Hitachi Unified Compute Platform for GKE Enterprise

Enabling a Google Distributed Cloud Solution at the Datacenter Edge

Reference Architecture Guide

MK-SL-266-03
December 2023

© 2023 Hitachi Vantara LLC. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including copying and recording, or stored in a database or retrieval system for commercial purposes without the express written permission of Hitachi, Ltd., or Hitachi Vantara LLC (collectively “Hitachi”). Licensee may make copies of the Materials provided that any such copy is: (i) created as an essential step in utilization of the Software as licensed and is used in no other manner; or (ii) used for archival purposes. Licensee may not make any other copies of the Materials. “Materials” mean text, data, photographs, graphics, audio, video and documents.

Hitachi reserves the right to make changes to this Material at any time without notice and assumes no responsibility for its use. The Materials contain the most current information available at the time of publication.

Some of the features described in the Materials might not be currently available. Refer to the most recent product announcement for information about feature and product availability, or contact Hitachi Vantara LLC at https://support.hitachivantara.com/en_us/contact-us.html.

Notice: Hitachi products and services can be ordered only under the terms and conditions of the applicable Hitachi agreements. The use of Hitachi products is governed by the terms of your agreements with Hitachi Vantara LLC.

By using this software, you agree that you are responsible for:

1. Acquiring the relevant consents as may be required under local privacy laws or otherwise from authorized employees and other individuals; and
2. Verifying that your data continues to be held, retrieved, deleted, or otherwise processed in accordance with relevant laws.

Notice on Export Controls. The technical data and technology inherent in this Document may be subject to U.S. export control laws, including the U.S. Export Administration Act and its associated regulations, and may be subject to export or import regulations in other countries. Reader agrees to comply strictly with all such regulations and acknowledges that Reader has the responsibility to obtain licenses to export, re-export, or import the Document and any Compliant Products.

Hitachi and Lumada are trademarks or registered trademarks of Hitachi, Ltd., in the United States and other countries.

AIX, AS/400e, DB2, Domino, DS6000, DS8000, Enterprise Storage Server, eServer, FICON, FlashCopy, GDPS, HyperSwap, IBM, Lotus, MVS, OS/390, PowerHA, PowerPC, RS/6000, S/390, System z9, System z10, Tivoli, z/OS, z9, z10, z13, z14, z15, z16, z/VM, and z/VSE are registered trademarks or trademarks of International Business Machines Corporation.

Active Directory, ActiveX, Bing, Excel, Hyper-V, Internet Explorer, the Internet Explorer logo, Microsoft, Microsoft Edge, the Microsoft corporate logo, the Microsoft Edge logo, MS-DOS, Outlook, PowerPoint, SharePoint, Silverlight, SmartScreen, SQL Server, Visual Basic, Visual C++, Visual Studio, Windows, the Windows logo, Windows Azure, Windows PowerShell, Windows Server, the Windows start button, and Windows Vista are registered trademarks or trademarks of Microsoft Corporation. Microsoft product screen shots are reprinted with permission from Microsoft Corporation.

All other trademarks, service marks, and company names in this document or website are properties of their respective owners.

Copyright and license information for third-party and open source software used in Hitachi Vantara products can be found in the product documentation, at https://www.hitachivantara.com/en-us/company/legal.html or https://knowledge.hitachivantara.com/Documents/Open_Source_Software.

Feedback

Hitachi Vantara welcomes your feedback. Please share your thoughts by sending an email message to SolutionLab@HitachiVantara.com. To assist the routing of this message, use the paper number in the subject and the title of this white paper in the text.

Revision history

Changes	Date
Updated for Google Distributed Cloud Virtual and GKE Enterprise	December 7, 2023
Updated cluster worker node to data-plane node and made other edits.	September 12, 2023
Support for bare metal.	June 2, 2023

Reference Architecture Guide

Executive overview

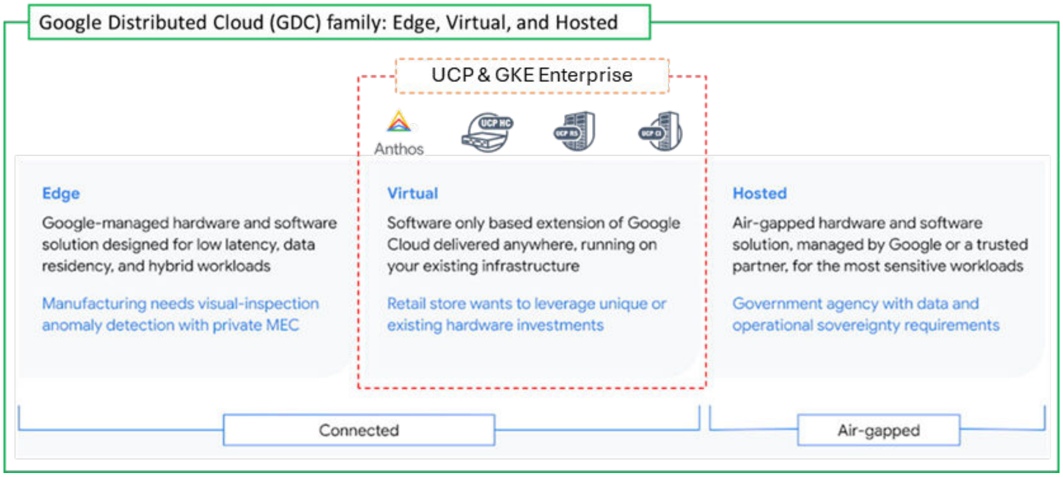
Hitachi Vantara has successfully integrated GKE Enterprise, formerly Anthos, the cloud-native container platform developed by Google Cloud, with the innovative Hitachi Unified Compute Platform (UCP) to achieve GKE Enterprise Ready status. Hitachi UCP offers enterprise-class converged, hyper-converged infrastructure, and rack-scale systems that deliver agility, scalability, reliability, resilience, and high-performance to meet the dynamic needs of modern businesses.

GKE Enterprise (Anthos) is a robust cloud-native container platform built on open-source technologies, such as Kubernetes, Istio, and Knative, providing secure, scalable, and consistent application development and deployment across on-premises and cloud environments. GKE Enterprise enables businesses to accelerate application development across hybrid edge-core-cloud environments, promoting flexibility and agility to stay ahead in a competitive market.

GKE Enterprise for on-premises is delivered as part of Google's Distributed Cloud (GDC) family, a portfolio of hardware, software, and services that brings Google infrastructure services to the edge and into your data centers. This provides a uniform set of experiences for development, security, and management across any IT environment backed by a common set of Anthos APIs. With this GKE Enterprise Ready status, customers can now select Hitachi UCP when choosing to use GDC Virtual approach to deploy and manage GKE on VMware and GKE on Bare Metal services.

By validating GKE Enterprise on UCP, Hitachi Vantara provides customers with a powerful portfolio of solutions that combines the benefits of the GKE Enterprise cloud-native container platform with the uncompromised performance and reliability of UCP. GKE Enterprise on UCP can leverage either hyperconverged or converged configurations with external storage, complete with our rich set of data services for container storage. This integration has been successfully tested, validated, and now officially listed as an GKE Enterprise Ready platform, helping businesses to modernize their IT infrastructure, develop new applications, and meet the evolving demands of the market with ease. For more information see https://cloud.google.com/anthos/docs/resources/partner-platforms#hitachi.

This document details how to deploy and manage a hybrid multi-cloud environment using the Hitachi UCP platform with GKE Enterprise on-prem from Google (GKE on GDC Virtual on-prem).



Google-managed hardware and software solution designed for low latency, data residency, and hybrid workloads

Manufacturing needs visual-inspection anomaly detection with private MEC

Software only based extension of Google Cloud delivered anywhere, running on your existing infrastructure

Retail store wants to leverage unique or existing hardware investments

Air-gapped hardware and software solution, managed by Google or a trusted partner, for the most sensitive workloads

Government agency with data and operational sovereignty requirements

For more information see https://cloud.google.com/blog/topics/anthos/anthos-on-prem-and-bare-metal-are-now-gdc-virtual.

Overview

This reference architecture serves as a proof point that Hitachi Unified Compute platform has been tested and validated as an GKE Enterprise Ready platform that supports the latest version. It also provides the foundation for GKE on VMware (previously named Anthos on-prem/Anthos clusters on VMware) and GKE on Bare Metal (previously named Anthos clusters on bare metal), components of Google Distributed Cloud Virtual (GDC Virtual). This document describes how to deploy and manage a hybrid multi-cloud environment using Google's GKE Enterprise and Hitachi UCP platform.

This paper covers the functional aspects of GKE Enterprise core components and provides an architecture overview and implementation of GKE Enterprise on top of the Hitachi UCP environment. In addition, it provides an example deployment of a stateful application with persistent volumes on Hitachi Virtual Storage Platform (VSP) and VMware vSAN, as well as the integration with third-party Kubernetes clusters deployed on top of UCP, all managed with Google Cloud console.

A key element in the successful deployment of a container platform is having a robust and flexible infrastructure that can meet a wide variety of requirements in a highly dynamic environment. Hitachi UCP provides converged solutions with VSP and hyper-converged solutions certified for different ecosystem stacks with a variety of configurable options that can meet any application workload and business needs. Hitachi infrastructure together with GKE Enterprise capabilities provides a highly available, high-performance infrastructure, scalable, centralized management, hybrid, and multi-cluster management for containerized workloads.

The intended audience of this document is IT administrators, system architects, consultants, and sales engineers to assist in planning, designing, and implementing the UCP product portfolio with Google GKE Enterprise solutions – Unified Compute Platform for GKE Enterprise.

Solution components

This section outlines the components used in this reference architecture.

Hitachi Unified Compute Platform deployment options

The following Hitachi Unified Compute Platform deployment options are used in this solution.

Hitachi Unified Compute Platform CI

Hitachi Unified Compute Platform CI (UCP CI) is an optimized, preconfigured, and pretested converged infrastructure appliance for both VMware vSphere and bare metal. It offers a broad range of compute and storage components that can be scaled and configured independently to eliminate overprovisioning. You have a choice of operating environments to maximize your flexibility.

With Unified Compute Platform CI, you can choose between single-rack configurations and multi-rack configurations. More details about UCP CI can be found at https://www.hitachivantara.com/en-us/products/integrated-systems/converged-infrastructure.html.

Hitachi Unified Compute Platform RS

To simplify your hybrid cloud journey, Hitachi Unified Compute Platform RS (UCP RS) provides a turnkey solution that reduces total cost of ownership (TCO) and improves security. The software-defined data center solution accelerates the time to market with a natively integrated cloud infrastructure stack. It comes prepackaged with management software, to provide automated, policy-based IT operations.

Unified Compute Platform RS has automation that enables the deployment of an entire cloud infrastructure in hours, not weeks or months. There is rapid and repeatable application deployment.

Move your workload across data centers to meet changing business needs. Manage your applications across private and public cloud from a common toolset. Scale your data enter without increasing IT headcount. Automate your data center with policies.

More details about UCP RS can be found at https://www.hitachivantara.com/en-us/products/integrated-systems/cloud-foundation.html.

Hitachi Unified Compute Platform HC

Hitachi Unified Compute Platform HC (UCP HC) is an integrated turnkey appliance that combines compute, storage, and virtualization to deliver certainty for edge to core to cloud operations.

This market-proven Hitachi solution provides a scalable, seamless, and simplified cloud foundation for enterprise and mid-market customers. Advanced automation and intelligence for day 0-2 operations accelerate innovation and improve productivity while lowering the TCO.

More details about UCP HC can be found at https://www.hitachivantara.com/en-us/products/integrated-systems/hyper-converged-infrastructure.html.

Hitachi UCP hardware components

The following tables list the versions of hardware and software tested in this reference architecture.

Additional compute options such as HA8xx G3 are also available but for most recent compatibility information, see Hitachi Vantara Support UCP Product Compatibility at https://compatibility.hitachivantara.com/ and https://compatibility.hitachivantara.com/assets/vmware.

Hardware	Description	Version	Quantity
Hitachi Advanced Server HA810 G2 (for VMware compute cluster)	▪ 2 × Intel Xeon Gold 6338 CPU @ 2.00GHz processors ▪ 8 × 32 GB DIMM, 256 GB memory ▪ NS204i-r NVMe OS Boot Device (Two 480 GB M.2) ▪ Emulex LPe36000 Fibre Channel Adapter ▪ 1 × Intel(R) Ethernet Controller E810-XXV for SFP NIC dual-port ▪ For vSAN configuration: • 1 × SAS SSD 1.92TB (cache) • 3 × SAS SSD 1.92TB (capacity)	iLO 5: 2.65 System ROM: U46 v1.58	3
Hitachi Advanced Server HA810 G2 (for Anthos bare metal)	▪ 2 × Intel Xeon Gold 6338 CPU @ 2.00GHz processors ▪ 8 × 32 GB DIMM, 256 GB memory ▪ NS204i-r NVMe OS Boot Device (Two 480 GB M.2)	iLO 5: 2.65 System ROM: U46 v1.58	5

Hardware	Description	Version	Quantity
	▪ Emulex LPe36000 Fibre Channel Adapter ▪ 1 × Intel(R) Ethernet Controller E810-XXV for SFP NIC dual-port		
Hitachi Virtual Storage Platform E1090	▪ 1 TB cache ▪ 8 × 15 TB NVMe drives ▪ 4 × 32 Gbps Fibre Channel ports	93-06-42-80/00	1
Cisco Nexus 9332C switch (spine)	▪ 32-port 40/100 GbE ▪ 2-port 1/10 GbE	NXOS 9.3.5	2
Cisco Nexus 93180YC-FX switch (leaf)	▪ 48-port 10/25 GbE ▪ 6-port 40/100 GbE	NXOS 9.3.5	2
Cisco Nexus 92348	▪ 48-port 1 GbE ▪ 4-port 1/10/25 GbE ▪ 2-port 40/100 GbE	NXOS 9.3.5	1
Brocade G620	▪ 48-port 16/32 Gbps Fibre Channel switch	9.0.0b	2

Software components

The following table lists the key software components for GKE on VMware.

Software	Version
Hitachi Storage Virtualization Operating System RF	90-05-02-00/01 83-05-33-40/00
VMware vSphere	7.0 Update 3 or newer
VMware vSAN	7.0 Update 3 or newer
Anthos on VMware	1.14.1-gke.39
Kubernetes	1.25.5-gke.100
F5 Big-IP Virtual Edition	17.0.0.1

The following table lists the key software components tested with GKE on Bare Metal.

Software	Version
Hitachi Storage Virtualization Operating System RF	90-05-02-00/01 83-05-33-40/00
Red Hat Enterprise Linux	8.4
Ubuntu	20.04 LTS
Anthos	1.14.3
Kubernetes	1.25.6-gke.1000
Hitachi Storage Plug-in for Containers	3.11

Google Cloud GKE Enterprise

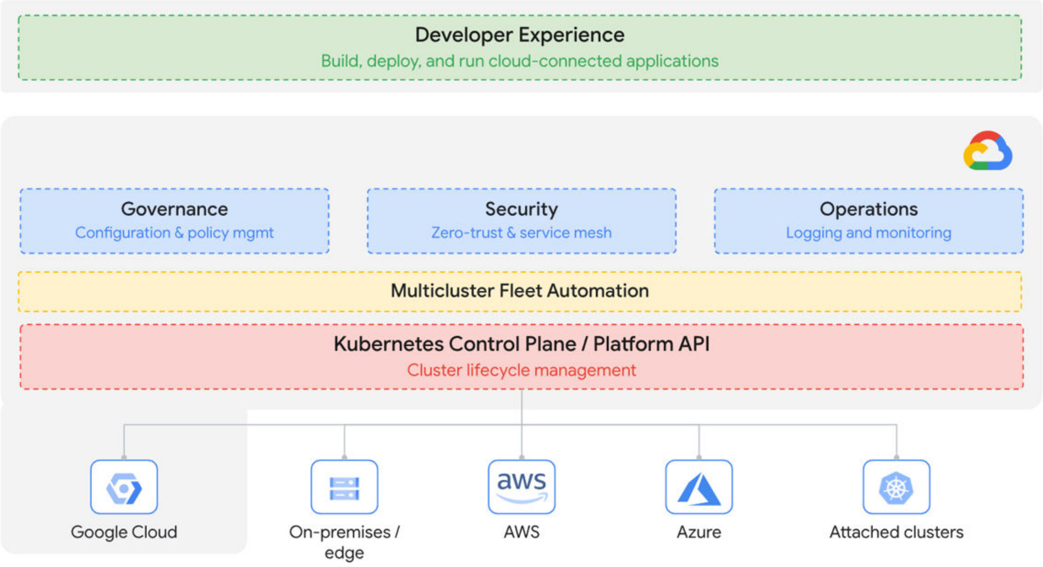
GKE Enterprise from Google Distributed Cloud (GCP) is a cloud-centric container platform that provides you with a consistent platform to construct and manage modern hybrid and multi-cloud environments through a single pane of glass with Google Cloud console. GKE Enterprise can run on-premises in a VMware vSphere-based or bare metal environment.

GKE on VMware and GKE on Bare Metal, components of Google Distributed Cloud Virtual (GDC Virtual), are software that bring Google Kubernetes Engine (GKE) to on-premises data centers such as Hitachi UCP platform, which has been qualified as an GKE Enterprise Ready platform.

The following figure shows a GKE Enterprise solution and its capabilities to manage your fleet clusters and the applications that run on them. A fleet can be made up of GKE clusters on Google Cloud or include clusters outside Google Cloud running on-premises or other public clouds such as Amazon AWS and Microsoft Azure. GKE Enterprise helps simplify working across multiple clusters and infrastructure providers, and provides the following features:

- Configuration and policy management
- Fleet-wide networking features
- Identity management features
- Observability features

Anthos Service Mesh provides powerful tools for application security, networking, and observability.


Developer Experience
Build, deploy, and run cloud-connected applications
Governance
Configuration & policy mgmt
Security
Zero-trust & service mesh
Operations
Logging and monitoring
Multicluster Fleet Automation
Kubernetes Control Plane / Platform API
Cluster lifecycle management
Google Cloud
On-premises / edge
aws
AWS
Azure
Attached clusters

GKE Enterprise deployment options

Google Cloud and GKE Enterprise features can be used on the following GKE environments:

- Google Kubernetes Engine (GKE) on Google Cloud
- Google Distributed Cloud Virtual (GKE Enterprise on-premises):
  - GKE on VMware
  - GKE on Bare Metal
- Google Distributed Cloud Edge
- Multi-cloud:
  - GKE on AWS
  - GKE on Azure
- Attached clusters, these are third-party Kubernetes clusters (EKS, AKS, and other Kubernetes clusters) registered to your fleet.

This paper focuses on the deployment of GKE on VMware and bare metal on top of Hitachi Unified Compute Platform.

GKE on VMware cluster components

The following components make up a GKE on VMware installation:

- Admin cluster

  The admin cluster is where the Kubernetes control planes for the admin cluster and its associated user clusters run, as well as any add-ons. A single admin cluster can manage multiple user clusters.

  The following nodes are in the admin cluster:

  - Admin cluster control plane — runs the control plane for the admin cluster. The machines that run the admin control plane are called control-plane nodes.
  - User cluster control plane — runs the control plane for a user cluster. There will be a VM for each deployed user cluster.

    > **Note:** Since version 1.14, Anthos, now GKE Enterprise, introduced support for user clusters with Controlplane V2. When enabling this mode, the control plane for a user cluster runs on one or more nodes in the user cluster itself instead of in the admin cluster. Controlplane V2 has become the default and recommended setting for cluster creation. See https://cloud.google.com/anthos/clusters/docs/on-prem/latest/how-to/create-user-cluster-controlplane-v2 for more details.

  - Add-ons — run add-ons such as logging and monitoring components.
- User clusters

  A user cluster is where you deploy and run your organization‘s workloads and services. Each node in a user cluster is called a data-plane node. The number (default 3 nodes) and resources for these nodes in a user cluster depend on the workloads your organization plans to run.
- Admin workstation

  The admin workstation is a separate VM with the tools cluster creators and developers need to manage GKE on VMware. The following tools are used from the admin workstation:

  - `Kubectl` — used to interact with your admin and user clusters, including deploying and managing workloads.
  - `Gkectl` — used to create and update clusters and perform other administrative tasks.

As an alternative to logging into the admin workstation, Google Cloud console provides a web interface where you can perform a subset of GKE on VMware administrative tasks, including creating new user clusters.

vSphere Cloud Native Storage (CNS)

Cloud Native Storage (CNS) integrates vSphere and Kubernetes and offers capabilities to create and manage container volumes deployed in a vSphere environment. CNS consists of two components, a CNS component in vCenter Server and a vSphere volume driver (also called the vSphere CSI driver) in Kubernetes, called vSphere Container Storage Plug-in.

- CNS enables vSphere and vSphere storage (VMFS, vVols, NFS), including vSAN, as a platform to run stateful applications. CNS enables access of this data path to Kubernetes and brings an understanding of Kubernetes volume and pod abstractions to vSphere. CNS uses several components to work with vSphere storage; this includes VMFS or vVols provided by the Hitachi Storage Provider for VMware vCenter. After you create PVs, you can review them and the virtual disks that back them in the vSphere Client and monitor their storage policy compliance.
- The vSphere Container Storage Plug-in has different components that provide an interface used by the Container Orchestrators such as GKE to manage the lifecycle of vSphere volumes. It also allows you to create, expand and delete volumes, attach, and detach volumes to the data-plane node VMs and use bind mounts for the volumes inside the pods.

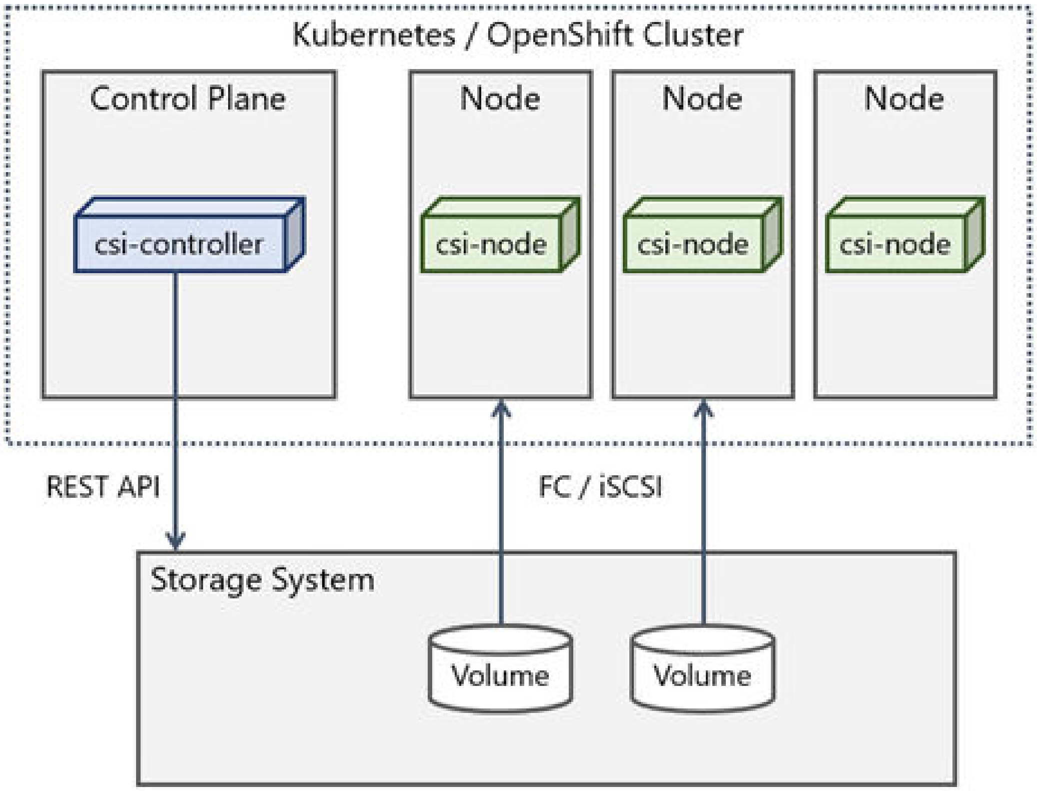
GKE on VMware requires installation of the vSphere CSI driver. This CSI driver is installed automatically in GKE Enterprise clusters when the clusters are provisioned.

Hitachi Storage Plug-in for Containers

Hitachi Storage Plug-in for Containers is a software component that contains libraries, settings, and commands that you can use to create a container to run your stateful applications. It enables stateful applications to persist and maintain data after the lifecycle of the container has ended. Storage Plug-in for Containers provides persistent volumes from Hitachi Dynamic Provisioning (HDP) or Hitachi Thin Image (HTI) pools to bare metal or hybrid deployments using the Fibre Channel protocol. Storage Plug-in for Containers can also provide virtual environments with persistent storage if iSCSI is used.

Storage Plug-in for Containers integrates Kubernetes with Hitachi storage systems using Container Storage Interface (CSI).

The following diagram illustrates a container environment where Storage Plug-in for Containers is deployed.



Solution design

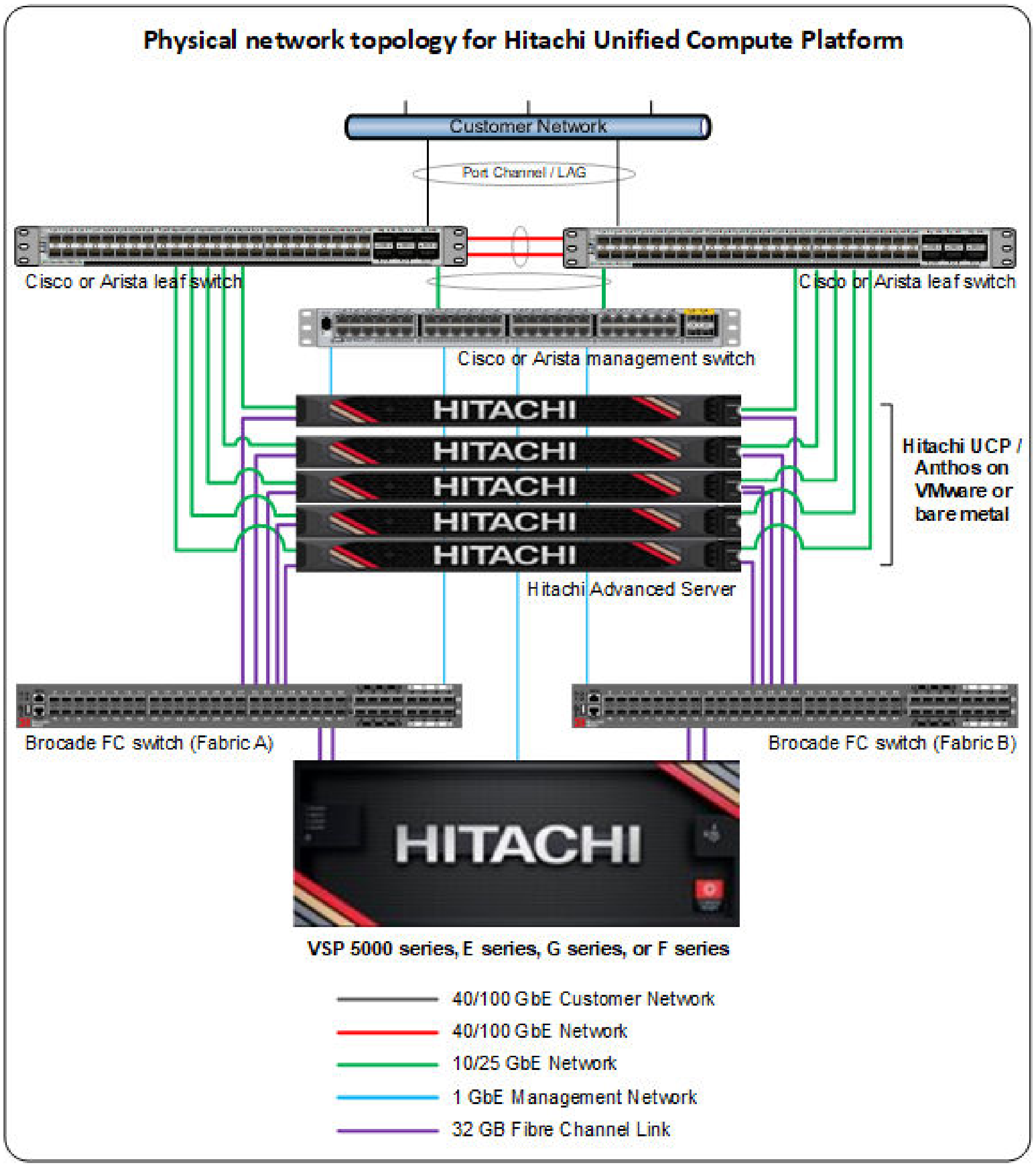
This section describes the detailed solution example for the Hitachi Unified Compute Platform and Google GKE Enterprise.

UCP infrastructure components

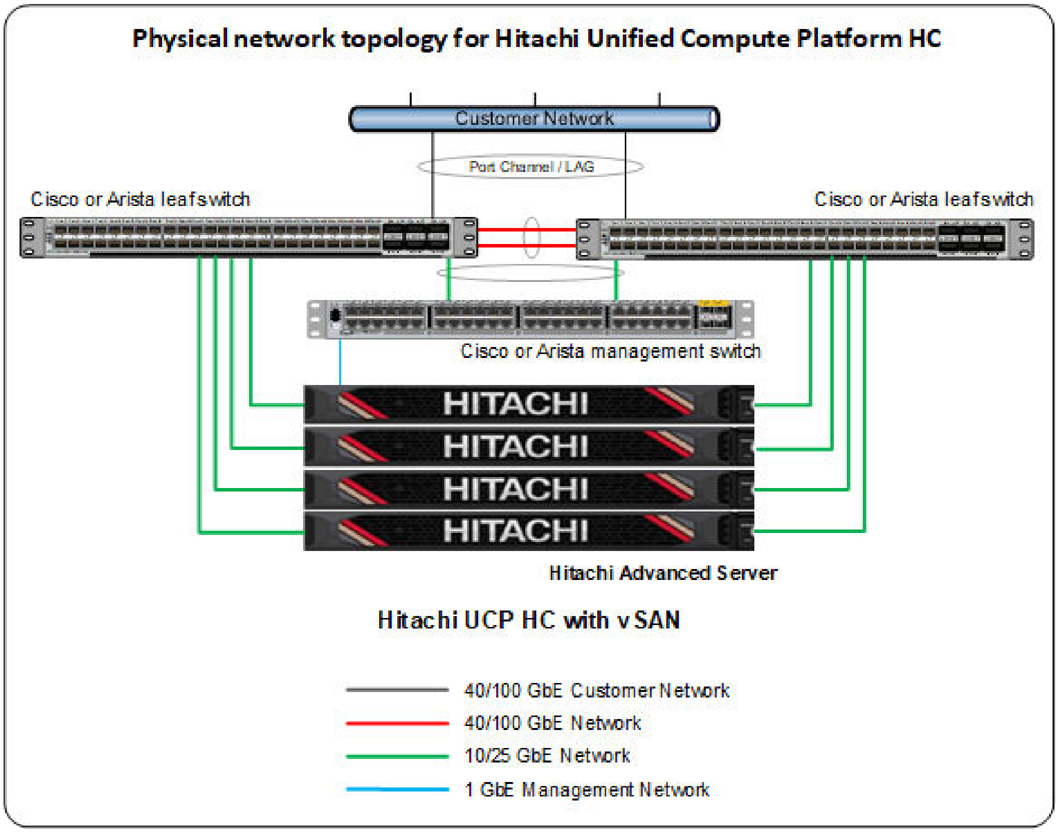
The following figure shows a high availability configuration of Hitachi Unified Compute Platform used to validate the Google GKE Enterprise on-prem solution. It includes the following components:

- Two Cisco 9332C or Arista 7050CX3 spine Ethernet switches.
- Two Cisco 93180YC-FX or Arista 7050SX3 leaf Ethernet switches.
- One Cisco 92348 or Arista 7010T management switch.
- Three or four Hitachi Advanced Server models for the vSAN cluster.
    - For vSAN compute nodes, leverage supported internal drives. These compute nodes are vSAN Ready Node Certified as UCP HC.
    - For vVols or VMFS compute nodes, leverage the HBA PCIe card, which is optionally configured together with the UCP HC vSAN Ready Nodes, or when configuring UCP Fibre Channel-only nodes in UCP RS.
- One Hitachi Virtual Storage Platform storage system for the UCP CI option.

The following diagram represents a standard architecture for the Hitachi Unified Compute Platform product portfolio that supports both VMware and bare metal deployments.


Physical network topology for Hitachi Unified Compute Platform
Customer Network
Port Channel / LAG
Cisco or Arista leaf switch
Cisco or Arista leaf switch
Cisco or Arista management switch
HITACHI
HITACHI
HITACHI
HITACHI
HITACHI
Hitachi UCP / Anthos on VMware or bare metal
Hitachi Advanced Server
Brocade FC switch (Fabric A)
Brocade FC switch (Fabric B)
HITACHI
VSP 5000 series, E series, G series, or F series
40/100 GbE Customer Network
40/100 GbE Network
10/25 GbE Network
1 GbE Management Network
32 GB Fibre Channel Link

The following diagram represents a standard architecture for the Hitachi Unified Compute Platform HC (with VMware vSAN).


Physical network topology for Hitachi Unified Compute Platform HC
Customer Network
Port Channel / LAG
Cisco or Arista leaf switch
Cisco or Arista leaf switch
Cisco or Arista management switch
HITACHI
HITACHI
HITACHI
HITACHI
Hitachi Advanced Server
Hitachi UCP HC with vSAN
40/100 GbE Customer Network
40/100 GbE Network
10/25 GbE Network
1 GbE Management Network

The configuration with Hitachi Virtual Storage Platform is described in Unified Compute Platform product portfolio documentation. See the Hitachi Unified Compute Platform CI for VMware vSphere Reference Architecture Guide at https://knowledge.hitachivantara.com/Documents/Application_Optimized_Solutions/VMWare/Unified_Compute_Platform_CI_for_VMware_vSphere_Reference_Architecture_Guide for more information regarding Unified Compute Platform CI configurations.

Hitachi UCP Advisor (optional)

Hitachi Unified Compute Platform Advisor (UCP Advisor) brings simplified IT administration to virtualized, converged, and hyperconverged systems from Hitachi. UCP Advisor supports guided life-cycle management to the server, network, and storage elements within supported Unified Compute Platform systems.

VMware vVols and storage policy-based management (SPBM) (optional)

Storage Provider for VMware vCenter (VASA Provider) enables organizations to deploy Hitachi Storage infrastructure with VMware vSphere virtual volumes (vVols) to bring customers on a reliable enterprise journey to a software-defined, policy-controlled data center.

Hitachi storage policy-based management allows automated provisioning of virtual machines (VMs) and quicker adjustment to business changes. Virtual infrastructure (VI) administrators can make changes to policies to reflect changes in their business environment, dynamically matching storage policy requirements for VMs to available storage pools and services. The vVols solution reduces the operational burden between VI administrators and storage administrators with an efficient collaboration framework leading to faster and better VM and application services provisioning.

To use VMware vVols with Hitachi storage, install Hitachi Storage Provider for VMware vCenter. See *VMware vSphere Virtual Volumes (vVols) with Hitachi Virtual Storage Platform Quick Start and Reference Guide* at https://knowledge.hitachivantara.com/Documents/Application_Optimized_Solutions/VMWare/VMware_vSphere_Virtual_Volumes_(vVols)_with_Hitachi_Virtual_Storage_Platform_Quick_Start_and_Reference_Guide for details.

See *Storage Provider for VMware vCenter (VASA)* https://knowledge.hitachivantara.com/Documents/Adapters_and_Drivers/Storage_Adapters_and_Drivers/VMware/Storage_Provider_for_VMware_vCenter_(VASA) to deploy this environment.

Scalability considerations on the UCP infrastructure

Scalability for GKE on UCP adheres to the following Google Cloud limitations.

GKE on VMware

- Nodes per cluster: Supports a minimum of 3 nodes and a maximum of 500 nodes using bundled load balancing. With integrated load balancing, a maximum of 250 nodes is supported.
- Pods per node: Each node can accommodate up to 110 pods.
- Total Pods: A user cluster can support up to 15,000 pods.

GKE on Bare Metal

- Nodes per cluster: Supports a minimum of 1 node and a maximum of 500 nodes per cluster. For optimal performance, do not exceed 200 nodes.
- Pods per node: Each node can accommodate up to 110 pods.
- Total Pods: A user cluster can support up to 15,000 pods.
- Registered clusters: By default, a maximum of 15 clusters can be registered to GCP. For additional clusters, a request to increase the limit should be submitted to GCP.

See the official Google Cloud documentation for GKE on VMware at https://cloud.google.com/anthos/clusters/docs/on-prem/latest/concepts/scalability and GKE on Bare Metal at https://cloud.google.com/anthos/clusters/docs/bare-metal/latest/limits for the most accurate and up-to-date scalability information.

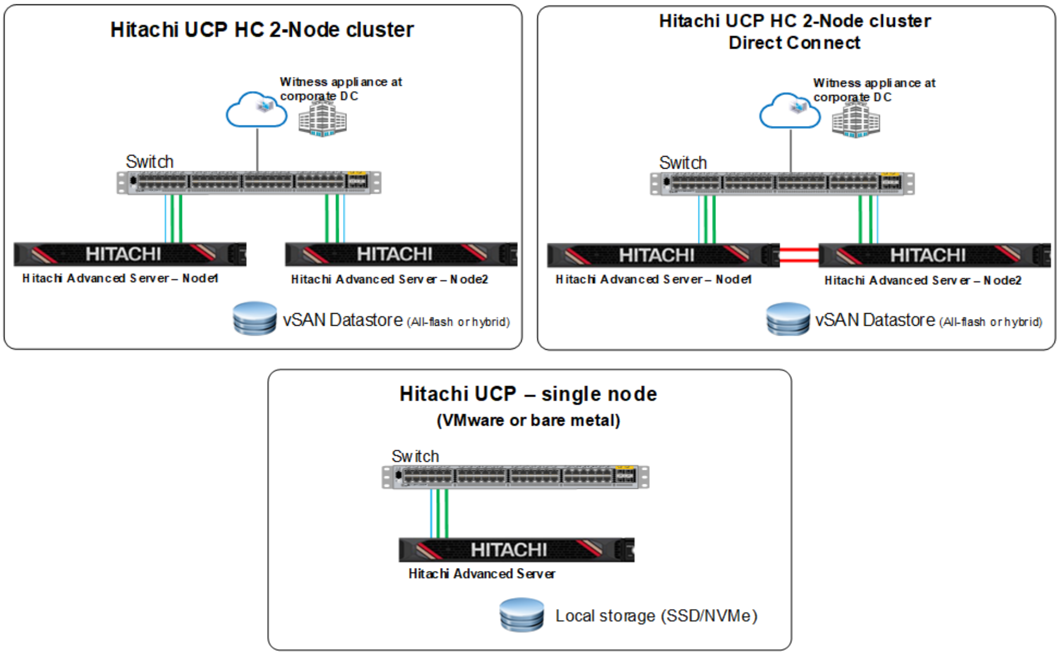
UCP infrastructure with reduced resource footprint for edge locations

A UCP infrastructure can be implemented in locations where minimal configuration is a requirement.

- UCP HC 2-node vSAN cluster provides the same high availability characteristics as a regular vSAN cluster. This can be deployed with either a switch configuration or direct-connect configuration.
- UCP bare metal single node is another option for non-HA environments. And this can be either VMware or bare metal.

GKE on VMware can be deployed in either of these 2 types of UCP infrastructure with reduced resources footprint.

Also, GKE on Bare Metal can be deployed in a single node using the standalone cluster configuration with the edge profile. In this case, the single node cluster lets you run workloads on the same control plane node.


Hitachi UCP HC 2-Node cluster
Witness appliance at corporate DC
Switch
HITACHI
Hitachi Advanced Server – Node1
HITACHI
Hitachi Advanced Server – Node2
vSAN Datastore (All-flash or hybrid)
Hitachi UCP HC 2-Node cluster Direct Connect
Witness appliance at corporate DC
Switch
HITACHI
Hitachi Advanced Server – Node1
HITACHI
Hitachi Advanced Server – Node2
vSAN Datastore (All-flash or hybrid)
Hitachi UCP – single node (VMware or bare metal)
Switch
HITACHI
Hitachi Advanced Server
Local storage (SSD/NVMe)

GKE on VMware (Anthos clusters on VMware)

Overview and prerequisites

GKE on VMware, formerly Anthos clusters on VMware, a component of Google Distributed Cloud Virtual (GDC Virtual), is software that brings Google Kubernetes Engine (GKE) to on-premises data centers. It can be installed on VMware using Hitachi Unified Compute Platform (UCP) converged or hyperconverged infrastructure.

The following components make up an Anthos GKE Enterprise cluster on VMware installation:

- Admin cluster

  The admin cluster is where the Kubernetes control planes for the admin cluster and its associated user clusters run, as well as any add-ons. A single admin cluster can manage multiple user clusters.

  The following nodes are in the admin cluster:

  - Admin cluster control plane — runs the control plane for the admin cluster. The machines that run the admin control plane are called control-plane nodes.
  - User cluster control plane — runs the control plane for a user cluster. There will be one (for non-HA mode) or three VMs (for HA mode) for each deployed user cluster.
  - Add-ons — run add-ons such as logging and monitoring components.

- User clusters

  A user cluster is where you deploy and run your organization's workloads and services. Each node in a user cluster is called a data-plane node. The number of (default 3 nodes) and resources for these nodes in a user cluster depend on the workloads your organization plans to run.

- Admin workstation

  The admin workstation is a separate VM with the tools cluster creators and developers need to manage GKE on VMware. The following tools are used from the admin workstation:

  - `kubectl` — used to interact with your admin and user clusters, including deploying and managing workloads.
  - `gkectl` — used to create and update clusters and perform other administrative tasks.

As an alternative to logging into the admin workstation, Google Cloud console provides a web interface where you can perform a subset of GKE on VMware administrative tasks, including creating new user clusters.

As indicated in previous sections, the GKE on-prem deployment consists of three types of virtual machines:

- Admin workstation VM — used to configure and manage the GKE on VMware clusters.
- Admin cluster VMs — used to run the admin control plane, user cluster’s control plane, and add-ons.
- User Cluster VMs — used to run user workloads and services.

The following tables list the minimum requirements for these different types of virtual machines, basically using the default values.

Node	Requirements	Purpose
Admin workstation	▪ 4 vCPUs ▪ 8 GiB RAM ▪ 100 GiB	This is a standalone VM with the tools and resources needed to create GKE on VMware clusters in your vSphere environment.

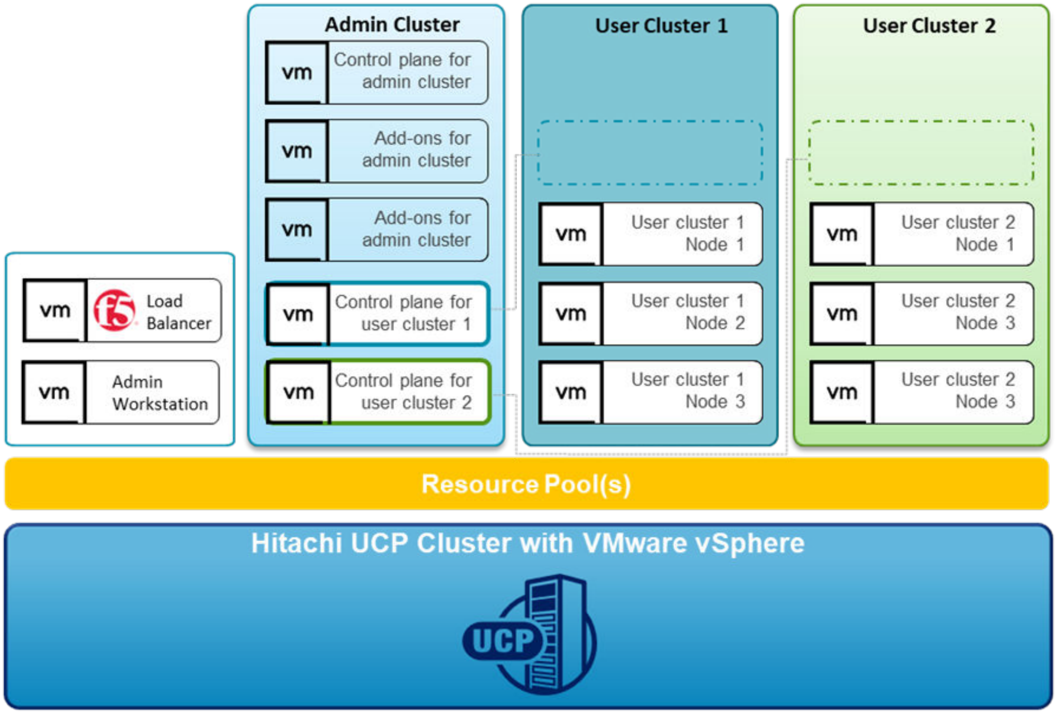
Node	Requirements	Purpose
Admin cluster control-plane	▪ 4 vCPUs ▪ 16 GiB RAM ▪ 40 GiB disk ▪ 100 GiB disk	One VM, runs the control plane for the admin cluster
Add-ons	▪ 4 vCPUs ▪ 16 GiB RAM ▪ 40 GiB disk	Two VMs that run the admin cluster's add-ons.
User cluster control-plane	▪ 4 vCPUs ▪ 8 GiB RAM ▪ 40 GiB disk	For each user cluster, one or three VMs. Runs the control plane for user clusters.

Node	Requirements	Purpose
User cluster data-plane node	▪ 4 vCPU(s) ▪ 8 GiB ▪ 40 GiB	A user cluster node is where your workloads run. These values are the default. The number of nodes and resources required will depend on the workloads you plan to run.

See https://cloud.google.com/anthos/clusters/docs/on-prem/latest/how-to/cpu-ram-storage for additional details about hardware requirements for GKE on VMware (Anthos clusters on VMware).

GKE on VMware deployment example:

In this solution, an admin cluster and two user clusters have been deployed on-premises on top of Hitachi UCP with VMware vSphere. The following figure shows that for this deployment, the admin cluster consists of five virtual machines, one control plane for the admin cluster, one control plane for user cluster-1, one control plane for user cluster-2, and two add-ons for the admin cluster. Both user clusters consist of three virtual machines or data-plane nodes for user workloads. The environment includes the admin workstation virtual machine and F5 Big-IP Load Balancer.



Deploy GKE on VMware (GKE Enterprise on-prem on VMware)

On-premises GKE Enterprise clusters can be installed on VMware using Hitachi Unified Compute Platform (UCP) converged or hyperconverged infrastructure, depending on your application and business needs. For complete guides to GKE Enterprise on-premises options with Google Distributed Cloud Virtual, including cluster setup and administration, see the following resource:

- GKE on VMware: https://cloud.google.com/anthos/gke/docs/on-prem

This section of this guide covers the setup of GKE on VMware on VMware using the Hitachi Unified Compute Platform portfolio.

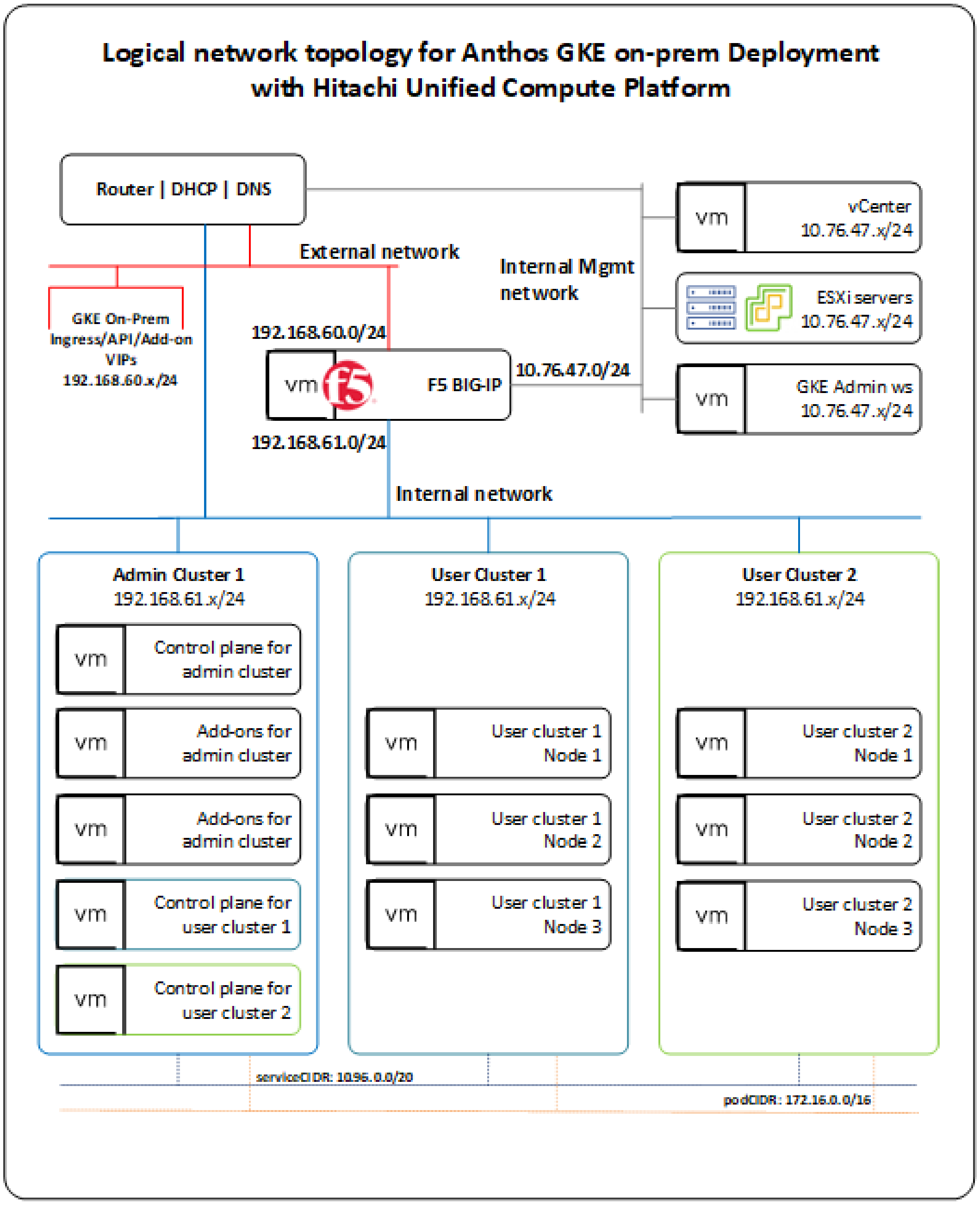
Set up Hitachi UCP CI or UCP HC clusters

To validate this reference architecture, a UCP CI/UCP HC cluster type was configured as described in UCP infrastructure components (on page 14). The VMware cluster was configured following best practices as described in the Hitachi UCP documentation.

The deployment environment consists of the following components:

- VMware vCenter cluster configured to support block storage (with Hitachi Virtual Storage Platform) and VMware vSAN storage:
  - UCP CI deployment for block storage — ESXi cluster connected to Hitachi Virtual Storage Platform (using Fibre channel)
  - UCP HC deployment (vSAN) — ESXi/vSAN cluster (vSAN Ready Nodes)
- Load balancer
- GKE Enterprise (Anthos) admin workstation
- Red Hat client workstation (or jump server)
- DNS server
- DHCP server

The following illustration shows a high-level logical network topology for the deployment of GKE Enterprise on-prem on top of Hitachi Unified Compute Platform.


Logical network topology for Anthos GKE on-prem Deployment with Hitachi Unified Compute Platform
Router | DHCP | DNS
External network
Internal Mgmt network
GKE On-Prem Ingress/API/Add-on VIPs 192.168.60.x/24
192.168.60.0/24
vm F5 BIG-IP
10.76.47.0/24
192.168.61.0/24
vm vCenter 10.76.47.x/24
ESXi servers 10.76.47.x/24
vm GKE Admin ws 10.76.47.x/24
Internal network
Admin Cluster 1 192.168.61.x/24
vm Control plane for admin cluster
vm Add-ons for admin cluster
vm Add-ons for admin cluster
vm Control plane for user cluster 1
vm Control plane for user cluster 2
User Cluster 1 192.168.61.x/24
vm User cluster 1 Node 1
vm User cluster 1 Node 2
vm User cluster 1 Node 3
User Cluster 2 192.168.61.x/24
vm User cluster 2 Node 1
vm User cluster 2 Node 2
vm User cluster 2 Node 3
serviceCIDR: 10.96.0.0/20
podCIDR: 172.16.0.0/16

Deploy and configure a load balancer

There are several load balancing options supported by GKE on VMware, and you can choose the option most suited according to your needs.

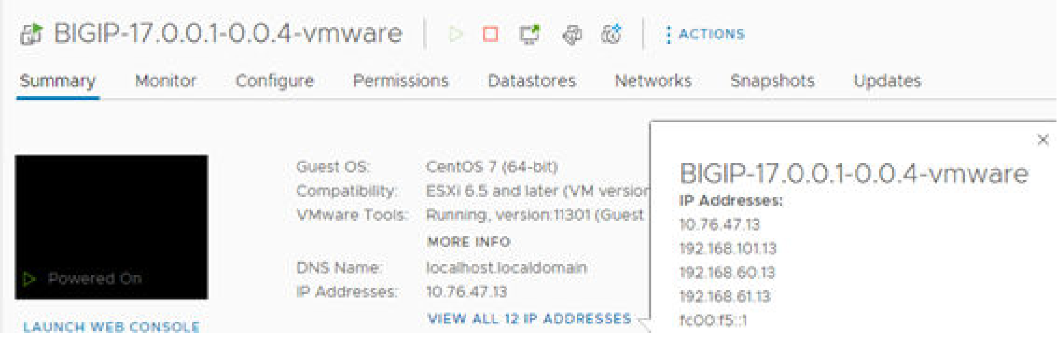
In this validation, GKE on VMware was configured to be integrated with F5 Big-IP. When choosing this option, GKE on VMware clusters automatically configure the required VIPs on the load balancer.

The following summarizes the steps to deploy and configure F5 Big-IP Virtual Edition appliance.

Procedure

1. Download the F5 Big-IP Virtual Edition (OVA) from F5.

   This requires registration and login to the official F5 site.
2. Deploy the OVA into the UCP cluster using either DHCP or static for the management interface.

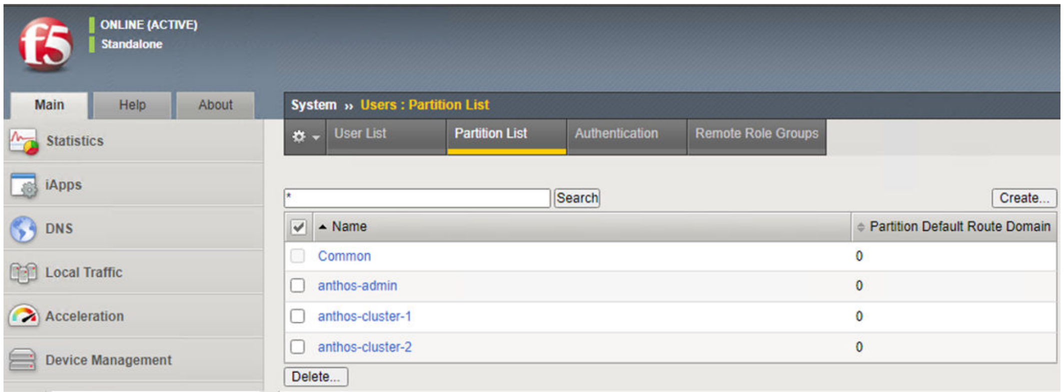


Guest OS: CentOS 7 (64-bit)
Compatibility: ESXi 6.5 and later (VM versior
VMware Tools: Running, version:11301 (Guest
DNS Name: localhost.localdomain
IP Addresses: 10.76.47.13

BIGIP-17.0.0.1-0.0.4-vmware
IP Addresses:
10.76.47.13
192.168.101.13
192.168.60.13
192.168.61.13
fc00:f5::1

3. Log in using the management IP and activate the license.
4. Configure the internal network, external network, and HA network if deploying multiple virtual appliances for the solution.
5. Create individual partitions for the GKE on VMware admin cluster and for each user cluster to be deployed.

   The partition list should look like the following.



☑	▲ Name	Partition Default Route Domain
□	Common	0
□	anthos-admin	0
□	anthos-cluster-1	0
□	anthos-cluster-2	0

See *Installing F5 BIG-IP ADC for GKE on VMware* at https://cloud.google.com/architecture/partners/installing-f5-big-ip-adc-for-gke-on-prem for specific details about the setup of F5 Big-IP for GKE Enterprise.

> **Note:** The F5 Big-IP load balancer is not bundled with GKE Enterprise, so you must get a license and set up the load balancer separately from installing GKE on VMware. The load balancer must be configured before configuring GKE on VMware clusters.

Set up Google Cloud resources

After the UCP on-prem infrastructure has been configured, you can start the GKE Enterprise deployment process. This requires certain tools that are in the Google CLI and several other prerequisites are needed to deploy and access the solution. See https://cloud.google.com/anthos/clusters/docs/on-prem/latest for specific details.

To prepare the environment for GKE on-prem on VMware, follow these steps.

Procedure

1. Create a Google Cloud project, following the steps from *Creating and managing projects* at https://cloud.google.com/resource-manager/docs/creating-managing-projects.

   > **Note:** Request that your cloud administration team create a project configured for access to GKE on VMware. All projects intended for use with GKE Enterprise must be whitelisted by Google.

2. Deploy a client workstation to manage the installation of GKE on VMware.

   This client workstation can be Linux, MacOS, or Windows. The validation for this paper was done on a client workstation using Red Hat Enterprise Linux 8.4. This workstation must be able to communicate with the VMware vCenter server and the Internet.

3. Install Google Cloud CLI and related tooling on the client workstation.

   a. Install Google Cloud CLI, but skip the `gcloud init` command, and follow instructions at https://cloud.google.com/sdk/docs or see https://cloud.google.com/sdk/docs/downloads-interactive#linux-mac to use the Google Cloud CLI installer in an interactive mode.

   b. After the Cloud CLI has been installed, verify the installed components with the following command:

   ```
   gcloud components list
   ```

   c. If needed, update the gcloud CLI using the following command:

   ```
   gcloud components update
   ```

   d. Install anthos-auth and kubectl using the following commands:

   ```
   gcloud components install kubectl
   gcloud components install anthos-auth
   ```

4. After the workstation has been configured with Google Cloud CLI and related tooling, log in to Google Cloud using the credentials from your organization. Enter the login command and it will display a URL that can be copied into a browser to allow sign-in to Google services. After login, it will present an authorization code that you can copy and paste back into the client workstation and then press **Enter**, as follows:

```
[root@jp-gke-adminws ~]# gcloud auth login
Go to the following link in your browser:
    https://accounts.google.com/o/oauth2/auth?
response_type=code&client_id=32555940559.apps.googleusercontent.com&redirect_uri=
https%3A%2F%2Fsdk.cloud.google.com%2Fauthcode.html&scope=openid+https%3A%2F
%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email+https%3A%2F%2Fwww.googleapis.com
%2Fauth%2Fcloud-platform+https%3A%2F%2Fwww.googleapis.com%2Fauth
%2Fappengine.admin+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fsqlservice.login
+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcompute+https%3A%2F
%2Fwww.googleapis.com%2Fauth
%2Faccounts.reauth&state=JpRTK02S4iYbGcRk0xD8AhIeTkVeZo&prompt=consent&access_typ
```

```
e=offline&code_challenge=PuLG0MKEYxyDpWi1H22C7nZsyDNk4Xoci98v95PT2mA&code_challen
ge_method=S256
Enter authorization code:
4/0AWtgzh7cbWJCGroizNyu6Pl0tP0hh6ByEbmIttvcrhpJVjHYxNozMaODOvU_AtaspOdrtw


                                                                            You
are now logged in as
[cccc.ccccc2@hitachivantara.com].
                                                                   Your current
project is [hv-ucp-anthos].  You can change this setting by
running:
                    $ gcloud config set project PROJECT_ID
```

5. Enable Google APIs in your Cloud project so that your on-prem environment can communicate with Google Cloud.

   For this validation, we used the project hv-ucp-anthos. The following example shows how to enable Google APIs in your Cloud project:

```
gcloud services enable --project hv-ucp-anthos \
anthos.googleapis.com \
anthosgke.googleapis.com \
anthosaudit.googleapis.com \
cloudresourcemanager.googleapis.com \
container.googleapis.com \
gkeconnect.googleapis.com \
gkehub.googleapis.com \
serviceusage.googleapis.com \
stackdriver.googleapis.com \
opsconfigmonitoring.googleapis.com \
monitoring.googleapis.com \
logging.googleapis.com \
iam.googleapis.com \
storage.googleapis.com \
connectgateway.googleapis.com
```

6. Create service accounts and grant required roles.

   Before you create your admin and user clusters, you must create these service accounts:

Service Account Name	Purpose
Component access service account	This service account is used to download cluster components on your behalf, from the Container Registry.
Connect-register service account	This service account is used to register your clusters with Google Cloud.
Logging-monitoring service account	This service account is used to export logs and metrics from clusters to Cloud Logging and Cloud Monitoring.

Depending on the features you want to enable, you might also need to have some optional service accounts. See https://cloud.google.com/anthos/clusters/docs/on-prem/latest/how-to/service-accounts#optional_service_accounts for details.

The following steps provide examples of how to manually create these service accounts and grant the required roles to these service accounts. For each service account, first create the service account, then create a JSON key, and then grant the required roles.

7. Create a component access service account.

```
gcloud iam service-accounts create component-access-sa \
--display-name "Component Access Service Account" \
--project hv-ucp-anthos

gcloud iam service-accounts keys create component-access-key.json \
--iam-account component-access-sa@hv-ucp-anthos.iam.gserviceaccount.com \
--project hv-ucp-anthos
```

> **Note:** Depending on your organization policies, service account key creation might be disabled; check with your cloud administrator to create the JSON keys if necessary.

8. After the accounts have been created, grant Identity and Access Management (IAM) roles to your component access service account.

   The following roles are required so GKE on VMware can do preflight checks:

   - `serviceusage.serviceUsageViewer`
   - `iam.roleViewer`
   - `iam.serviceAccountViewer`
   - `compute.viewer`

```
gcloud projects add-iam-policy-binding hv-ucp-anthos \
--member "serviceAccount:component-access-sa@hv-ucp-
anthos.iam.gserviceaccount.com" \
--role "roles/serviceusage.serviceUsageViewer"

gcloud projects add-iam-policy-binding hv-ucp-anthos \
--member "serviceAccount:component-access-sa@hv-ucp-
anthos.iam.gserviceaccount.com" \
--role "roles/iam.roleViewer"

gcloud projects add-iam-policy-binding hv-ucp-anthos \
--member "serviceAccount:component-access-sa@hv-ucp-
anthos.iam.gserviceaccount.com" \
--role "roles/iam.serviceAccountViewer"

gcloud projects add-iam-policy-binding hv-ucp-anthos \
--member "serviceAccount:component-access-sa@hv-ucp-
anthos.iam.gserviceaccount.com" \
--role "roles/compute.viewer"
```

9. Create a connect-register service account.

```
gcloud iam service-accounts create connect-register-sa \
--display-name "Connect-register Service Account" \
--project hv-ucp-anthos

gcloud iam service-accounts keys create connect-register-key.json \
--iam-account connect-register-sa@hv-ucp-anthos.iam.gserviceaccount.com \
--project hv-ucp-anthos
```

10. The connect-register service account must be granted the gkehub.admin role on your fleet host project. This is the Cloud project where you view and manage your clusters.

```
gcloud projects add-iam-policy-binding hv-ucp-anthos \
--member "serviceAccount:connect-register-sa@hv-ucp-
anthos.iam.gserviceaccount.com" \
--role "roles/gkehub.admin"
```

11. Create a logging-monitoring service account.

```
gcloud iam service-accounts create logging-monitoring-sa \
--display-name "Logging-monitoring Service Account" \
--project=hv-ucp-anthos

gcloud iam service-accounts keys create logging-monitoring-key.json \
--iam-account logging-monitoring-sa@hv-ucp-anthos.iam.gserviceaccount.com \
--project hv-ucp-anthos
```

The logging-monitoring service account must be granted the following roles on your logging-monitoring project. This is the Cloud project where you view logs for your clusters.

- `stackdriver.resourceMetadata.writer`
- `opsconfigmonitoring.resourceMetadata.writer`
- `logging.logWriter`
- `monitoring.metricWriter`
- `monitoring.dashboardEditor`

```
gcloud projects add-iam-policy-binding hv-ucp-anthos \
--member "serviceAccount:logging-monitoring-sa@hv-ucp-
anthos.iam.gserviceaccount.com" \
--role "roles/stackdriver.resourceMetadata.writer"

gcloud projects add-iam-policy-binding hv-ucp-anthos \
--member "serviceAccount:logging-monitoring-sa@hv-ucp-
anthos.iam.gserviceaccount.com" \
--role "roles/opsconfigmonitoring.resourceMetadata.writer"

gcloud projects add-iam-policy-binding hv-ucp-anthos \
--member "serviceAccount:logging-monitoring-sa@hv-ucp-
```

```
anthos.iam.gserviceaccount.com" \
--role "roles/logging.logWriter"

gcloud projects add-iam-policy-binding hv-ucp-anthos \
--member "serviceAccount:logging-monitoring-sa@hv-ucp-
anthos.iam.gserviceaccount.com" \
--role "roles/monitoring.metricWriter"

gcloud projects add-iam-policy-binding hv-ucp-anthos \
--member "serviceAccount:logging-monitoring-sa@hv-ucp-
anthos.iam.gserviceaccount.com" \
--role "roles/monitoring.dashboardEditor"
If needed, use the following command to list the created service accounts:

[root@jp-gke-adminws ~]# gcloud iam service-accounts list

DISPLAY NAME
EMAIL                                                        DISABLED
Connect-register Service Account        connect-register-sa@hv-ucp-
anthos.iam.gserviceaccount.com    False
Component Access Service Account        component-access-sa@hv-ucp-
anthos.iam.gserviceaccount.com    False
Logging-monitoring Service Account      logging-monitoring-sa@hv-ucp-
anthos.iam.gserviceaccount.com  False
```

Note: In these examples, make sure to substitute your Project ID and service account name.

Deploy admin workstation on Hitachi UCP

An admin workstation is required to create GKE on VMware (Anthos clusters on VMware). The admin workstation is a standalone VM that is deployed within your Hitachi UCP cluster and is preinstalled with all the tools and resources required to create GKE on the VMware solution.

In this validation we used the `gkeadm` command-line tool, which is available for Linux, Windows, or MacOS. To deploy the admin workstation, follow these steps.

Procedure

1. Download the `gkeadm` tool from https://cloud.google.com/anthos/clusters/docs/on-prem/latest/how-to/download-gkeadm.

   Version 1.14.1 was the latest available at the time of this validation.

   ```
   [root@jp-gke-adminws ~]# gsutil cp gs://gke-on-prem-release/gkeadm/1.14.1-gke.39/
   linux/gkeadm ./
   Copying gs://gke-on-prem-release/gkeadm/1.14.1-gke.39/linux/gkeadm...
   \ [1 files][ 84.4 MiB/ 84.4 MiB]
   Operation completed over 1 objects/84.4 MiB.
   ```

```
[root@jp-gke-adminws ~]# chmod +x gkeadm
```

2. Get the vCenter CA root certificate, which is used by gkeadm and GKE on-prem to authenticate to the vCenter.

```
true | openssl s_client -connect vcsoleng.sce.lab:443 -showcerts 2>/dev/null |
sed -ne '/-BEGIN/,/-END/p' > vcsoleng-sce-lab.pem
```

3. Copy the vCenter certificate file to the location of your choice. The path will be used on the configuration file when creating the admin workstation.
4. To view the decoded certificate, use the following command:

```
openssl x509 -in vcsoleng-sce-lab.pem -text -noout
```

   Another way to get the certificate is described in *Getting your vCenter CA root certificate* at https://cloud.google.com/anthos/clusters/docs/on-prem/latest/how-to/vcenter-ca-cert-path.

5. Use the `gkeadm` tool to generate the following template configuration files: `credential.yaml` and `admin-ws-config.yaml`.

```
[root@jp-gke-adminws ~]# ./gkeadm create config
Created credential template at "credential.yaml".
Created config template at "admin-ws-config.yaml".
```

   a. Update the `credential.yaml` file with the vCenter server's username and password:

```
kind: CredentialFile
# list of credentials
items:
# reference name for this credential entry
- name: vCenter
  username: "administrator@vsphere.local"
  password: "vCenterAdminPassword"
```

   b. Update the `admin-ws-config.yaml` configuration file with the values specific to your environment:

   - Path to the JSON key file for your component access service account
   - vCenter IP address or hostname, datacenter, datastore, cluster, resource pool, folder, and network
   - Path to the root CA certificate for your vCenter server
   - IP allocation mode: static or DHCP, in this case we used “static”

- `IP address, netmask, gateway, and DNS for the admin workstation`
- `NTP server address`

The following is an example of the `admin workstation` file edited for this validation:

```
gcp:
  # Path of the component access service account's JSON key file
  componentAccessServiceAccountKeyPath: "/root/gke-files/component-access-key.json"
# Specify which vCenter resources to use
vCenter:
  # The credentials and address GKE On-Prem should use to connect to vCenter
  credentials:
    address: "vcsoleng.sce.lab"
    # reference to vCenter credentials file
    fileRef:
      # read credentials from this file
      path: credential.yaml
      # entry in the credential file
      entry: vCenter
  datacenter: "scdc1"
  datastore: "vsp-1090-mgmt"
  cluster: "HA810G2-GKE-CL1"
  network: "DPortGroup-ha810g2-mgmt"
  # vSphere vm folder to deploy vms into. defaults to datacenter top level folder
  folder: "ucp-gke"
  resourcePool: "Anthos-Resource-Pool"
  # Provide the path to vCenter CA certificate pub key for SSL verification
  caCertPath: "/root/gke-files/certs/lin/vcsoleng-sce-lab.pem"
# The URL of the proxy for the jump host
proxyUrl: ""
adminWorkstation:
  name: gke-admin-ws-221202-142644
  cpus: 4
  memoryMB: 8192
  # The boot disk size of the admin workstation in GB. It is recommended to use a
  # disk with at least 100 GB to host images decompressed from the bundle.
  diskGB: 100
  # Name for the persistent disk to be mounted to the home directory (ending in .vmdk).
  # Any directory in the supplied path must be created before deployment.
  dataDiskName: gke-on-prem-admin-workstation-data-disk/gke-admin-ws-221202-142644-data-disk.vmdk
  # The size of the data disk in MB.
  dataDiskMB: 512
  network:
    # The IP allocation mode: 'dhcp' or 'static'
    ipAllocationMode: "static"
    # # The host config in static IP mode. Do not include if using DHCP
    hostConfig:
    #   # The IPv4 static IP address for the admin workstation
      ip: "10.76.47.16"
    #   # The IP address of the default gateway of the subnet in which the
admin workstation
    #   # is to be created
      gateway: "10.76.47.12"
    #   # The subnet mask of the network where you want to create your
admin workstation
    #   # (e.g. 255.255.255.0)
      netmask: "255.255.255.0"
    #   # The list of DNS nameservers to be used by the admin workstation
      dns:
      - "10.76.46.10"
  # The URL of the proxy for the admin workstation
  proxyUrl: ""
  ntpServer: "10.76.47.1"
```

6. Create the admin workstation using the following command:

```
[root@jp-gke-adminws ~]# ./gkeadm create admin-workstation
Using config file "admin-ws-config.yaml"...
Running preflight validations...
- Validation Category: Tools
    - [SUCCESS] gcloud
    - [SUCCESS] ssh
    - [SUCCESS] ssh-keygen
    - [SUCCESS] scp

- Validation Category: Config Check
...

- Validation Category: vCenter
    - [SUCCESS] Credentials
    - [SUCCESS] vCenter Version
    - [SUCCESS] ESXi Version
    - [SUCCESS] Datacenter
    - [SUCCESS] Datastore
    - [SUCCESS] Resource Pool
    - [SUCCESS] Folder
    - [SUCCESS] Network
    - [SUCCESS] Datadisk

All validation results were SUCCESS.

Downloading OS image ...
Creating admin workstation VM ...
...
******************************************
```

```
Admin workstation VM successfully created:
...
- SSH Key: /root/.ssh/gke-admin-workstation
******************************************
...

Preparing "admin-cluster.yaml" for gkectl...
Preparing "user-cluster.yaml" for gkectl...

********************************************************************
Admin workstation is ready to use.

Admin workstation information saved to /root/gke-admin-ws-221202-142644
This file is required for future upgrades
SSH into the admin workstation with the following command:
ssh -i /root/.ssh/gke-admin-workstation ubuntu@10.76.47.16
********************************************************************
```

7. Connect to the admin workstation.

   Use the command displayed in the previous output to SSH to your admin workstation. For example:

   ```
   ssh -i /root/.ssh/gke-admin-workstation ubuntu@10.76.47.16
   ```

8. After you are connected to the admin workstation, verify that the following generated files are in the home directory:

   - `admin-cluster.yaml` — a template config file for creating your admin cluster.
   - `user-cluster.yaml` — a template config file for creating your user cluster.
   - The JSON key for the component service account. If you let `gkeadm` create the service accounts (when using the `--auto-create-service-accounts` flag), the folder should have all the JSON key files. Otherwise you must manually copy the remaining JSON key files from the client workstation to the GKE on VMware admin workstation. Make note of the name and path because you will need them later to create the clusters.
   - `credential.yaml` — a template config file with vCenter credentials. This file needs to be updated with the load balancer (for example F5 BigIP) and private registry credentials.
   - vCenter cert file

Deploy GKE on VMware admin clusters on Hitachi UCP

An admin cluster must be created before creating any user cluster to run your workloads. The admin cluster runs the Kubernetes control plane for the admin cluster itself and for the user clusters.

Follow these steps to create GKE Enterprise admin cluster.

Procedure

1. On the admin workstation, make a copy of the admin-cluster.yaml template with a new name (for example admin-cluster-ucp.yaml) and start editing with the corresponding IP addresses and load balancing information.

   Most of the fields are already filled in with the values used when you created the admin workstation.

   See https://cloud.google.com/anthos/clusters/docs/on-prem/latest/how-to/create-admin-cluster for an example of an admin cluster configuration file.

2. When the edits are complete, run the following command to validate the configuration file:

   ```
   gkectl check-config --config admin-cluster-ucp.yaml
   ```

3. After the configuration checks have passed, run the following command to initialize your vSphere environment.

   This will import the OS images to vSphere and mark them as templates. If an issue is identified during the configuration check, and if the issue has already been remediated, you can skip the validation using the `--skip-validation-all` flag.

   ```
   gkectl prepare --config admin-cluster-ucp.yaml --skip-validation-all
   ```

4. If you have chosen to use Seesaw load balancer, create and configure the VMs for your Seesaw load balancer with the following command; otherwise skip this command:

   ```
   gkectl create loadbalancer --config admin-cluster-ucp.yaml
   ```

5. Create the GKE Enterprise admin cluster using the following command:

   ```
   gkectl create admin --config admin-cluster-ucp.yaml --skip-validation-all
   ```

   The `gkectl` command creates a kubeconfig file named `kubeconfig` in the current directory. This is the kubeconfig file that must be used to interact with the admin cluster using kubectl or run a diagnosis with `gkectl`. For example, you can list the cluster or list the nodes in the admin cluster using `kubectl`.

   The following is the output for these commands:

   ```
   ubuntu@gke-admin-ws-221202-142644:~$ kubectl --kubeconfig kubeconfig get clusters
   NAME            AGE
   gke-admin-ucp   64d

   ubuntu@gke-admin-ws-221202-142644:~$ kubectl --kubeconfig kubeconfig get nodes
   NAME                                                 STATUS
   ROLES                  AGE   VERSION
   gke-admin-master-75tlg                               Ready    control-plane,
   master   17h   v1.25.5-gke.100
   gke-admin-node-6cf77f44f4-hzkhf                      Ready
   <none>                 16h   v1.25.5-gke.100
   gke-admin-node-6cf77f44f4-t929t                      Ready
   <none>                 16h   v1.25.5-gke.100
   ```

Deploy GKE on VMware user clusters on Hitachi UCP

User clusters can be created using GKE on VMware on-prem API clients, `gkectl`, and Control plane V2. For this validation we created the clusters using the `gkectl` methods described in *Create a user cluster* at https://cloud.google.com/anthos/clusters/docs/on-prem/latest/how-to/create-user-cluster.

Follow these steps to create GKE on VMware user clusters.

Procedure

1. On the admin workstation, make a copy of the `user-cluster.yaml` template with a new name (for example `user-cluster-1.yaml`) and start editing with the corresponding IP addresses, load balancing information, cluster name, and service accounts.

   Most of the fields are already filled in with the values used when you created the admin workstation.

   See https://cloud.google.com/anthos/clusters/docs/on-prem/latest/how-to/create-user-cluster for an example of a user cluster configuration file.

2. When the edits are complete, run the following command to validate the configuration file:

   ```
   gkectl check-config --kubeconfig ADMIN_CLUSTER_KUBECONFIG --config
   USER_CLUSTER_CONFIG
   Replace the ADMIN_CLUSTER_KUBECONFIG  with the path of the kubeconfig file for
   your admin cluster, and the USER_CLUSTER_CONFIG with the file name of your user
   cluster configuration file as shown in the following example.
   gkectl check-config --kubeconfig kubeconfig --config user-cluster-1.yaml
   ```

3. If you have chosen to use Seesaw load balancer, create and configure the VMs for your Seesaw load balancer with the following command, otherwise skip this command:

   ```
   gkectl create loadbalancer --kubeconfig kubeconfig --config user-cluster-1.yaml
   ```

4. Create the first GKE on VMware user cluster using the following command:

   ```
   gkectl create cluster --kubeconfig kubeconfig --config user-cluster-1.yaml
   ```

   The `gkectl` tool creates a kubeconfig file named `USER_CLUSTER_NAME-kubeconfig` in the current directory. This is the kubeconfig file that must be used to interact with the user cluster using `kubectl` or run a diagnosis with `gkectl`. For example, you can list the cluster or list the nodes in the user cluster using `kubectl`.

   The following is the output for these commands:

   ```
   ubuntu@gke-admin-ws-221202-142644:~$ kubectl --kubeconfig anthos-user-ucpcluster-
   1-kubeconfig get clusters
   NAME                        AGE
   anthos-user-ucpcluster-1    64d

   ubuntu@gke-admin-ws-221202-142644:~$ kubectl --kubeconfig anthos-user-ucpcluster-
   1-kubeconfig get nodes
   ```

```
NAME                                                  STATUS   ROLES    AGE   VERSION
anthos-user-ucpcluster-1-pool-1-787b9d7d4f-rgvpx   Ready    <none>   21h
v1.25.5-gke.100
anthos-user-ucpcluster-1-pool-1-787b9d7d4f-sjvrd   Ready    <none>   21h
v1.25.5-gke.100
anthos-user-ucpcluster-1-pool-1-787b9d7d4f-zfsdz   Ready    <none>   21h
v1.25.5-gke.100
```

Also, the `gkectl` tool can be used to diagnose the cluster:

```
ubuntu@gke-admin-ws-221202-142644:~$ gkectl diagnose cluster --kubeconfig
kubeconfig --cluster-name anthos-user-ucpcluster-1
Preparing for the diagnose tool...
Diagnosing the cluster......  DONE
Diagnose result is saved successfully in /home/ubuntu/diagnose-user-anthos-user-
ucpcluster-1-20230209235440.json

- Validation Category: User Cluster F5 BIG-IP
Checking f5 (credentials, partition)...SUCCESS

- Validation Category: OS Images
Checking User cluster OS images exist...SUCCESS

- Validation Category: VCenter
Checking Credentials...SUCCESS
Checking VSphere CSI Driver...SUCCESS
Checking vCenter Version...SUCCESS
Checking ESXi Version...SUCCESS
Checking Datacenter...SUCCESS
Checking Resource pool...SUCCESS
Checking Folder...SUCCESS
Checking Network...SUCCESS

- Validation Category: Datastore
Checking Datastore...SUCCESS

- Validation Category: Cluster Healthiness
Checking user cluster and node pools...SUCCESS
Checking user cluster certificates...SUCCESS
...
Checking anthos-identity-service pods...SUCCESS
Checking gke-managed-metrics-server pods...SUCCESS
Checking cert-manager pods...SUCCESS
Checking kube-public pods...SUCCESS
Checking GKE Hub Membership...SUCCESS
Checking all poddisruptionbudgets...SUCCESS
Checking storage...SUCCESS
Checking resource...SUCCESS
Checking virtual machine resource contention...SUCCESS
Checking host resource contention...SUCCESS
```

```
- Validation Category: Connectivity
Checking VMs TOD (availability)...SUCCESS
Some validations were SKIPPED. Check the report above.
Cluster is healthy!
ubuntu@gke-admin-ws-221202-142644:~$
```

5. To create additional user clusters in your solution, follow these steps:

a. Copy the original `user-cluster.yaml` template or the configuration file used for user-cluster-1 to a new file (for example, `user-cluster-2.yaml`) and start editing with the corresponding IP addresses, load balancing information, and new user cluster name.

b. When the edits are complete, run the following command to validate the configuration file:

```
gkectl check-config --kubeconfig kubeconfig --config user-cluster-2.yaml
```

c. Create an additional GKE Enterprise user cluster with the following command:

```
gkectl create cluster --kubeconfig kubeconfig --config user-cluster-1.yaml
```

d. Verify the new cluster and its nodes. Make sure to use the kubeconfig file corresponding to the newly created user cluster:

```
ubuntu@gke-admin-ws-221202-142644:~$ kubectl --kubeconfig anthos-user-
ucpcluster-2-kubeconfig get clusters
NAME                       AGE
anthos-user-ucpcluster-2   8d

ubuntu@gke-admin-ws-221202-142644:~$ kubectl --kubeconfig anthos-user-
ucpcluster-2-kubeconfig get nodes
NAME                                                STATUS   ROLES    AGE
VERSION
37nthos-user-ucpcluster-2-pool-1-5cb9b895dc-4czj8    Ready    <none>   17h
v1.25.5-gke.100
37nthos-user-ucpcluster-2-pool-1-5cb9b895dc-nr7jz    Ready    <none>   17h
v1.25.5-gke.100
37nthos-user-ucpcluster-2-pool-1-5cb9b895dc-sbdjb    Ready    <none>   17h
v1.25.5-gke.100
```

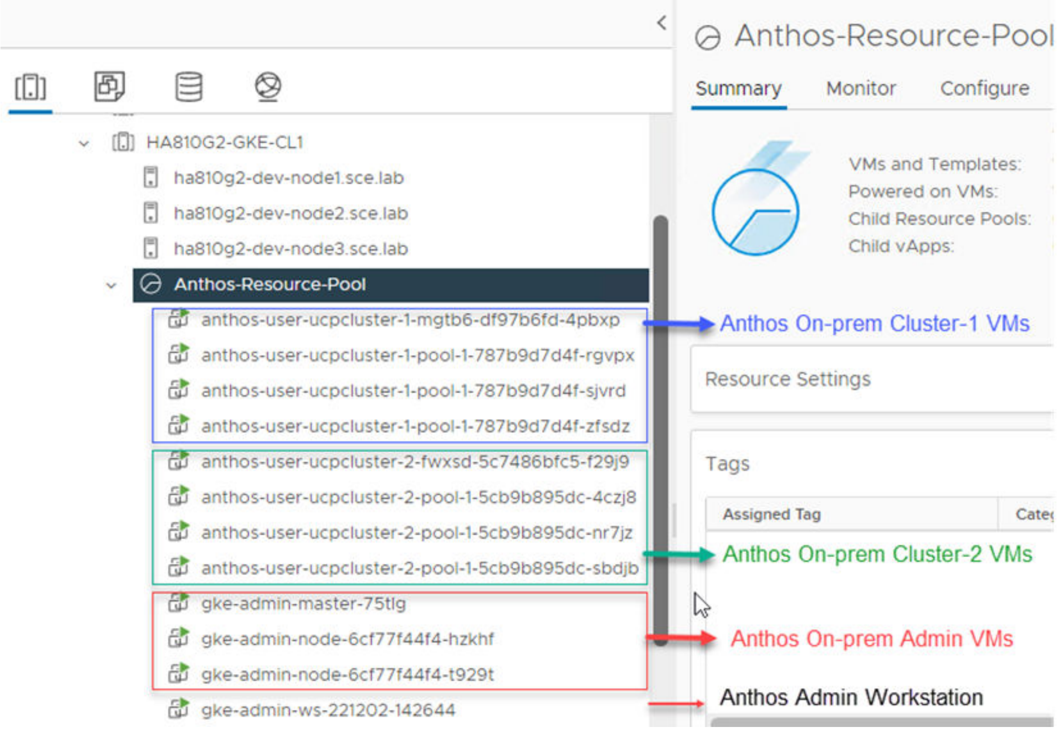
In addition to the kubectl, you can use the `gkectl` tool to list additional details about the clusters:

```
ubuntu@gke-admin-ws-221202-142644:~$ gkectl list admin --kubeconfig
kubeconfig
NAME            AGE   VERSION
gke-admin-ucp   66d   1.14.1-gke.39

ubuntu@gke-admin-ws-221202-142644:~$ gkectl list clusters --kubeconfig
kubeconfig
```

```
NAMESPACE                                   NAME
READY   STATE     AGE   VERSION
anthos-user-ucpcluster-1-gke-onprem-mgmt   anthos-user-ucpcluster-1
True    RUNNING   66d   1.14.1-gke.39
anthos-user-ucpcluster-2-gke-onprem-mgmt   anthos-user-ucpcluster-2
True    RUNNING   9d    1.14.1-gke.39
```

You can see all the deployed GKE Enterprise (Anthos) VMs on the vCenter that is managing the Hitachi UCP/GKE Enterprise environment. These include the GKE Enterprise admin workstation, the admin cluster VMs, and the user cluster VMs under the resource pool defined for the GKE Enterprise on-prem environment.

For this validation, the clusters were created using the default number of nodes. The following illustration shows the default number of nodes for each of the clusters. Also, a control-plane VM is deployed and added to the admin cluster for each user cluster.


HA810G2-GKE-CL1
ha810g2-dev-node1.sce.lab
ha810g2-dev-node2.sce.lab
ha810g2-dev-node3.sce.lab
Anthos-Resource-Pool
anthos-user-ucpcluster-1-mgtb6-df97b6fd-4pbxp
anthos-user-ucpcluster-1-pool-1-787b9d7d4f-rgvpx
anthos-user-ucpcluster-1-pool-1-787b9d7d4f-sjvrd
anthos-user-ucpcluster-1-pool-1-787b9d7d4f-zfsdz
anthos-user-ucpcluster-2-fwxsd-5c7486bfc5-f29j9
anthos-user-ucpcluster-2-pool-1-5cb9b895dc-4czj8
anthos-user-ucpcluster-2-pool-1-5cb9b895dc-nr7jz
anthos-user-ucpcluster-2-pool-1-5cb9b895dc-sbdjb
gke-admin-master-75tlg
gke-admin-node-6cf77f44f4-hzkhf
gke-admin-node-6cf77f44f4-t929t
gke-admin-ws-221202-142644
Anthos-Resource-Pool
Summary
Monitor
Configure
VMs and Templates:
Powered on VMs:
Child Resource Pools:
Child vApps:
Anthos On-prem Cluster-1 VMs
Resource Settings
Tags
Assigned Tag
Categ
Anthos On-prem Cluster-2 VMs
Anthos On-prem Admin VMs
Anthos Admin Workstation

GKE on Bare Metal (Anthos clusters on bare metal)

Overview and prerequisites

GKE on Bare Metal, formerly Anthos clusters on bare metal, a component of Google Distributed Cloud Virtual (GDC Virtual), is software that brings Google Kubernetes Engine (GKE) to on-premises data centers. It can be installed without the need for a vSphere environment on Hitachi Unified Compute Platform (UCP) using one of the GKE Enterprise (Anthos) supported Linux operating systems for bare metal.

The following components make up an GKE cluster on bare metal installation:

- Admin cluster

  The admin cluster is where the Kubernetes control planes for the admin cluster runs, it consists of one or more nodes. Each node is a physical machine running a supported Linux OS.

- User clusters

  A user cluster is where you deploy and run your organization's workloads and services. Each user cluster consists of at least one control-plane node and one data-plane node.

- Admin workstation

  The admin workstation is a separate machine with the tools cluster creators and developers need to manage GKE on Bare Metal. Also, note that an admin workstation can be used as an admin control plane node for one of your clusters, just make sure it meets the requirements as described. The following tools are used from the admin workstation:

  - `kubectl` — used to interact with your admin and user clusters, including deploying and managing workloads.
  - `bmctl` — used to create and update clusters and perform other administrative tasks on bare metal clusters.

As an alternative to logging into the admin workstation, Google Cloud console provides a web interface where you can perform a subset of GKE on Bare Metal administrative tasks, including creating new user clusters.

When installing GKE on Bare Metal, you can create the following types of clusters:

- Admin cluster — used to create and controls user clusters to run workloads.
- User Cluster — used to run user workloads and services.
- Standalone cluster — a single cluster that can manage and run workloads but cannot create or manage other user clusters.
- Hybrid — a single cluster that can manage and run workloads. A hybrid cluster can also create and manage additional user clusters.

In addition to the cluster types, you can choose the installation profile, and there are different resource requirements based on the chosen profile:

- Default — this profile has standard system resource requirements.
- Edge — the edge profile has reduced system resource requirements. This is recommended for edge devices with limited resources.

The following table lists the minimum requirements for the admin workstation, as described at https://cloud.google.com/anthos/clusters/docs/bare-metal/latest/installing/workstation-prerequisites.

Resource	Minimum	Recommended
CPUs / vCPUs	2 core	4 core
RAM	RHEL: 6 GiB	RHEL: 12 GiB

Resource	Minimum	Recommended
	Ubuntu: 4 GiB	Ubuntu: 8 GiB
Storage	128 GiB	256 GiB

The following table lists the resource requirements for all cluster types using default profile. Consider additional resources for your workloads to operate normally.

Resource	Minimum	Recommended
CPUs / vCPUs	4 core	8 core
RAM	16 GiB	32 GiB
Storage	128 GiB	256 GiB

See https://cloud.google.com/anthos/clusters/docs/bare-metal/latest/installing/node-machine-prerequisites for additional details about hardware requirements for GKE on Bare Metal.

Deploy GKE on Bare Metal

On-premises GKE Enterprise clusters can be installed on bare metal using Hitachi Unified Compute Platform (UCP) and one of the GKE Enterprise supported Linux operating systems for bare metal, depending on your application and business needs. For complete guides to GKE Enterprise on-premises options with Google Distributed Cloud Virtual, including cluster setup and administration, see the following resource:

- GKE on Bare Metal: https://cloud.google.com/anthos/clusters/docs/bare-metal

This section of this guide covers the setup of GKE on Bare Metal using the Hitachi Unified Compute Platform portfolio.

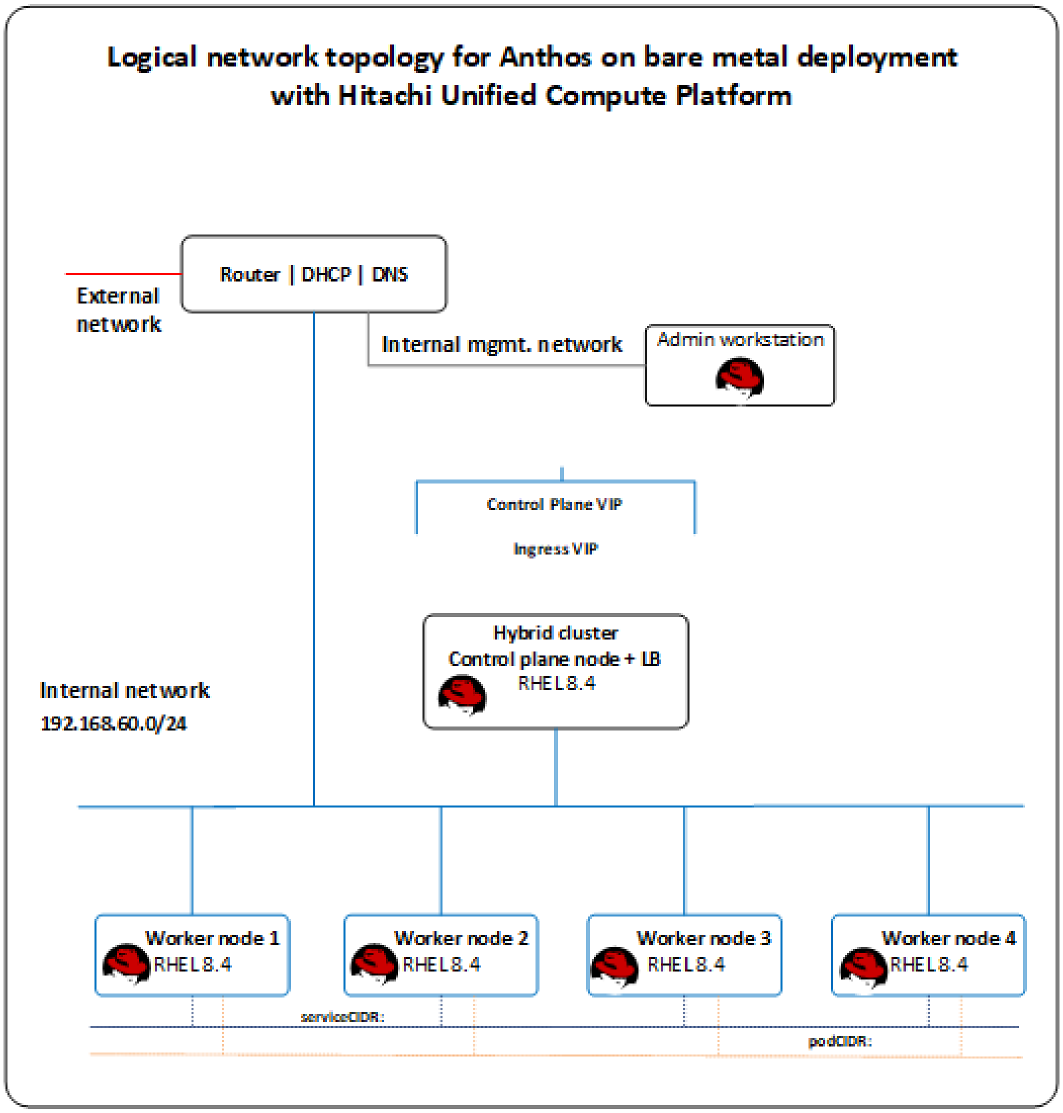
Set up Hitachi UCP bare metal nodes

To validate this reference architecture, a UCP CI cluster type was configured as described in UCP infrastructure components (on page 14). The VMware cluster was configured following best practices as described in the Hitachi UCP documentation.

The deployment environment consists of the following components:

- Hitachi UCP Platform bare metal configured to support block storage (with Hitachi Virtual Storage Platform):
    - UCP CI bare metal deployment using Hitachi VSP storage
    - Operating System and support matrix, see Anthos Ready platform partner and Anthos Ready storage partner
- Network access to Google APIs. See https://cloud.google.com/anthos/clusters/docs/bare-metal/latest/concepts/network-reqs for additional details about internal and external network requirements.
- Load balancer
- GKE on Bare Metal admin workstation, with utilities installed and access to Google Cloud, and access to all your cluster nodes.
- DNS server
- DHCP server

The following illustration shows a high-level logical network topology for the deployment of GKE on Bare Metal with a non-HA control plane, on top of Hitachi Unified Compute Platform.



Prepare the Admin Workstation and set up Google Cloud resources

Installation of GKE on Bare Metal requires certain tools that are in the Google CLI and several other prerequisites are needed to deploy and access the solution. See https://cloud.google.com/anthos/clusters/docs/bare-metal/latest for specific details.

To prepare the Linux admin workstation and set up Google Cloud resources for GKE on Bare Metal, follow these steps.

Procedure

1. Create a Google Cloud project, following the steps from *Creating and managing projects* at https://cloud.google.com/resource-manager/docs/creating-managing-projects.

   > **Note:** Request that your cloud administration team create a project configured for access to Anthos on bare metal.

2. Prepare a Linux admin workstation using one of the supported Linux distributions. The validation for this paper was done on a workstation using Red Hat Enterprise Linux 8.4. This workstation must be able to communicate with all the nodes and the Internet.
3. Install Google Cloud CLI and related tooling on the client workstation.
   a. Install docker version 19.03 or later.
   b. Install Google Cloud CLI, but skip the `gcloud init` command, and follow instructions at https://cloud.google.com/sdk/docs or see https://cloud.google.com/sdk/docs/downloads-interactive#linux-mac to use the Google Cloud CLI installer in an interactive mode.
   c. After the Cloud CLI has been installed, verify the installed components with the following command:

      ```
      gcloud components list
      ```

   d. If needed, update the gcloud CLI using the following command:

      ```
      gcloud components update
      ```

   e. Install anthos-auth and kubectl using the following commands:

      ```
      gcloud components install kubectl
      ```

   f. Install `bmctl`. Make sure to install the version of the cluster that you are creating or operating.

      Before installing `bmctl`, create a baremetal directory, then change (**`cd`**) to the baremetal directory, and proceed with the installation of `bmctl`.

      For details how to install `bmctl` see https://cloud.google.com/anthos/clusters/docs/bare-metal/latest/downloads.
4. After the workstation has been configured with Google Cloud CLI and related tooling, log in to Google Cloud using the credentials from your organization. Enter the login command and it will display a URL that can be copied into a browser to allow sign-in to Google services. After login, it will present an authorization code that you can copy and paste back into the client workstation and then press **Enter**, as follows:

   ```
   gcloud auth application-default login
   ```

5. Enable Google APIs in your Cloud project so that your on-prem environment can communicate with Google Cloud.

   For this validation, we used the project hv-ucp-anthos. The following example shows how to enable Google APIs in your Cloud project:

   ```
   gcloud services enable --project hv-ucp-anthos \
   anthos.googleapis.com \
   anthosgke.googleapis.com \
   anthosaudit.googleapis.com \
   cloudresourcemanager.googleapis.com \
   container.googleapis.com \
   gkeconnect.googleapis.com \
   gkehub.googleapis.com \
   ```

```
gkeonprem.googleapis.com \
serviceusage.googleapis.com \
stackdriver.googleapis.com \
opsconfigmonitoring.googleapis.com \
monitoring.googleapis.com \
logging.googleapis.com \
iam.googleapis.com \
storage.googleapis.com \
connectgateway.googleapis.com
```

6. Create service accounts and grant required roles.

Before you create your admin and user clusters, you must create these service accounts:

ServiceAccount Name	Purpose
Container registry service account	This account is used to download container images from Container Registry.
Connect-agent service account	Used to maintain a connection between your cluster and Google Cloud.
Connect-register service account	This service account is used to register your clusters with Google Cloud.
Cloud-ops service account	This service account is used to export logs and metrics from clusters to Logging and Monitoring.
Storage-agent service account	Used to automatically store snapshots of clusters to Cloud Storage

Service accounts and their respective roles can be created either manually using the `gcloud` CLI or automatically using the `bmctl create config` command.

To create service accounts and roles automatically using the `bmctl`, you can include the `--create-service-accounts` flag when you run `bmctl create config` to have `bmctl` create the service accounts with the required IAM roles. See https://cloud.google.com/anthos/clusters/docs/bare-metal/latest/installing/creating-clusters/admin-cluster-creation for more details.

The following steps provide examples of how to manually create these service accounts and grant the required roles to these service accounts. For each service account, first create the service account, then create a JSON key, and then grant the required roles.

Make sure you are in the `baremetal` directory before proceeding to the next steps.

7. Create a service account for pulling container images.

```
gcloud iam service-accounts create anthos-baremetal-gcr \
--display-name " ucp-anthos-baremetal-gcr service account" \
--project hv-ucp-anthos
```

```
gcloud iam service-accounts keys create anthos-baremetal-gcr.json \
--iam-account anthos-baremetal-gcr@hv-ucp-anthos.iam.gserviceaccount.com \
--project hv-ucp-anthos
```

> **Note:** No special role is required for this `gcr` service account.

8. Create a service account for use with Connect.

```
gcloud iam service-accounts create anthos-baremetal-connect \
--display-name "ucp-anthos-baremetal-connect service account" \
--project hv-ucp-anthos

gcloud iam service-accounts keys create anthos-baremetal-connect.json \
--iam-account anthos-baremetal-connect@hv-ucp-anthos.iam.gserviceaccount.com \
--project hv-ucp-anthos
```

> **Note:** Depending on your organization policies, service account key creation might be disabled; check with your cloud administrator to create the JSON keys if necessary.

After the account has been created, grant Identity and Access Management (IAM) roles to your connect service account.

The `gkehub.connect` role is required for this account:

```
gcloud projects add-iam-policy-binding hv-ucp-anthos \
--member "serviceAccount: anthos-baremetal-connect@hv-ucp-
anthos.iam.gserviceaccount.com" \
--role "roles/gkehub.connect"
```

9. Create a connect-register service account.

```
gcloud iam service-accounts create anthos-baremetal-register \
--display-name "ucp-anthos-baremetal-register service account" \
--project hv-ucp-anthos

gcloud iam service-accounts keys create anthos-baremetal-register.json \
--iam-account anthos-baremetal-register @hv-ucp-anthos.iam.gserviceaccount.com \
--project hv-ucp-anthos
```

10. The connect-register service account must be granted the `gkehub.admin` role on your fleet host project. This is the Cloud project where you view and manage your clusters.

```
gcloud projects add-iam-policy-binding hv-ucp-anthos \
--member "serviceAccount:connect-register-sa@hv-ucp-
anthos.iam.gserviceaccount.com" \
--role "roles/gkehub.admin"
```

11. Create a service account to audit logs and monitor projects.

```
gcloud iam service-accounts create anthos-baremetal-cloud-ops \
--display-name "ucp-anthos-baremetal-cloud-ops service account " \
--project=hv-ucp-anthos

gcloud iam service-accounts keys create anthos-baremetal-cloud-ops.json \
--iam-account anthos-baremetal-cloud-ops@hv-ucp-anthos.iam.gserviceaccount.com \
--project hv-ucp-anthos
```

This cloud ops service account must be granted the following roles on your project:

- `stackdriver.resourceMetadata.writer`
- `opsconfigmonitoring.resourceMetadata.writer`
- `logging.logWriter`
- `monitoring.metricWriter`
- `monitoring.dashboardEditor`

```
gcloud projects add-iam-policy-binding hv-ucp-anthos \
--member "serviceAccount: anthos-baremetal-cloud-ops @hv-ucp-
anthos.iam.gserviceaccount.com" \
--role "roles/stackdriver.resourceMetadata.writer"

gcloud projects add-iam-policy-binding hv-ucp-anthos \
--member "serviceAccount: anthos-baremetal-cloud-ops @hv-ucp-
anthos.iam.gserviceaccount.com" \
--role "roles/opsconfigmonitoring.resourceMetadata.writer"

gcloud projects add-iam-policy-binding hv-ucp-anthos \
--member "serviceAccount: anthos-baremetal-cloud-ops @hv-ucp-
anthos.iam.gserviceaccount.com" \
--role "roles/logging.logWriter"

gcloud projects add-iam-policy-binding hv-ucp-anthos \
--member "serviceAccount: anthos-baremetal-cloud-ops @hv-ucp-
anthos.iam.gserviceaccount.com" \
--role "roles/monitoring.metricWriter"

gcloud projects add-iam-policy-binding hv-ucp-anthos \
--member "serviceAccount: anthos-baremetal-cloud-ops @hv-ucp-
```

```
anthos.iam.gserviceaccount.com" \
--role "roles/monitoring.dashboardEditor"
```

12. Create a service account that can access a Cloud Storage bucket.

    ```
    gcloud iam service-accounts create anthos-baremetal-storage \
    --display-name "ucp-anthos-baremetal-register service account" \
    --project hv-ucp-anthos

    gcloud iam service-accounts keys create anthos-baremetal-storage.json \
    --iam-account anthos-baremetal-storage @hv-ucp-anthos.iam.gserviceaccount.com \
    --project hv-ucp-anthos
    ```

    This service account must be granted the `storage.admin` role so that it can upload data to a Cloud Storage bucket or import VM images.

    ```
    gcloud projects add-iam-policy-binding hv-ucp-anthos \
    --member "serviceAccount:anthos-baremetal-storage@hv-ucp-
    anthos.iam.gserviceaccount.com" \
    --role "roles/storage.admin"
    ```

    Depending on the version of GKE being deployed there could be a need to create a custom role with specific permissions for the storage service account. See https://cloud.google.com/anthos/clusters/docs/bare-metal/latest/installing/configure-sa for details.

Next steps

If needed, use the following command to list the created service accounts:

```
gcloud iam service-accounts list
```

Deploy and configure a load balancer

GKE on Bare Metal supports two types of load balancers: bundled and manual.

- Bundled load balancer: GKE on Bare Metal deploys an L4 load balancer during the cluster installation. There is no need to deploy an external load balancer.
- Manual load balancer: GKE on Bare Metal does not deploy load balancers. When using this method, you must deploy and configure a load balancer before installing the cluster.

See https://cloud.google.com/anthos/clusters/docs/bare-metal/latest/installing/load-balance for more details.

In this validation we used the bundled load balancer mode.

Configure operating systems on the cluster nodes

Select and configure the base operating systems on the node machines that will be used for GKE on Bare Metal.

Follow the steps to configure the selected operating system as indicated at https://cloud.google.com/anthos/clusters/docs/bare-metal/latest/installing/os-reqs.

After all the node machines have been configured, make sure to set up root SSH access to all the nodes. See https://cloud.google.com/anthos/clusters/docs/bare-metal/latest/installing/workstation-prerequisites#set_up_root_ssh_access_to_nodes for details.

Deploy GKE on Hitachi UCP

As indicated earlier, GKE on Bare Metal can be deployed in multiple types of clusters: admin, user, standalone or hybrid. Make sure to choose the deployment model that meet your needs.

In this validation we used a hybrid type deployment. To deploy the admin workstation, follow these steps.

Procedure

1. Log in to the admin workstation and change to the `baremetal` directory created when installing `bmctl`.

   Note: If the JSON key files for the service accounts were created separately, make sure to download and copy to the `baremetal` directory.

2. Log in to Google Cloud using the credentials from your organization.

   ```
   gcloud auth application-default login
   ```

3. Create a cluster config file using the `bmctl` command. The following command will create a cluster config file inside the `bmctl-workspace/<CLUSTER_NAME>` directory:

   ```
   bmctl create config -c CLUSTER_NAME
   ```

4. Modify cluster config file with the parameters specific to your environment.

   See https://cloud.google.com/anthos/clusters/docs/bare-metal/latest/installing/creating-clusters/hybrid-cluster-creation#edit_the_cluster_config_file for details about editing the cluster config file.

   See https://cloud.google.com/anthos/clusters/docs/bare-metal/latest/reference/config-samples for cluster configuration examples.

5. Create the cluster using your cluster config. Ensure you are in the `baremetal` directory:

   ```
   bmctl create cluster -c CLUSTER_NAME
   ```

   The `bmctl` will perform a series of preflight checks based on the cluster config file and if all the checks are successful, the cluster will be created. If there are any issues, check the logs which are located in the `bmctl-workspace/<CLUSTER_NAME>/log` directory.

   After the cluster is successfully created, you can use `kubectl` to see details of the new cluster. During cluster creation, the `bmctl` generates a kubeconfig file for the cluster that can be used with `kubectl` to interact with the cluster. This `kubeconfig` file is written to `bmctl-workspace/<CLUSTER_NAME>/CLUSTER_NAME-kubeconfig`.

   Here is an example:

   ```
   [root@jp-gke-admin-bm baremetal]# kubectl --kubeconfig bmctl-workspace/user-
   cluster1/user-cluster1-kubeconfig get nodes
   NAME               STATUS   ROLES                  AGE   VERSION
   ucpbm-gke-admin1   Ready    control-plane,master   35d   v1.25.6-gke.1000
   ucpbm-gke-wnode1   Ready    worker                 35d   v1.25.6-gke.1000
   ```

```
ucpbm-gke-wnode2   Ready    worker                 35d   v1.25.6-gke.1000
ucpbm-gke-wnode3   Ready    worker                 35d   v1.25.6-gke.1000
ucpbm-gke-wnode4   Ready    worker                 35d   v1.25.6-gke.1000
[root@jp-gke-admin-bm baremetal]#
```

Version 1.14.3 was the latest available at the time of this validation.

Set up storage using Hitachi Storage Plug-in for Containers

Hitachi Storage Plug-in for Containers can be easily deployed to GKE on Bare Metal using the following steps:

- Install Hitachi Storage Plug-in for Containers
- Configure Secret settings to access Hitachi VSP Storage system
- Configure StorageClass settings
- Configure Multipathing (Fibre Channel or iSCSI)

> **Note:** If there is a previous version of Storage Plug-in for Containers, remove it before performing the installation procedure.

Installing Storage Plug-in for Containers on GKE on Bare Metal

To install Hitachi Storage Plug-in for Containers using the Operator, perform the steps described in the "Installation on Kubernetes" section on https://knowledge.hitachivantara.com/Documents/Adapters_and_Drivers/Storage_Adapters_and_Drivers/Containers/Storage_Plug-in_for_Containers. Here some of these steps:

Procedure

1. Download the Storage Plug-in for Containers package from Hitachi Support Connect.
2. Copy and extract the Storage Plug-in for Containers package to the GKE on Bare Metal admin workstation and move it to the directory `yaml/operator`.
3. Create the namespace for the Operator.

```
kubectl create -f hspc-operator-namespace.yaml
```

4. Create a secret to get the container images from Red Hat registry.

```
kubectl create secret docker-registry regcred-redhat-com \
--namespace=hspc-operator-system \
--docker-server=registry.connect.redhat.com \
--docker-username=<user> \
--docker-password=<password>
```

5. Create the Operator and verify that it is running:

```
kubectl get deployment -n hspc-operator-system
NAME                               READY   UP-TO-DATE   AVAILABLE   AGE
hspc-operator-controller-manager   1/1     1            1           24s
```

6. If deploying on the same namespace as the Operator, you need to create one more secret to `registry.redhat.io`. If deploying in a separate namespace, create two secrets on the previous URL. For this validation we installed on the same namespace.

```
kubectl create secret docker-registry regcred-redhat-io \
--namespace=${SPC_NAMESPACE} \
--docker-server=registry.redhat.io \
--docker-username=<user> \
--docker-password=<password>
```

7. Verify that `hspc_v1_hspc.yaml` has the correct namespace, and proceed to deploy Storage Plug-in for Containers with the following command:

```
kubectl create -f hspc_v1_hspc.yaml
```

8. Use the following command verify that all the PODs are in the running state:

```
[root@jp-gke-admin-bm operator]# kubectl get pods -n hspc-operator-system
NAME                                              READY STATUS  RESTARTS   AGE
hspc-csi-controller-69c68db7d5-hgd47                  6/6   Running   0  2m54s
hspc-csi-node-7jxgp                                   2/2   Running   0  2m54s
hspc-csi-node-bdjnt                                   2/2   Running   0  2m54s
hspc-csi-node-lw4mw                                   2/2   Running   0  2m54s
hspc-csi-node-stl48                                   2/2   Running   0  2m54s
hspc-operator-controller-manager-6b8684c94d-c8sn7 1/1   Running   0  69m
```

Result

Hitachi Storage Plug-in for Containers has been successfully installed. If you want to make an advanced configuration, see Configuration of Storage Plug-in for Containers.

Configure secret

The secret contains storage system information that enables access to Storage Plug-in for Containers. It contains the storage URL (VSP REST API), user, and password settings. The following is an example of the YAML manifest file:

```
apiVersion: v1
kind: Secret
metadata:
  name: hitachi-vsp-secret
  namespace: ucp-anthos
type: Opaque
data:
  url: aHR0cHM6Ly8xNzIuMjUuNDQuMTE3
  user: dWNwZ2tldXNyMQ==
  password: SGl0YWNoaTEh
```

The URL, user, and password are base64 encoded. The following is an example of how to get the base64 encoded of a user called `ocpusr1`. Do the same for the URL and password:

```
[root@jp-gke-admin-bm baremetal]# echo -n " ucpgkeusr1" | base64
dWNwZ2tldXNyMQ==
```

One way to create a secret is using the following command:

```
kubectl create ns ucp-anthos
kubectl create -f <secret-manifest-file>
```

Configure StorageClass

The StorageClass contains storage settings that are necessary for Storage Plug-in for Containers to work with your environment. The following YAML manifest file provides information about the required parameters for Storage Plug-in for Containers with Hitachi VSP storage:

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
 name: hitachi-vsp-sc
 annotations:
  kubernetes.io/description: Hitachi Storage Plug-in for Containers
provisioner: hspc.csi.hitachi.com
reclaimPolicy: Delete
allowVolumeExpansion: true
volumeBindingMode: Immediate
parameters:
 serialNumber: "715021"
 poolID: "2"
 portID: CL5-A,CL6-A
 connectionType: fc csi.storage.k8s.io/fstype: ext4
 csi.storage.k8s.io/provisioner-secret-namespace: "ucp-anthos"
 csi.storage.k8s.io/provisioner-secret-name: "hitachi-vsp-secret"
 csi.storage.k8s.io/node-stage-secret-name: "hitachi-vsp-secret"
 csi.storage.k8s.io/controller-expand-secret-name: "hitachi-vsp-secret"
 csi.storage.k8s.io/node-publish-secret-namespace: "ucp-anthos"
 csi.storage.k8s.io/controller-publish-secret-name: "hitachi-vsp-secret"
 csi.storage.k8s.io/controller-publish-secret-namespace: "ucp-anthos"
 csi.storage.k8s.io/node-publish-secret-name: "hitachi-vsp-secret"
 csi.storage.k8s.io/controller-expand-secret-namespace: "ucp-anthos"
 csi.storage.k8s.io/node-stage-secret-namespace: "ucp-anthos"
```

Here are additional details about some of these parameters:

- serialNumber: VSP serial number
- provisioner: For Storage Plug-in for Containers, the default is hspc.csi.hitachi.com
- poolID: Pool ID on the VSP used to carve dynamically persistent volumes
- portID: VSP storage ports, use a comma separator for multipath
- connectionType: It is the connection type between storage and nodes. Fibre Channel and iSCSI are supported. If blank, Fibre Channel is set.
- fstype: set filesystem type as ext4
- secret-name: define VSP secret name
- secret-namespace: enter the same namespace used for the secret

One way to create a StorageClass is using the following command:

```
kubectl create -f <storage-class-manifest-file>
```

Use the following command to list the storage classes on GKE on Bare Metal clusters:

```
[root@jp-gke-admin-bm baremetal]# kubectl get sc NAME PROVISIONER RECLAIMPOLICY
VOLUMEBINDINGMODE ALLOWVOLUMEEXPANSION AGE anthos-system kubernetes.io/no-provisioner
Delete WaitForFirstConsumer false 38d hitachi-vsp-sc hspc.csi.hitachi.com Delete
Immediate true 37d local-disks kubernetes.io/no-provisioner Delete
WaitForFirstConsumer false 38d local-shared kubernetes.io/no-provisioner Delete
WaitForFirstConsumer false 38d [root@jp-gke-admin-bm baremetal]#
```

Configure Multipathing

For data-plane nodes connected to Hitachi VSP storage using Fibre Channel or iSCSI, it is recommended that multipathing is enabled. The requirement is to create the `multipath.conf` and ensure that the user_friendly_names option is set to yes and the `multipathd.service` is enabled.

Consider the following before applying the multipath configuration:

- For Fibre Channel, ensure that Fibre Channel switches are configured with proper zoning for the compute nodes and Hitachi VSP storage systems are accessible to each other.
- For iSCSI, ensure the Hitachi VSP storage is properly configured for iSCSI and the compute nodes can access the iSCSI targets. Also, for iSCSI, check the Hitachi Storage Plug-in for Containers Release Notes for additional considerations regarding IQN configurations.
- RHEL already includes the device-mapper-multipath package which is required to support multipathing. For solutions with iSCSI, RHEL already has the iSCSI initiator tools installed by default. There is no need to install an additional package, just apply the configurations as indicated in this section.

Here is an example of how to enable multipath on a RHEL node using the mpathconf utility. The following command creates the `/etc/multipath.conf` file:

```
mpathconf --enable --with_multipathd y
```

The multipath.conf file will look like this:

```
[root@ucpbm-gke-wnode2 ~]# cat /etc/multipath.conf
...
defaults {
        user_friendly_names yes
        find_multipaths yes
        enable_foreign "^$"
}
blacklist_exceptions {
        property "(SCSI_IDENT_|ID_WWN)"
}
blacklist {
}
```

To check status of `multipathd.service`, use the following command:

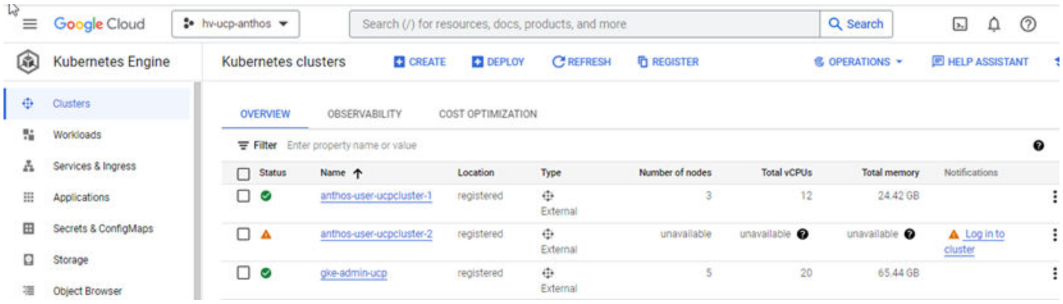
```
systemctl status multipathd.service
```

If you are using iSCSI, make sure the `iscsid.service` is enabled.

Manage clusters from the Google Cloud console

GKE on VMware or GKE on Bare Metal are registered to a fleet with Google Cloud at creation time, and they are displayed in the console in your fleet host project, along with other fleet clusters such as GKE on Google Cloud. All your clusters are displayed on a single dashboard on the “Clusters” page in the console.

The following illustration shows a view of the GKE Enterprise on-prem clusters on the Google Cloud console.



Status	Name	Location	Type	Number of nodes	Total vCPUs	Total memory	Notifications
✓	anthos-user-ucpcluster-1	registered	External	3	12	24.42 GB	
⚠	anthos-user-ucpcluster-2	registered	External	unavailable	unavailable	unavailable	⚠ Log in to cluster
✓	gke-admin-ucp	registered	External	5	20	65.44 GB	

To manage the GKE Enterprise clusters from Google Cloud console, you must set up authentication and grant some specific roles so you can log in to these clusters directly from the Google Cloud console. There are different authentication methods as described in *Manage clusters from the Google Cloud console* at https://cloud.google.com/anthos/clusters/docs/on-prem/latest/how-to/connect-to-cluster-console. For this validation we used the bearer token authentication method.

Complete the following steps to enable access to the GKE Enterprise on-prem clusters.

Procedure

1. Grant IAM roles for access through the Google Cloud console.

   The recommended roles are:

   - roles/container.viewer
   - roles/gkehub.viewer
   - roles/gkeonprem.admin

   In the following example, change the project ID and user's email based on your organization:

   ```
   gcloud projects add-iam-policy-binding hv-ucp-anthos \
       --member="user:jose.perez2@hitachivantara.com" \
       --role=roles/container.viewer
   gcloud projects add-iam-policy-binding hv-ucp-anthos \
       --member="user:jose.perez2@hitachivantara.com" \
       --role=roles/gkehub.viewer
   gcloud projects add-iam-policy-binding hv-ucp-anthos \
         --member="user:jose.perez2@hitachivantara.com" \
         --role=roles/gkeonprem.admin
   ```

2. Configure role-based access control (RBAC).

   a. Create a `cloud-console-reader.yaml` file and apply it to the cluster:

      ```
      cat <<EOF > cloud-console-reader.yaml
      kind: ClusterRole
      apiVersion: rbac.authorization.k8s.io/v1
      metadata:
        name: cloud-console-reader
      rules:
      - apiGroups: [""]
        resources: ["nodes", "persistentvolumes", "pods"]
        verbs: ["get", "list", "watch"]
      - apiGroups: ["storage.k8s.io"]
        resources: ["storageclasses"]
        verbs: ["get", "list", "watch"]
      EOF
      ```

b. Apply this clusterRole to the cluster. Make sure the kubeconfig file corresponds to the cluster to which you want to log in. The following is an example for cluster-2:

```
kubectl apply -f cloud-console-reader.yaml --kubeconfig anthos-user-
ucpcluster-2-kubeconfig
```

c. Create and authorize a Kubernetes service account (KSA):

```
KSA_NAME=KSA_NAME
kubectl create serviceaccount ${KSA_NAME}
kubectl create clusterrolebinding VIEW_BINDING_NAME \
   --clusterrole view --serviceaccount default:${KSA_NAME}
kubectl create clusterrolebinding CLOUD_CONSOLE_READER_BINDING_NAME \
   --clusterrole cloud-console-reader --serviceaccount default:${KSA_NAME}
```

The following is an example for cluster-2 created in the previous steps:

```
KSA_NAME=ucp-gke-user01
kubectl create serviceaccount ${KSA_NAME} --kubeconfig anthos-user-
ucpcluster-2-kubeconfig

kubectl create clusterrolebinding ucp-gke-user-view \
   --clusterrole view --serviceaccount default:${KSA_NAME} --kubeconfig
anthos-user-ucpcluster-2-kubeconfig

kubectl create clusterrolebinding ucp-gke-user-cloudconsole-reader \
   --clusterrole cloud-console-reader --serviceaccount default:${KSA_NAME} -
-kubeconfig anthos-user-ucpcluster-2-kubeconfig
```

If admin permissions are needed, such as deploying a Kubernetes application from Cloud Marketplace, bind the cluster-admin role to the KSA:

```
kubectl create clusterrolebinding ucp-gke-user-admin \
   --clusterrole cluster-admin --serviceaccount default:${KSA_NAME} --
kubeconfig anthos-user-ucpcluster-2-kubeconfig
```

3. After the service accounts and role bindings have been created, retrieve the KSA’s bearer token with the following commands:

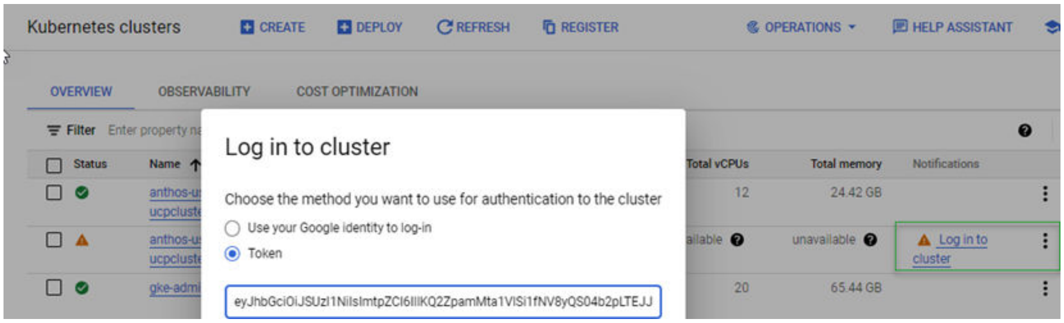
```
SECRET_NAME=${KSA_NAME}-token

kubectl apply -f - << __EOF__
apiVersion: v1
kind: Secret
metadata:
  name: "${SECRET_NAME}"
  annotations:
    kubernetes.io/service-account.name: "${KSA_NAME}"
type: kubernetes.io/service-account-token
__EOF__
```

```
until [[ $(kubectl get -o=jsonpath="{.data.token}" "secret/${SECRET_NAME}") ]];
do
  echo "waiting for token..." >&2;
  sleep 1;
done

kubectl get secret ${SECRET_NAME} -o jsonpath='{$.data.token}' | base64 --decode
```

The following example shows how to get the bearer token for ucpcluster-1:

```
ubuntu@gke-admin-ws-221202-142644:~$ SECRET_NAME=ucp-gke-user01-token
ubuntu@gke-admin-ws-221202-142644:~$ kubectl get secret ${SECRET_NAME} --
kubeconfig anthos-user-ucpcluster-2-kubeconfig -o jsonpath='{$.data.token}' |
base64 --decode
eyJhbGciOiJSUzI1NiIsImtpZCI6IkU2UzhSc1lYUUR1dnplQmkxZnlYWlhPRjNLVjRMRnI2R2M0VjdOY
zBJRkEifQ.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZXRlcy5pby9zZXJ
2aWNlYWNjb3VudC9uYW1lc3BhY2UiOiJkZWZhdWx0Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3Vu
dC9zZWNyZXQubmFtZSI6InVjcC1na2UtdXNlcjAxLXRva2VuIiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlY
WNjb3VudC9zZXJ2aWNlLWFjY291bnQubmFtZSI6InVjcC1na2UtdXNlcjAxIiwia3ViZXJuZXRlcy5pby
9zZXJ2aWNlYWNjb3VudC9zZXJ2aWNlLWFjY291bnQudWlkIjoiMDc1M2Q2NTItMDg0OC00YmQ3LTgzMjA
tYTJkOWNhMjk5YWFjIiwic3ViIjoic3lzdGVtOnNlcnZpY2VhY2NvdW50OmRlZmF1bHQ6dWNwLWdrZS11
c2VyMDEifQ.e7ePsPTLh2hPQrqUKis06VQOT9Vi6wUTUeYYucv0rObyJgqUbQrrLYs68C8Bvb9pUbYGCs
e0RsJwO9xEAqyesUcuxtBH41TaJreWjgAB-25M7ZCXA0GM-igUcKvGf7JROcvq5QTz1Hbl9-
4h6G7uvLLnDU2lDlrVyNcOX6rbi3sH6duVGS0Di-
PX3MMFeXMz3NtJfCoddl5ZCetHzZV1TJVqKDjJ2U0qhsT003x6vlBuzqZDVlAqcDfnp_Tz2auwh0od4uQ
IbgJ_8jQp0FvgPcwiC--qB7etNUiAMYfmh9V6AwPpFvZiAHD7h5UuivBvW--AGGubVY6dWW_ROhU2aQ
```

4. Copy the token and save so it can be used to log in to the Google Cloud console. In the Google Cloud console, on the GKE Clusters page, click the 3 dots next to the registered cluster and click **Log in**, select **Token**, enter the token obtained in the previous step, and then click **Login**.



Log in to cluster

Choose the method you want to use for authentication to the cluster

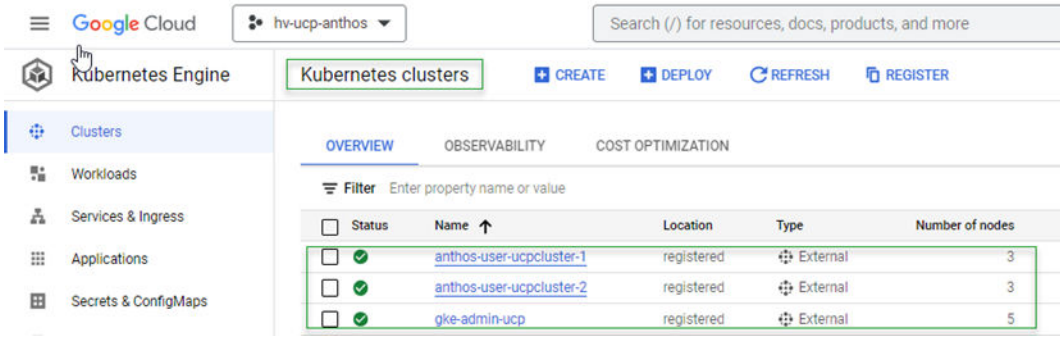
○ Use your Google identity to log-in

● Token

eyJhbGciOiJSUzI1NiIsImtpZCI6IlIKQ2ZpamMta1VISi1fNV8yQS04b2pLTEJJ

5. Repeat this process for each of the GKE Enterprise clusters you want to manage from the Google Cloud console.

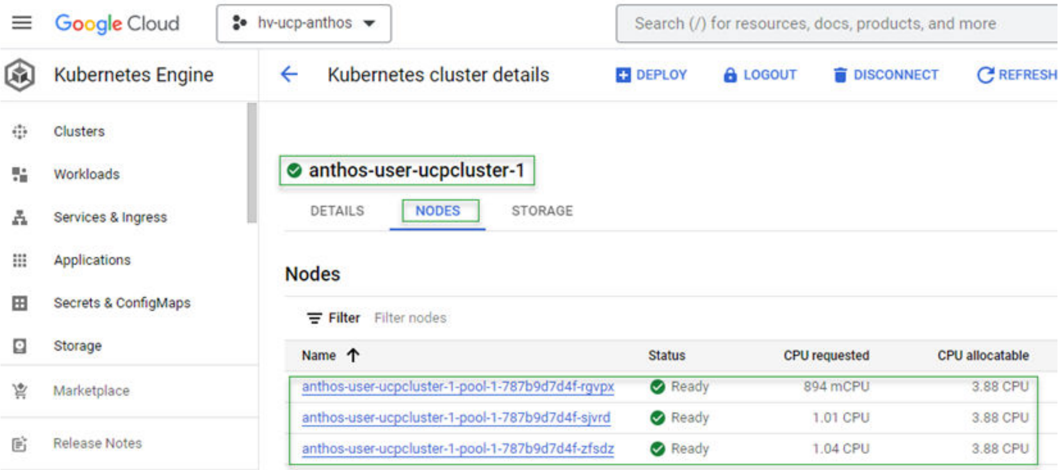
   The following is a view of the GKE Enterprise on-prem clusters (GKE on VMware) running as vSphere VMs on Hitachi UCP platform, all with green check marks. Hitachi UCP platform, all with green check marks.



Kubernetes clusters

Status	Name ↑	Location	Type	Number of nodes
✓	anthos-user-ucpcluster-1	registered	External	3
✓	anthos-user-ucpcluster-2	registered	External	3
✓	gke-admin-ucp	registered	External	5

6. Click a specific cluster to display additional information about the cluster such as cluster nodes, Kubernetes version, storage classes, and persistent volumes.

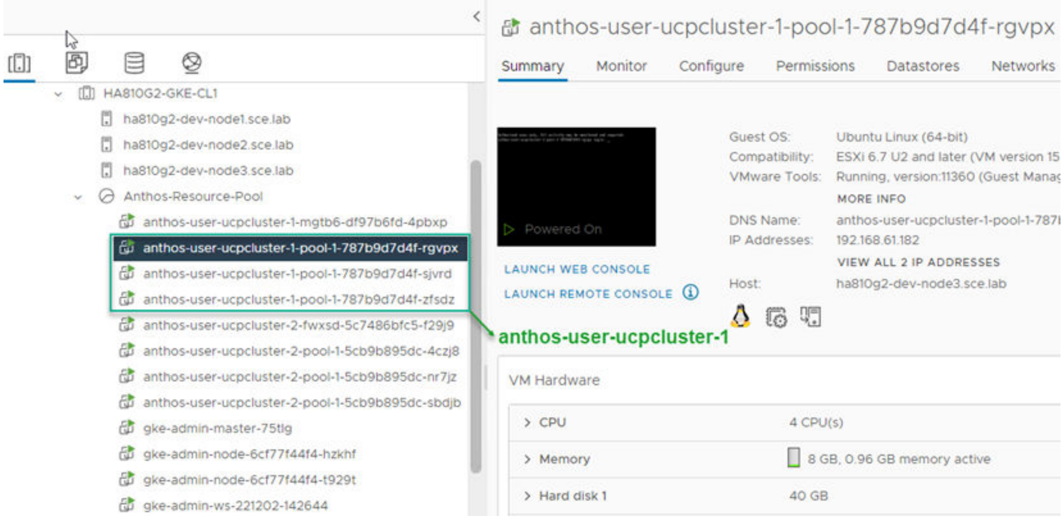


anthos-user-ucpcluster-1

Nodes

Name ↑	Status	CPU requested	CPU allocatable
anthos-user-ucpcluster-1-pool-1-787b9d7d4f-rgvpx	✓ Ready	894 mCPU	3.88 CPU
anthos-user-ucpcluster-1-pool-1-787b9d7d4f-sjvrd	✓ Ready	1.01 CPU	3.88 CPU
anthos-user-ucpcluster-1-pool-1-787b9d7d4f-zfsdz	✓ Ready	1.04 CPU	3.88 CPU

The same cluster data-plane nodes can be seen with the same name on the vCenter/UCP environment.



HA810G2-GKE-CL1
ha810g2-dev-node1.sce.lab
ha810g2-dev-node2.sce.lab
ha810g2-dev-node3.sce.lab
Anthos-Resource-Pool
anthos-user-ucpcluster-1-mgtb6-df97b6fd-4pbxp
anthos-user-ucpcluster-1-pool-1-787b9d7d4f-rgvpx
anthos-user-ucpcluster-1-pool-1-787b9d7d4f-sjvrd
anthos-user-ucpcluster-1-pool-1-787b9d7d4f-zfsdz
anthos-user-ucpcluster-2-fwxsd-5c7486bfc5-f29j9
anthos-user-ucpcluster-2-pool-1-5cb9b895dc-4czj8
anthos-user-ucpcluster-2-pool-1-5cb9b895dc-nr7jz
anthos-user-ucpcluster-2-pool-1-5cb9b895dc-sbdjb
gke-admin-master-75tlg
gke-admin-node-6cf77f44f4-hzkhf
gke-admin-node-6cf77f44f4-t929t
gke-admin-ws-221202-142644
Summary
Monitor
Configure
Permissions
Datastores
Networks
Powered On
LAUNCH WEB CONSOLE
LAUNCH REMOTE CONSOLE
MORE INFO
VIEW ALL 2 IP ADDRESSES
anthos-user-ucpcluster-1

anthos-user-ucpcluster-1-pool-1-787b9d7d4f-rgvpx

Guest OS: Ubuntu Linux (64-bit)
Compatibility: ESXi 6.7 U2 and later (VM version 15
VMware Tools: Running, version:11360 (Guest Manag
DNS Name: anthos-user-ucpcluster-1-pool-1-787l
IP Addresses: 192.168.61.182
Host: ha810g2-dev-node3.sce.lab

VM Hardware

CPU	4 CPU(s)
Memory	8 GB, 0.96 GB memory active
Hard disk 1	40 GB

Solution validation

If you have followed the guidance in the Solution Design section, your infrastructure is prepared, and you can try these example deployments. This reference architecture was validated by the following:

- Deploying stateful applications on GKE Enterprise on-prem clusters, deployed on a Hitachi UCP platform, using the Google Cloud console and Google Cloud Marketplace.
- Connecting and registering an existing Red Hat OpenShift cluster deployed on Hitachi UCP platform, leveraging the GKE Enterprise attached cluster features to demonstrate how easy it is to connect, register any Kubernetes cluster running anywhere, and manage from a single-pane-of-glass using the Google Cloud console.

Deploy applications on UCP for GKE Enterprise

After the GKE Enterprise on-prem clusters have been deployed and registered, you can start deploying workloads using the Google Cloud console or the command line.

Google Cloud Marketplace is a catalog of curated container applications that you can use for easy deployment to your GKE Enterprise clusters running anywhere.

Deploy a multi-instance of MariaDB with Persistent Volumes on Hitachi Virtual Storage Platform

This example deploys a stateful multi-instance MariaDB with replication. The deployment includes two StatefulSets, a primary (read/write access), and a secondary (read-only access).

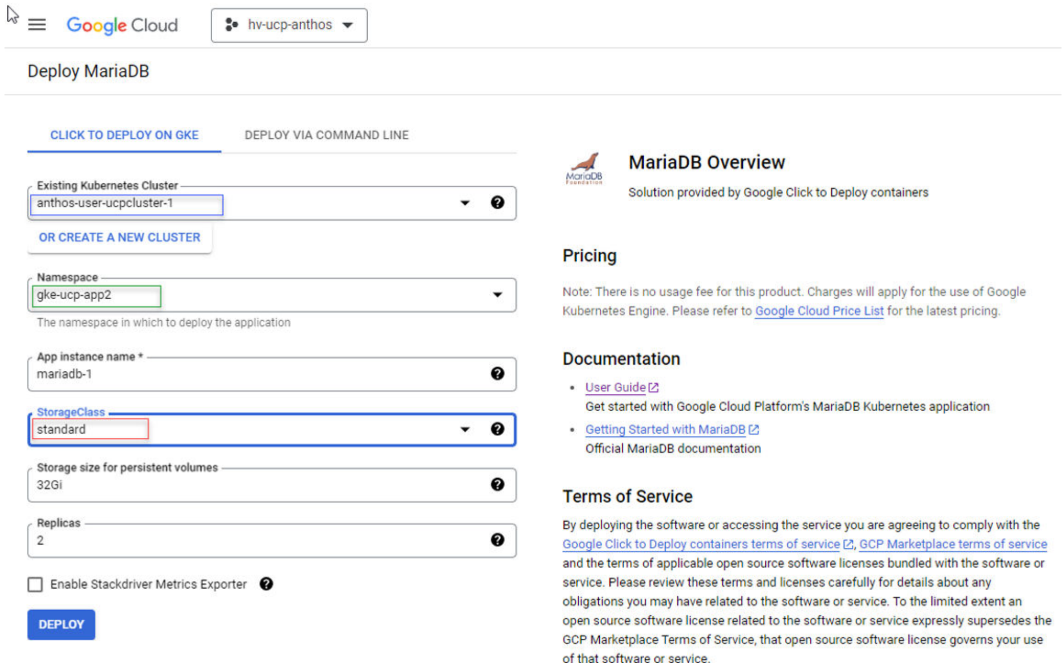
As indicated previously, we can deploy an application with a few clicks from the Google Cloud console using Google Cloud Marketplace.

The following example shows how to quickly deploy MariaDB using Google Cloud console.

Procedure

1. On the Google Cloud console, click **Marketplace** and then select **Kubernetes Apps**.

   You will be presented with a list of available applications that are ready to deploy.
2. Click the MariaDB app.
3. Click **Configure** and do the following:
   a. Select the cluster.
   b. Select or create a namespace.
   c. Enter the app instance name.
   d. Select the storage class.
   e. Enter the capacity for the persistent volumes and number of replicas.



Deploy MariaDB

Existing Kubernetes Cluster: anthos-user-ucpcluster-1

Namespace: gke-ucp-app2

The namespace in which to deploy the application

App instance name *: mariadb-1

StorageClass: standard

Storage size for persistent volumes: 32Gi

Replicas: 2

☐ Enable Stackdriver Metrics Exporter

MariaDB Overview

Solution provided by Google Click to Deploy containers

Pricing

Note: There is no usage fee for this product. Charges will apply for the use of Google Kubernetes Engine. Please refer to Google Cloud Price List for the latest pricing.

Documentation

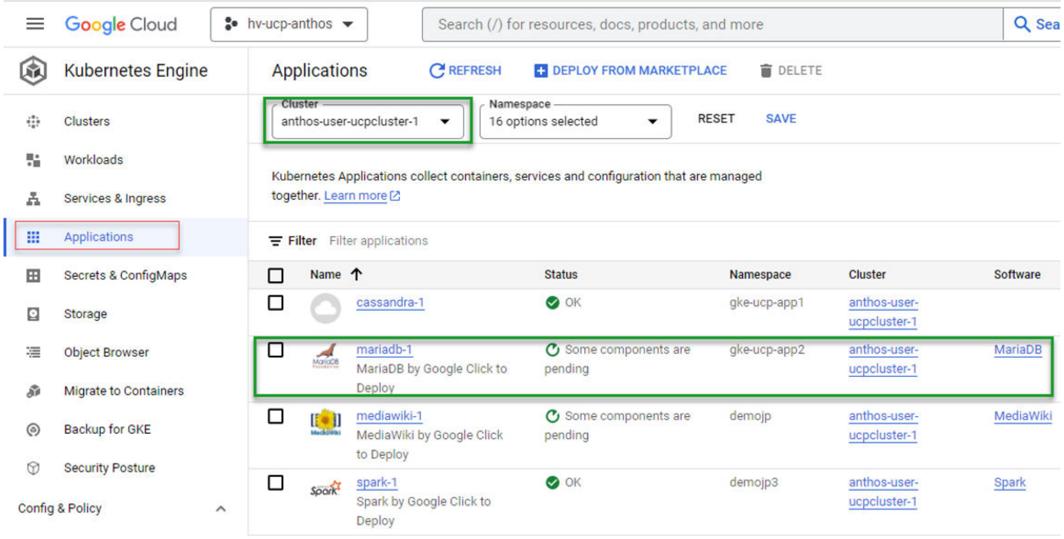
- User Guide
  Get started with Google Cloud Platform's MariaDB Kubernetes application
- Getting Started with MariaDB
  Official MariaDB documentation

Terms of Service

By deploying the software or accessing the service you are agreeing to comply with the Google Click to Deploy containers terms of service, GCP Marketplace terms of service and the terms of applicable open source software licenses bundled with the software or service. Please review these terms and licenses carefully for details about any obligations you may have related to the software or service. To the limited extent an open source software license related to the software or service expressly supersedes the GCP Marketplace Terms of Service, that open source software license governs your use of that software or service.

4. Click **Deploy** to start the deployment process.
5. After the deployment is complete, click Applications to verify, and filter by cluster if needed.

   The following illustration shows the new MariaDB app deployed into the `Anthos-user-ucpcluster-1 cluster`.



Cluster: anthos-user-ucpcluster-1

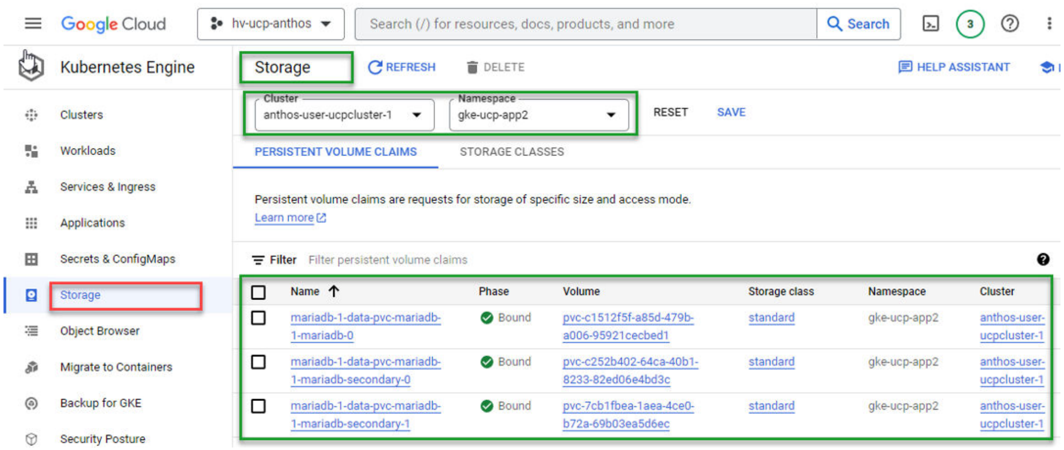
Namespace: 16 options selected

Kubernetes Applications collect containers, services and configuration that are managed together. Learn more

☐	Name ↑	Status	Namespace	Cluster	Software
☐	cassandra-1	✓ OK	gke-ucp-app1	anthos-user-ucpcluster-1	
☐	mariadb-1 MariaDB by Google Click to Deploy	Some components are pending	gke-ucp-app2	anthos-user-ucpcluster-1	MariaDB
☐	mediawiki-1 MediaWiki by Google Click to Deploy	Some components are pending	demojp	anthos-user-ucpcluster-1	MediaWiki
☐	spark-1 Spark by Google Click to Deploy	✓ OK	demojp3	anthos-user-ucpcluster-1	Spark

6. To see the persistent volumes claims you can select Storage and then filter by cluster and even namespace.

   In the following illustration you can see the PVCs/PVs using the StorageClass standard, which is backed by a VMFS datastore on the Hitachi UCP cluster using Hitachi Virtual Storage Platform.



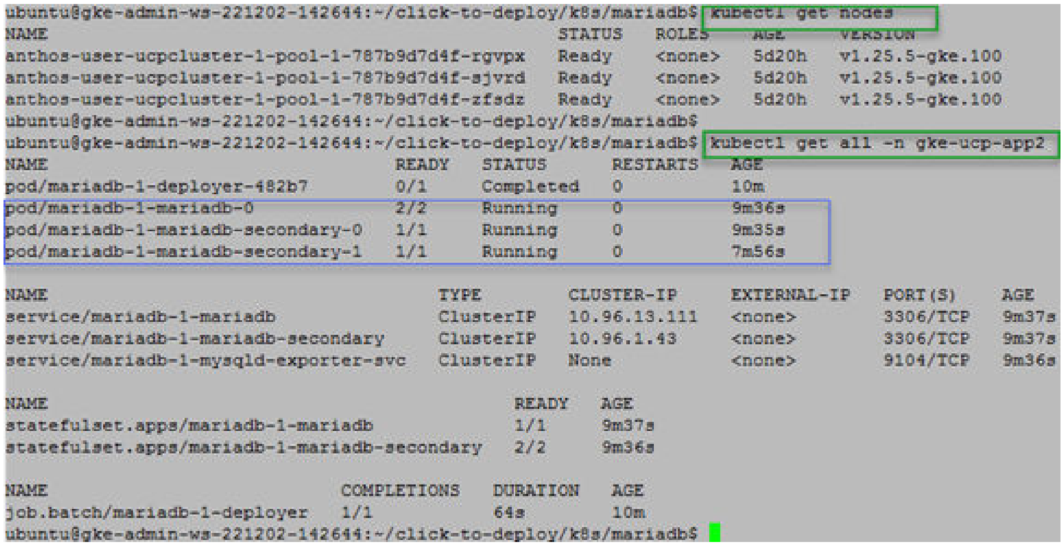
Cluster: anthos-user-ucpcluster-1

Namespace: gke-ucp-app2

Persistent volume claims are requests for storage of specific size and access mode.
Learn more

☐	Name ↑	Phase	Volume	Storage class	Namespace	Cluster
☐	mariadb-1-data-pvc-mariadb-1-mariadb-0	✓ Bound	pvc-c1512f5f-a85d-479b-a006-95921cecbed1	standard	gke-ucp-app2	anthos-user-ucpcluster-1
☐	mariadb-1-data-pvc-mariadb-1-mariadb-secondary-0	✓ Bound	pvc-c252b402-64ca-40b1-8233-82ed06e4bd3c	standard	gke-ucp-app2	anthos-user-ucpcluster-1
☐	mariadb-1-data-pvc-mariadb-1-mariadb-secondary-1	✓ Bound	pvc-7cb1fbea-1aea-4ce0-b72a-69b03ea5d6ec	standard	gke-ucp-app2	anthos-user-ucpcluster-1

You can also see these details from the command line using the `kubectl` tool.



```
ubuntu@gke-admin-ws-221202-142644:~/click-to-deploy/k8s/mariadb$ kubectl get nodes
NAME                                           STATUS   ROLES    AGE    VERSION
anthos-user-ucpcluster-1-pool-1-787b9d7d4f-rgvpx   Ready    <none>   5d20h   v1.25.5-gke.100
anthos-user-ucpcluster-1-pool-1-787b9d7d4f-sjvrd   Ready    <none>   5d20h   v1.25.5-gke.100
anthos-user-ucpcluster-1-pool-1-787b9d7d4f-zfsdz   Ready    <none>   5d20h   v1.25.5-gke.100
ubuntu@gke-admin-ws-221202-142644:~/click-to-deploy/k8s/mariadb$
ubuntu@gke-admin-ws-221202-142644:~/click-to-deploy/k8s/mariadb$ kubectl get all -n gke-ucp-app2
NAME                                 READY   STATUS      RESTARTS   AGE
pod/mariadb-1-deployer-482b7         0/1     Completed   0          10m
pod/mariadb-1-mariadb-0              2/2     Running     0          9m36s
pod/mariadb-1-mariadb-secondary-0    1/1     Running     0          9m35s
pod/mariadb-1-mariadb-secondary-1    1/1     Running     0          7m56s

NAME                                   TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
service/mariadb-1-mariadb              ClusterIP   10.96.13.111    <none>        3306/TCP   9m37s
service/mariadb-1-mariadb-secondary    ClusterIP   10.96.1.43      <none>        3306/TCP   9m37s
service/mariadb-1-mysqld-exporter-svc  ClusterIP   None            <none>        9104/TCP   9m36s

NAME                                           READY   AGE
statefulset.apps/mariadb-1-mariadb             1/1     9m37s
statefulset.apps/mariadb-1-mariadb-secondary   2/2     9m36s

NAME                           COMPLETIONS   DURATION   AGE
job.batch/mariadb-1-deployer   1/1           64s        10m
ubuntu@gke-admin-ws-221202-142644:~/click-to-deploy/k8s/mariadb$
```

A similar process can be used to deploy stateful applications on GKE on Bare Metal using Hitachi Storage Plug-in for Containers and Hitachi VSP storage.

Connect and manage an on-prem OCP cluster with GKE Enterprise and Google Cloud

GKE on VMware, GKE on Bare Metal, and multi-cloud GKE (AWS and Azure) are automatically registered to your project fleet on Google Cloud when they are created. However, GKE clusters on Google Cloud, EKS clusters (AWS), AKS clusters (Azure), and other third-party Kubernetes clusters (also called attached clusters) must be manually registered to join your project fleet on Google Cloud.

Use the GKE Enterprise attached cluster feature to manage any standard, Cloud Native Computing Foundation (CNCF) compliant Kubernetes cluster from the Google Cloud console, across multiple cloud providers, along with your GKE Enterprise clusters, and enable GKE Enterprise features such as centralized configuration control with Config Management and Microservices with Anthos Service Mesh.

The following is a summary of the steps required to register third party Kubernetes clusters into your project fleet on Google Cloud. For specific details see *GKE Enterprise (Anthos attached clusters)* at https://cloud.google.com/anthos/clusters/docs/multi-cloud/attached.

Procedure

1. Download and install Google Cloud CLI, and then use `gcloud` for registration.
2. Install `kubectl`.

   The recommendation is to install `kubectl` with Google Cloud CLI.
3. Enable APIs.

   The following APIs are required to be enabled in your fleet host project:

   - `container.googleapis.com`
   - `gkeconnect.googleapis.com`
   - `gkehub.googleapis.com`, also known as the Fleet API. This is the Google Cloud service that manages cluster registration and fleet membership.
   - `cloudresourcemanager.googleapis.com`
4. Grant access permissions.

   Cluster registration requires both permission to register the cluster, and admin permissions on the cluster itself.
5. Create a Google Cloud service account and create a JSON key file that contains the service account credentials. Make sure to bind the corresponding roles.
6. For Red Hat OpenShift, create a custom Security Context Constraints (SCCs) before registering the cluster to allow installing Connect Agent in your OCP cluster.
7. To register the third-party cluster, run the following command:

   ```
   gcloud container hub memberships register [MEMEBERSHIP_NAME] \
                   --context=[CLUSTER_CONTEXT] \
                   --service-account-key-file=[LOCAL_KEY_PATH] \
                   --kubeconfig=[KUBECONFIG_PATH] \
                   --project=[PROJECT_ID}
   ```

   Replace the following:

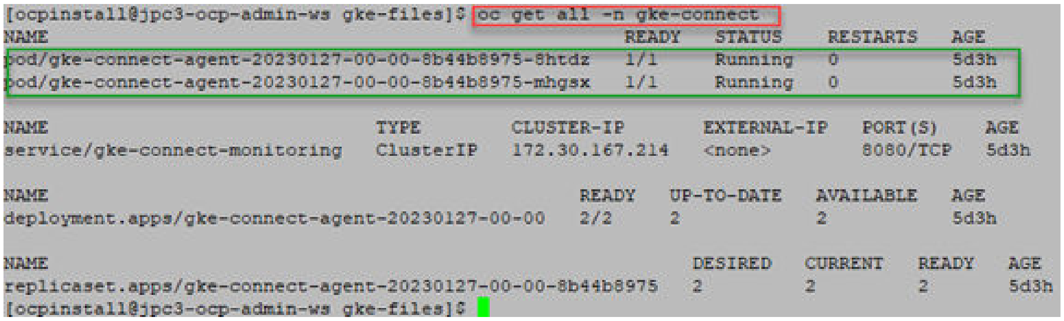
   - MEMBERSHIP_NAME: the name that you choose for your cluster being registered to the fleet.
   - SERVICE_ACCOUNT_KEY_PATH: the local file path to the service account's downloaded private key JSON file.
   - KUBECONFIG_CONTEXT: the cluster context of the cluster being registered as it appears in the kubeconfig file.
   - KUBECONFIG_PATH: the local file path where your kubeconfig containing an entry for the cluster being registered is stored.

The following is an example of the registration of an on-prem OCP cluster deployed on top of Hitachi UCP:

```
[ocpinstall@jpc3-ocp-admin-ws gke-files]$ gcloud container hub memberships
register hitachi-ucp-ocp-onpremcluster1 \ > --context=default/api-jpc3-ocp-hvlab-
local:6443/cluster_admin \ > --service-account-key-file=/home/ocpinstall/gke-
files/connect-register-key.json \ > --kubeconfig=/home/ocpinstall/ocp-upi-
install/auth/kubeconfig \ > --project=hv-ucp-anthos Waiting for membership to be
created...done. Created a new membership [projects/hv-ucp-anthos/locations/
global/memberships/hitachi-ucp-ocp-onpremcluster1] for the cluster [hitachi-ucp-
ocp-onpremcluster1] Generating the Connect Agent manifest... Deploying the
Connect Agent on cluster [hitachi-ucp-ocp-onpremcluster1] in namespace [gke-
connect]... Deleting namespace [gke-connect] in the cluster...done. Deployed the
Connect Agent on cluster [hitachi-ucp-ocp-onpremcluster1] in namespace [gke-
connect]. Finished registering the cluster [hitachi-ucp-ocp-onpremcluster1] with
the fleet. [ocpinstall@jpc3-ocp-admin-ws gke-files]$
```

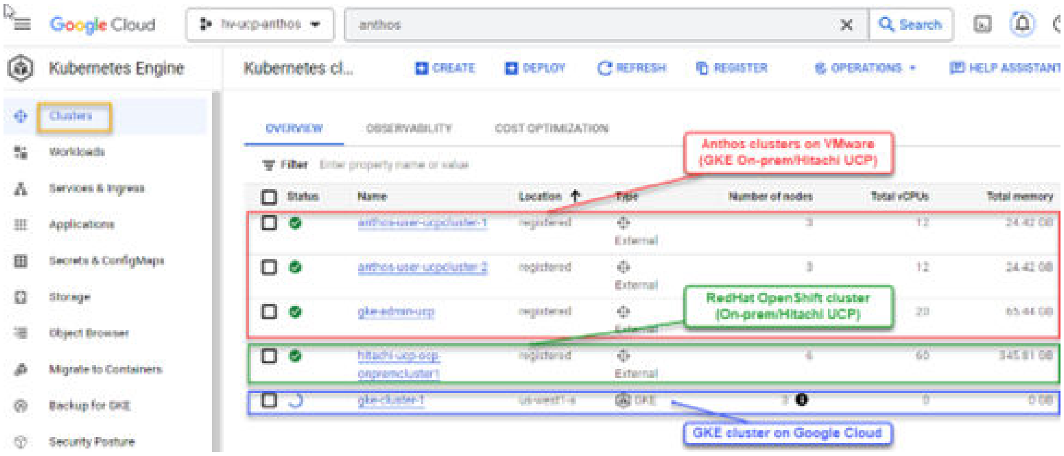
8. After registration, verify that the Connect Agent is running on the namespace `gke-connect`:

```
oc get all -n gke-connect
```



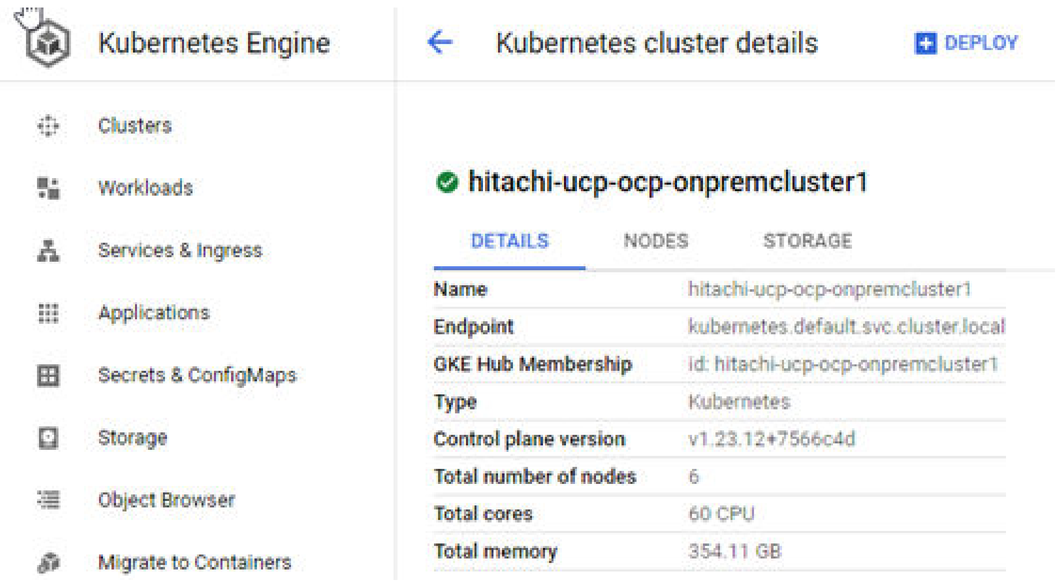
```
[ocpinstall@jpc3-ocp-admin-ws gke-files]$ oc get all -n gke-connect
NAME                                                 READY   STATUS    RESTARTS   AGE
pod/gke-connect-agent-20230127-00-00-8b44b8975-8htdz   1/1     Running   0          5d3h
pod/gke-connect-agent-20230127-00-00-8b44b8975-mhgsx   1/1     Running   0          5d3h

NAME                             TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)    AGE
service/gke-connect-monitoring   ClusterIP   172.30.167.214   <none>        8080/TCP   5d3h

NAME                                               READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/gke-connect-agent-20230127-00-00   2/2     2            2           5d3h

NAME                                                          DESIRED   CURRENT   READY   AGE
replicaset.apps/gke-connect-agent-20230127-00-00-8b44b8975   2         2         2       5d3h
[ocpinstall@jpc3-ocp-admin-ws gke-files]$
```

After the registration is complete, your cluster will appear in the GKE and GKE Enterprise cluster pages in the Google Cloud console, with the other GKE Enterprise clusters, as shown.



Status	Name	Location	Type	Number of nodes	Total vCPUs	Total memory
✓	anthos-user-ucpcluster-1	registered	External	3	12	24.42 GB
✓	anthos-user-ucpcluster-2	registered	External	3	12	24.42 GB
✓	gke-admin-ucp	registered	External		20	65.44 GB
✓	hitachi-ucp-ocp-onpremcluster1	registered	External	4	60	345.81 GB
	gke-cluster-1	us-west1-a	GKE	3	0	0 GB

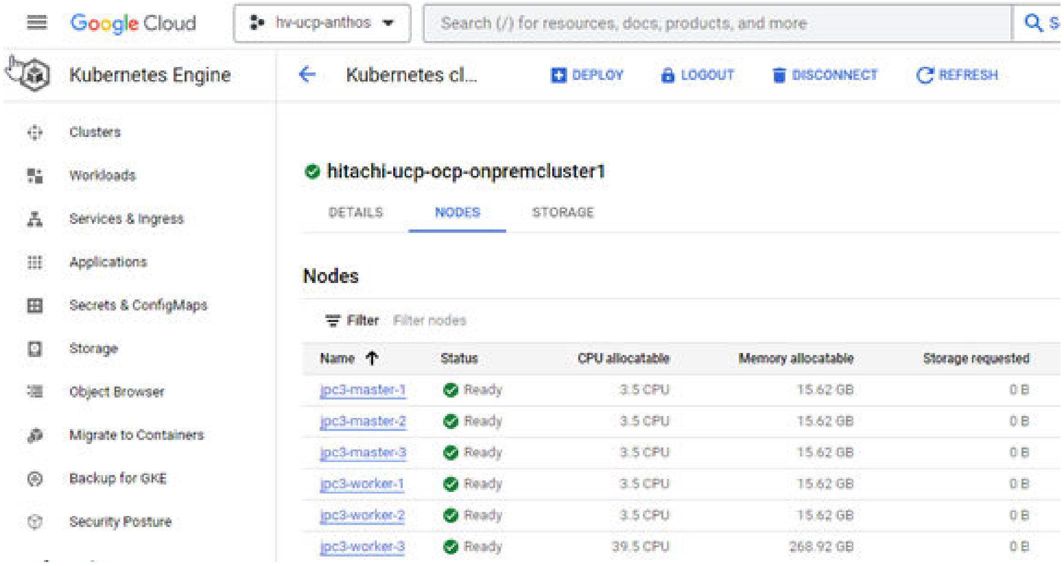
From here we can see more details of the cluster:



hitachi-ucp-ocp-onpremcluster1

Name	hitachi-ucp-ocp-onpremcluster1
Endpoint	kubernetes.default.svc.cluster.local
GKE Hub Membership	id: hitachi-ucp-ocp-onpremcluster1
Type	Kubernetes
Control plane version	v1.23.12+7566c4d
Total number of nodes	6
Total cores	60 CPU
Total memory	354.11 GB

The following shows the cluster nodes:

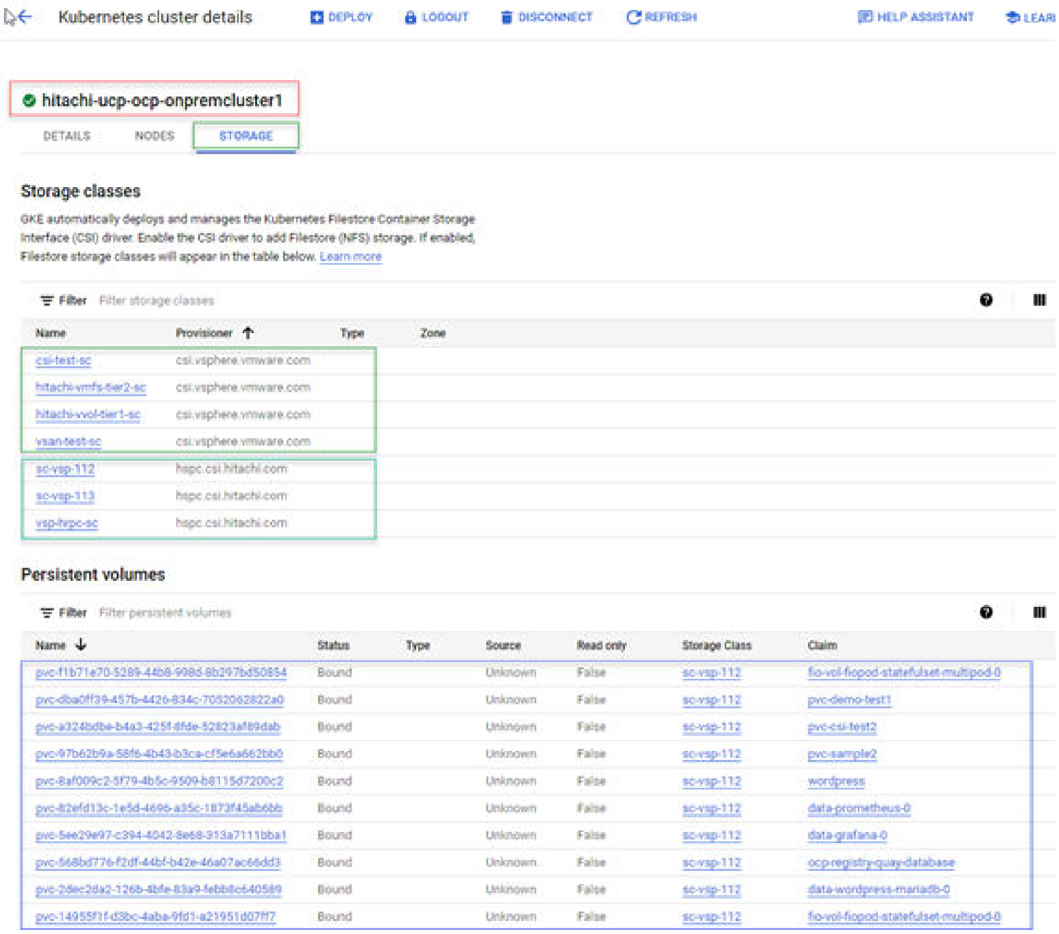


hitachi-ucp-ocp-onpremcluster1

Nodes

Name ↑	Status	CPU allocatable	Memory allocatable	Storage requested
jpc3-master-1	✓ Ready	3.5 CPU	15.62 GB	0 B
jpc3-master-2	✓ Ready	3.5 CPU	15.62 GB	0 B
jpc3-master-3	✓ Ready	3.5 CPU	15.62 GB	0 B
jpc3-worker-1	✓ Ready	3.5 CPU	15.62 GB	0 B
jpc3-worker-2	✓ Ready	3.5 CPU	15.62 GB	0 B
jpc3-worker-3	✓ Ready	39.5 CPU	268.92 GB	0 B

Additional details can be seen from the cluster such as Storage Classes and Persistent Volumes.



Kubernetes cluster details
DEPLOY
LOGOUT
DISCONNECT
REFRESH
HELP ASSISTANT
LEARI
hitachi-ucp-ocp-onpremcluster1
DETAILS
NODES
STORAGE

Storage classes

GKE automatically deploys and manages the Kubernetes Filestore Container Storage Interface (CSI) driver. Enable the CSI driver to add Filestore (NFS) storage. If enabled, Filestore storage classes will appear in the table below. Learn more

Filter Filter storage classes

Name	Provisioner ↑	Type	Zone
csi-test-sc	csi.vsphere.vmware.com		
hitachi-vmfs-tier2-sc	csi.vsphere.vmware.com		
hitachi-vvol-tier1-sc	csi.vsphere.vmware.com		
vsan-test-sc	csi.vsphere.vmware.com		
sc-vsp-112	hspc.csi.hitachi.com		
sc-vsp-113	hspc.csi.hitachi.com		
vsp-hrpc-sc	hspc.csi.hitachi.com		

Persistent volumes

Name ↓	Status	Type	Source	Read only	Storage Class	Claim
pvc-f1b71e70-5289-44b8-998d-8b297bd50854	Bound		Unknown	False	sc-vsp-112	fio-vol-fiopod-statefulset-multipod-0
pvc-dba0ff39-457b-4426-834c-7052062822a0	Bound		Unknown	False	sc-vsp-112	pvc-demo-test1
pvc-a324bdbe-b4a3-425f-8fde-52823af89dab	Bound		Unknown	False	sc-vsp-112	pvc-csi-test2
pvc-97b62b9a-58f6-4b43-b3ca-cf5e6a662bb0	Bound		Unknown	False	sc-vsp-112	pvc-sample2
pvc-8af009c2-3f79-4b5c-9509-b8115d7200c2	Bound		Unknown	False	sc-vsp-112	wordpress
pvc-82efd13c-1e5d-4696-a35c-1873f45ab6bb	Bound		Unknown	False	sc-vsp-112	data-prometheus-0
pvc-5ee29e97-c394-4042-8e68-313a7111bba1	Bound		Unknown	False	sc-vsp-112	data-grafana-0
pvc-568bd776-f2df-44bf-b42e-46a07ac66dd3	Bound		Unknown	False	sc-vsp-112	ocp-registry-quay-database
pvc-2dec2da2-126b-4bfe-83a9-febb8c640589	Bound		Unknown	False	sc-vsp-112	data-wordpress-mariadb-0
pvc-14955f1f-d3bc-4aba-9fd1-a21951d07ff7	Bound		Unknown	False	sc-vsp-112	fio-vol-fiopod-statefulset-multipod-0

Result

At this point you can enable any of the supported features such as GKE Enterprise Service Mesh and Config Management.

Conclusion

This reference architecture validates how Hitachi Unified Compute Platform, Hitachi Virtual Storage Platform, and Google Cloud GKE Enterprise (Anthos) combine to deliver a powerful and flexible Kubernetes platform for a secure and enterprise-ready hybrid multi-cloud solution.

For customers looking to implement an enterprise class hybrid multi-cloud solution, Hitachi UCP for Google GKE Enterprise provides the best platform that can integrate with other cloud providers, leverage existing hardware investments in their own data centers, and manage with a modern hybrid multi-cloud framework through a single pane of glass.

With Hitachi Storage Plug-in for Containers and Hitachi VSP Storage, your organization can dynamically provision and deliver enterprise shared storage for containers that persists beyond the timeline of a single container host.

Product descriptions

This section includes information about the hardware and software components used in this solution.

Unified Compute Platform CI

Hitachi Unified Compute Platform CI (UCP CI) is an optimized and preconfigured converged infrastructure platform. It offers a broad range of compute and storage components that can be scaled and configured independently to eliminate overprovisioning. With Unified Compute Platform CI, you can optimize your data center to run any container application workload, at any scale.

Unified Compute Platform HC

Unified Compute Platform HC (UCP HC) is an integrated turnkey appliance that combines compute, storage, and optional network switching to deliver certainty for edge to core to cloud operations. This market-proven Hitachi solution provides a scalable, seamless, and simplified cloud foundation for enterprise and mid-market customers. Advanced automation and intelligence for day 0-2 operations accelerate innovation and improve productivity while lowering the TCO.

Hitachi Unified Compute Platform RS

To simplify your hybrid cloud journey, Hitachi Unified Compute Platform RS (UCP RS) provides a turnkey solution that reduces total cost of ownership (TCO) and improves security. The software-defined data center solution accelerates the time to market with a natively integrated cloud infrastructure stack. It comes prepackaged with management software, to provide automated, policy-based IT operations.

UCP RS has automation that enables the deployment of an entire cloud infrastructure in hours, not weeks or months. There is rapid and repeatable application deployment.

Move your workload across data centers to meet changing business needs. Manage your applications across private and public cloud from a common toolset. Scale your data center without increasing IT headcount. Automate your data center with policies.

There is a hypervisor-enabled firewall with Unified Compute Platform CI for enhanced security. Granular security prevents east-west breach. Security policies align with workload, regardless of physical location.

Hitachi Virtual Storage Platform E1090

The Hitachi Virtual Storage Platform E1090 (VSP E1090) storage system is a high-performance, large-capacity data storage system. The VSP E1090 all-flash arrays (AFAs) support NVMe and SAS solid-state drives (SSDs). The VSP E1090H hybrid models can be configured with both SSDs and hard disk drives (HDDs).

- The NVMe flash architecture delivers consistent, low-microsecond latency, which reduces the transaction costs of latency-critical applications and delivers predictable performance to optimize storage resources.
- The hybrid architecture allows for greater scalability and provides data-in-place migration support.

Hitachi Advanced Server

Designed to unlock the full benefits of the hybrid cloud, Hitachi Advanced Server models deliver high performance and enhanced security while reducing operational costs. Global enterprises, cloud service providers, and governments trust Hitachi servers to run bare metal, virtualized, or containerized applications. Powered by industry-leading scalable processors, Hitachi servers are ideal to deliver edge, core, and cloud IT services.

Hitachi servers are designed and optimized to maximize performance for VMware, Oracle, Bare Metal, Virtual Desktop Infrastructure (VDI), SAP, analytics, high-performance computing (HPC), and other enterprise workloads.

Cisco Nexus switches

The Cisco Nexus switch product line provides a series of solutions that make it easier to connect and manage disparate data center resources with software-defined networking (SDN). Leveraging the Cisco Unified Fabric, which unifies storage, data and networking (Ethernet/IP) services, the Nexus switches create an open, programmable network foundation built to support a virtualized data center environment.

Brocade switches from Broadcom

Brocade and Hitachi Vantara have partnered to deliver storage networking and data center solutions. These solutions reduce complexity and cost, as well as enable virtualization and cloud computing to increase business agility.

Brocade Fibre Channel switches deliver industry-leading performance, simplifying scale-out network architectures. Get the high-performance, availability, and ease of management you need for a solid foundation to grow the storage network you want.

Hitachi Unified Compute Platform Advisor

Hitachi Unified Compute Platform Advisor (UCP Advisor) is a comprehensive cloud infrastructure management and automation software that enables IT agility and simplifies day 0-N operations for edge, core, and cloud environments. The fourth-generation UCP Advisor accelerates application deployment and drastically simplifies converged and hyperconverged infrastructure deployment, configuration, life cycle management, and ongoing operations with advanced policy-based automation and orchestration for private and hybrid cloud environments.

The centralized management plane enables remote, federated management for the entire portfolio of converged, hyperconverged, and storage data center infrastructure solutions to improve operational efficiency and reduce management complexity. Its intelligent automation services accelerate infrastructure deployment and configuration, significantly minimizing deployment risk and reducing provisioning time and complexity, automating hundreds of mandatory tasks.

UCP Advisor improves predictability with guided lifecycle management capabilities for the complete data center infrastructure stack, including servers and switches from Arista, Brocade, and Cisco, and non-disruptively patches and upgrades infrastructure.

UCP Advisor provides deep integration with VMware management software, improving administrator productivity with intuitive and intelligent operations and automation. It complements VMware vRealize software to further streamline the administration and automation of software-defined data center (SDDC). Automated workflows deliver IT agility using UCP Advisor REST APIs and vRealize Orchestrator and when used with vRealize Automation, enable self- services multi-cloud environments.

It provides comprehensive visibility and monitoring of the infrastructure for collective insight into health and operational efficiency. It automates network configuration operations and system monitoring including generating reports for compliance. UCP Advisor and the integrations with vRealize Log Insight provide rich log analytics and auditability enabling comprehensive visibility of the infrastructure for better resource planning.

Hitachi Vantara

Corporate Headquarters
2535 Augustine Drive
Santa Clara, CA 95054 USA

HitachiVantara.com/contact