Creating HPE Container Solutions Learner Guide

Rev. 21.41

Hewlett Packard
Enterprise

HPE and Red Hat OpenShift Solutions
Module 3

Objectives

This module will help you learn how to design HPE solutions to support Red Hat OpenShift Container Platform (OCP). You will learn some important background information about the OCP architecture. You will then learn about ways that HPE distinguishes itself as the foundation for the OCP platform, as well as best practices for designing the solution.

After completing this module, you will be able to:

- Describe the use cases, features, and architecture of Red Hat OCP
- Explain the advantages of running Red Hat OCP on HPE optimized infrastructure
- Explain how HPE solutions integrate with Red Hat OCP
- Plan and deploy HPE solutions for Red Hat OCP
- Given a set of customer requirements for Red Hat OCP, determine the appropriate software-defined orchestration and automation tools

Red Hat OpenShift Architecture

You will begin by learning more about OpenShift and its architecture.

Red Hat OpenShift Container Platform (OCP)

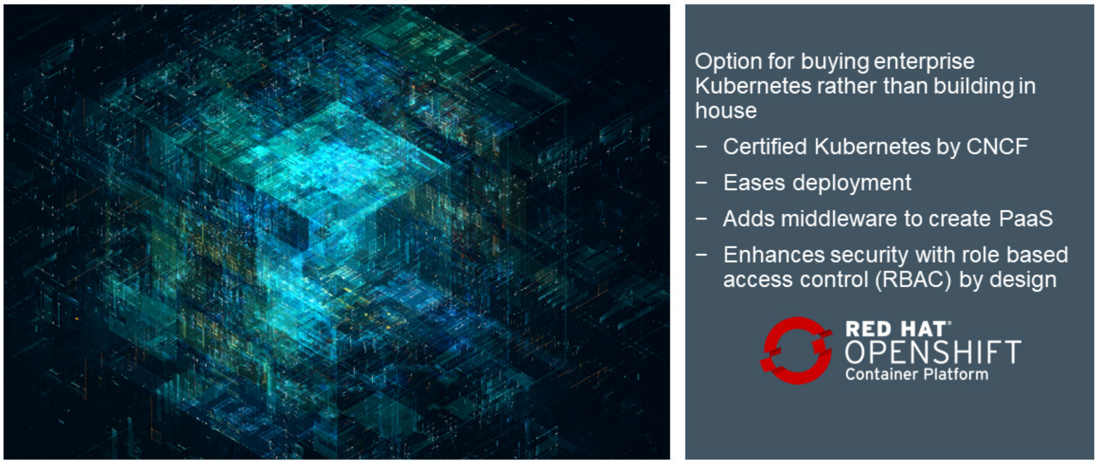

Figure 3-1: Red Hat OpenShift Container Platform (OCP)

Red Hat OpenShift Container Platform (OCP) is a comprehensive container framework solution. Built on the Red Hat Enterprise Linux OS, it delivers an enterprise-class Kubernetes solution to manage containers on that Linux OS. The OCP Kubernetes solution has been certified by the Cloud Native Computing Foundation (CNCF).

OCP eases deployment of the container runtime and Kubernetes components with scripts. It also adds middleware—such as service brokers and automation tools—to create a complete platform as a service (PaaS) solution.

Finally, OCP features role-based access control (RBAC) by design. Many enterprise customers require their container orchestration platform to support RBAC so that they can comply with internal security policies and possibly regulatory requirements.

OCP architecture

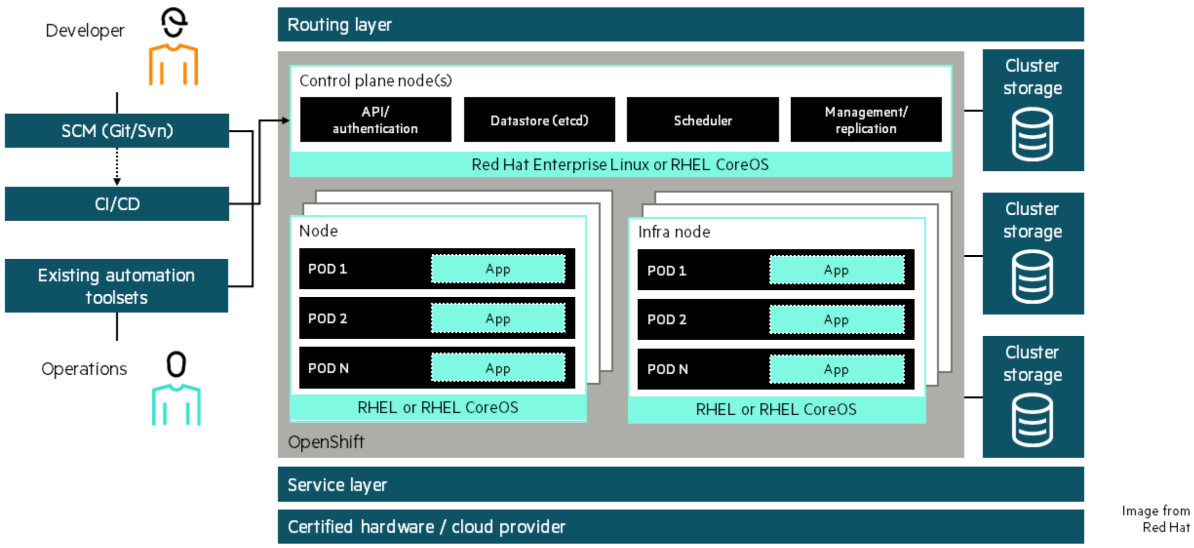

Figure 3-2: OCP architecture (image from Red Hat)

OCP builds on the Kubernetes architecture. One or more control plane nodes support the Kubernetes control plane components, including the etcd datastore, API server, scheduler, and controllers.

The control plane nodes control worker nodes that support the Kubernetes pods. Worker nodes can be established on physical servers or VMs, on-prem or in the cloud. OpenShift provides node bootstrapping, in which nodes pull down their configuration and certificates from a centralized node. Nodes run RHEL Server or RHEL CoreOS. RHEL CoreOS is a Linux-based OS that is designed specifically for clusters that hosts containers; it promotes security and ease of management. (After Red Hat acquired CoreOS, it merged its Atomic Project with CoreOS Container Linux to create RHEL CoreOS.)

The architecture might also feature infrastructure nodes, which host services such as routers.

The OpenShift architecture supports several storage options, including Gluster, Ceph, and other storage vendors. It might also use OpenShift Data Foundation, which is a software-defined storage solution.

A routing layer helps to handle communications between cluster components and between the cluster and external devices.

In addition to supporting other registries, OpenShift provides the OpenShift Container Registry (OCR), which integrates with the OCP, notifying it about new images and making it easier to automate pulling down and deploying new images.

Admins can manage the platform with the OpenShift CLI or OpenShift Console. But operators can also use existing automation toolkits to interact with API while developers often use CI/CD tools. Red Hat OCP supports a number of operators, which integrate various services and frameworks into the platform. GitLab is a popular source code management (SCM) solution, and the GitLab operator for Red Had OCP helps companies to establish a CI/CD platform for accelerating DevOps.

Benefits of the HPE and Red Hat Partnership

Next, you will learn more about the relationship that HPE has with Red Hat, and what exactly this relationship entails for you and your customers.

A single point of purchase for Red Hat OCP with HPE

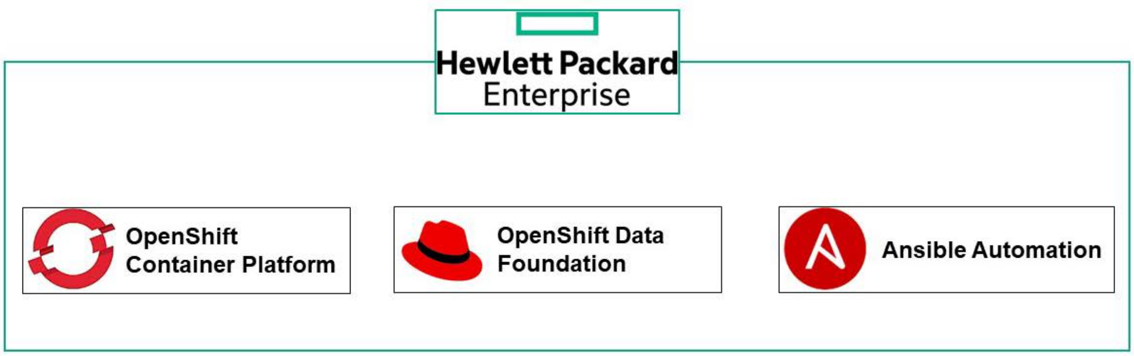

Figure 3-3: A single point of purchase for Red Hat OCP with HPE

Customers who purchase OCP will likely want to purchase more than just the container platform alone.

If they need persistent storage for their containers, then customers will also want to purchase Red Hat OpenShift Data Foundation, formerly called OpenShift Container Storage. Data Foundation provides software-defined persistent storage, which is suitable for data-intensive workloads with file, block, or object format data created by enterprise Kubernetes users.

Whether they need persistent storage or not, many customers will also want to leverage Red Hat Ansible Automation to accelerate and simplify their deployment. Ansible Automation combines Ansible Tower and Ansible Engine to provide both agility and quick implementation for automation.

With HPE, customers can purchase a package that includes any or all of these solutions, and that is purpose-built to smoothly run OCP on HPE hardware.

Support for software is provided directly from Red Hat

Figure 3-4: Support for software is provided directly from Red Hat

HPE provides its customers support during the delivery process for solutions that include Red Hat OCP and its related products. After delivery, HPE continues to provide support for the hardware, while support for software is provided directly from Red Hat.

Multiple levels of support are available from Red Hat: from 9 AM to 5 PM Monday through Friday, or 24x7 support. 9 to 5 support is available for the OCP with 2 cores or 4vCPU solution, for the Data Foundation with 2 cores solution, and for the Ansible Automation Platform with 100 nodes solution. 24x7 support is available for OCP with 2 cores or 4vCPU, and for Data Foundation with 2 cores.

The support from Red Hat lasts for the duration of a customer’s subscription, and most subscriptions can be either 1 or 3 years. These offerings include OCP with 2 cores or 4 vCPU, Data Foundation with 2 cores, and the Ansible Automation Platform with 100 nodes.

General Planning Considerations

You will next learn about considerations for designing the container platform. This section discusses considerations that might apply to other container platforms, not just Red Hat OCP.

Deployment: VMs vs bare metal

One of the primary considerations for deploying containers in the infrastructure is where the containers will run. They can run on either VMs, on-prem or in the cloud, or on bare metal nodes. The OS simply needs to support the container runtime and other orchestration components like kubelet and kube-proxy.

Read each example of container deployment below to compare VM and bare metal deployment.

VM workers

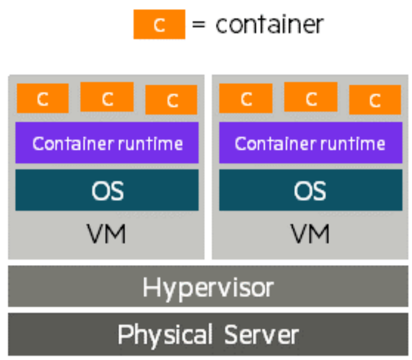

Figure 3-5: VM workers

Customers might choose to deploy containers on VMs because they like the ease of management of a virtualized environment. They can create a VM template for the worker node and then deploy that template for as many hosts as they need. Customers often already manage a virtualized environment, and they understand well how to manage and maintain that environment, as well as deal with issues such as failed hardware.

Using virtualized workers can also offer additional isolation and flexibility in worker node deployment. For example, the company can easily create different VMs for the development teams and production teams. In addition, the company is not forced to dedicate the physical machine to containers. It can run container worker node VMs and other types of VMs side by side.

When companies want to deploy containers in a public cloud, they need to use VMs provisioned on that cloud.

For all these reasons, many companies find it simplest to start by deploying the container platform on a virtualized environment.

Bare metal workers

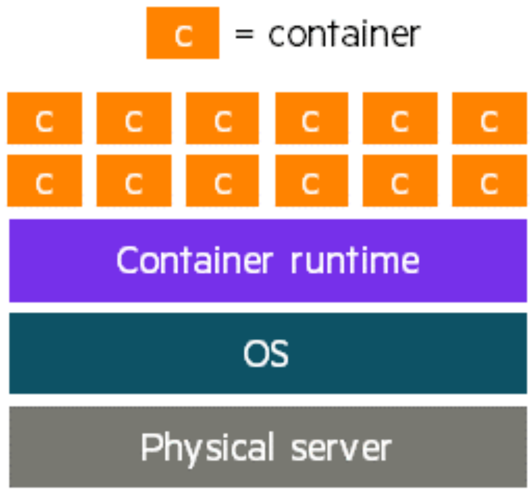

Figure 3-6: Bare metal workers

Deploying containers on bare metal offers advantages as well.

Most customers use VMware vSphere for the virtualization layer. The VMware licensing then becomes an additional cost for the solution, beyond the container platform subscription. By eliminating this sometimes hefty cost, customers can reduce the TCO of the solution.

Without the hypervisor layer, the bare metal host can provide better performance for containers with demanding workloads. An HPE study has shown about a 9% performance boost when a container runs on bare metal versus on a VM. Running containers on VMs can also reduce latency, which can be critical for latency sensitive applications. In short, bare metal deployment provides a better cost to performance ratio.

Deployment: Pure container environment or mixed environment?

	Pure container environment	Mixed (container and non-container) environment
Resource sharing	Resources are dedicated to the specific environment and may not be optimally shared	Resources can be optimally shared. However, sometimes resource allocated to containerized environments may be higher than optimal and may result in idle time.
Performance	Optimal performance can be established	Performance of any of the workloads may be impacted as required system resources are shared
Upgrades	Upgrades are easy to schedule	Downtime for non-containerized workloads must be considered while scheduling upgrades

Figure 3-7: Deployment: Pure container environment or mixed environment?

Another important consideration for deploying containers to infrastructure is whether the containers will exist in a purely containerized environment, or if they will share the environment with other, non-containerized workloads. Both deployments have their advantages and disadvantages.

When customers create a separate environment for containers, the company does not obtain the most efficient utilization for their total resources. When the containerized environment is running at lower than average utilization, for example, it cannot share extra resources with the non-containerized environment. As a result, companies typically have to provision both environments to meet their maximum—plus buffer—requirements, leading to a higher total cost. However, the containerized environment receives the performance benefits of dedicated resources. With proper planning, the company can deploy sufficient resources to ensure optimal performance for the container workloads. Upgrades are also easier to schedule. Kubernetes can execute a rolling upgrade that prevents any downtime in the containerized workloads; because all workloads are containerized in this environment, all workloads are protected.

A mixed container and non-container environment provides better resource sharing, which potentially reduces total costs for the customer. However, to obtain these benefits, admins must carefully tune. If they allocate too many resources to the containerized environment, resources still might sit idle sometimes. On the other hand, both containerized and non-containerized workloads might have their performance impacted because they are sharing resources. And scheduling upgrades is more complicated since the non-containerized workloads will presumably incur downtime.

Deployment: Stateless or stateful apps?

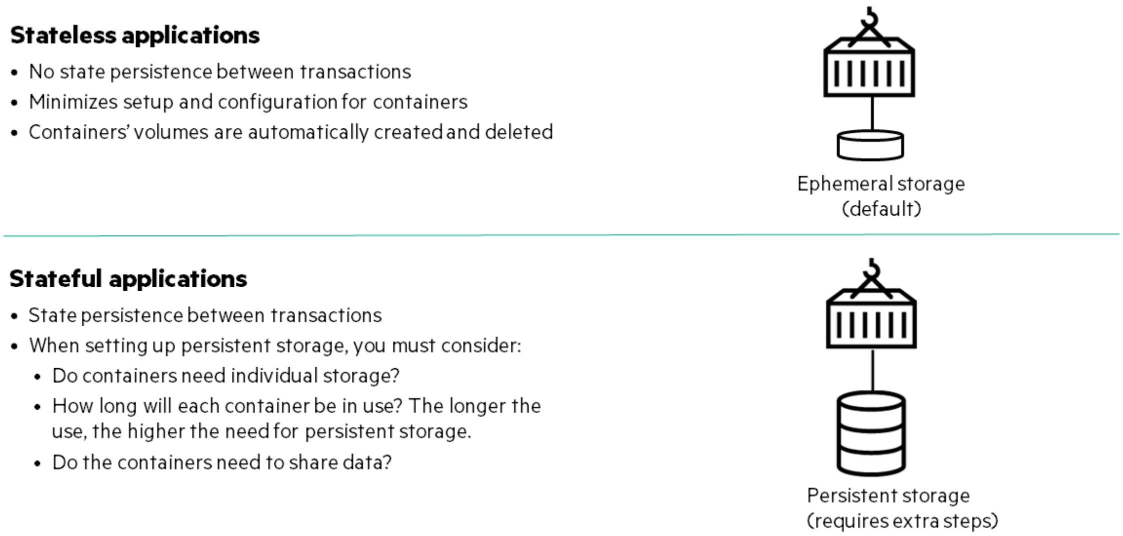

Figure 3-8: Deployment: Pure container environment or mixed environment?

Yet another consideration for deployment of containers is the nature of the application they will be running—whether it is stateless or stateful.

Containers are ephemeral in nature. They can be stopped, destroyed, rebuilt, and replaced with minimal setup and configuration. By default, they use ephemeral storage volumes, meaning that for applications, there is no state persistence between transaction; data from one session cannot be referenced in a later session.

This behavior can work well for stateless applications, which do not require state maintenance between transactions. Using the default ephemeral storage for stateless applications provides easy setup and configuration. Users can simply define an empty ephemeral volume for a Kubernetes pod. This volume is automatically created when the pod starts, and a container mounted to the volume can write "scratch" data to it. When the pod stops, the volume is automatically deleted, which works because the stateless application does not need to store the "scratch" data.

But some applications require state persistence between transactions; data from one session must be referenced in a later session. When stateful applications run in the containers, the containers need access to persistent storage, which requires extra steps to implement.

When setting up persistent storage, you must consider: do the containers need individual storage? How long will each container be in use? The longer the use, the higher the need for persistent storage. And do the containers need to share data? The answers to each of these questions will inform how you implement persistent storage.

Planning Red Hat Deployments

This section explains decisions that architects must make when preparing for a Red Hat deployment.

The broadest and deepest OpenShift stacks from HPE

1. **HPE Synergy, HPE ProLiant DL, HPE Apollo, and HPE SimpliVity solutions**
2. **Options for software-defined container storage or HPE storage arrays**
3. **Support for up-to-date OpenShift versions**

Figure 3-9: The broadest and deepest OpenShift stacks from HPE

Red Hat OCP customers benefit from the broad range of options that HPE infrastructure provides with choices optimized for each customer environment. As you will see in more detail on the next page, you can design solutions based on HPE Synergy, HPE ProLiant DL, HPE Apollo, or HPE SimpliVity.

You can also offer customers servers that support software-defined storage for the containers. Or, as you will learn in more detail in a later module, you can deliver enterprise-class storage for containers on HPE storage arrays.

HPE infrastructure solutions support the up-to-date OpenShift software that your customers want to run. If you choose to use some of the HPE value adds, such as automation scripts in the "hpe-solutions-openshift" Github site, you will need to check the site for the current version supported.

Selecting the HPE compute solution

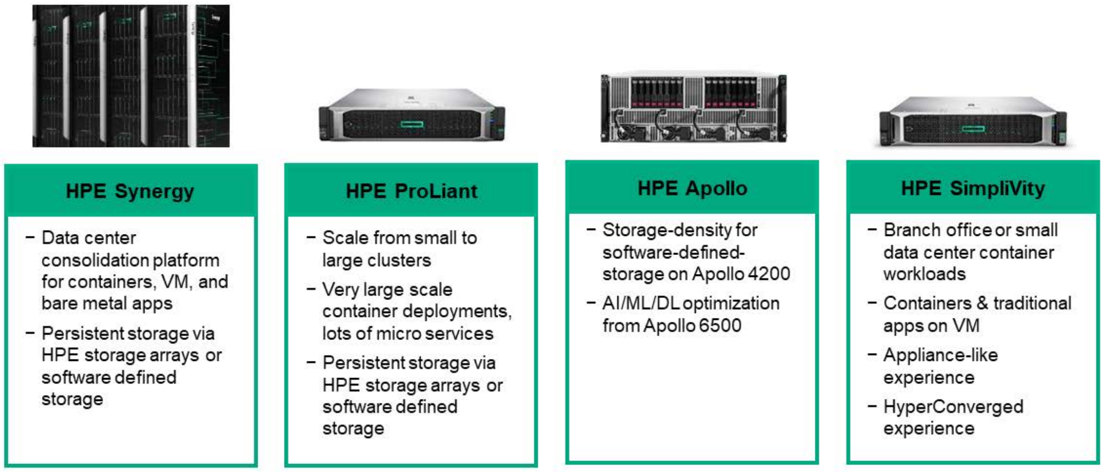

Figure 3-10: Selecting the HPE compute solution

HPE offers a broad range of compute options for supporting the Red Hat OpenShift environment.

You should choose Synergy for customers that want a blade format for their servers. Synergy is also particularly useful for data center consolidation. Customers can consolidate containers, VMs, and bare metal apps on Synergy. When you and your customers have determined that some of their container workloads need to be deployed on bare metal, Synergy can help customers simplify management of the bare metal worker nodes. For the storage layer, you can deploy OpenShift Data Foundation on Synergy compute modules connected to Synergy D3940 drives; the Synergy architecture provides great flexibility for the number of drives mapped to each compute module. Or you can add HPE storage arrays to the solution.

HPE ProLiant, on the other hand, is the right choice for customers that want rack servers. While Synergy is mainly positioned for large enterprises, you can design ProLiant DL solutions for smaller clusters or for large clusters. You can position ProLiant for customers with very large-scale container deployments that utilize lots of microservices. HPE offers flexible storage options for the ProLiant-based solution as well. You can establish a software-defined storage layer on the ProLiant servers with OpenShift Data Foundation. Or you can attach HPE storage arrays to the solution.

HPE Apollo servers can fulfill specialized use cases. For example, if customers are looking for storage-density for the software-defined storage layer, you can offer HPE Apollo 4200 Systems. HPE Apollo 6500 Systems

Lastly, SimpliVity is best for customers that need to run containers in the branch office or run container workloads in a small data center. SimpliVity helps customers run containers and traditional apps side-by-side on VMs. HPE SimpliVity offers customers the appliance-like experience of a hyperconverged solution. Customers simply deploy the SimpliVity nodes, which automatically form a cluster of compute and storage resources on which admins can easily deploy VMs. From there they can quickly deploy the Red Hat OpenShift solution. Note that HPE SimpliVity, like most hypercoverged solutions, requires customers to scale compute and storage resources together. In other words, they add another SimpliVity node to the cluster, which adds more processors, memory, and storage.

Deploying Red Hat OCP on HPE Synergy

This section provides an RA for deploying Red Hat OCP on HPE Synergy. It also explains how HPE helps to automate parts of the deployment.

HPE Synergy: The ideal container platform

- Deploy at cloud speed
- Flex containers up and down
- Centralize container lifecycle management
  - Software-defined all the way to bare metal
  - Synergy integration with Docker and tools like Chef and Ansible
- Obtain advanced container data management
  - HPE storage arrays to provide persistent storage and high availability

Figure 3-11: HPE Synergy: The ideal container platform

HPE Synergy provides the ideal platform for enterprise container environments.

A container platform helps to accelerate app deployment, but only starting from an infrastructure that is already prepared to run the containers. In the real world, infrastructure provisioning can create a bottleneck that slows business down. Because HPE Synergy makes the infrastructure programmable, it dramatically reduces provisioning time and helps customers to operate truly at cloud speed.

With traditional solutions, provisioning compute and storage takes so much time that IT wants to do it once and leave it as is. But that approach does not mesh well with containerized applications meant to scale up and down. A solution like Kubernetes or Red Hat OCP can quickly scale containers up and down, but to maintain good performance, the workers might need to scale too. With HPE Synergy and service profile templates (SPTs), IT can quickly deploy new compute modules to scale out the solution based on the current need.

If they need to scale down later, they can deploy a different image to those compute modules and use them for a different solution.

Even better, HPE Synergy extends automation down to the physical hardware. Synergy solutions are software-defined and integrate with Docker, as well as third-party automation tools like Chef and Ansible. Customers can manage the containers’ complete life cycle centrally. Lifecycle management becomes simpler and faster; for example, admins can reduce updates from hours to minutes.

As customers use containers for more and more mission critical applications, their needs for advanced data protection grow. HPE Synergy solutions work with HPE storage solutions, which, as you will learn in a later module, deliver persistent container storage that is easy to deploy and highly available.

Use HPE container orchestration platform RAs and RCs

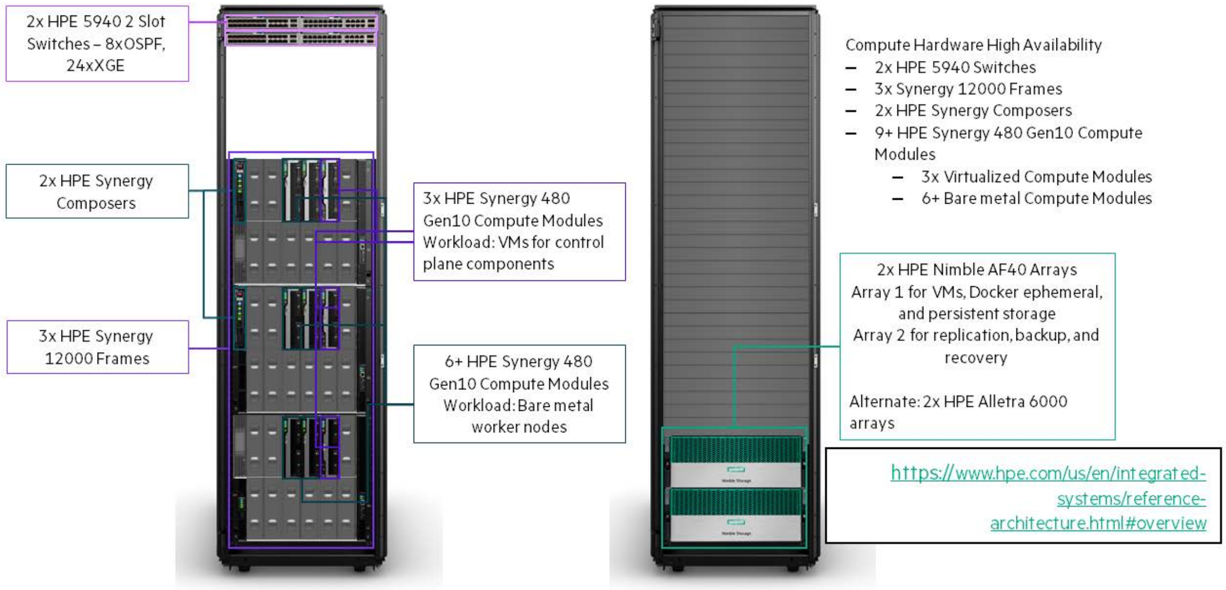

Figure 3-12: Use HPE container orchestration platform RAs and RCs

HPE makes its considerable expertise available to you in reference architectures (RAs) that provide validated designs and configurations for running Red Hat OpenShift on HPE infrastructure. You should look for RAs and RCs at this link:

https://www.hpe.com/us/en/integrated-systems/reference-architecture.html#overview

You can then use those documents as starting points for designing solutions for your particular customer.

The figure shows the HPE components used in *HPE Reference Configuration for Red Hat OpenShift Container Platform on HPE Synergy and HPE Nimble Storage* (published 2019). This RA uses a virtualization layer and deploys control plane components on VMs, and it uses bare metal worker nodes. If you are interested in an RA that uses bare metal master nodes, refer to a similar RA, *HPE Reference Configuration for Red Hat OpenShift Container Platform on HPE Synergy and HPE Storage Systems* (published in 2021).

Key points include:

- **HPE Synergy 480 Gen 10 compute modules**—Several modules act as ESXi hosts to support control plane VMs. Other modules act as bare metal workers. Like most RAs, this RA provides a basic building block that you can expand based on your customer requirements. For example, your customer might require more than three modules to support the virtualization platform or more than six bare metal workers. You will learn more about sizing guidelines in a moment.
- **Supporting Synergy hardware**—All Synergy solutions require supporting components such as Synergy Composers (two are recommended for high availability), frames, and interconnect modules. Refer to the RA for details.
- **HPE Nimble arrays**—This RA recommends a Nimble array to provide the backend storage for the virtualized compute modules. It hosts VMs data in datastores, and it hosts Docker ephemeral storage. It also hosts Kubernetes Persistent Volumes (PVs) for the platform. (You will learn more about how HPE storage integrations support that role in a later module). You could add a second array for replication, backup, and recovery for a highly available solution.

  You might alternatively use HPE Alletra 6000 arrays or, for higher availability, HPE Primera or Alletra 9000. Note these recommendations differ from the alternative suggested in the RA, a 3PAR array, as

3PAR arrays are now legacy solutions. You should be ready to make similar substitutions as you use RAs as a guide, but not a prescriptive list.

- **HPE 5940 switches**—Remember to include top of rack (ToR) switches to provide connectivity between racks. The RA recommends 5940s, but you might alternatively recommend Aruba 83x0

Design Red Hat OCP on HPE Synergy

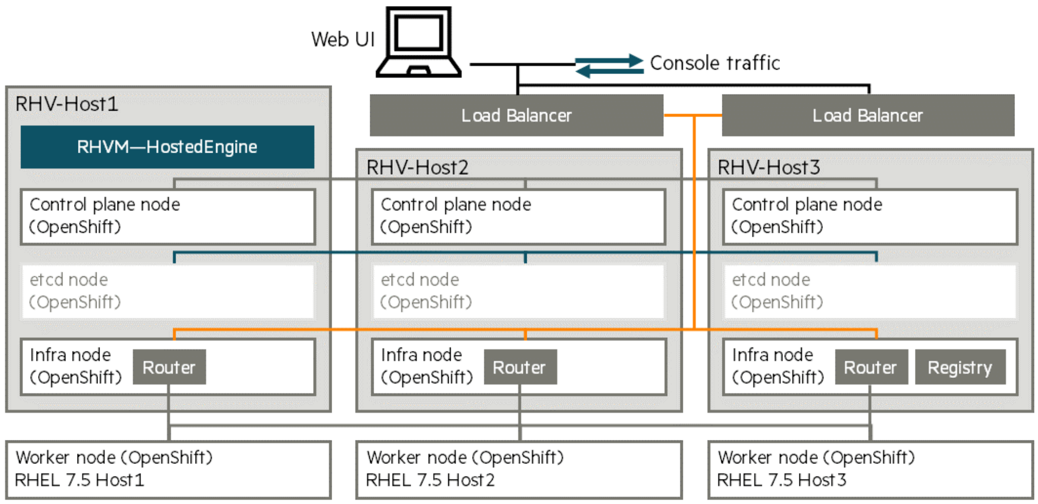

Figure 3-13: Design Red Hat OCP on HPE Synergy

The figure above shows the Red Hat OCP architecture as deployed in the RA. The cluster of three virtualized SY480 compute modules, listed in the figure as RHV Host1, Host2 and Host3, support multiple VMs. This RA recommends distributing control plane components over a few different types of VMs called control plane (or sometimes master) nodes, etcd nodes, and infrastructure nodes. Learn more about some of these components below.

Control plane nodes

These nodes run most of the master components associated with Kubernetes, including the API server, the scheduler, and the controller master and the associated controllers. These nodes might also maintain the etcd database. Or customers can choose to deploy the database on dedicated etcd nodes, as shown in the figure above.

It is recommended that you deploy three control plane nodes per cluster. If you use VMs for these nodes, us affinity rules to place those VMs on different hosts.

Infrastructure nodes

These nodes run the routing services and also maintain the OCP registry. It is recommended that you deploy three etcd node VMs with affinity rules placing those VMs on different hosts.

Load balancers

The load balancer, or HA proxy, nodes load balance requests across the infrastructure nodes. The RA used two VMs for load balancing, but in a production environment you should use a dedicated third-party software or hardware solution.

Worker nodes

Worker nodes run kubelet, kube-proxy, and the Docker runtime. They can be bare metal SY480 compute modules, as shown here. Or they can be VMs running on a cluster of virtualized compute modules.

The figure shows three bare metal worker nodes, but you should scale this number based on the number of pods that your customer's solution needs to support, as will be described in more detail later in this section.

Size worker nodes

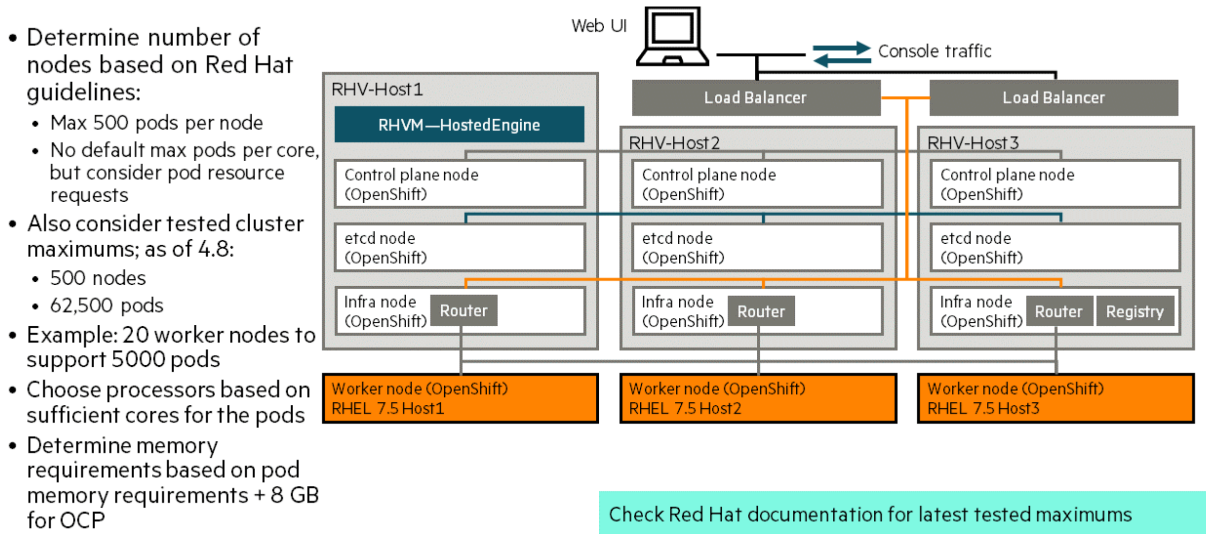

Figure 3-14: Size worker nodes

You will now drill down on guidelines for sizing each of the node types, beginning with the worker nodes, as the number of worker nodes will affect sizing for the control plane.

To determine the number of worker nodes that a cluster requires, you and the customer will need to discuss the number of pods that the customer plans to deploy, as well as the requirements for each of those pods. In OCP v4.8, a worker node can support up to 500 pods. However, all pods do not consume the same number resources, so you will need to take the customer's containerized workload requirements into account.

Kubernetes, on which OCP is based, allows users to specify resource requests and resource limits for resources such as CPU, memory, and storage. The pod can only run if a worker in the cluster has sufficient resources to fulfill the request. One pod might request a full core (bare metal worker) or vCPU (virtualized worker, while another pod might request only a fraction of a core or vCPU. You need to understand the various sizes of pods that your customer plans to run, as well as the number for each.

Once you understand the total amount of compute and memory requirements that pods will impose on the cluster, you can start to plan the number of worker nodes and the components for those worker nodes. For example, a company might need to support 5000 pods. In theory, 10 worker nodes could support this number of pods. However, you and the customer might agree on deploying 20 worker nodes because some of the pods will be compute- or memory-intensive and because the customer wants to accommodate growth.

You can choose appropriate processors based on providing sufficient cores for the pods. Consider a bare metal example. If the worker must support 200 pods, each of which requires 100 mCPU (one-tenth of a core), and 50 more pods, each of which requires 200 mCPU (one-fifth of a core), the worker will need about 30 cores to support the pods. Then you would also plan a core or two for overhead. Similarly, you can plan the memory requirements based on pod memory requirements, plus at least 8GB for overhead for running OCP.

If you are sizing virtualized workers, you would plan vCPU and memory allocated to the worker VM, rather than cores and memory.

For storage, you will need to consider how much data the customer's pods will store in ephemeral volumes and how much in persistent volumes, as well as whether that data will be stored on local drives, a storage array, or both. You will learn more in Module 5.

Size control plane nodes

- Follow Red Hat guidelines
- Sizing depends on factors such as:
  - Number of worker nodes
  - Number of Kubernetes namespaces and objects
- Storage sizing based on factors such as
  - Whether main control plane nodes or dedicated nodes are hosting the database

Recommendations from Red Hat (OCP v4.8)

Worker nodes	Cluster load (namespaces)	CPU cores	Memory (GB)
25	500	4	16
100	1000	8	32
250	4000	16	96

Figure 3-15: Size control plane nodes

To size control plane nodes, look to the current Red Hat documentation for recommended host practices for scalability and performance. The figure above shows the Red Hat recommendations for control node CPU cores and memory for various sized clusters in OCP 4.8.

As you see the sizing depends on both the number of worker nodes in the cluster and the number of Kubernetes namespaces and objects. Red Hat tested each cluster size with the cluster load indicated in the figure (for example, 500 namespaces for 25 worker nodes). Red Hat tested with a relatively small number of objects per namespace (12 image streams, 3 build configurations, 6 builds, 1 deployment with 2 pod replicas, 2 deployments with 1 pod replica, 3 services, 3 routes, 10 secrets, and 10 config maps), so a company's cluster might have fewer namespaces but more objects per namespace.

While companies might be able to get away with smaller sized nodes for limited lab environments, you should always follow the guidelines for real-world environments, including both testing and production environments.

Also take into consideration whether the control plane nodes running other control plane services will also be hosting the etcd database. You will need to take that fact into consideration when planning the storage. If you are deploying separate etcd nodes, also size them based on Red Hat guidelines.

You should typically plan three control plane nodes per cluster (or even five for very large clusters).

Size infrastructure nodes

- Follow Red Hat guidelines
- 3 per cluster
- Sizing
  - Depends on number of worker nodes
  - Might depend on number of objects
    - But tested by Red Hat with a large number of objects

Recommendations from Red Hat (OCP v4.8)

Worker nodes	CPU cores	Memory (GB)
25	4	16
100	8	32
250	16	128
500	32	128

Figure 3-16: Size infrastructure nodes

You should generally deploy three infrastructure nodes per cluster to support services such as routing.

Red Hat provides recommendations for sizing these nodes as well. This sizing depends primarily on the number of workers. While requirements might also depend on the number of objects used in the cluster, Red Hat has tested the sizes shown here with a large number of objects (for the 250- and 500-worker node clusters, 10,000 namespaces with 61,000 pods and 10,000 deployments; see Red Hat host best practices for more details). Therefore, if you follow these guidelines for each cluster size shown, the infrastructure nodes should work well.

HPE Solution Sales Enablement Tool (SSET)

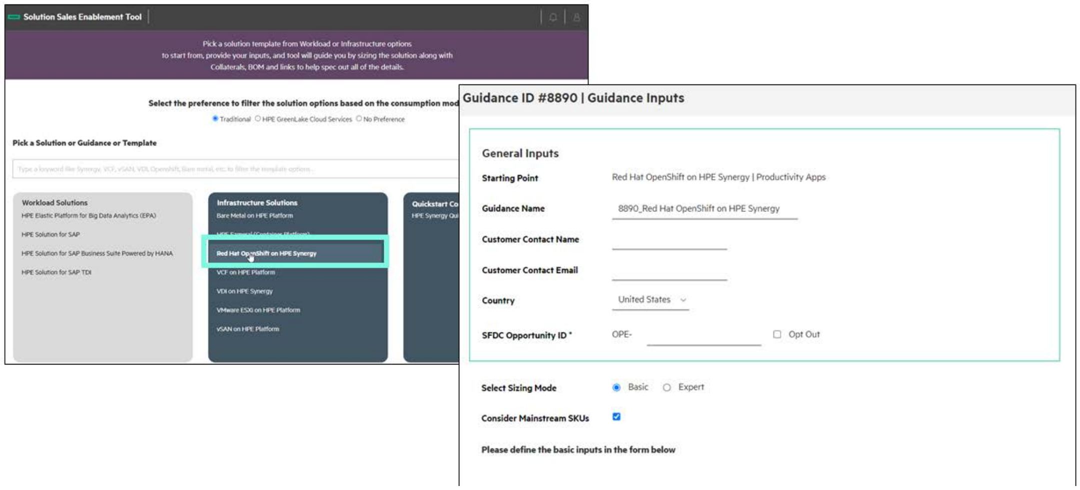

Figure 3-17: HPE Solution Sales Enablement Tool (SSET)

If you are deploying Red Hat OCP on HPE Synergy, you can also use the HPE Solution Sales Enablement Tool (SSET) to size the solution. As you can see here, using this interface, you simply select the solution you are interested in building. For sizing OCP on HPE Synergy, this tool provides Basic and Expert options, providing more or less guidance as needed. Please note that the solution recommends HPE Nimble for storage.

You may want to open HPE SSET and explore the process of sizing OCP on HPE Synergy.

Example configuration for virtual hosts

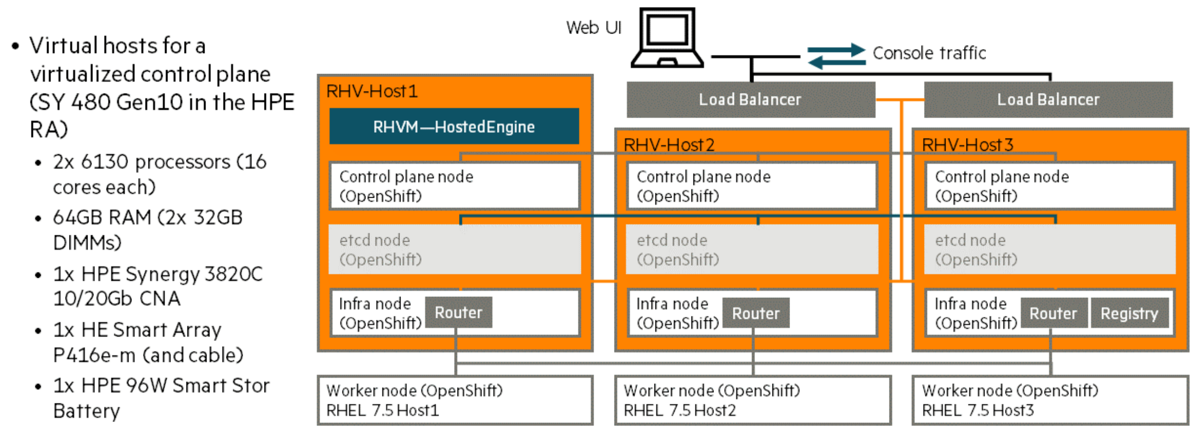

Figure 3-18: Example configuration for virtual hosts

If you are running any nodes as VMs, whether control plane nodes, infrastructure nodes, or worker nodes, you need to size the virtual hosts that will host them. Generally you should use a cluster of at least three hosts for availability purposes. However, you can further scale that cluster based on the number of VMs the customer requires and those VMs' profiles.

The figure shows the virtual host configuration for the *HPE Reference Configuration for Red Hat OpenShift Container Platform on HPE Synergy and HPE Nimble Storage*. Note that this RA uses bare metal workers, so the virtual hosts only support the control plane. If your company uses virtualized workers, you would need to further scale the number of ESXi hosts.

Container monitoring and security with Sysdig

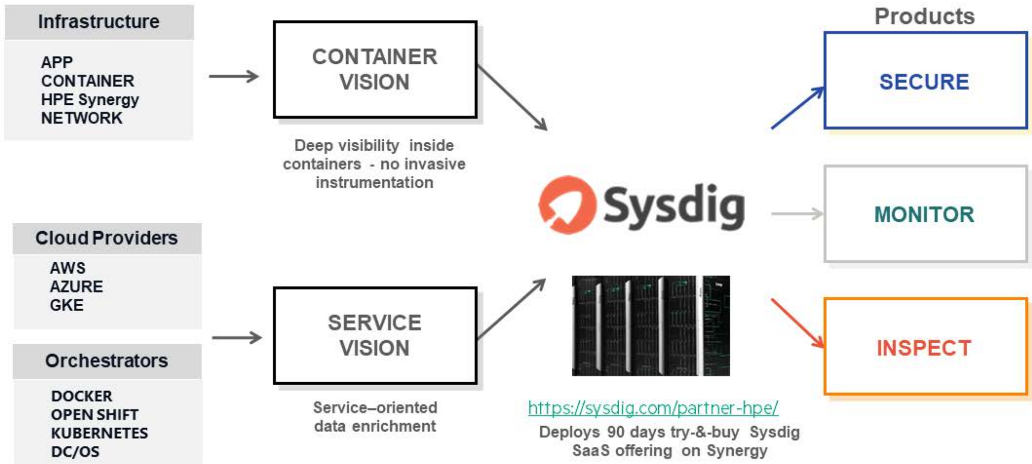

Figure 3-19: Container monitoring and security with Sysdig

You can help the customer integrate their OpenShift solution with HPE partner, Sysdig. Sysdig integrates with the infrastructure solutions, cloud provides, and container orchestration solutions in the customer's containerized environment. It aims to help customers surmount some of their key issues with containers: difficulty managing complex microservices and particularly difficulty measuring performance of applications running inside containers distributed across many environments and even clouds.

Sysdig ContainerVision provides deep visibility into containers, network, application, and system activity without invasive instrumentation—no need for application plug-ins to get application visibility. ServiceVision ties all the metadata provided from orchestration solutions like OCP into the data collected by ContainerVision. The Sysdig backend, which can be deployed on premises or used as a SaaS service, aggregates, correlates, and organizes all of the data collected by ContainerVision and ServiceVision. Customers can then use that data with three key Sysdig products.

Sysdig Secure provides run-time security, actively looks for and prevents threats in the production environment. Sysdig Secure uses behavioral analysis to detect anomalies, enabling it to stop zero day threats and internal threats. It can also prevent lateral movement within environments—in other words, prevent hackers from gaining a foothold in one container or machine and moving out across the network to compromise more resources. Sysdig Secure also provides forensic security, which helps security teams look back after a breach occurred. The data that Sysdig collects is critical for real time incident response (IR) and also helps customers keep records for passing audits and proving compliance—no other vendor has capabilities like this at this time. Sysdig Secure can also integrate with customers' existing SEIM solutions.

Sysdig Monitor gives customers the power to analyze data in relation to the logical applications and services that users actually care about. It automatically finds applications as they move around the environment. Customers can write a small app-check for a custom application and Sysdig then automatically discover the application as it spins up in new containers. Sysdig Monitor helps customers answer questions like what is the performance of their Cassandra database across 20 pods and two data centers. It yields container data, enriched by orchestration metadata, on the fly, allowing customers to slice-and-dice data by service, pod, namespace, host, or anything else that's relevant for troubleshooting.

Sysdig Inspect enhances Sysdig's forensic capabilities.

For more information about HPE and Sysdig partnership, click here.

Physical environment configuration

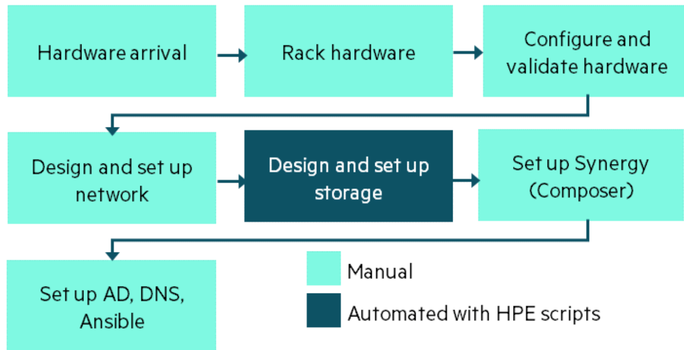

Figure 3-20: Physical environment configuration

You will now look at an overview of the steps for deploying Red Hat OCP on HPE Synergy. This course will not go into the detail, but give you a general idea of the tasks to complete with an emphasis on how HPE helps to automate many of them.

The first steps involve racking the hardware, making connections, and setting up the networks. HPE provides scripts to help set up Nimble storage, if you are using that option. At this point, you also set up Synergy Composer. The customer should also set up supporting services such as AD and DNS. If the customer does not already use Ansible, deploy Ansible Tower.

Ansible is an open-source software provisioning, configuration, management, and application-deployment tool that enables infrastructure as code. Among other things, companies like HPE can use the Ansible declarative language to describe system configurations and deliver those descriptions as playbooks. These Ansible playbooks allow Ansible to perform operations on managed nodes.

OCP deployment

- Create and capture golden image for workers
- Use PXE to install RHV on compute modules that will host control plane, etcd, and infrastructure nodes
- Use HPE scripts and Ansible playbooks to automate many deployment tasks such as:
  - Deploying VMs for various OCP roles
  - Installing OCP roles
  - Installing OpenShift Container Storage (OCS)
  - Validating the solution
  - Installing supporting services such as Sysdig

Figure 3-21: OCP deployment

Next, you set up images and deploy the operating systems on the Synergy compute modules.

Create and capture golden images for any bare metal workers. Also Set up RHV images for any compute modules that will act as virtual hosts, including control plane nodes, infrastructure nodes, and possibly workers. Use PXE to deploy RHV to these compute modules.

You can then use Ansible playbooks, developed by HPE, to automate host configuration. You can find these playbooks here: https://github.com/hewlettpackard/hpe-solutions-openshift

Follow the instructions provided with the scripts and playbooks in GitHub. At a high level, the HPE scripts and playbooks help you automate tasks such as deploying VMs that support various OpenShift roles to the virtual hosts, installing OCP roles, installing OpenShift Container Storage (OCS) (older term for OpenShift Data Foundation), and installing supporting services such as Sysdig.

Automation tools

Obtain scripts for deploying OCP at:

- https://github.com/hewlettpackard/hpe-solutions-openshift

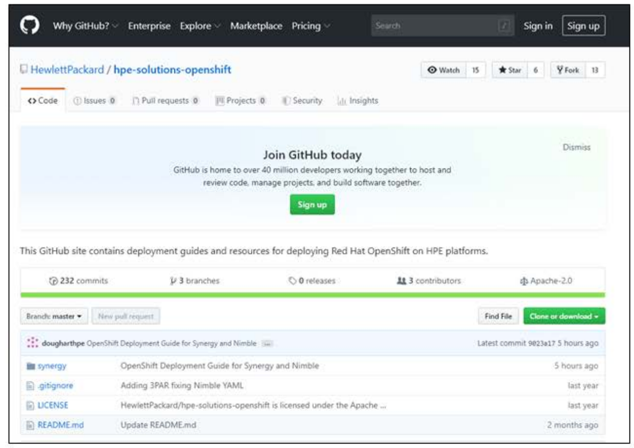

Figure 3-22: Automation tools

You can obtain the scripts mentioned on the previous pages from Github at:

https://github.com/hewlettpackard/hpe-solutions-openshift

You can also search Github for HPE scripts for deploying other container platform-related RAs.

You might need to navigate through a tree to find the script specific to your customer's solution.

Subscriptions

- HPE-branded Red Hat subscriptions
  - Identical to Red Hat subscriptions
  - Support delivered by Red Hat
- Red Hat OpenShift Container Platform
  - Entitles customer for Red Hat OpenShift Container Platform + supporting components such as Red Hat Enterprise Linux (RHEL) OS
  - # depends on physical cores or vCPUs across all workers
- Red Hat OpenShift Storage
  - Entitles customer for the Red Hat software-defined-storage for OpenShift
  - # depends on physical cores or vCPUs across all workers

Figure 3-23: Subscriptions

Remember that you can sell a complete Red Hat OCP solution on HPE infrastructure in which HPE is the single point of purchase. Sell the HPE-branded Red Hat subscriptions as part of the solution. These subscriptions are identical to Red Hat native software subscriptions, and Red Hat delivers the software support related to these subscriptions. (If customers ask, the subscriptions are available in physical and electronic delivery forms, and they are transferrable to other supported HPE servers as needed).

As of the release of this course, HPE offers three options for the HPE-branded Red Hat OCP subscriptions:

- Red Hat OpenShift Container Platform 2 Cores or 4 vCPUs 1yr Subscription 9x5 RH Support
- Red Hat OpenShift Container Platform 2 Cores or 4 vCPUs 3yr Subscription 24x7 RH Support
- Red Hat OpenShift Container Platform 2 Cores or 4 vCPUs 3yr Subscription 9x5

These subscriptions entitle customers to run Red Hat OCP, as well as supporting components such as the Red Hat Enterprise Linux (RHEL) OS. In other words, a separate RHEL OS licenses is *not* required. However, the Red Hat OCP nodes should only be running Red Hat OCP.

The number of required subscriptions depends on the worker nodes. If customers are using bare metal workers, they need one subscription for every two cores on those workers. If they are using virtual workers, they need one subscription for every four vCPUs assigned to those workers. Customers do *not* have to include control plane nodes and infrastructure nodes in the core/vCPU count. They are entitled to run a Red Hat OS and Red Hat OCP on the control plane and infrastructure nodes as part of their Red Hat OCP solution. However, those nodes should not run container workloads or other applications.

If you are including OpenShift Data Foundation in the solution, remember to include subscriptions for OpenShift Container Storage. HPE offers these options:

- Red Hat OpenShift Container Storage 2 Cores 1-year Subscription 9x5 RH Support
- Red Hat OpenShift Container Storage 2 Cores 3-year Subscription 24x7 RH Support
- Red Hat OpenShift Container Storage 2 Cores 3-year Subscription 9x5 RH Support

The required number of subscriptions depends on the number of cores or vCPUs on all the workers that support OpenShift Data Foundation.

If you need more information on planning the number of subscriptions, refer to the Red Hat OpenShift subscription sizing guide.

Deploying Red Hat OCP on HPE ProLiant and Apollo

You will now learn about deploying Red Hat OCP on other HPE servers, including HPE ProLiant and HPE Apollo. You can follow many of the fundamental principles about which you learned in the previous sections. This section focuses on the particular benefits of and guidelines for HPE ProLiant and Apollo deployments.

Benefits of HPE ProLiant for Red Hat OpenShift

Trusted hardware platform

- Thousands of businesses worldwide trust mission-critical workloads to HPE ProLiant

Simplicity

- HPE ProLiant offers simple, familiar technologies for rapid container deployment.
- Optimize performance using suggestions from OneView

ML and AI ready

- Maximized performance for GPU-ready containerized workloads.

Robust performance

- HPE ProLiant servers are fully certified for use with OpenShift

Cost-effective solutions

- Rack-ready, high density servers.
- Cost-effective hardware lowers the barrier to entry

Respond to change

- Seamlessly scale the solution to allow for deployment of additional services and capacity as needed.

Figure 3-24: Benefits of HPE ProLiant for Red Hat OpenShift

Running Red Hat OpenShift on HPE ProLiant offers your customers several benefits.

Trusted, reliable, HPE ProLiant servers are Red Hat certified. They offer GPU acceleration, ideal for intensive ML and AI applications.

HPE ProLiant servers are designed for simple optimization for the workload. You or your customers simply choose the correct workload template, and the server automatically optimizes itself for that workload. Excellent performance and reliability number among the reasons that thousands of businesses worldwide trust mission-critical workloads to HPE ProLiant. At the same time these rack-ready, high-density servers offer a cost-effective solution that lower the barrier to entry; with HPE ProLiant, customers of many sizes can enjoy the benefits of containers.

Because many customers are already familiar with HPE ProLiant servers—as these are some of the most widely deployed servers in the industry—their IT staff will have a smaller learning curve for managing the solution. Using OneView, you and your customers can pre-configure templates for the Red Hat OCP servers. They can also use Red Hat Tower to reduce deployment to a single click. Customers can get the solution up and running quickly and rapidly deploy containers.

In addition to speeding deploying, these powerful, granular tools allow customers to respond to change more quickly and effectively. Red Hat OpenShift and HPE ProLiant will allow customers to scale capacity as your container demands change. For example, OneView templates will help admins scale the solution seamlessly and with much lower risk of operator error. And admins can use again use Red Hat Tower to automate the configuration of the new servers.

Overview of Reference Architecture implementation

Figure 3-25: Overview of Reference Architecture implementation

You will now focus on one example RA for deploying Red Hat OCP on HPE ProLiant DL and Apollo servers: *HPE Reference Architecture for Accelerate AI/ML on HPE ProLiant DL Servers and HPE Apollo 6500 Gen10 Server*. This RA outlines a solution that not only runs containers on Linux, orchestrated with Kubernetes, but also offers persistent storage using OCP Data Foundation. It is specifically designed for customers who need to accelerate artificial intelligence (AI) and machine learning (ML).

The solution offers several key benefits. It accelerates ML modeling with a multi-tenant, self-service environment. It reduces development time for AI-powered applications with DevOps capabilities. And it speeds up procurement, deployment, and scaling of AI infrastructure.

If you want to deploy a Red Hat OCP solution on HPE ProLiant DL servers for a different use case, you can refer to an alternative RA such as *HPE Reference Architecture for Red Hat OpenShift on HPE ProLiant DL380 Gen10 and HPE ProLiant DL360 Gen10 Servers*.

Solution hardware components

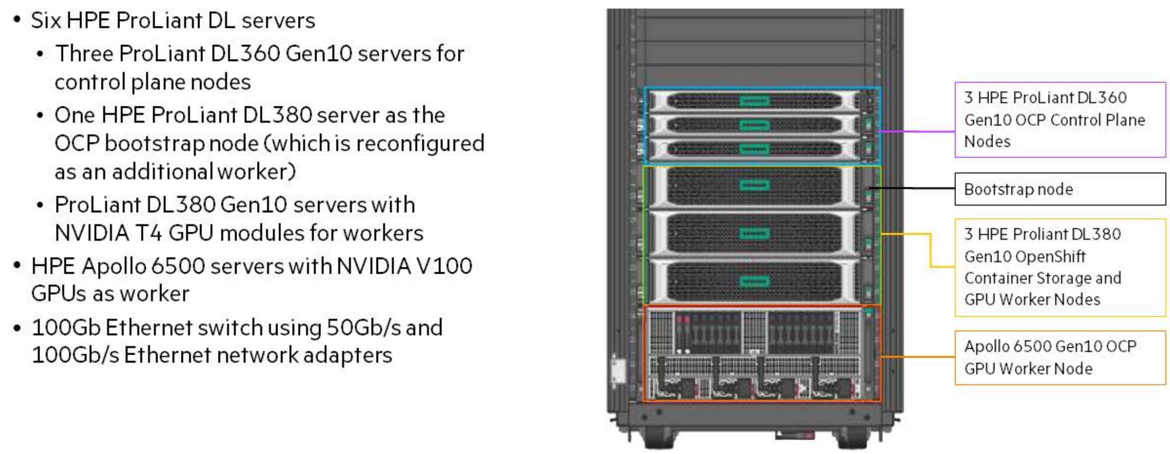

Figure 3-26: Solution hardware components

The figure illustrates the components in the RA mentioned on the previous page. The RA recommends three HPE ProLiant DL 360 Gen10 servers to act as the control plane nodes, as well as one more ProLiant DL 360 Gen10 server to act as a bootstrap node to help deploy the solution. After the deployment, the bootstrap server can be repurposed as an additional worker. The ProLiant DL360 servers use this configuration in the RA:

- 2 Intel Xeon-Gold 5218 processors (2.3 GHz/16-core)
- 32 GB RAM (4x 8GB)
- 2 HPE 960GB SATA SSDs

The RA defines configurations for two workers that are optimized for supported AI/ML container workloads:

- ProLiant DL380 Gen10 servers
  - 2x Intel Xeon-Gold 6242 processors (2.8 GHz/16-core)
  - 192 GB RAM (12x 16GB)
  - 3x HPE 960GB SATA SSDs
  - 4x NVIDIA T4 GPU modules
- ProLiant Apollo 6500 servers
  - 2x Intel Xeon-Gold 6240F processors (2.6 GHz/18-14-8-core)
  - 192 GB RAM (6x 32GB)
  - 3x HPE 960GB SATA SSDs
  - 4x NVIDIA V100 GPU modules

The RA also includes 100Gb Ethernet switches using 50Gb/s and 100Gb/s Ethernet adapters to support high speeds between nodes in the cluster.

If you would like more details about the configurations used in the RA, refer to it. As always, be aware that the RA provides a starting point and guidelines. You might, for example, update processors to more up-to-date equivalents. Or you might deploy Gen10 Plus, rather than Gen10, servers.

Solution software components

Figure 3-27: Solution software components

This solution supports the same software components that you examined for earlier solutions, including Red Hat OCP and OpenShift Data Foundation. It can also include RHEL CoreOS, which is a host OS optimized for running containers.

Logical view

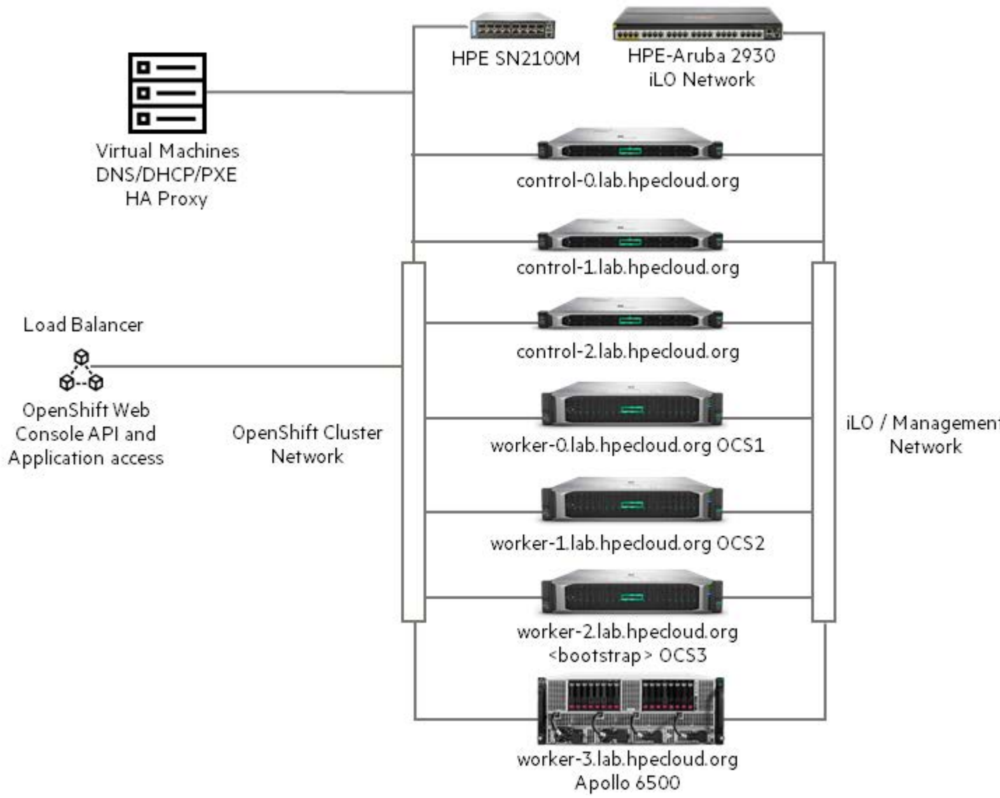

Figure 3-28: Logical view

The figure illustrates the logical layout for the RA covered in this section. It includes just a few workers, but you could, of course, add more workers for a real-world solution.

As you see, control nodes and workers communicate on an OpenShift cluster networker. This network also connects to a load balancer, through which external users and applications can manage the cluster and access applications running on it. Control nodes and workers also connect to an iLO network, which permits admins to manage and monitor the HPE infrastructure.

Deployment: Options

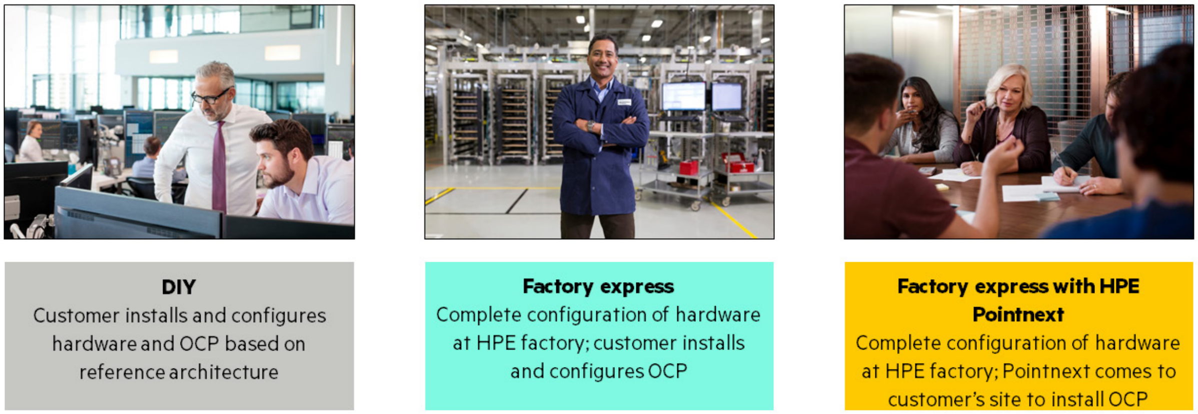

Figure 3-29: Deployment: Options

You can offer customers several options for deploying the Red Hat OCP solution on HPE ProLiant DL and Apollo servers.

If your customers prefer to handle deployment in-house, they can certainly do so. To set them up for success, provide them with the RA as well as guidance in using the RA. Make sure that the customer knows where to look for HPE scripts in Github.

However, many customers lack in-house expertise and want a faster track to a fully operational solution. You can offer these customers one of two services, depending on how much of the setup the customer wants to offload. With the Factory Express service, which you can add to the solution in HPE One Config Advanced (OCA), HPE completely configures the hardware for the OCP solution at the factory. However, the customer remains responsible for installing and configuring OCP onsite. (You might also offer your own partner services and install and configure OCP for the customer.) If the customer wants HPE to also install OCP, add a Factory Express service to the solution in OCA and then use Pointnext links to add HPE Pointnext services.

Deployment: Summary of process

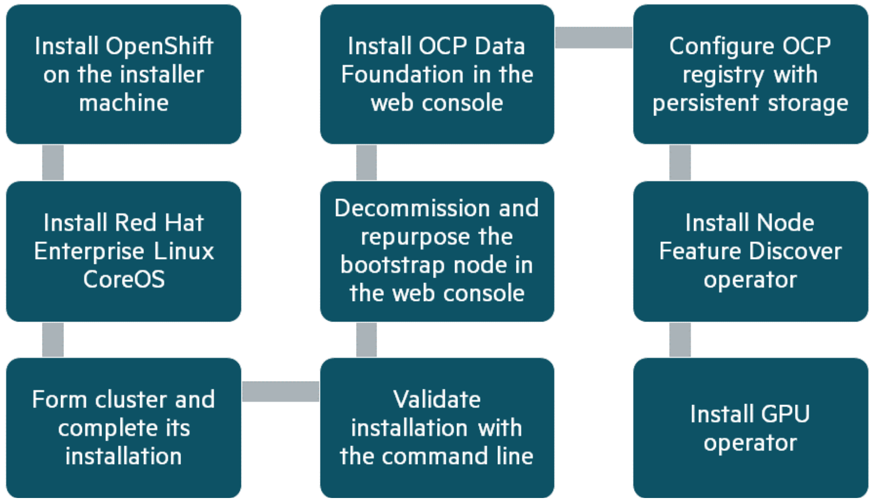

Figure 3-30: Deployment: Summary of process

Depending on the services that you have worked out with the customer, you or other members in your partner organization might be responsible for installing and configuring OCP. You should have a general understanding of the process.

At a high level, installing and configuring Red Hat OCP according to the RA discussed in this topic begins by installing an OpenShift install utility on an installer machine. You then install RHEL CoreOS on the future control plane, worker nodes, and bootstrap node using automated process such as PXE. (You could also deploy Red Hat OCP in a virtualized environment, as discussed in the previous topic; however, this RA uses a bare metal deployment.) The nodes will then form a cluster and install OCP using the bootstrap node.

After validating the installation from the command line, you are ready to decommission the bootstrap node. You can then repurpose the server for another purpose such as a worker in the cluster. The OpenShift Console is now available, and you can install Data Foundation from there. You can then deploy the OCP registry with persistent storage.

As you have learned Red Hat OCP supports multiple operators for various use cases. To support GPUs, you must install these two operators: Node Feature Discover operator and GPU operator. The customer can later install other operators, as needed.

Note that you can also visit GitHub to look for HPE automation scripts and Ansible playbooks to help you deploy Red Hat OCP on ProLiant DL servers: https://github.com/hewlettpackard/hpe-solutions-openshift

Summary

In this module, you first learned about the architecture for Red Hat OpenShift, and how it builds off of Kubernetes architecture. Then, you learned about how you and your customers can enjoy one single purchase point with HPE for Red Hat products, including Red Hat OCP, Data Foundation, and Ansible Automation. You then learned about planning Red Hat deployments, based on criteria ranging from the stateful or stateless nature of the application, to the HPE hardware that your customer will want to use for their workloads.

Learning checks

1. What is a consideration for deploying containers on bare metal servers versus VMs?
   a. Bare metal servers provide the best option for public cloud deployments.
   b. Bare metal servers can offer better performance for latency sensitive workloads.
   c. Companies tend to be more familiar with automating provisioning of bare metal servers.
   d. Companies must use bare metal servers if they want to use Kubernetes orchestration.
2. Match the HPE compute option to its use case in a Red Hat OCP environment.

   ___ HPE Synergy

   ___ HPE ProLiant

   ___ HPE Apollo

   ___ HPE SimpliVity

   a. The best choice for customers who want rack servers
   b. The best choice for customers who are looking for storage-density for the software-defined storage layer
   c. The best choice for customers who want a blade format for their servers
   d. The best choice for customers who need to run container workloads in small data centers