for accelerating and securing specific workloads:

- **Intel® Advanced Vector Extensions 512 (Intel® AVX-512)** expands the number of registers and adds instructions to streamline processing. The result is high performance for mixed workloads.
- **Intel® Transactional Synchronization Extensions New Instructions (Intel® TSX-NI)** provides high performance of multi-threaded workloads.
- **Intel® Trusted Execution Technology (Intel® TXT)** provides the necessary underpinnings to evaluate the computing platform and its security.
- **Intel® Platform Trust Technology (Intel® PTT)** provides more secure encryption keys in a Trusted Platform Module (TPM) integrated directly into the chipset.

2nd Generation Intel Xeon Scalable processors were selected for the RA because they meet the performance needs for demanding, mission-critical workloads.

4.2 Intel® Ethernet Network Adapter X710

Intel Ethernet Network Adapter X710 delivers excellent performance with a throughput of 40 gigabits per second (Gb/s) in a PCIe* v3.0 x8 slot.[1]

Optimized performance vectors and key uses include:

- **Small Packet Performance:** Achieves excellent throughput on smaller payload sizes
- **Virtualized Performance:** Alleviates hypervisor I/O bottlenecks by providing flow separation for VMs
- **Network Virtualization:** Offloads network-virtualization overlays, including Virtual Extensible LAN (VXLAN), Network Virtualization Using Generic Routing Encapsulation (NVGRE), GENEVE*, Multiprotocol Label Switching (MPLS), and VXLAN-GPE with Network Service Header (NSH)

4.3 Intel® Data Center SSDs

The Intel Reference Solution for IBM Cloud Private for Data uses all-flash storage to minimize storage latency and increase storage availability.

- **Intel SSD DC S3520:** Provides data integrity, performance consistency, and drive reliability for read-intensive workloads in the Intel Reference Solution for IBM Cloud Private for Data, such as boot, web server, low-rate operational databases, and analytics. IBM and Intel selected the Intel SSD DC S3520 for the Intel Reference Solution for IBM Cloud Private for Data because it provides an ideal combination of performance and cost.
- **Intel SSD DC P4510:** Is a cloud-inspired Intel 3D NAND SSD optimized for cloud infrastructures, and it offers high quality, reliability, advanced manageability, and serviceability to minimize service disruptions for the Intel Reference Solution for IBM Cloud Private for Data.
- **Intel Optane SSD DC P4800X**: Allows bigger, more affordable datasets and helps eliminate data center storage bottlenecks for the Intel Reference Solution for IBM Cloud Private for Data. IBM and Intel selected the Intel Optane SSD DC P4800X for the Intel Reference Solution for IBM Cloud Private for Data in order to accelerate application performance, reduce transaction costs for latency-sensitive workloads, and help improve overall total cost of ownership (TCO).

5. Configuration Details

This section details some of the specific configuration steps taken in producing this reference architecture. Additional configuration details can be found in VMware and IBM documentation (linked to as appropriate).

IBM Cloud Private for Data can be installed either as a stand-alone application or as an application running in an existing or new IBM Cloud Private environment. This RA provides details for installing in an existing IBM Cloud Private environment. Figure 4 shows an overview of the installation steps. If you wish to install IBM Cloud Private for Data as a stand-alone package, refer to the IBM Knowledge Center website at ibm.com/support/knowledgecenter/SSQNUZ_1.2.1/com.ibm.icpdata.doc/zen/install/standovu.html#standovu.

5.1 Step 1: Prepare the Environment for VMware vSphere and VMware vSAN*

5.1.1 Summary of Tasks

See Appendix A for detailed instructions on configuring VMware vSphere and vSAN provisioning and orchestration.

5.2 Step 2: Install IBM Cloud Private Cluster

This step in the install process installs all the necessary components for IBM Cloud Private. Follow these steps to deploy IBM Cloud Private. For detailed instructions, see Appendix B or follow the IBM documentation at ibm.com/support/knowledgecenter/SSBS6K_3.1.2/installing/install_containers.html.

Summary of Tasks

- Prepare a Red Hat* Enterprise Linux* (RHEL*) 7.5 VM template
- Create an RHEL 7.5 boot-node VM
- Configure VMs for IBM Cloud Private
- Load the IBM Cloud Private Docker images on the boot machine
- Prepare the IBM Cloud Private configuration file
- Prepare the Ansible* host file
- Start the IBM Cloud Private Ansible installer in the Docker container on the boot node
- Run a health check after the installation to ensure all services are running properly

5.3 Step 3: Install IBM Cloud Private for Data Core Components Including Extensions

IBM Cloud Private for Data uses a modules mechanism in order to extend the functionality of the platform, provide a common way for partners to introduce new features, and simplify the installation.

Therefore, the process usually consists of the following steps:

- Make sure all requirements are met
- Obtain the installation packages
- Install the core IBM Cloud Private for Data components/services
- Install additional modules

For more information on the installation process and options, consult the documentation: ibm.com/support/knowledgecenter/SSQNUZ_1.2.1/com.ibm.icpdata.doc/zen/install/ovu.html.

Modules are packages that can be “add-ons” (priced components) or “extensions” (to which customers are entitled with their base IBM Cloud Private for Data Enterprise or Cloud Native editions).

5.3.1 Prerequisites

The prerequisites that must be fulfilled in order to install IBM Cloud Private for Data are described in the “System requirements for IBM Cloud Private for Data in an existing IBM Cloud Private installation” article at ibm.com/support/knowledgecenter/SSQNUZ_1.2.1/com.ibm.icpdata.doc/zen/install/reqs-exist-icp-inst.html#reqs-exist-icp-inst.

In short, you should make sure the following requirements are met:

IBM Cloud Private:

- IBM Cloud Private version 3.1.2 is installed
- IBM Cloud Private command-line interface (CLI) is installed (for more information, see IBM Cloud Private CLI in the IBM Cloud Private documentation)
- User ID used for installation of IBM Cloud Private for Data has administrator privileges in IBM Cloud Private
- Image enforcement policy, including the entry mycluster.icp:8500/zen/* has been created

Docker registry:

- Ensure Docker registry has a minimum of 150 GB of storage space
- For add-on installation in addition to IBM Cloud Private for Data, storage space must be allocated to the Docker registry

System:

- On each node, Linux kernel semaphore needs to be configured (*kernel.sem = 250 1024000 32 4096*)
- On first of the master nodes, you should have mounted two disks with at least 300 GB free space:
  - /ibm/
  - /installers/

Storage:

- Network File System (NFS)-based persistent storage needs to be installed and configured.
- A Kubernetes provisioner for NFS storage is configured (to support dynamic provisioning of persistent volumes).
- Storage should have about 600 GB available space.

5.3.2 Getting Installation Packages

IBM Cloud Private for Data comes in two versions:

- Cloud Native Edition
- Enterprise Edition

This RA uses Enterprise edition.

In order to obtain and prepare the installation files, do the following:

1. Download ICP4DATA_ENT_Req_PreExICPx86_Vnnn.bin on your master node in /installers/ dir.
2. Make the file executable (**chmod +x ICP4DATA_ENT_Req_PreExICPx86_Vnnn.bin)**.
3. Run the binary file **(./ICP4DATA_ENT_Req_PreExICPx86_Vnnn.bin)**. Here, you need to accept the license and the download will start. The package is about 40 GB, so it will take some time.
4. Extract the downloaded package to /ibm/ dir **(tar -xvf icp4d_ee_nnn.tar -C /ibm)**. Go to /ibm/ and make the installer executable **(chmod +x installer_filename)**.

5.3.3 Core Components Installation

Before you install IBM Cloud Private for Data, ensure that the installation files are available on the master node (master-1) of your cluster.

Complete the following steps:

1. Use a Secure Shell (SSH) connection to connect to the host where you have installation files, and then go into the /ibm folder.
2. Log on to Helm/kubectl and to Docker:

```
cloudctl login -a https://172.17.255.240:8443
--skip-ssl-validation docker login mycluster.
icp:8500
```

3. Run the installer file. Accept all licenses, choose the storage class configured for NFS, Zen* namespace, and the mycluster.icp:8500 registry. Installation will take about two hours.

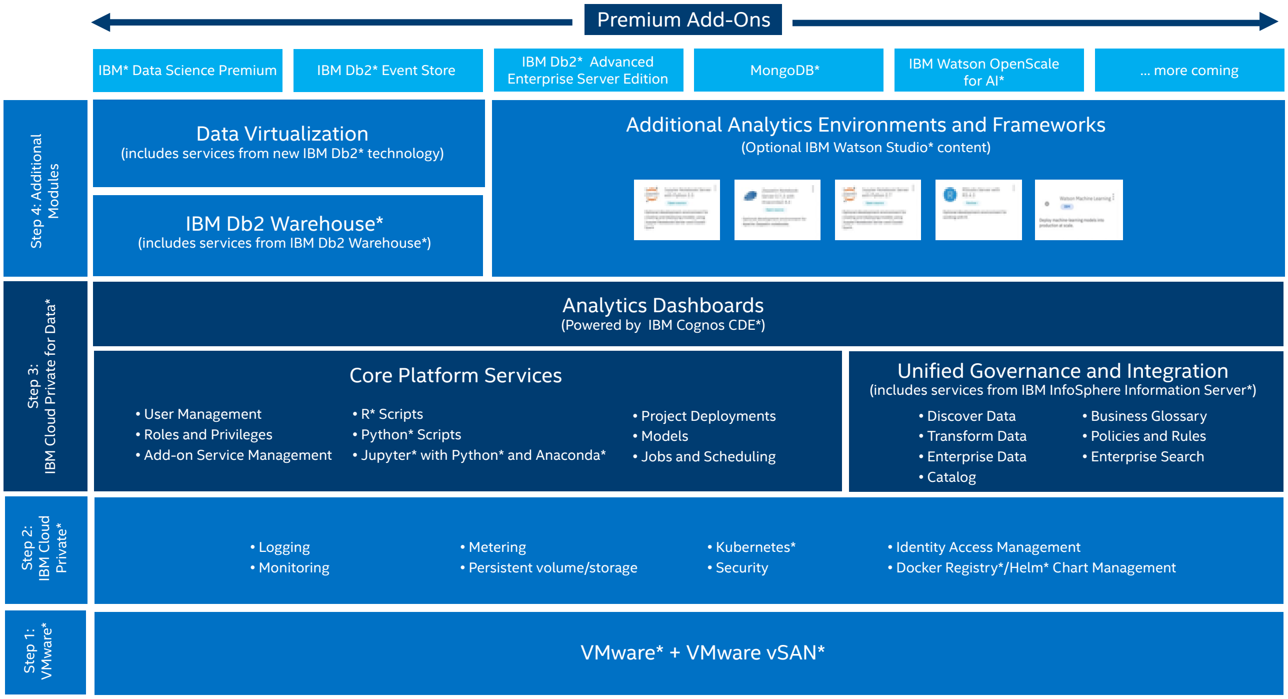

Premium Add-Ons
IBM* Data Science Premium
IBM Db2* Event Store
IBM Db2* Advanced Enterprise Server Edition
MongoDB*
IBM Watson OpenScale for AI*
... more coming
Step 4: Additional Modules
Data Virtualization
(includes services from new IBM Db2* technology)
IBM Db2 Warehouse*
(includes services from IBM Db2 Warehouse*)
Additional Analytics Environments and Frameworks
(Optional IBM Watson Studio* content)
Step 3: IBM Cloud Private for Data*
Analytics Dashboards
(Powered by IBM Cognos CDE*)
Core Platform Services
• User Management
• Roles and Privileges
• Add-on Service Management
• R* Scripts
• Python* Scripts
• Jupyter* with Python* and Anaconda*
• Project Deployments
• Models
• Jobs and Scheduling
Unified Governance and Integration
(includes services from IBM InfoSphere Information Server*)
• Discover Data
• Transform Data
• Enterprise Data
• Catalog
• Business Glossary
• Policies and Rules
• Enterprise Search
Step 2: IBM Cloud Private*
• Logging
• Monitoring
• Metering
• Persistent volume/storage
• Kubernetes*
• Security
• Identity Access Management
• Docker Registry*/Helm* Chart Management
Step 1: VMware*
VMware* + VMware vSAN*

Figure 4. Modularized installation of IBM Cloud Private for Data*

5.3.4 Installing IBM Cloud Private for Data Add-on Modules

For this RA, the following additional components were chosen:

- Db2 Warehouse
- Data virtualization
- RStudio* analytics environment

5.3.4.1 Db2 Warehouse

You can optionally create one or more databases within IBM Cloud Private for Data, IBM Db2 Warehouse being one of them. For this RA, one-node IBM Db2 Warehouse was chosen, because it provides a high-performing analytic engine that combines in-memory processing with integrated in-database analytics.

To create an integrated database, you must complete the following tasks:

- Download the database installation package (ibm.com/support/knowledgecenter/SSQNUZ_current/com.ibm.icpdata.doc/zen/admin/download-db-pkg.html?view=kc#download-db2-warehouse-pkg)
- Prepare your cluster for the database (ibm.com/support/knowledgecenter/SSQNUZ_current/com.ibm.icpdata.doc/zen/admin/create-dedicated-node.html?view=kc#create-dedicated-node)
- Configure database storage (ibm.com/support/knowledgecenter/SSQNUZ_current/com.ibm.icpdata.doc/zen/admin/database-storage.html?view=kc#task_sqy_f4q_1fb)
- Create a database deployment on the cluster (ibm.com/support/knowledgecenter/SSQNUZ_current/com.ibm.icpdata.doc/zen/admin/provision-db.html?view=kc#provision-db2-warehouse-db)

For more Db2 Warehouse information, see IBM Knowledge Center for details at ibm.com/support/knowledgecenter/SSQNUZ_current/com.ibm.icpdata.doc/zen/admin/create-db.html.

5.3.4.2 Data Virtualization

To enable data virtualization, the following tasks need to be completed:

- Complete any additional preparation.
- Download the data-virtualization installation package (ibm.com/support/knowledgecenter/SSQNUZ_current/com.ibm.icpdata.doc/dv/download-dv-pkg.html?view=kc). The software for installing the data virtualization add-on in your IBM Cloud Private for Data cluster is provided as a separate package, which you must download to your cluster. For more information, see: ibm.com/support/knowledgecenter/SSQNUZ_1.2.1/com.ibm.icpdata.doc/dv/download-dv-pkg.html.
- Provision data virtualization (ibm.com/support/knowledgecenter/SSQNUZ_current/com.ibm.icpdata.doc/dv/provision-dv.html?view=kc). Before using data virtualization, you must deploy and provision the add-on to IBM Cloud Private for Data.
- Give users access to data virtualization (ibm.com/support/knowledgecenter/SSQNUZ_current/com.ibm.icpdata.doc/dv/dv_user_management.html?view=kc). IBM Cloud Private for Data users that need to use data-virtualization functions must be assigned specific data-virtualization roles based on their job descriptions.

In the example shown above, the default settings of Docker prevented the installation from succeeding and some additional preparation was required.

To install data virtualization, “Base Device Size” `(docker info | grep Base)` should be larger than 15 GB.

To change this, add `dm.basesize=20G` to `/etc/docker/daemon.json`. The section responsible for the setting should look similar to this:

```
{
        “storage-opts”: [
        “dm.thinpooldev=/dev/mapper/docker-
thinpool”,
        “dm.basesize=20G”,
        “dm.use _ deferred _ removal=true”,
        “dm.use _ deferred _ deletion=true”
        ]
}
```

After the change, configuration update and service restart is required:

```
systemctl daemon-reload
systemctl restart docker
```

Note: This should be done on a host where you have installation files.

For other steps (downloading, provisioning, and giving access), see the product documentation at ibm.com/support/knowledgecenter/SSQNUZ_1.2.1/com.ibm.icpdata.doc/dv/install-dv-add-on.html.

5.3.5 Analytics Development Environments

By default, IBM Cloud Private for Data includes Jupyter Notebook Server* with Python 3.6. You can optionally install other development environments.

You can install one or more of the following development environments on top of IBM Cloud Private for Data:

- Jupyter Notebook Server with Python 2.7
- Jupyter Notebook Server with Python 3.5
- RStudio Server with R3.4.3
- Apache Zeppelin Notebook Server 0.7.3* with Anaconda2 4.4*

To install a development environment:

1. SSH into the master node (master-1) of your cluster as root: `ssh root@MASTER _ 1 _ IP`
2. Download ICP4DATA_Vnnn_AnalyticsEnv.bin from IBM Passport Advantage*.
3. Change to the directory where the file was downloaded and make the BIN file executable:

   chmod +x ICP4DATA_Vnnn_AnalyticsEnv.bin
4. Run the BIN file:

   ./ICP4DATA_Vnnn_AnalyticsEnv.bin

   This downloads the following TAR file: analytics-environments.tar.
5. Extract the contents of the TAR file into the /modules directory that you created in your installation files partition. For example:

   tar -xvf analytics-environments.tar -C /IBM/modules/

   The contents of the analytics-environments.tar file are extracted into the /IBM/modules/ analytics-environments subdirectory.
6. Change to the **IBM/InstallPackage/ components** directory.
7. Run the deploy.sh script to deploy each environment that you want to use, refer to Table 11.

Table 11. Notebook settings for deploy.sh

Environment	Command
Jupyter Notebook Server* with Python 2.7*	./deploy.sh /IBM/modules/ analytics-environments/ 0150-py27spark202.tar
Jupyter Notebook Server with Python 3.5	./deploy.sh /IBM/modules/ analytics-environments/ 0160-py35spark221.tar
RStudio Server* with R3.4.3*	./deploy.sh /IBM/modules/ analytics-environments/ 0180-rstudio.tar
Apache Zeppelin Notebook Server* 0.7.3 with Anaconda2 4.4*	./deploy.sh /IBM/modules/ analytics-environments/ 0140-zeppelin.tar

For more Jupyter Notebook Server setting information see: ibm.com/support/knowledgecenter/SSQNUZ_1.2.1/com.ibm.icpdata.doc/zen/admin/enable-dev-environments.html#enable-dev-environments.

6. Cluster Verification

To validate the IBM Cloud Private for Data cluster, you need a benchmarking methodology that can validate key elements of IBM Cloud Private for Data: data collection, organization, and analysis.

6.1 Data Collection and Analysis Validation

To validate the ability to load data to the cluster and analyze it, this RA uses a set of tests based on the TPC-DS* benchmark. This gives the ability to run demanding, complex queries, often joining multiple tables, sorting large amounts of data, and computing aggregate data.

The benchmark is well known and adopted; its workloads are designed to challenge upward boundaries of system performance (I/O, CPU, and memory utilization). It also provides tools for generating test data and test queries, which makes it easy to use.

Note that this RA does not describe the TPC-DS benchmark; instead, it uses some of the underlying methods/sources to validate the cluster.

6.2 Prepare the Test

In order to run the tests, some preparation is needed:

- Obtain the package.
- Build the tools.
- Generate the data and queries.

The package provides the How_To_Guide.doc document, which contains a detailed description of the procedure.

6.2.1 Obtain and Build the Sources

The sources can be obtained from the official site. From this site, download the TPC-DS_Tools_v2.10.0.zip package. Then build the binaries, after unpacking the archive, according to the guide. In this case, the commands were:

```
cd tools

cp Makefile.suite Makefile

make
```

6.2.2 Generate Test Data

A dsdgen tool can be used for creating test datasets. For example, the following commands generate 100 GB worth of data with nine parallel streams:

```
./dsdgen -scale 100 -dir /tpc/data -parallel
9 -child 1 &

./dsdgen -scale 100 -dir /tpc/data -parallel
9 -child 2 &

./dsdgen -scale 100 -dir /tpc/data -parallel
9 -child 3 &

./dsdgen -scale 100 -dir /tpc/data -parallel
9 -child 4 &

./dsdgen -scale 100 -dir /tpc/data -parallel
9 -child 5 &

./dsdgen -scale 100 -dir /tpc/data -parallel
9 -child 6 &

./dsdgen -scale 100 -dir /tpc/data -parallel
9 -child 7 &

./dsdgen -scale 100 -dir /tpc/data -parallel
9 -child 8 &

./dsdgen -scale 100 -dir /tpc/data -parallel
9 -child 9 &
```

6.2.3 Generate Test Queries

The package includes templates for a variety of queries targeted to different target relational database management systems (RDBMSs). In this case, creating all 99 queries can be achieved with the following:

```
for i in $(seq 1 99)
do
  ./dsqgen -template query$i.tpl \
     -directory ../query_templates \
     -dialect db2 \
     -scale 1 \
     -output_dir /tpc/queries
  mv /tpc/queries/query_0.sql /tpc/queries/
query$i.sql
done
```

6.3 Run the Tests

This RA tests two aspects of the cluster:

- **Data collection** by loading the test data into Db2 Warehouse
- **Data analysis** by running test queries against Db2 Warehouse

For these to run, the RA uses command-line utilities in Db2 Warehouse with the help of a containerized Db2 client.

Get the image from Docker Hub*:

```
docker login -u <<user>> -p <<password>>

docker pull store/ibmcorp/db2wh_ce:v3.2.0-
db2wh_client-linux
```

Run a container with a volume with generated data mounted to it, and point it to the integrated Db2 address:

```
docker run -dit --net=host \
  -v /tpc:/tpc \
  --name=client \
  -e REMOTE_DB=<<IP>>:<<PORT>> \
  store/ibmcorp/db2wh_ce:v3.2.0-db2wh_client-
  linux
```

From there, you can connect to the database from the Docker container.

To enter the container, run:

```
docker exec -it client cli
```

6.3.1 Load Data

First, create the schema:

```
time db2 -tvmf tools/tpcds.sql
time db2 -tvmf tools/tpcds_ri.sql
time db2 -tvmf tools/tpcds_source.sql
```

Then, load the data with the following command:

```
time for fil in `ls | grep ‘\.dat$’` ; do db2
load client from “/tpc/data/${fil}” OF DEL
MODIFIED BY COLDEL”|” insert into ${fil::-8} ;
done
```

6.3.2 Run a Single Stream of Queries

In this test, simply run all the queries (generated earlier), one by one:

```
for i in $(seq 1 99)
do
time=`db2batch -d BLUDB -a user999/
wu75Z0z*E3F_-%wX -f /tpc/queries/query${i}.sql
| grep Elapsed | sed “s/* Elapsed Time is:
//”`
echo $i $time
done
```

6.3.3 Run Queries in Parallel

For this test, run multiple parallel sessions, each running a single query stream.

Putting Bash* code from the test above into the `/tpc/queries/query_runner.sh` file allows for running the test with:

```
for i in {1..7}
do
   (/tpc/queries/query_runner.sh >/tpc/
   queries/tests/test$i.log 2>&1 &)
done
```

7. Summary and Conclusions

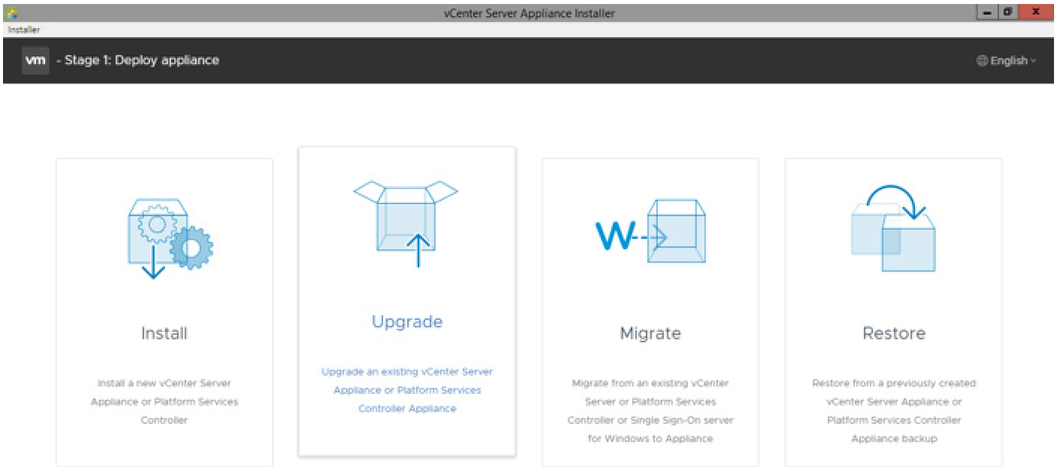
Deploying the Intel Reference Solution for IBM Cloud Private for Data provides an integrated cloud solution in which you can run VMs and containerized applications side by side. This RA provided prescriptive guidance on how to prepare, provision, deploy, and manage an IBM Cloud Private for Data solution on VMware vSAN ReadyNodes that utilizes Intel hardware and technologies with the performance and security necessary to provide an economical, enterprise-grade way forward for digital transformation.

Appendix A

A.1 Configuring VMware vSphere

Begin your Intel Reference Solution for IBM Cloud Private for Data installation by deploying VMware ESXi and connecting it to the management network. Next, install VMware vCenter* using the VMware vCenter Server Appliance* installer media.

1. Download the ISO image containing the VMware vCenter Server 6.7U1* installer media from https://my.vmware.com/web/vmware/details?productId=742&rPId=24636&downloadGroup=VC67U1.
2. Confirm that the md5sum is correct for your downloaded ISO file.
3. Mount the ISO file and run the installer files. There are two installer modes available: Disk Image Mounter (graphical user interface [GUI]) and terminal mode. For ease of use, Disk Image Mounter is recommended.
4. Click **Install**, and then complete the next nine steps. This RA uses the **Embedded Platform Services Controller** option and a **medium VSCA Deployment** size, and it makes use of the ability to bootstrap vSAN on a single host, placing vCenter Server Appliance on the newly created vSAN data store.


vCenter Server Appliance Installer
Installer
vm
- Stage 1: Deploy appliance
English

Install

Install a new vCenter Server Appliance or Platform Services Controller

Upgrade

Upgrade an existing vCenter Server Appliance or Platform Services Controller Appliance

Migrate

Migrate from an existing vCenter Server or Platform Services Controller or Single Sign-On server for Windows to Appliance

Restore

Restore from a previously created vCenter Server Appliance or Platform Services Controller Appliance backup

Figure 5. VMware vCenter Server Appliance* installer GUI

After a successful installation of vCenter, follow these basic steps to complete its deployment:

1. **Configure VMware vSphere Distributed Switch* networking:** In vSphere, there are two methods of connecting VMs to physical networks. The first is based on vSphere standard switches; the second is based on vSphere Distributed Switch. This RA uses vSphere Distributed Switch. This allows VMs to maintain their network settings even if they are migrated to a different host, as the settings of a distributed switch are propagated to all hosts associated with the switch.

   vSphere Distributed Switch separates the data plane and the management plane. Management resides centrally on the vCenter Server, and it lets users administer the networking configuration of all ESXi hosts in a data center. The data plane remains local to every host that is connected to the distributed switch as the host proxy switches. The networking configuration that you create on vCenter Server is automatically pushed down to all host proxy switches.

   vSphere Distributed Switch has a lot of other features that distinguish it from a standard vSwitch. One example is the Network I/O Control feature, which allows users to provide quality of service (QoS) on the various traffic types, like the vSAN traffic.

   vSphere Distributed Switch specifies two abstractions: Uplink Port Groups and Distributed Port Groups. The first one maps the physical NICs of each ESXi host to uplinks on the distributed switch; the second one provides network connectivity to VMs and forwards the VMkernel traffic. You can configure NIC teaming, failover, load balancing, VLAN, security, traffic shaping, and other policies on distributed port groups.

2. **Add and configure VMkernel interfaces for VMware vSphere vMotion* and vSAN networks on each ESXi host**: Each ESXi host needs to configure the VMkernel interfaces for vSphere vMotion and vSAN networks. Both of these networks are separated and reside on different VLANs. vSphere vMotion provides the functionality of moving VMs automatically across the available pool of resources according to a set of rules, if defined by the administrator.

   In vSphere, you can easily create these rules on each ESXi host with the PowerCLI* tool, which is the PowerShell* interface for managing the whole VMware vSphere stack. Assuming you have created a vSphere Distributed Switch named "DSwitch," then you can use the following commands:

   ```
   New-VirtualPortGroup -Name "vSAN" -VirtualSwitch "DSwitch" -VLanId 12

   New-VMHostNetworkAdapter -VMHost 172.17.255.12 -PortGroup "vSAN" -VirtualSwitch "DSwitch" -IP
   192.168.8.12 -SubnetMask 255.255.255.0 -Mtu 9000 -VsanTrafficEnabled $true

   New-VirtualPortGroup -Name "vMotion" -VirtualSwitch "DSwitch" -VLanId 11

   New-VMHostNetworkAdapter -VMHost 172.17.255.12 -PortGroup "vMotion" -VirtualSwitch "DSwitch"
   -IP 192.168.4.12 -SubnetMask 255.255.255.0 -Mtu 9000 -VMotionEnabled $true
   ```

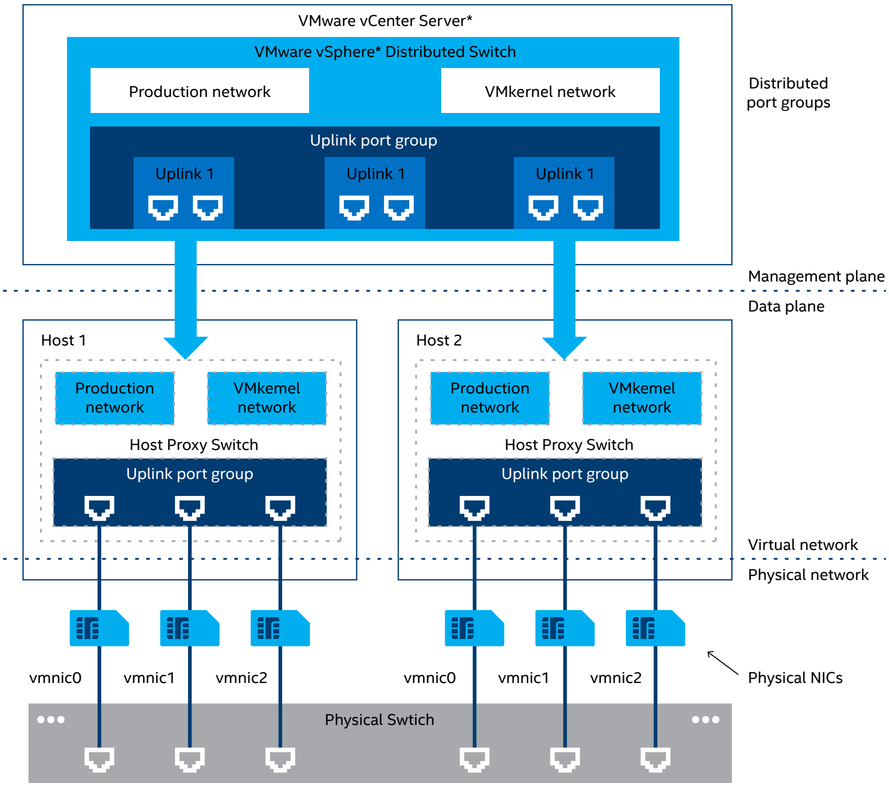


Figure 6. VMware vSphere* distributed switch architecture

3. **Configure a cluster with vSphere High Availability* and VMware vSphere Distributed Resource Scheduler* for better high availability (HA) and scheduling of VMs:** A vSphere Distributed Resource Scheduler cluster is a collection of ESXi hosts and VMs with all shared resources and management interfaces. It offers three cluster-level resource-management capabilities: load balancing, power management, and VM placement. vCenter monitors distribution and usage of CPU and memory resources for all hosts and VMs in the cluster, and then vSphere Distributed Resource Scheduler compares these metrics with the current demand and the imbalance target for the desired resource utilization. vSphere Distributed Resource Scheduler then performs or recommends, depending how is it configured, VM migrations or new VM placement.

   vSphere High Availability monitors the pool of ESXi hosts and VMs, and, in the event of host failure, it migrates VMs to hosts with available capacity using vSphere vMotion. When you add new VMs to a vSphere High Availability cluster, vSphere High Availability checks if there are enough resources to power the VM on in case of host failure.

   This RA enables HA with host monitoring and vSphere Distributed Resource Scheduler with partially automated settings, where only the placement of VMs onto hosts is automated; the migration recommendations need to be manually applied.

4. Configure vSAN and claim disks for its cache and capacity tier.

A.2 VMware vSAN Provisioning and Orchestration

In this setup, vSAN is configured in an all-flash mode, where one flash device is used for cache, while other flash devices are used for the capacity layer. This greatly improves the performance of VMs running on vSAN, in comparison to a hybrid vSAN configuration. In an all-flash configuration, the cache tier is used for the write cache.

Each ESXi host in the cluster has a configured VMkernel interface for vSAN traffic on separate distributed port groups. This RA created vSAN disk groups by claiming each server's disks; in this case, Intel Optane SSD DC P4800X disks are matched as cache tier and Intel SSD DC P4510 disks as capacity tier, and each disk group contains a maximum of one flash cache device and up to seven capacity disks.

Table 12 shows how vSAN is configured for each server in the cluster (SKU plus configuration); there are two disk groups with this configuration.

Table 12. VMware vSAN* configuration for this RA

Disk Group	Storage SKU	Tier	Capacity
Disk Group 1	1 x Intel® Optane™ SSD DC P4800X (2.5-in. PCIe*)	Cache tier	350 GB
	3 x Intel® SSD DC P4510 (2.5-in. PCIe 3.1)	Capacity tier	5.46 TB
Disk Group 2	1 x Intel Optane SSD DC P4800X (2.5-in. PCIe)	Cache tier	350 GB
	3 x Intel SSD DC P4510 (2.5-in. PCIe 3.1)	Capacity tier	5.46 TB

An all-flash configuration allows users to turn on the deduplication and compression modes, which can help improve TCO by reducing the data stored on your physical disks.

Appendix B

Follow these steps to deploy IBM Cloud Private. For detailed instructions, see the IBM documentation at ibm.com/support/knowledgecenter/SSBS6K_3.1.2/installing/install_containers.html.

1. **Prepare a RHEL 7.5 VM template:** You need to prepare RHEL 7.5, or another operating system, from the template available at ibm.com/support/knowledgecenter/en/SSBS6K_3.1.2/supported_system_config/supported_os.html, which will be used as the base of all IBM Cloud Private nodes. In vCenter Server, create a new Linux VM named icp-temp—this name will be used later with the Terraform* deployment. The guest operating system version should be RHEL 7 (64-bit). During customization of the virtual hardware, you only need to specify virtual disk size, as the CPU and memory will be modified for each VM in later steps, so you can leave the default settings. This RA's large environment uses 500 GB size for the template according to sizing provided by ibm.com/support/knowledgecenter/en/SSBS6K_3.1.2/installing/plan_capacity.html. In any case, use thin provisioning so the template does not use unnecessary disk space. This will also speed up creating new VMs from this template. The template VM also needs a single virtual network adapter, which should be connected to the network created for IBM Cloud Private VMs. For usage of persistent volumes on vSphere, configure the diskUUID.enable=TRUE parameter in the advanced configuration parameters in the VM's settings.

Mount the RHEL installation image and start standard installation. Make sure to configure the Logical Volume Manager (LVM) disk sizes according to ibm.com/support/knowledgecenter/en/SSBS6K_3.1.2/supported_system_config/hardware_reqs.html. Configure networking accordingly for your setup; this RA uses a Dynamic Host Configuration Protocol (DHCP) server and set values in `/etc/sysconfig/network-scripts/ifcfg-ens192` to:

```
BOOTPROTO=dhcp

ONBOOT=yes
```

Now, you should configure access to RHEL yum repositories or Red Hat Satellite*, update RHEL to the latest patch level, and install the following tools and packages:

- Python 2.7.x
- PyYAML*
- bind-utils
- nfs-utils
- Perl*
- open-vm-tools (VMware Tools* can be installed optionally via vCenter)
- ntp

IBM Cloud Private requires Docker to be installed on all cluster nodes. Per IBM Cloud Private documentation (ibm.com/support/knowledgecenter/en/SSBS6K_3.1.2/installing/install_docker.html), you can choose to either let the IBM Cloud Private installer set it up or install it manually from the package provided with IBM Cloud Private—IBM Cloud Private 3.1.2 Docker for Linux (x86_64) from IBM Software Access Catalog downloads (ibm.com/partnerworld/program/benefits/software-access-catalog).

This RA pre-installed Docker on the template, using the Docker install binary that comes with IBM Cloud Private. It is the version of Docker that has been tested with the given release of IBM Cloud Private and is supported by IBM—IBM Cloud Private 3.1.2 is matched with Docker 18.03. Download the package to the template VM and run these commands to install:

```
chmod +x icp-docker-18.03.1_x86_64.bin

sudo ./icp-docker-18.03.1_x86_64.bin --install
```

For the pre-install Docker, it must be configured with the devicemapper or overlay2 storage driver. The storage driver for the supplied IBM Cloud Private Docker package is set to loop-lvm by default. For production deployments, you must change to a different storage option: either direct-lvm or overlay2. This RA uses devicemapper in supported direct-lvm mode, so it has provided another disk to Docker. The recommended starting disk size is 100 GB.

Create LVM for Docker on a new disk and configure thin pools:

```
pvcreate /dev/sdb

vgcreate docker /dev/sdb

lvcreate --wipesignatures y -n thinpool docker -l 95%VG

lvcreate --wipesignatures y -n thinpoolmeta docker -l 1%VG

lvconvert -y --zero n -c 512K --thinpool docker/thinpool --poolmetadata docker/thinpoolmeta
```

Set up direct-lvm:

```
mkdir -p /etc/docker

cat >/etc/docker/daemon.json <<EOF

{

 “storage-driver”: “devicemapper”,

 “storage-opts”: [

   “dm.thinpooldev=/dev/mapper/docker-thinpool”,

   “dm.use_deferred_removal=true”,

   “dm.use_deferred_deletion=true”

   ]

 }

 EOF
```

The map_max_count determines the maximum number of memory map areas a process can have. Docker requires that the max_map_count be substantially greater than the default (65530). This RA set this parameter in the template:

```
echo "vm.max_map_count=262144" >> /etc/sysctl.conf
```

Configure password-less SSH for root from boot-master to all nodes, including boot-master (assuming a root install). For non-root install, the user must have "no password" sudo privileges to all commands.

With the configured VM, you can now export to the template in vCenter, for later use by Terraform in the deployment process.

2. **Create an RHEL 7.5 boot node VM:** To simplify installation and configuration of IBM Cloud Private, this RA created a separate VM from the previously created template, which will serve as a boot node.
3. **Copy VMs from the template and configure VMs for IBM Cloud Private:** Use the prepared template VM to create new VMs that you will use as IBM Cloud Private nodes; you will need to configure their vCPU, memory, and disk resources. It is strongly recommended that you use automation tools; at scale, automation will significantly ease and speed up your deployment process.

   To deploy IBM Cloud Private in a large deployment with HA, this RA uses Terraform scripts from the IBM Cloud Architecture GitHub*: github.com/ibm-cloud-architecture/terraform-icp-vmware.

   The RA uses the vSphere provider to provision and configure VMs on the vSphere stack and deploy IBM Cloud Private on top of them. This RA uses this repository only to deploy VMs; after deploying the VMs, continue the installation by using the IBM documentation: ibm.com/support/knowledgecenter/en/SSBS6K_3.1.2/installing/install_containers.html.
4. **Download and load the IBM Cloud Private Docker images on the boot machine:** Download the IBM Cloud Private 3.1.2 for Linux (x86_64) Docker package from IBM Passport Advantage (ibm.com/software/passportadvantage/) onto the boot node and then extract the images and load them into the Docker:

   ```
   tar xf ibm-cloud-private-x86_64-3.1.2.tar.gz -O | sudo docker load
   ```

   Create the installation directory and extract the configuration file from the installer image:

   ```
   sudo mkdir /opt/ibm-cloud-private-3.1.2

   cd /opt/ibm-cloud-private-3.1.2

   sudo docker run -v $(pwd):/data -e LICENSE=accept \ibmcom/icp-inception-amd64:3.1.2-ee \cp -r
   cluster /data
   ```
5. **Prepare the IBM Cloud Private configuration file, *config.yaml:*** The IBM Cloud Private configuration file incorporates all IBM Cloud Private cluster configuration settings, even when nonexpendable for a deployment process. Below are additional parameter settings provided for installation customization. The rest of the parameters and values are left as default; these are collected in the official IBM Cloud Private 3.1.2 documentation: ibm.com/support/knowledgecenter/en/SSBS6K_3.1.2/installing/install_containers.html.

   Network management is set as Calico* value: *network_type: calico*

   The IPv4 network consistent with Classless Inter-Domain Routing (CIDR) format is set to the entire network using *network_cidr: 10.1.0.0/16*

   The Kubernetes service cluster IP range, which is a virtual network also in CIDR format, allocates a block of reserved IPs for cluster services: s*ervice_cluster_ip_range: 10.0.0.0/16*

   Kubelet requires additional argument configuration for disabling swap. It is used to pack instances as tightly as possible in the utilized environment. Moreover, the scheduler shouldn't use swap until cluster performance decreases: *kubelet_extra_args: ["--fail-swap-on=false"]*

   The following etcd parameters can disable additional wait duration before closing a nonresponsive connection, disable the frequency duration to ping if a server-to-client connection is still alive, and determine the number of committed transactions for which to do a disk snapshot: *etcd_extra_args: ["--grpc-keepalive-timeout=0", "--grpc-keepalive-interval=0", "--snapshot-count=10000"]*

   The following commands set cluster-administrator privilege (name and password):

   ```
   default_admin_user: admin

   default_admin_password: admin
   ```

   Choose the service that will be managing the virtual IP (VIP) in the case of an HA environment on any combination of master and proxy nodes: *vip_manager: keepalived*

   Two of the master-node HA settings for setting the correct interface and virtual IP address in the IBM Cloud Private HA cluster are *vip_iface: ens192* and *cluster_vip: 172.17.255.240*

Two of the proxy-node HA settings for setting the virtual IP interface and address for a proxy node in the IBM Cloud Private HA environment are *proxy_vip_iface: ens192* and *proxy_vip: 172.17.255.241*

Cloud-provider settings and configuration files for VMware Cloud Providers* were chosen because of compatibility with VMware components (vCenter especially), in addition to their interface format, which is POSIX compliant. It is worth remembering the prerequisites before setting any storage options; these are described in detail for VMware Cloud Providers in the official documentation for IBM Cloud Private v3.1.2: ibm.com/support/knowledgecenter/en/SSBS6K_3.1.2/manage_cluster/vsphere_prereq.html. The user that is indicated in the vSphere cloud configuration file (user: “administrator@vsphere.local”) requires privileges to interact with vCenter:

```
kubelet_nodename: hostname
cloud_provider: vsphere

vsphere_conf:
  user: “administrator@vsphere.local”
  password: <type-your-password-here>
  server: vcenter.icp.igk
  port: 443
  insecure_flag: 1
  datacenter: ICP-dc
  datastore: vsanDatastore
  working_dir: ICP-dc
monitoring:
 mode: managed
 prometheus:
   scrapeInterval: 1m
   evaluationInterval: 1m
   retention: 24h
   persistentVolume:
     enabled: false
     storageClass: “-”
   resources:
     limits:
       cpu: 500m
       memory: 8192Mi
 alertmanager:
   persistentVolume:
     enabled: false
     storageClass: “-”
   resources:
     limits:
       cpu: 200m
       memory: 256Mi
 grafana:
   user: “admin”
```

```
    password: “admin”
    persistentVolume:
      enabled: false
      storageClass: “-”
    resources:
      limits:
        cpu: 500m
        memory: 2048Mi
```

6. **Prepare /etc/hosts, sharing it between all IBM Cloud Private nodes, and prepare the Ansible host file:** For /etc/hosts, add the IP addresses and host names for all nodes in the cluster, commenting out the lines that begins with 127.0.1.1 and ::localhost.

```
<master_nodes_IP_address> <master_nodes_host_name> <worker_nodes_1_IP_address> <worker_
nodes_1_host_name> <proxy_nodes_IP_address> <proxy_nodes_host_name>
<management_nodes_IP_address> <management_nodes_host_name>
<va_nodes_IP_address> <va_nodes_host_name>
```

   This file needs to be shared across all IBM Cloud Private nodes and updated if you add more nodes in the future.

   The Ansible hosts were prepared according to IBM Cloud Private documentation (ibm.com/support/knowledgecenter/en/SSBS6K_3.1.2/installing/hosts.html) and copied to the cluster folder within the boot-node VM. It is strongly recommend that you use automation tools in order to ease the deployment process and allow for faster preparation to scale out your IBM Cloud Private cluster.

7. **Start the IBM Cloud Private Ansible installer in the Docker container on the boot node:** Before starting the installation process, it is important to share SSH keys among cluster nodes. Per IBM Cloud Private documentation (ibm.com/support/knowledgecenter/SSBS6K_3.1.2/installing/ssh_keys.html), you must generate the SSH key on a boot node and share the public key among all IBM Cloud Private nodes:

```
ssh-copy-id -i ~/.ssh/id_rsa.pub <user>@<node_ip_address>
```

   Copy the private key to the cluster folder:

```
sudo cp ~/.ssh/id_rsa ./cluster/ssh_key
```

   By default, the command to deploy your environment is set to deploy 15 nodes at a time. If your cluster has more than 15 nodes, the deployment might take longer to finish. If you want to speed up your deployment, you can specify a higher number of nodes to be deployed at a time. To do so, use the argument *-f <number of nodes to deploy>* with the command:

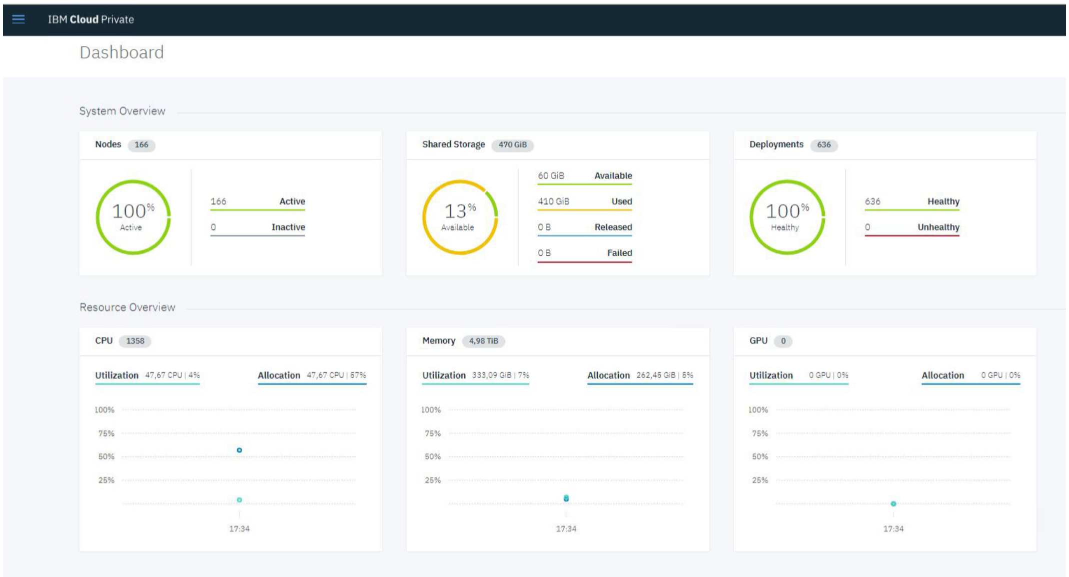
```
sudo docker run --net=host -t -e LICENSE=accept \
-v “$(pwd)”:/installer/cluster ibmcom/icp-inception-amd64:3.1.2-ee install –vv –f 23
```

8. **Run a health check after the installation to ensure all services are running properly:** After a successful deployment, the access information for your cluster will be displayed:

```
UI URL is https://<ip_address>:8443 , default username/password is admin/admin
```

   You should also run a health check to ensure that all services are up and running:

```
sudo docker run --net=host -t -e LICENSE=accept -v “$(pwd)”:/installer/cluster ibmcom/icp-
inception-amd64:3.1.2-ee healthcheck –vv –f 23
```



IBM Cloud Private

Dashboard

System Overview

Nodes 166
100% Active
166 Active
0 Inactive

Shared Storage 470 GiB
13% Available
60 GiB Available
410 GiB Used
0 B Released
0 B Failed

Deployments 636
100% Healthy
636 Healthy
0 Unhealthy

Resource Overview

CPU 1358
Utilization 47,67 CPU | 4%
Allocation 47,67 CPU | 57%

Memory 4,98 TiB
Utilization 333,09 GiB | 7%
Allocation 262,45 GiB | 5%

GPU 0
Utilization 0 GPU | 0%
Allocation 0 GPU | 0%

Figure 7. IBM Cloud Private* dashboard after a successful deployment and health check

B.1 Networking Details

Use the following section to configure the network environment for your Intel Reference Solution for IBM Cloud Private for Data deployment. The tables under Network Architecture present a sample deployment using two NICs on each server (Tables 13 and 14).

B.1.1 Network Architecture

Table 13. Intel® Reference Solution for IBM Cloud Private for Data* recommended network addresses

VLANs to Configure on Redundant Switches	Server Ports	Sample VLAN Sample IP Range	CIDR	VMware vSphere* Distributed Switch
IBM Cloud Private*	NIC1, NIC2	Untagged	100.65.80.0/22	DPortGroup-ICP
VMware ESXi* (management)	NIC1, NIC2	Untagged	100.65.80.0/22	DPortGroup-MGMT
VMware vSAN*	NIC1, NIC2	11	192.168.8.0/24	DPortGroup-vSAN
VMware vSphere vMotion*	NIC1, NIC2	12	192.168.12.0/24	DPortGroup-vMotion

Table 14. Details for all networks on top of VMware vSphere* using VMware vSphere Distributed Switch* and four separate distributed port groups

VLANs to Configure on Redundant Switches	Server Ports	Sample VLAN	Sample IP Range	VMware vSphere* Distributed Switch
IBM Cloud Private* (VMs)	NIC 1, NIC2	Untagged	172.17.255.0/24	DPort Group: ICP
VMware ESXi* (management)	NIC 1, NIC2	Untagged	172.17.255.0/24	DPort Group: MGMT
VMware vSAN*	NIC 1, NIC2	11	192.168.8.0/24	DPort Group: vSAN
VMware vSphere vMotion*	NIC 1, NIC2	12	192.168.12.0/24	DPort Group: vMotion

Additional Networks			
Hardware network for Intelligent Platform Management Interface (IPMI) management	192.168.24.0/24		
IBM Cloud Private Calico* networks on top of the IBM Cloud Private (VMs) network (overlay): • IBM Cloud Private Calico network for pods • IBM Cloud Private Calico network for services	10.1.0.0/16 10.0.0.0/16		

B.1.2 Management Network

The management network uses a separate network and cards because it has no dependencies on the other networks in the Intel Reference Solution for IBM Cloud Private for Data. You can use your organizations' own best practices and processes in provisioning it.

B.2 Monitoring, Logging, and Metering in IBM Cloud Private

The IBM Cloud Private software platform provides all necessary tools for monitoring, logging, metering, and alerting. The platform makes it easy to manage processes without specialized expertise.

B.2.1 Logging

The ELK* stack in IBM Cloud Private combines Elasticsearch*, Logstash*, and Kibana* into one package to use with IBM Cloud Private for logging Kubernetes and Docker events. You can deploy ELK from the IBM Cloud Private catalog to one of the namespaces in the cluster in order for it to log messages from Docker applications.

B.2.2 Monitoring and Alerting

IBM Cloud Private uses Prometheus* and Grafana* to gather and visualize data from the cluster and applications. Prometheus or add-on tools can be used to provide alerts, with or without notifications.

The Grafana dashboard, which is part of the cluster monitoring dashboard provided by IBM Cloud Private, offers 14 different types of dashboard templates to monitor the status of applications deployed through the cluster. Examples include:

- **Kubernetes Pod Overview,** which contains pod-specific metrics, such as CPU, memory, and network status, in addition to how many restarts a single pod has had.

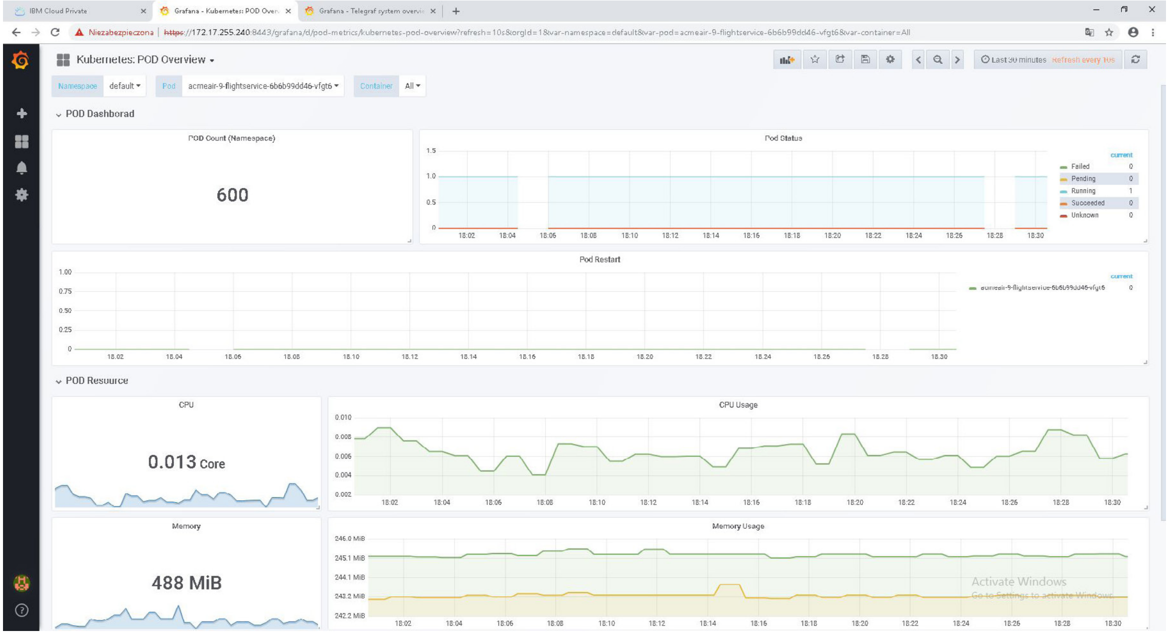


Figure 8. IBM Cloud Private* Grafana* dashboard—Kubernetes* POD overview

- **Kubernetes Cluster Monitoring**, which provides information collected by Prometheus about the Kubernetes cluster and includes CPU, memory, and file system usage. Moreover, the dashboard shows essential information about components, such as individual pods, containers, and system services.

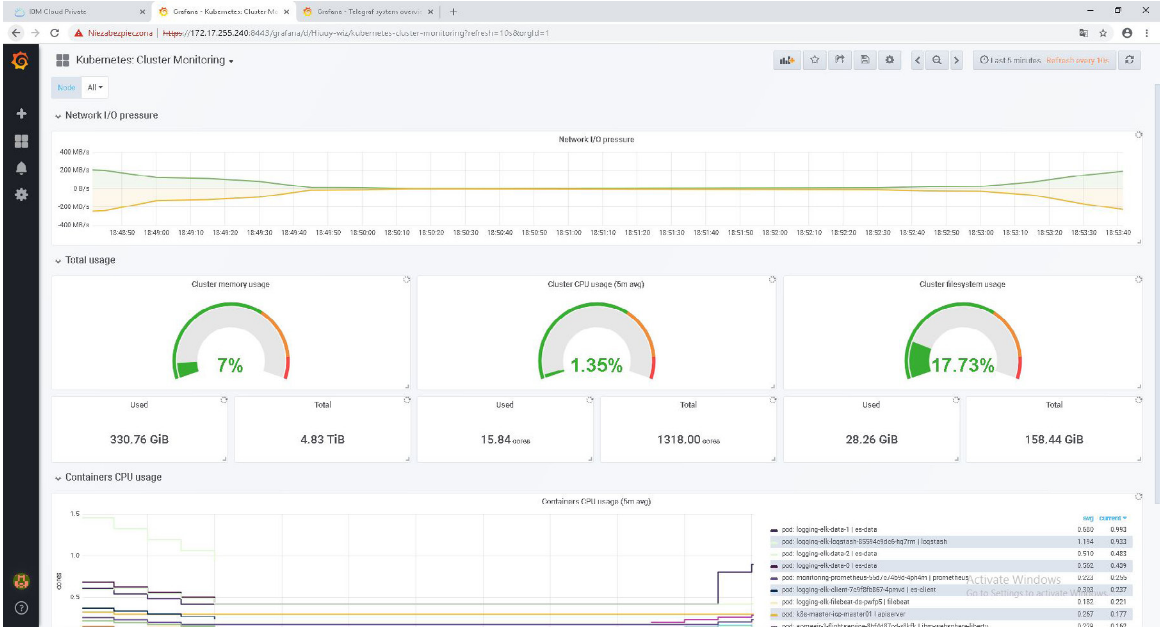
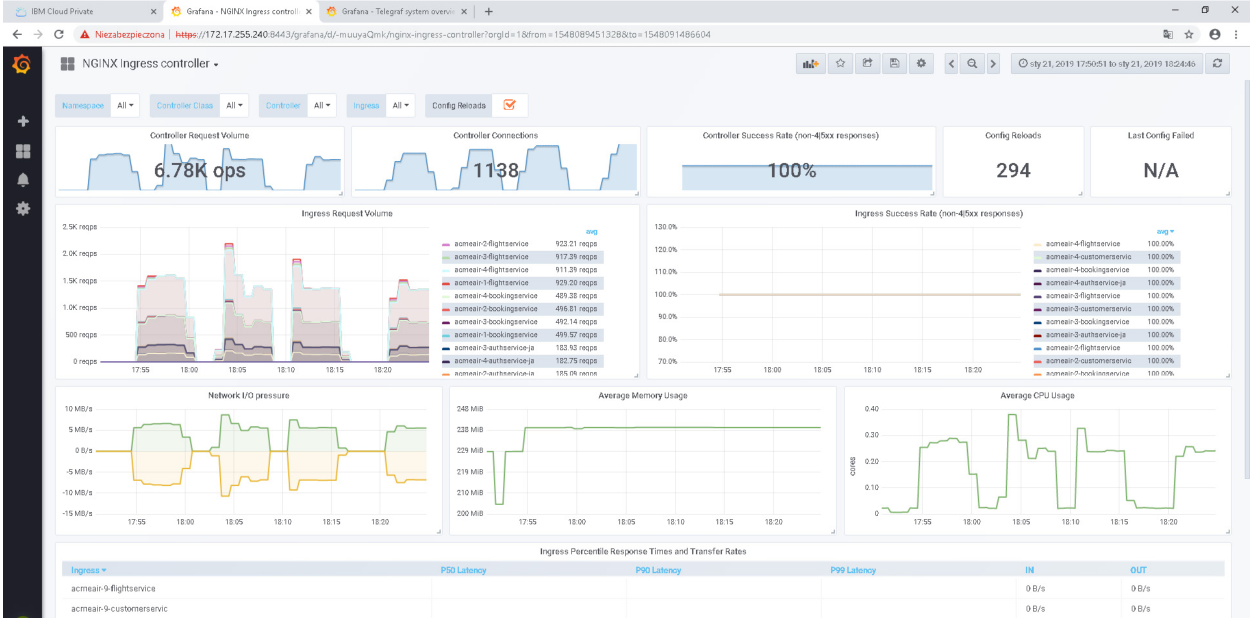

Kubernetes: Cluster Monitoring
Node All
Last 5 minutes Refresh every 10s
Network I/O pressure
400 MB/s
200 MB/s
0 B/s
-200 MB/s
-400 MB/s
Total usage
Cluster memory usage
7%
Cluster CPU usage (5m avg)
1.35%
Cluster filesystem usage
17.73%
Used
330.76 GiB
Total
4.83 TiB
Used
15.84 cores
Total
1318.00 cores
Used
28.26 GiB
Total
158.44 GiB
Containers CPU usage
Containers CPU usage (5m avg)
cores
avg current
pod: logging-elk-data-1 | es-data 0.680 0.993
pod: logging-elk-logstash-85594c9dc6-hq7rm | logstash 1.194 0.933
pod: logging-elk-data-2 | es-data 0.510 0.483
pod: logging-elk-data-0 | es-data 0.502 0.439
pod: monitoring-prometheus-55d7c74b9d-4ph4m | prometheus 0.223 0.255
pod: logging-elk-client-7c9f8fb867-4pmvd | es-client 0.303 0.237
pod: logging-elk-filebeat-ds-pwfp5 | filebeat 0.182 0.221
pod: k8s-master-icp-master01 | apiserver 0.267 0.177
Activate Windows
Go to Settings to activate Windows.

Figure 9. IBM Cloud Private* Grafana* dashboard—Kubernetes* cluster monitoring

- **NGINX Ingress Controller***, which includes an array of metrics about the ingress controller, including requests, volume, and network I/O pressure.


IBM Cloud Private
Grafana - NGINX Ingress controll
Grafana - Telegraf system overvi
Niezabezpieczona | https://172.17.255.240:8443/grafana/d/-muuyaQmk/nginx-ingress-controller?orgId=1&from=1548089451328&to=1548091486604
NGINX Ingress controller
sty 21, 2019 17:50:51 to sty 21, 2019 18:24:46
Namespace All
Controller Class All
Controller All
Ingress All
Config Reloads
Controller Request Volume
6.78K ops
Controller Connections
1138
Controller Success Rate (non-4|5xx responses)
100%
Config Reloads
294
Last Config Failed
N/A
Ingress Request Volume
2.5K reqps
2.0K reqps
1.5K reqps
1.0K reqps
500 reqps
0 reqps
17:55
18:00
18:05
18:10
18:15
18:20
avg
acmeair-2-flightservice 923.21 reqps
acmeair-3-flightservice 917.39 reqps
acmeair-4-flightservice 911.39 reqps
acmeair-1-flightservice 929.20 reqps
acmeair-4-bookingservice 489.38 reqps
acmeair-2-bookingservice 496.81 reqps
acmeair-3-bookingservice 492.14 reqps
acmeair-1-bookingservice 499.57 reqps
acmeair-3-authservice-ja 183.93 reqps
acmeair-4-authservice-ja 182.75 reqps
acmeair-2-authservice-ja 185.09 reqps
Ingress Success Rate (non-4|5xx responses)
130.0%
120.0%
110.0%
100.0%
90.0%
80.0%
70.0%
17:55
18:00
18:05
18:10
18:15
18:20
avg
acmeair-4-flightservice 100.00%
acmeair-4-customerservic 100.00%
acmeair-4-bookingservice 100.00%
acmeair-4-authservice-ja 100.00%
acmeair-3-flightservice 100.00%
acmeair-3-customerservic 100.00%
acmeair-3-bookingservice 100.00%
acmeair-3-authservice-ja 100.00%
acmeair-2-flightservice 100.00%
acmeair-2-customerservic 100.00%
acmeair-2-bookingservice 100.00%
Network I/O pressure
10 MB/s
5 MB/s
0 B/s
-5 MB/s
-10 MB/s
-15 MB/s
17:55
18:00
18:05
18:10
18:15
18:20
Average Memory Usage
248 MiB
238 MiB
229 MiB
219 MiB
210 MiB
200 MiB
17:55
18:00
18:05
18:10
18:15
18:20
Average CPU Usage
0.40
0.30
0.20
0.10
0
cores
17:55
18:00
18:05
18:10
18:15
18:20

Ingress Percentile Response Times and Transfer Rates

Ingress	P50 Latency	P90 Latency	P99 Latency	IN	OUT
acmeair-9-flightservice				0 B/s	0 B/s
acmeair-9-customerservic				0 B/s	0 B/s

Figure 10. IBM Cloud Private* Grafana* dashboard—NGINX Ingress Controller*

B.2.3 Metering

Metering is provided by IBM Cloud Private for tracking application and cluster metrics.

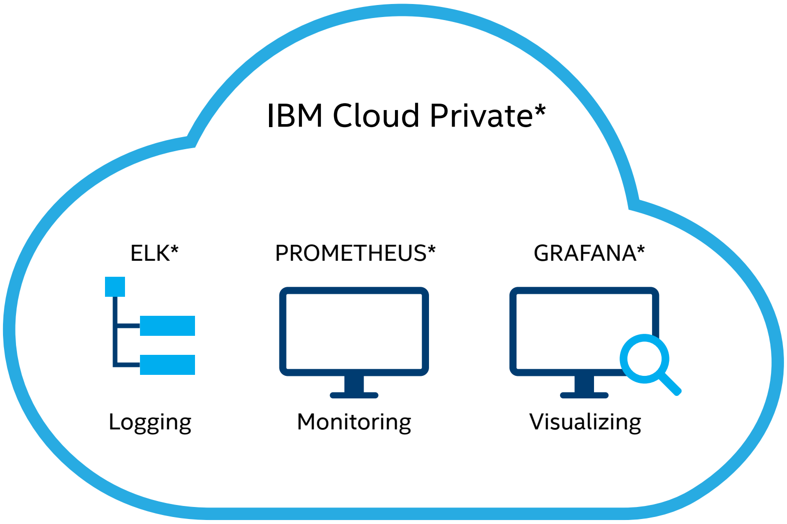


Figure 11. Logging and monitoring tools in IBM Cloud Private*

B.3 Network Configuration

The default network for IBM Cloud Private pods is provided by Calico, which is configured automatically during the installation process of IBM Cloud Private on top of the network configured for VMs in vSphere.

Calico is a new approach to virtual networking and network security for containers, VMs, and bare-metal servers. It provides a rich set of security-enforcement capabilities that run on a highly scalable and efficient virtual network. The Calico node runs in a Docker container on the Kubernetes master node and on each Kubernetes worker node in the cluster. The calico-cni plugin integrates directly with the Kubernetes kubelet process on each node to discover which pods are created and add them to Calico networking.

For this RA's setup, Calico is configured with IP-in-IP encapsulation. You must set this parameter even if all the nodes belong to the same subnet. This configuration enables encapsulation of pod-to-pod traffic over the underlying network infrastructure.

The calico_tunnel_mtu parameter must be set based on the maximum transmission unit (MTU) of the interface that is configured for use by Calico.

If the IP-in-IP encapsulation parameter is set to true, 20 bytes are used for the IP-IP tunnel header. You must set the calico_ tunnel_mtu parameter to be at least 20 bytes less than the actual MTU of the interface.

IBM Cloud Private Calico networks on top of the IBM Cloud Private VMs network overlay:

- IBM Cloud Private Calico network for pods, 10.1.0.0/16—The IPv4 network to use for the entire pod network.
- IBM Cloud Private Calico network for services, 10.0.0.1/24—The Kubernetes service cluster IP range. This configuration allocates a block of IPs for services. These service IPs do not need to be routable, since kube-proxy converts service IPs to pod IPs before traffic leaves the node.

The IBM Cloud Private network for pods' underlying network for VMs and the IBM Cloud Private service cluster IP range must not be in conflict with each other.

Five system-configuration parameters on IBM Cloud Private nodes and the ingress controller must be set to support a large number of open files and TCP connections with large bursts of messages. Changes can be made using the /etc/rc.d/rc.local or /etc/sysctl.conf script to preserve changes after reboot.

1. **/proc/sys/fs/file-max:** The maximum number of concurrently open files.
2. **/proc/sys/net/ipv4/tcp_max_syn_backlog:** The maximum number of remembered connection requests, which did not receive an acknowledgment from the connecting client. The default value is 1,024 MB for systems with more than 128 MB of memory, and 128 MB for low-memory machines.
3. **/proc/sys/net/core/somaxconn:** Limit of socket listen() backlog, known in user space as SOMAXCONN. Defaults to 128.
4. **/proc/sys/net/ipv4/ip_local_port_range:** Increase system IP port limits to allow for more connections.
5. net.ipv4.ip_local_port_range = 1024 65535

Large numbers of workloads running concurrently can create problems on Linux systems—particularly for proxy nodes—concerning running out of space in the conntrack table, which can cause poor iptables performance. To avoid these issues, increase the maximum number of tracked connections on the proxy nodes:

```
echo 524288 > /proc/sys/net/netfilter/nf_conntrack_max
```

[1] Intel. "Intel® Ethernet Converged Network Adapters X710 DA-2/DA-4." March 2018. intel.com/content/www/us/en/ethernet-products/converged-network-adapters/ethernet-x710-brief.html.

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark* and MobileMark*, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit intel.com/benchmarks.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at intel.com.

Cost reduction scenarios described are intended as examples of how a given Intel- based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.

Optimization Notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice. Notice Revision #20110804

Printed in USA 0619/RM/PRW/PDF Please Recycle 340109-001US