13

OpenShift Plus – a Multi-Cluster Enterprise Ready Solution

In this book, we discussed in ***Chapter 1***, *Hybrid Cloud Journey and Strategies*, the main challenges related to public cloud usage, such as keeping cloud costs under control and having consistent, secure, and compliant platforms, no matter which cloud provider is being used. We also introduced some of the tools that are helpful to work with multi-clusters and help to address those challenges, from a technology perspective.

In ***Chapter 9***, *OpenShift Pipelines – Tekton*, and ***Chapter 10***, *OpenShift GitOps – ArgoCD*, we covered **CI/CD pipelines** and **GitOps** using OpenShift, including deployment into multiple clusters.

In ***Chapter 11***, *OpenShift Multi-Cluster GitOps and Management*, we saw how to use **Red Hat Advanced Cluster Management** to manage and observe several clusters from a single pane, deploy workloads into many clusters at once, and also use policies to ensure all your clusters are compliant.

Finally, in the previous chapter – ***Chapter 12***, *OpenShift Multi-Cluster Security* – we introduced the **Red Hat Advanced Cluster Security** tool, which helps to enhance the security of all your clusters, by implementing security policies, managing the known vulnerabilities, assessing the security risks and compliance, and managing the network traffic.

Now, we have two different focuses in this chapter, the first is to introduce the last important piece for the hybrid cloud strategy: **Red Hat Quay**, which will store all container images in a central image registry and makes your CI/CD process a bit easier and more robust; and the second is to discuss the **Red Hat OpenShift Plus** offering, which bundles all these pieces together to provide you a comprehensive and competitive solution.

Therefore, you will find the following in this chapter:

- Introducing Red Hat Quay
- Deploying Red Hat using the Quay Operator
- Using Red Hat Quay
- What is OpenShift Plus?
- OpenShift Plus: A practical use-case

Let's start now then!

Introducing Red Hat Quay

Red Hat Quay is an enterprise container registry platform that runs on Red Hat Enterprise Linux or OpenShift, on-premise or in the cloud. Red Hat Quay provides great features for an image registry, such as the following:

- **Image vulnerability scan**: Scan the images to find known vulnerabilities just after they are pushed to the registry using the Clair project. See more information about Clair in the *Further reading* section of this chapter.
- **Geo-replication**: Sync the image registry contents between two or more Quay instances, allowing multiple and geographically distributed Quay deployments to look like a single registry. This is especially helpful for environments spread over far distances and with high latency. Quay handles asynchronous data synchronization between the different instances to make images available transparently for the end user.
- **Repository mirroring**: This synchronizes one or more repositories between two Quay instances.
- **Access control**: Granular permission control allows you to define who can read, write, and administer groups and repositories. You can also leverage robot accounts to allow automated access to organizations and repositories using the Quay API.

Red Hat Quay is available as a managed offering on the **quay.io** portal or self-managed, where you deploy and maintain it.

Deploying Red Hat Quay using the Quay Operator

There are different ways to deploy Quay, as you can see in the product's official documentation link that you'll find in the *Further reading* section of this chapter. For didactical reasons, we decided to deploy it using the Quay Operator with fully-managed components – a comfortable way to start using Quay.

Some prerequisites are necessary to install Quay as an enterprise container registry.

Prerequisites

The prerequisites to install Quay using the Operator are as follows:

- An OpenShift cluster with a privileged account to deploy and set up Quay.
- Object storage: The supported object storage is Red Hat OpenShift Data Foundation, AWS S3, Google Cloud Storage, Azure Storage, Ceph/RadosGW Storage/Hitachi HCP storage, Swift storage, and NooBaa.
- Cluster capacity to host the following services:

- PostgreSQL or MySQL database. PostgreSQL is preferred due to enhanced features for Clair security scanning.
- A proxy server.
- A key-value database. Redis is the default option to store non-critical Red Hat Quay configuration.
- Quay, the application service itself.

With all prerequisites met, we can install the Quay Operator. See next how to install it.

Operator installation

The process to deploy the Quay Operator is simple. Follow the instructions in the next steps:

1. Navigate to **OperatorHub** and search for **`Red Hat Quay`**.

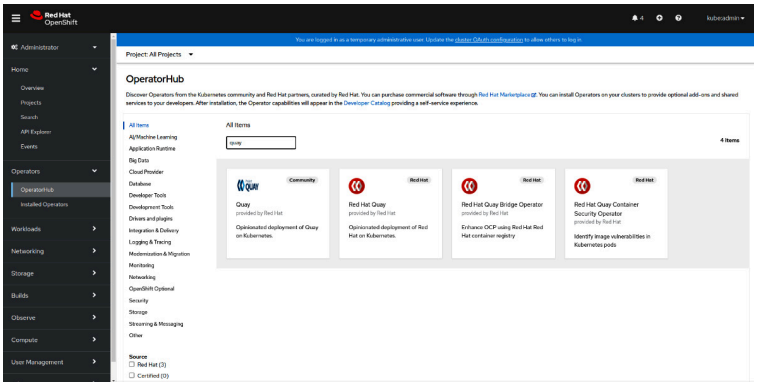

Figure 13.1 – Seach for the Red Hat Quay Operator

2. Click on the tile and then on the **Install** button.

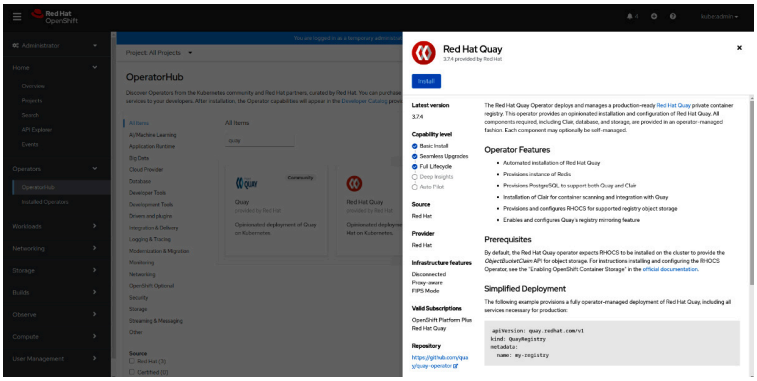

Figure 13.2 – Install Red Hat Quay

3. Leave the default options and click on the **Install** button.

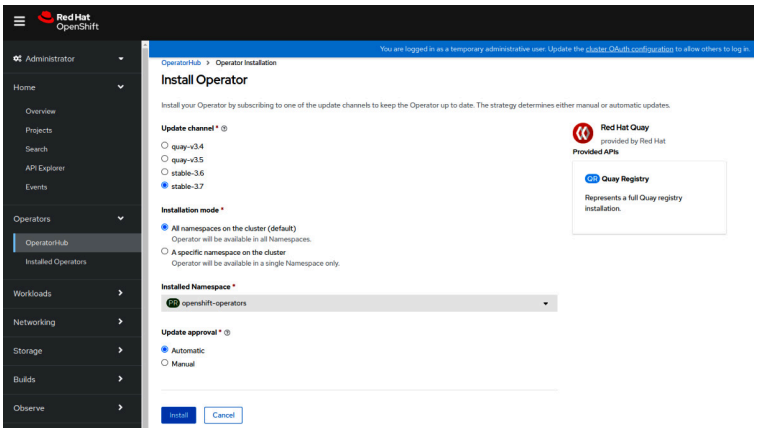

Figure 13.3 – Operator installation

4. When the Operator is installed, access the **Quay Registry** tab and click on **Create instance**.

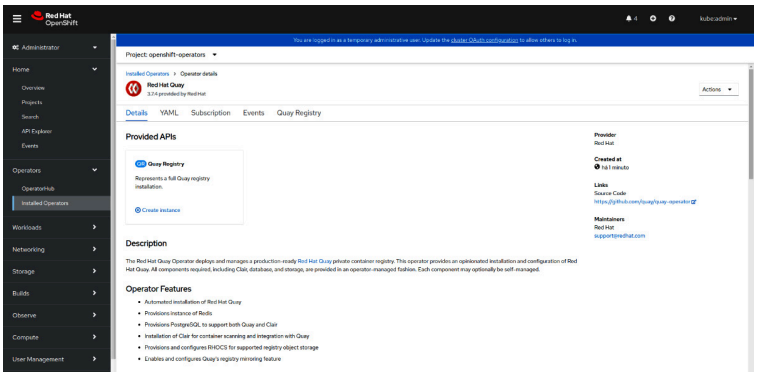

Figure 13.4 – Creating a Quay Registry instance

5. Optionally, change the name in the proper field and leave all default options.

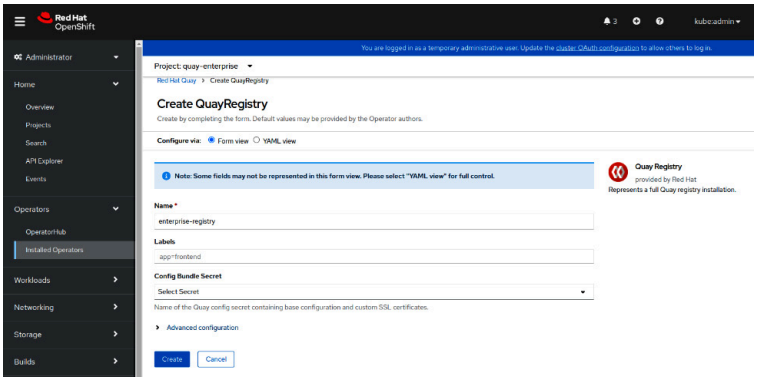

Figure 13.5 – Quay instance creation

6. Check the **Quay Registry** tab. You will see the **Status** column similar to the following screenshot.

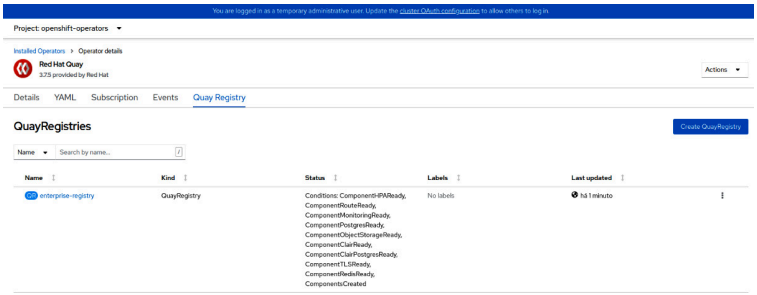

Figure 13.6 – Quay instance created

7. As soon as it finishes, click on the registry that has been just created and inspect the **Registry Endpoint** URL.

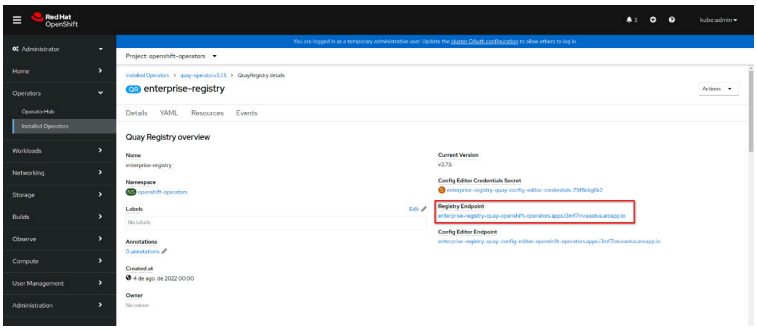

Figure 13.7 – Quay endpoint

Click on the registry endpoint address. The Quay Registry application will be shown to configure your multi-cluster registry.

Configuring Quay

Once the installation process finishes, you will have access to some important endpoints that are used to configure the integration between the Quay Operator, the Quay Registry instance, and OpenShift. To use them, you need to follow the next steps:

1. In the namespace in which you installed Quay (in our case, the **`openshift-operators`** namespace), navigate to **Workloads** | **Secrets** and search for a secret starting with **`<objectInstanceName>-quay-config-editor-credentials-<randomPodId>`**.

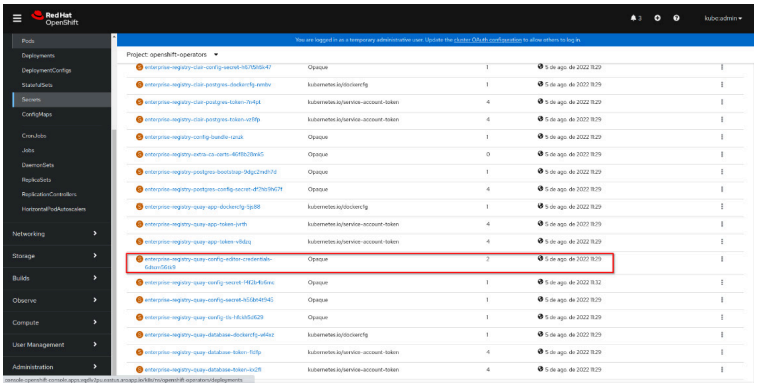

Figure 13.8 – Check Quay config credentials

2. Click on it and choose **Reveal Values**.

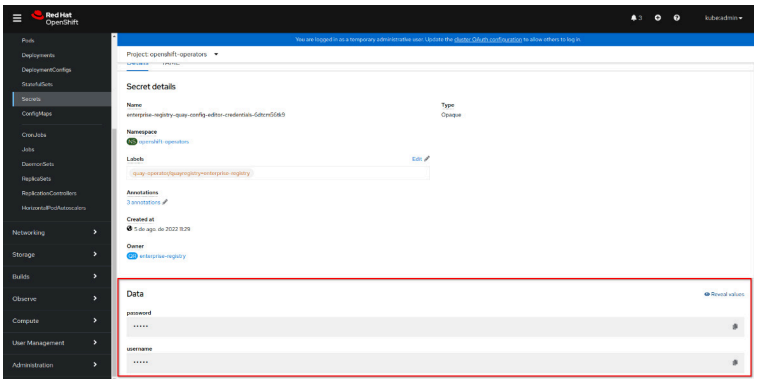

Figure 13.9 – Quay credential values

3. Take note of the credentials and navigate back to **Config Editor Endpoint** in the Quay Registry Operator instance.

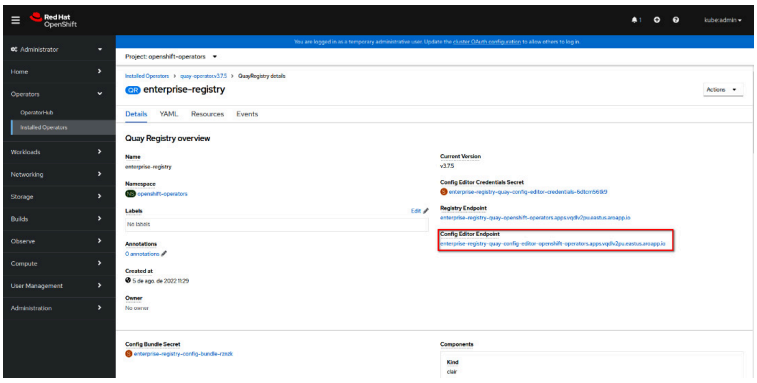

Figure 13.10 – Quay config endpoint

4. Click the link to access **Config Editor Tool**. Use the credentials you got in the previous step.

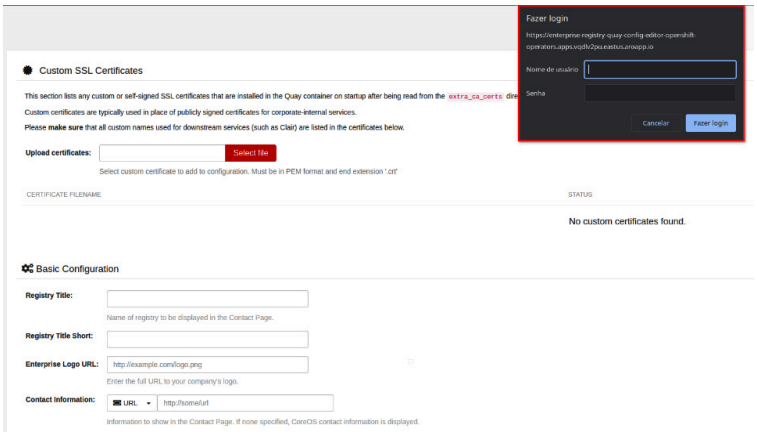

Figure 13.11 – Quay UI config tool

5. On this page, you can change many Quay configurations, such as SSL certificates, and so on – in our example, we changed the **Time Machine** expiration time. Click on **Validate Configuration Changes** after you make your changes.

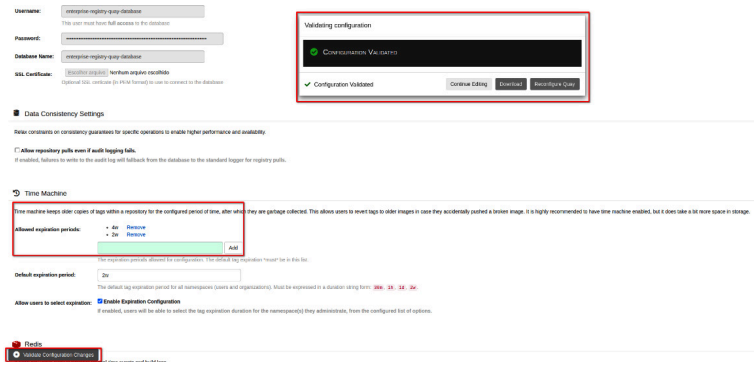

Figure 13.12 – Quay settings config tool

6. After the configurations are validated, choose **Reconfigure Quay**.

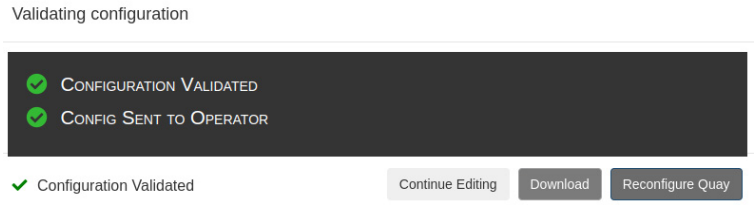

Figure 13.13 – Quay config tool validation

IMPORTANT NOTE

This step will restart the operator pods and Quay may become unavailable for a short period.

7. The Quay Config Editor tool will automatically update the **`quay-config-bundle`** secret, used to store the configuration parameters. To see it navigate to **Workloads** | **Secrets**, search for the **`quay-config-bundle`** secret and click on **Reveal values**. The changes made previously must be reflected inside this secret.

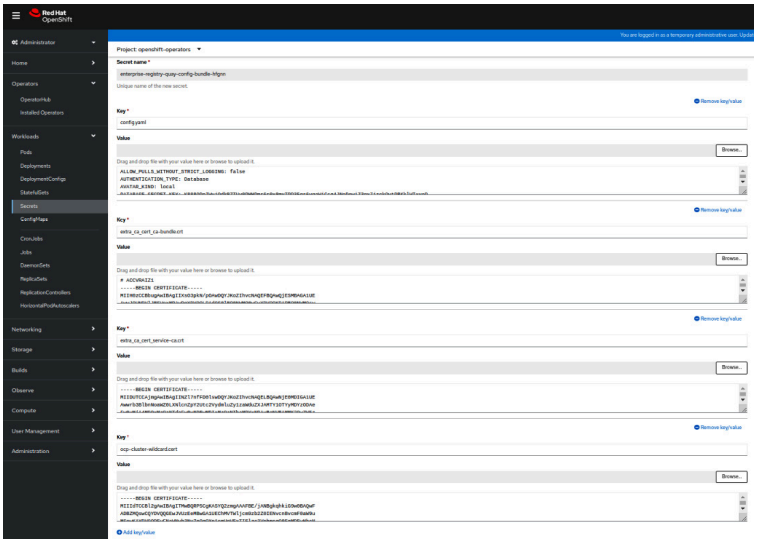

Figure 13.14 – Quay secret after config tool changes

If everything appears correct, you will be able to start using your Quay instance. Proceed with the next section to see instructions on how to use Quay.

Using Red Hat Quay

Red Hat Quay usage is very simple. A user-friendly interface will help you to configure additional global and security settings. We recommend you create an *organization* to organize your Quay registry by departments, regions, or any other division you want:

1. In this example, we created an organization named **multicluster-book**.

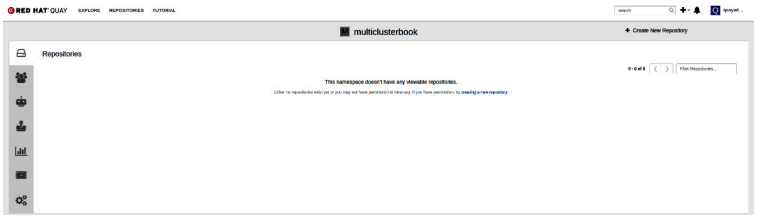

Figure 13.15 – Quay organization

2. Next, we created a private repository to upload our images.

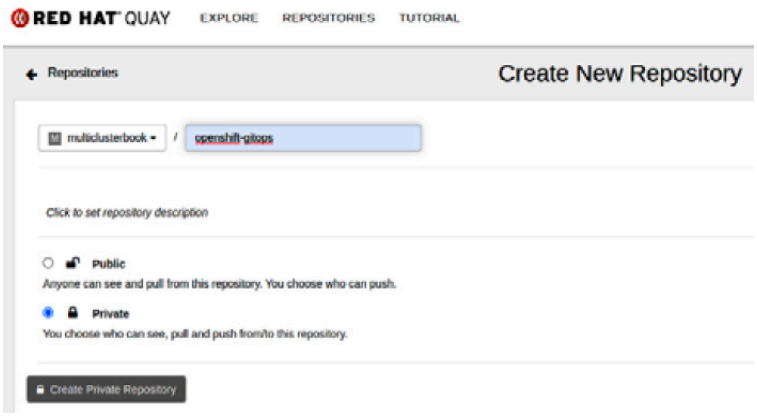

Figure 13.16 – Quay repository

A very helpful way to learn how to use Red Hat Quay is using the tutorial. You'll see how to do that.

Running the tutorial

Running the tutorial mentioned in the Quay UI is not required but it can be a good starting point to understanding Red Hat Quay as an enterprise container registry. The first step of this tutorial is to log in via the terminal to the Quay registry endpoint. We are using **podman** as the container engine but you can alternatively use **Docker** instead if you prefer, both will work similar, so you just need to replace podman with Docker in this case (for example, **`docker login`** instead of **`podman login`**, **`docker run`** instead of **`podman run`**, and so on):

```
$ podman login enterprise-registry-quay-openshift-operators.apps.vqdlv2pu.eastus.aroapp.ic
```

You should have a successful login, otherwise, check the credentials used. After, try to run a **BusyBox** container image.

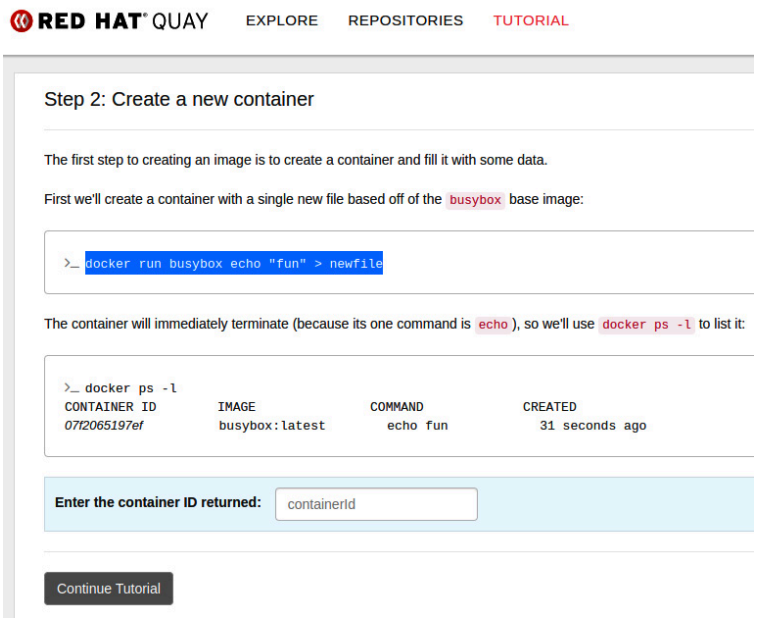

Figure 13.17 – Quay tutorial

Now, run the following commands to create a sample container image for this repository:

```
$ podman run busybox echo "fun" > newfile
$ podman ps -l
CONTAINER ID  IMAGE                                COMMAND     CREATED         STATUS
fd3b51f4c383  docker.io/library/busybox:latest  echo fun    16 seconds ago  Exited (0) 17
$ podman commit fd3b51f4c383 enterprise-registry-quay-openshift-operators.apps.vqdlv2pu.ea
$ podman push enterprise-registry-quay-openshift-operators.apps.vqdlv2pu.eastus.aroapp.io/
```

Simple, right?! Now your instance of Quay Container Registry is configured, tested, and ready to use. Start exploring it in your CI/CD pipelines, pushing images to it, and using it as the enterprise container registry for your DevOps and DevSecOps pipelines.

Now that we've already covered Quay, let's discuss a bit more about OpenShift Plus. Next, you'll continue to discover why you should care about and consider OpenShift Plus in your hybrid or multi-cloud strategy.

What is OpenShift Plus?

OpenShift Plus is an offering from Red Hat that bundles the OpenShift platform with the products you can see in the following diagram:

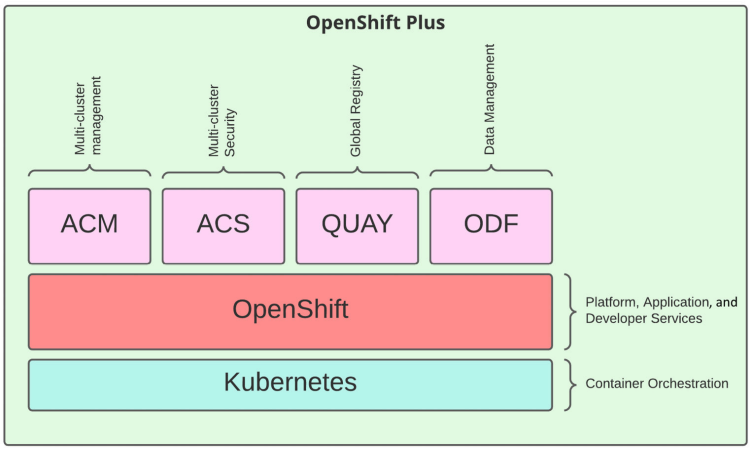

Figure 13.18 – OpenShift Plus products offering

These products are included with this offering(besides OpenShift itself):

- **Advanced Cluster Management**: Provides features to manage several OpenShift and Kubernetes clusters. The features include cluster management, observability, policy and compliance, and workload deployment. We dove into this product in ***Chapter 11***, *OpenShift Multi-Cluster GitOps and Management*. If you haven't seen that chapter yet, we highly encourage you to go back and walk through it now.
- **Advanced Cluster Security**: Enhance the security of your Kubernetes and OpenShift clusters with features such as Vulnerability Management, Network management, and others. This product was covered in ***Chapter 12***, *OpenShift Multi-Cluster Security*. We also encourage you to check this chapter now if you haven't done so yet.
- **Quay Enterprise Registry**: The global enterprise registry was covered at the beginning of this chapter.
- **OpenShift Data Foundation**: This is a storage layer that can provide different types of volumes to OpenShift clusters: file storage (for **`RWX`** persistent volumes) using **Cephfs**, block storage (for **`RWO`** persistent volumes) using **Ceph RBD**, and object storage provided by a solution named **Noobaa**. This product is not covered in this book. If you want to see more information about it, check the links we left in the *Further reading* section of this chapter.

If you have followed this book from the beginning, this is not new information for you. We will focus now on giving additional information that can help you to understand the value proposition behind OpenShift Plus and why you should consider OpenShift Plus, instead of *just* the OpenShift platform.

Value proposition: benefits

OpenShift Plus adds some great benefits to your multi-cluster strategy:

- **Consistency**: ACM provides the governance feature, which helps you to have consistent configurations among all your clusters. In terms of security, ACS also implements security policies you can define that will also help to keep all clusters compliant with them.

- **Portability**: Quay provides a global image registry that all clusters pull images from. ACM also can help you to deploy applications against many clusters at once and to inspect and monitor their status. By using it together with OpenShift GitOps, you can leverage **ArgoCD** to implement **GitOps** deployment practices and maintain the desired state of your applications, no matter in which OpenShift cluster and cloud provider you are running them. Of course, the portability between clusters also depends on the application, as the more decoupled the application is, the easier it will be to port between different OpenShift clusters and providers. The important thing to understand is that ACM and Quay help to make your applications portable much easier. OpenShift Data Foundation also helps to make stateful workloads easier to port, by providing a standard way for applications to have access to Persistent Volumes, regardless of the infrastructure or cloud providers the clusters are running on.
- **Management**: Managing several clusters in different environments is not an easy task in general. Companies usually have many people focused only on doing that. You can turn this activity into a bit of an easier task and perhaps decrease the costs associated with it by using ACM. Features such as cluster management, observability, and governance (policies) are helpful to manage several clusters from a single pane.
- **Security**: As we discussed in ***Chapter 12***, *OpenShift Multi-Cluster Security*, making a few clusters secure and compliant in general is not a hard task, however, when we scale to several clusters it becomes more challenging. ACS adds the capability to easily enforce security policies, detect vulnerabilities, and plan actions to remediate them.

Besides OpenShift Plus, Red Hat also provides some additional value-added products available in the Red Hat Hybrid Cloud portal (**cloud.redhat.com**) for any OpenShift customer:

- **OpenShift Cluster Management**: This allows you to view high-level cluster information, create new clusters using the Assisted Installer, configure Red Hat subscriptions, basic level monitoring, and more.
- **Cost Management**: Cost management is an interesting tool to track costs and usage of AWS, Google Cloud, Azure cloud providers, and OpenShift clusters as well. It helps you to have a consolidated view of the costs associated with multiple cloud accounts and also across your hybrid cloud infrastructure, to inspect cost trends, and to even use it as a basis for charging departments or teams by their usage. See the link for more information about this tool in the *Further reading* section of this chapter.

In the next screenshot, you see an example of what the **Cost Management Overview** screen looks like.

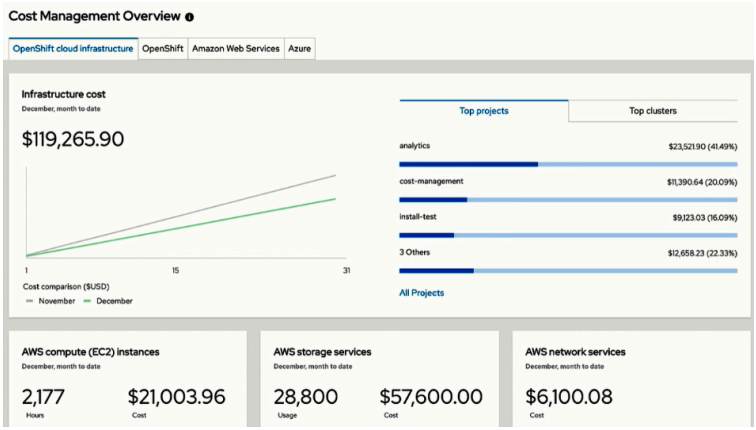

Figure 13.19 – Cost Management feature

- **Developer Sandbox**: This is a free OpenShift environment in a shared, multi-tenant OpenShift cluster pre-configured with a set of developer tools. Use it as a sandbox for your tests and as an environment to learn how to develop for OpenShift. You will find the link for it in the *Further reading* section of this chapter.

Besides the items listed in the preceding list, OpenShift Plus may make sense from the subscription cost as well. Any package is usually priced better than buying individual pieces separately. That being said, if you are interested in any software that is part of the *Plus* stack mentioned in this chapter, it might make sense to buy OpenShift Plus, instead of OpenShift and the software only. Consult your Red Hat account team and ask for a quote for OpenShift Plus.

Now that we've already discussed some of the benefits of having the OpenShift Plus stack and considerations for when it is really compelling, let's use a fictional use case as an example to make it more tangible.

OpenShift Plus – a practical use case

Let's consider the following use case. Try to use the concepts we covered in this book to identify which tool you could use to cover each requirement.

Use case: The company *HybridMyCloud* is developing an AI/ML application that has the following requirements:

- The application life cycle is comprised of Development, QA, Pre-prod, and Production environments.
- Development and QA can share the same underlying platform. Pre-prod and Production need to be separated due to regulatory needs. Production application needs to be in a private subnet, with no public direct access from the internet.
- The C-level and procurement department of *HybridMyCloud* signed a contract with AWS and, due to that, wants to host most resources there. However, they don't want to be locked into it and want to have the ability to move the application to Azure when needed.

- This AI/ML application is based on containers and stateless, however, it stores images in S3 object storage. It receives an X-Ray image, stores it in S3 object storage, and uses an ML-trained model to make a risk assessment of pneumonia.
- This application handles **Protected Health Information** (**PHI**), therefore *HybridMyCloud* wants to be compliant with the **Health Insurance Portability and Accountability Act** (**HIPAA**).
- This is a mission-critical application, so the enterprise architects of *HybridMyCloud* decided to have this application deployed not only on multi-availability zones but also using multiple regions on AWS.
- Other policies defined by *HybridMyCloud* are as follows:
  - All containers need to have resource requests and limits for CPU and memory.
  - Never use the latest tag images.
  - All images used need to come from the enterprise registry. Pulling images from any other sources other than the enterprise registry and Red Hat certified image registries is strictly prohibited.

So, how would you fulfill all these requirements? There are many different options to match these requirements; OpenShift Plus is definitely a good option that can comply with all of them. Let's discuss what the infrastructure would look like and how OpenShift Plus tools would be used with this use case:

Several OpenShift clusters are required to fulfill these requirements:

- One cluster for Development and QA
- One cluster for Pre-prod
- Two clusters for Production, each one in a different AWS region
- One spare cluster on Azure that can host the application at any given time

You could use the Cluster Lifecycle Management feature of **Advanced Cluster Management** first to deploy these clusters on AWS and Azure. For S3 storage, either the native cloud provider services can be used or, alternatively, use the **OpenShift Data Foundation** with *Multi-Cloud Object Gateway*, which would add the following benefits:

- Use Object Bucket Claims to create S3 volumes.
- Use the *Multi-Cloud Object Gateway* to mirror the volume between AWS and Azure.

With clusters in place, **Advanced Cluster Management** can also be used for the following:

- **Advanced Cluster Management** in conjunction with **OpenShift GitOps** can be used to deploy the application in multiple clusters at the same time.
- Policies can be created to set **LimitRange** objects in all namespaces of each cluster and also edit the **`allowedRegistries`** field of the **`image.config.openshift.io/cluster`** custom resource to only allow images that come from the enterprise registry to run.

- Use the observability feature to monitor the application and cluster health.

Quay can be used as the enterprise registry and the embedded image vulnerability scanning tool can be used as one of the sources to find known vulnerabilities. ACS can be used to detect known vulnerabilities in all clusters and assess them using HIPAA policies. It also can be used in the CI/CD pipeline to perform a static security analysis of manifests and to find anti-patterns, such as the usage of the latest tag image.

These are just a few examples. Other tools we covered in this book can also be used, such as OpenShift Pipelines to implement CI/CD pipelines, Cost Management to track costs with cloud providers, and Red Hat Container Catalog to provide the best container base image, and so on.

Summary

In this chapter, we had a brief overview of Red Hat Quay, an interesting option for a container image registry. We discussed the main benefits of OpenShift Plus and why it makes sense to at least consider OpenShift Plus with your hybrid or multi-cloud strategy. Finally, we used a fictional use case to think about how the OpenShift Plus stack could be used in a practical scenario.

Wow, what a great journey so far! We have covered many things already in this book and we are so pleased that you have come along with us on this journey. We are almost at the end of our long journey here, but the best part is coming: in the next chapter, we will exercise most of what we have learned in this book using one practical example. We will have a comprehensive review of OpenShift Pipelines, GitOps, and all other tools we covered in this book using an example, practice with the concepts and tools from scratch, and have a real functional application deployed into multiple clusters in the end.

We encourage you to move on to the next chapter and try the exercises we propose there.

Further reading

Looking for more information? Check the following references to get more information about Red Hat Quay:

- *Quay product documentation:* **https://access.redhat.com/documentation/en-us/red_hat_quay/3.7**
- *Quay Object Storage prerequisite:* **https://access.redhat.com/documentation/en-us/red_hat_quay/3.7/html/deploy_red_hat_quay_on_openshift_with_the_quay_operator/operator-preconfigure#operator-storage-preconfig**
- *Clair – Vulnerability Static Analysis for Containers:* **https://github.com/quay/clair**

Check the following references to get more information about Red Hat OpenShift Data Foundation:

- *OpenShift Data Foundation Smart Page:* **https://www.redhat.com/en/technologies/cloud-computing/openshift-data-foundation**
- *OpenShift Data Foundation Product Page:* **https://access.redhat.com/documentation/en-us/red_hat_openshift_data_foundation**

Check the following references to get more information about Red Hat OpenShift Plus:

- *OpenShift Plus Smart Page:* **https://www.redhat.com/en/technologies/cloud-computing/openshift/platform-plus**

Check the following references to get more information about Red Hat Cost Management:

- *Official documentation:* **https://access.redhat.com/documentation/en-us/cost_management_service/2022**
- *Upstream project:* **https://project-koku.github.io/**

Check the following references to get more information about OpenShift Developer Sandbox:

- *Getting started:* **https://developers.redhat.com/developer-sandbox**

Previous chapter
‹ Chapter 12: OpenShift Multi-Cluster Security

Next chapter
Chapter 14: Building a Cloud-Native Use Case on a Hybrid Cloud Environment ›