# 1

# Hybrid Cloud Journey and Strategies

Do you want to learn about and operate OpenShift in multiple environments? If you are reading this book, we suppose that the answer is yes! But before we go into the technical details, we want to start this book by making you a proposition: any house construction starts with a foundation, right? In this book, our approach will be the same. We will start by giving you the foundation to understand and create a much stronger knowledge base – you will develop critical thinking and be able to make the best decisions for your use case.

That is why we decided to start this book by not talking about OpenShift itself yet, but by unveiling the most popular (and important!) context that it operates in: **the hybrid cloud infrastructure**. Therefore, in this chapter, you will be introduced to the hybrid cloud journey, challenges, dilemmas, and why many organizations are struggling with it. Knowing about these challenges from the beginning is a determinant **success factor for hybrid cloud adoption**.

Transforming an IT business so that it's agile and scalable but is also stable is a must-have nowadays, but that is not a simple step; instead, it is a *journey from one star to another* in the vast outer space of IT that currently surrounds us. However, why do we need those changes?

Why is the market adopting the cloud massively and so rapidly? We'll discuss that shortly!

*It is a changing world!*

We are living in the age of fast changes! 10 years ago, most of the current big tech companies did not exist or were just small startups; several technologies that we have today also were only known within university research groups, such as 3D printing, artificial intelligence and machine learning, 5G, edge computing, and others – and there is much more to come! Technologies like the ones mentioned previously are becoming more popular and will create several demands that do not exist today, new job positions, and far more changes.

In this rapidly changing world, some new needs became important. Most companies were forced to change to be able to release new software and versions much faster than before, quickly scale resources, and have a global presence with responsive applications.

It was in this context that the public cloud providers have emerged with great success. However, several organizations that made huge investments in the cloud are experiencing some challenges. In a study conducted by *IDG* in 2020, among big companies in different industries and geographies, 40% of the respondents stated that controlling cloud costs is the biggest challenge when taking full advantage of it. This research has also shown data privacy and security as big obstacles. We will walk through some of these challenges in this chapter.

In this chapter, we will cover the following topics:

- Main challenges of the public cloud
- How a hybrid cloud strategy helps mitigate these challenges
- How containers, Kubernetes, and OpenShift help to implement the hybrid cloud
- OpenShift options

- Types of OpenShift installations
- Additional tools to support hybrid cloud adoption

# Main challenges of the public cloud

From small enterprises to big tech companies, most of them face some common challenges when it comes to using and taking full advantage of public cloud providers. Some of the main challenges are as follows:

- **Keeping cloud costs under control**: Estimating and managing the costs of applications running in a public cloud provider is not a simple thing – cloud providers' billing models are multifaceted, with hundreds of different options and combinations, each with a pricing factor. Finding the best cost-benefit for one application can take a significant amount of time. To make things even more complex, cloud costs are usually dynamic and flexible – this may change significantly from time to time by type, duration of the contract, type of computing resources, and so on.
- **Security**: Data privacy and security is one of the major concerns with public clouds, according to the *IDG*'s research – almost 40% of them classified it as the top challenge. That is, it is naturally much more difficult to secure an IT environment that comprises of multiple different providers than the old days, in which the IT department usually only had a few on-premises environments to manage.
- **Governance, compliance, and configuration management**: Multiple providers mean different offerings and standards, probably different teams working with each of them, and, consequently, heterogeneous environments.
- **Integration**: Organizations that have legacy services and want to integrate with their applications, which are hosted in the cloud, usually face some dilemmas on the best way to do those integrations. While cloud providers virtually have no limits, when you integrate your applications with your legacy infrastructure, you

might be creating a **harmful** dependency, which will limit their scalability. However, mainly for big enterprises, those integrations are inevitable, so how can we prevent dependency issues (or at least minimize them)?

- **Vendor lock-in**: A common concern when adopting cloud providers is often related to being locked in with a single vendor and the business risks associated with it. I would say that there is a thin line between getting the best price from the cloud provider and being locked into their services. What could happen to the business if the cloud provider decides to raise prices in the next contractual negotiation? Is this a risk your business can afford? How can we mitigate it? Here, the quote *you get what you pay for* is suitable!

- **Human resources and enablement**: Hiring and keeping talented people in IT has always been a hard task; cloud technologies are no different. Cloud engineer, Architect, SRE, Cloud Native Application Developer – these are just a few job positions that open every day, and most companies struggle to fill them. Hiring, training, and maintaining a skilled team to develop and operate applications in the cloud is a real challenge.

*REFERENCE*

*You can check out the complete IDG research at* **https://www.idg.com/tools-for-marketers/2020-cloud-computing-study/** *[Accessed 30 August 2021].*

# Benefits of the public cloud

We have seen some complex challenges so far. So, you might be thinking, *so you don't like cloud providers and want to convince me to avoid them, right?*

No, of course not! I am sure that without the advent of cloud providers, several companies we use every day (and love!) simply would not exist! Let's point out the good parts, then:

- **Scalability**: Cloud providers can offer almost unlimited and on-demand compute resources.
- **Lower CAPEX**: You don't need to buy any hardware and equipment to start any operations – you can do that with just a few clicks.
- **Resilience and global presence**: Even small companies can distribute services globally among different Availability Zones and Regions.
- **Modern technologies**: Public cloud providers are always looking to bring new and modern offerings, which helps an organization to always be at the edge of the technology.

# Is hybrid cloud the solution?

As we've already discussed, the public cloud, while it can solve some challenges, introduces others. It was in this context that the hybrid cloud emerged: to mitigate some of the challenges and take the best from each provider, from on-premises, private, or cloud providers. The *HashiCorp State of Cloud Strategy Survey*, which was made in 2021 with more than 3,200 technology practitioners, found that multi-cloud is already a reality. 76% of the respondents stated that they are using multiple cloud vendors, with expectations for this to rise to 86% by 2023.

*REFERENCE*

*You can check out the complete HashiCorp research at* **https://www.hashicorp.com/state-of-the-cloud** *[Accessed 31 August 2021].*

So, what are the characteristics of the hybrid cloud that help mitigate the challenges of public cloud adoption? Here are a few of them:

- Best-of-breed cloud services from different vendors can be combined, enabling a company to choose the best option for each workload.
- The ability to migrate workloads between different public and private cloud environments, depending on the actual circumstances.
- Being able to have a single, unified orchestration and management across all the environments for all providers.

The following table lists some of the challenges and hybrid cloud mitigations:

| Challenge | Hybrid Cloud Mitigation |
| --- | --- |
| Cloud costs | Migrating workloads between different environments enables a company to use the best cost-benefit option for each case. |
| Security | The ability to integrate public and private environments allows you to choose the most secure option for any case. |
| Governance, compliance, and configuration management | Hybrid cloud adoption will drive an organization to establish a unified layer of orchestration and management between different providers. This will also help ensure compliance between the different environments. |
| Integration | Hybrid cloud means that the different providers will be integrated somehow – there are multiple ways to do this. |
| Vendor lock-in | The hybrid cloud strategy will give users the flexibility to make the best choices and move between different providers if needed; they won't be locked into one provider. |
| Human resources and enablement | Enablement will always be a challenge for technology. However, establishing a unified layer of orchestration and management between the cloud providers makes an organization less dependent on the knowledge from each cloud provider. |

# Containers and Kubernetes – part of the answer!

Containers have successfully emerged as one of the most important tools to promote better flexibility between applications and infrastructure. A container can encapsulate applications dependencies within a container image, which helps an application be easily portable between different environments. Due to that, containers are im-

portant instruments for enabling the hybrid cloud, although they have several other applications.

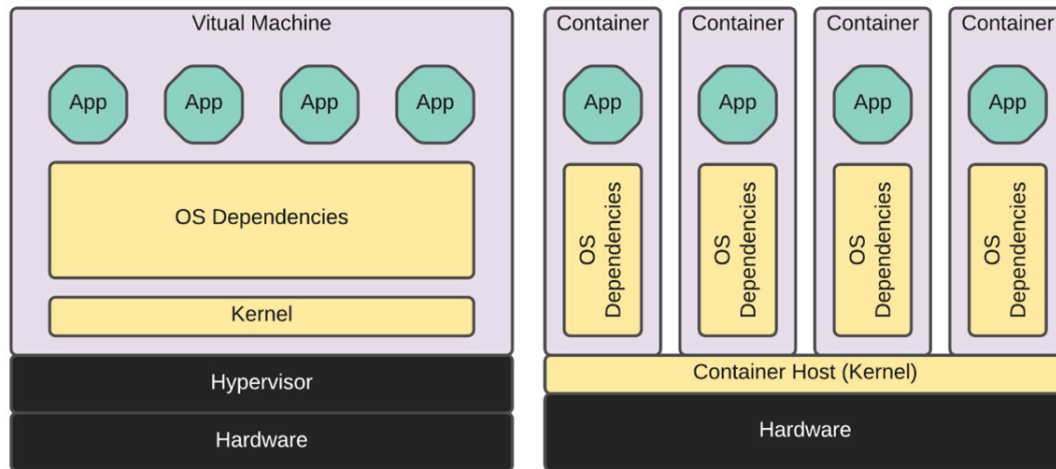The following diagram shows how a container differs from traditional VMs in this matter:



Figure 1.1 – Containers provide flexibility

While containers are beneficial, it is practically impossible to manage a large environment consisting of hundreds or thousands of containers without an orchestration layer. Kubernetes became the norm and it is a great orchestration tool. However, it is not simple to use. According to the *CNCF Survey 2020*, 41% of respondents see complexity as the top barrier for container adoption. When you decide to go for a vanilla Kubernetes implementation, some of the following will need to be defined (among a large set of options) and managed by you:

- Installation and OS setup, including configuration management
- Upgrades
- Security access and identity
- Monitoring and alerts
- Storage and persistence
- Egress, ingress, and network-related options
- Image scanning and security patches
- Aggregated logging tools

*REFERENCE*

*You can check out the complete CNCF Survey here:*
**https://www.cncf.io/blog/2020/11/17/cloud-native-survey-2020-containers-in-production-jump-300-from-our-first-survey/** *[Accessed 1 September 2021].*

# OpenShift – a complete option

OpenShift is one of the most popular platforms based on Kubernetes among enterprise customers. It was first released in 2011, even before Kubernetes was created. However, in 2015, with the release of OpenShift version 3, Red Hat decided to adopt Kubernetes as its container orchestration layer. Since then, they are actively collaborating with the Kubernetes community – Red Hat and Google are the top contributors to Kubernetes. Due to that, it is not a surprise that OpenShift is one of the most mature and complete solutions built on top of Kubernetes.

The following table summarizes some of the features that are included out-of-the-box with the Red Hat **OpenShift Container Platform** (**OCP**) (or easily pluggable):

| Feature | OpenShift Container Platform (OCP) Functionality |
| --- | --- |
| Built-in CI/CD Pipelines, Application Console | OpenShift pipelines(*), OpenShift GitOps(*), Developer Console. |
| Integrated Development Environment | OpenShift CodeReady Workspaces(*) and IDE extensions (VS Code and IntelliJ). |
| Serverless<br>Middleware | OpenShift Serverless(*). |
| Service Mesh | OpenShift Service Mesh(*). |
| Automated Container Builds | S2I, BuildConfig. |
| Dashboard | Administrator and Developer dashboards are available. |

| Feature | OpenShift Container Platform (OCP) Functionality |
|---|---|
| Monitoring | Prometheus and Grafana. |
| Log Aggregation | OpenShift Logging(*). |
| Integrated Container Registry | Included with the platform. |
| Marketplace | Throughout Red Hat's online marketplace (`http://marketplace.redhat.com/`) and also the Operatorhub (`http://operatorhub.io/`), there's a community-driven catalog of operators, which contains operators from Red Hat, certified partners, and also from the open source community. |
| Self-Service Catalog | Included with Developer and Administrator Console using templates, helm charts, builder images, devfiles, or operators. |
| Multi-Tenant Clusters | RBAC, ResourceQuotas, Limits, and NetworkPolicies. |
| Operating System | Support included that's provided by Red Hat. |
| Network | OpenvSwitch or OVN-Kubernetes; internal DNS; ingress controllers (using HAProxy); Multus (for multiple networks); SR-IOV; Egress IP. |
| Storage | Plugins included (in-tree) that integrate with several providers (AWS EBS, Azure Disk, NFS, VMware vSphere, and others). You can also integrate with certified storage vendors using a **Container Storage Interface** (**CSI**). |
| Machine Management | Automated server provisioning and patching. |
| Windows Containers | Support for workers using Windows Server (*). |
| Migration Toolkit for Containers | Stateful workload migration between clusters(*). |
| Kata Containers | OpenShift sandboxed containers(*). |
| Kubevirt (VMs) | OpenShift virtualization(*). |
| Over-the-Air Cluster Upgrades | Simplified upgrades of the entire stack (platform and OS) using the OTA feature. |

*(\*) Need to be installed on day 2*

These features are available for any customer that has a valid OpenShift subscription with Red Hat. However, if you don't have access to a Red Hat subscription, there are some alternatives (for studying purposes):

- You can use some of the trial options provided by Red Hat – check them out at **https://www.redhat.com/en/technologies/cloud-computing/openshift/try-it**.
- Use okd (**http://okd.io/**), which is the community distribution of OpenShift, also powered by Red Hat.
- Use Red Hat CodeReady Container in a VM on your desktop (requires an account on Red Hat's portal). More information can be found at **https://developers.redhat.com/products/codeready-containers/overview**.

We are going to see many of these great features in detail, along with practical examples, in this book.

*REFERENCE*

*The updated statistics about the contributions to the Kubernetes project, grouped by companies, can be found at* **https://k8s.devstats.cncf.io/d/9/companies-table**.

# OpenShift offerings – multiple options to meet any needs

An interesting factor about OpenShift is the vast range of platforms that are supported. With OpenShift version 4.11 (the version that was available when this book was written), you can have the following different combinations to choose from:
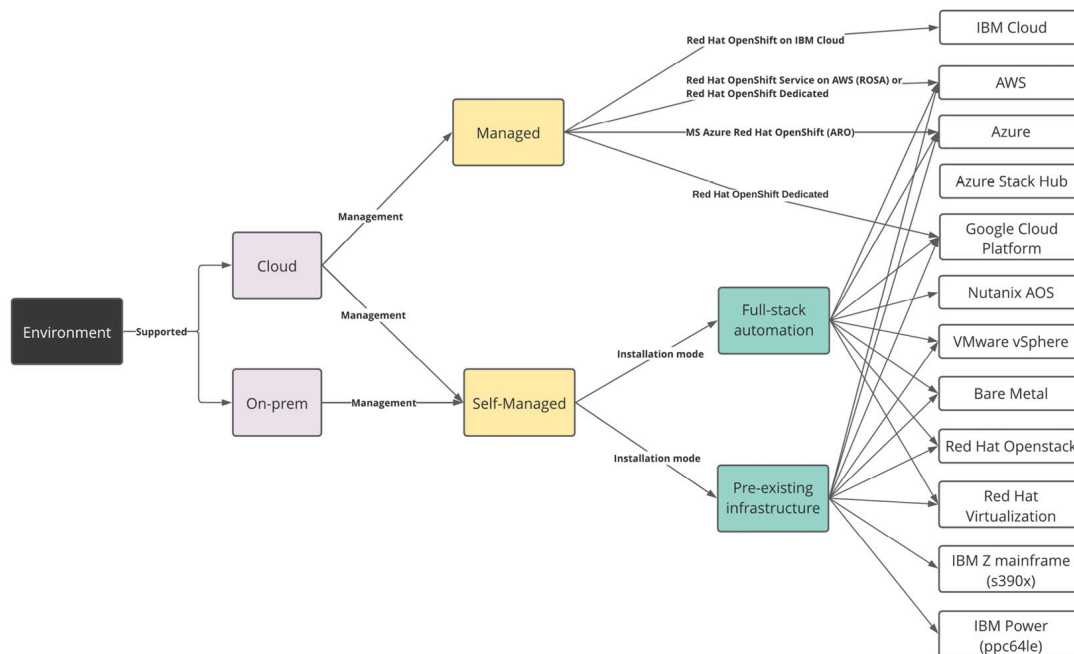


Figure. 1.2 – OpenShift offerings

In this section, we will walk through each of these options.

## OpenShift managed cloud services

In the old days, when we talked about using a certain technology, we also thought about how to deploy and manage it. Nowadays, this is not always true – almost everything now can be found in a Software as a Service model, which you can quickly and easily start using without caring about deployment and management.

The same applies to OpenShift: multiple managed cloud services allow an organization to focus on the application's development and the business while Red Hat and the cloud provider manage the rest.

The following table shows the existing managed offerings at the time of writing this book (check Red Hat for the current options):

| Offering Name | Cloud Provider | Billed By | Managed By | Supported By |
|---|---|---|---|---|
| Microsoft Azure Red Hat OpenShift | Azure | Microsoft | Red Hat and Microsoft | Red Hat and Microsoft |
| Red Hat OpenShift Dedicated | AWS or Google Cloud | 1. Red Hat for OpenShift<br><br>2. AWS or Google Cloud for infrastructure used | Red Hat | Red Hat |
| Red Hat OpenShift on IBM Cloud | IBM Cloud | IBM | IBM | Red Hat and IBM |
| Red Hat OpenShift Service on AWS | AWS | AWS | Red Hat and AWS | Red Hat and AWS |

*IMPORTANT NOTE*

*Note that Red Hat manages the full stack, not only the Kubernetes control plane. Red Hat provides management and version maintenance for the entire cluster, including masters, infrastructure, and worker nodes, though it's not limited to that: it also supports CI/CD, logging, metrics, and others.*

*There are other managed Kubernetes options on the market. Although this is not the focus of this book, keep in mind that some providers don't*

*manage and support the entire stack – only the control plane, for instance. When you're considering a Kubernetes managed solution, see if it is fully managed or only part of the stack.*

## Managed or self-managed – which is the best?

The answer is: it depends! There are several things you need to consider to find out the best for your case, but generally speaking, managed solutions are *not* the best option for organizations that need to have control over the servers and their infrastructure. For organizations that are more focused on application development and don't care about the platform, as long as it is safe and reliable, then managed solutions are probably a good fit.

Managed solutions could also be helpful for organizations that want to put their hands on the platform, evaluate it, and understand if it fits their needs but don't have skilled people to maintain it yet.

Most of this book has been written with a self-managed cluster in mind. However, excluding the chapters focused on platform deployment and troubleshooting, the rest of it will likely apply to any type of OpenShift cluster.

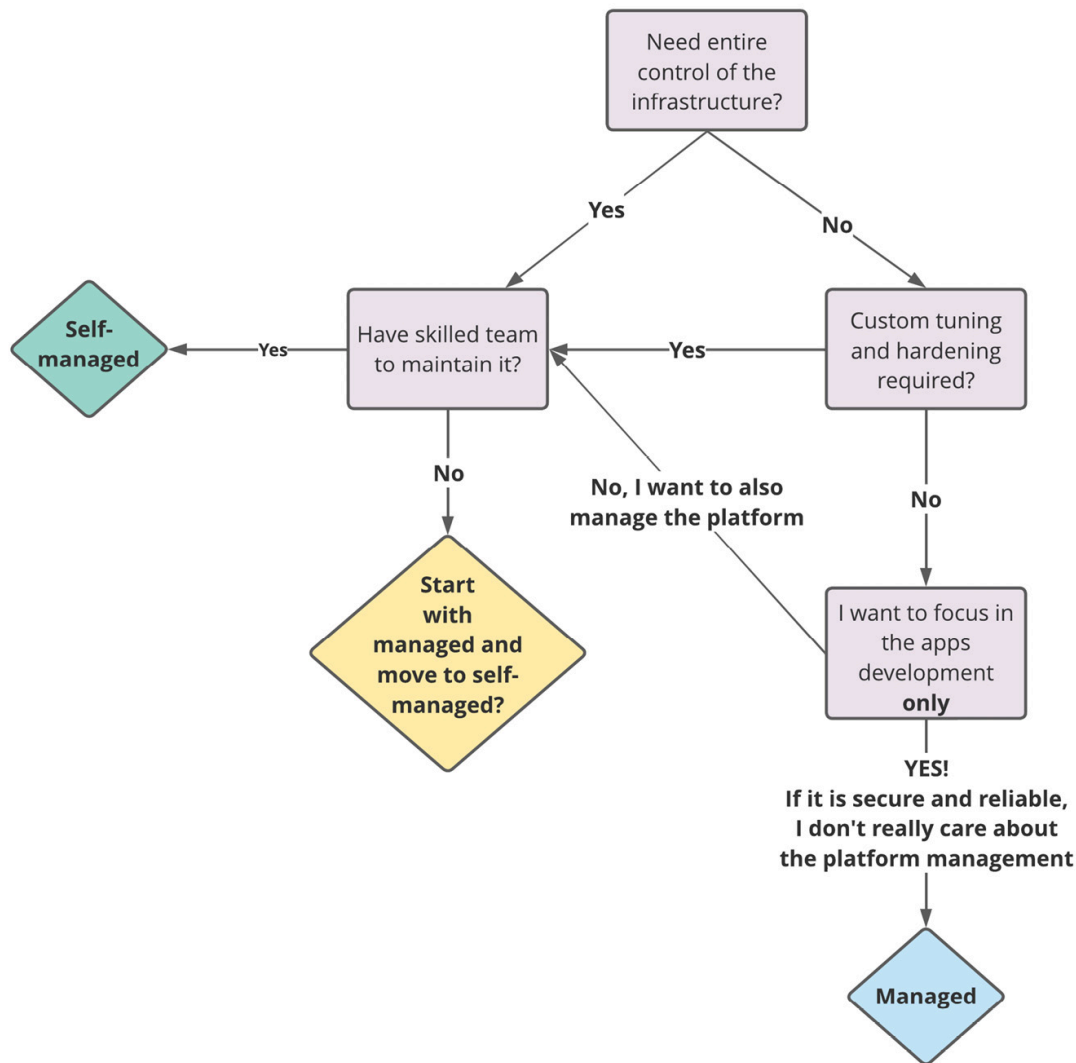The following diagram shows a workflow that aims to help you decide which strategy to go for:

Figure. 1.3 – Managed or self-managed decision workflow

## OpenShift installation modes

There are three installation modes you can use to deploy OpenShift in any of the supported providers, as follows:

- **Full-stack automated** (*installer-provisioned infrastructure*): In this mode, the installer will spin up all the required infrastructure automatically – the installer will integrate with the underlying virtualization or cloud provider to deploy all the machines that are required for the cluster. It is an opinionated fully automated solution that makes the deployment process a lot easier.

- **Pre-existing infrastructure** (*user-provisioned infrastructure*): With this installation, the machines are provisioned manually by follow-

ing some standard images and processes, on top of tested virtual-
ization or cloud providers.

- **Provider-agnostic** (also known as the *bare metal install method*):
  OpenShift is supported wherever Red Hat Enterprise Linux (*) is,
  though this doesn't mean that the installer and platform are tested
  (**) on every infrastructure layer combination that's supported
  with Red Hat Enterprise Linux. In such cases, you can use the
  provider-agnostic installation, which is a manual installation
  process with no integration between the installer and the platform
  with the virtualization or cloud provider.

(*) You can find a list of supported hypervisors for Red Hat Enterprise
Linux at **https://access.redhat.com/certified-hypervisors**.

(**) Please refer to this link for an updated list of tested providers and
integrations with OpenShift:
**https://access.redhat.com/articles/4128421**.

# OpenShift multi-cluster tools – going above and beyond

When it comes to supporting your hybrid or multi-cloud strategy, oth-
er great tools provide single and unified management, security, and
orchestration layers across all environments in all providers. We re-
served the last part of this book to take a deep dive into those tools, but
you must meet them from the beginning to understand the role of
each in the hybrid/multi-cloud picture.

## Red Hat Advanced Cluster Management for Kubernetes – unified management

As we mentioned previously, a single and unified management layer is
important to support the hybrid/multi-cloud strategy. Red Hat
Advanced Cluster Management lets us manage the life cycle, ensure

compliance using policies, and deploy applications on multiple Kubernetes clusters. The following are some of its main features:

- **Unified management**: Create, update, and delete Kubernetes clusters on top of different cloud providers. You can also access, find, and modify Kubernetes resources across the different clusters.
- **Governance, risk, and compliance**: Ensure compliance among multiple clusters using policies. Look for policy violations quickly and remediate them accordingly.
- **Application life cycle management**: Deploy applications across multiple clusters at once. Deploy complex applications by integrating Advanced Cluster Management with Red Hat Ansible Automation Platform to configure networks, load balancers, and other external dependencies.
- **Multi-cluster observability**: Check the health status of multiple clusters from a single point using out-of-the-box dashboards and metrics.

We will dive into Red Hat Advanced Cluster Management using practical examples in the last part of this book.

## Red Hat Advanced Cluster Security for Kubernetes – securing applications no matter where they are

Security is becoming increasingly important for Kubernetes users. When you have multiple Kubernetes clusters spread among different providers, ensuring security and having a real notion of the current vulnerabilities is a real challenge. Red Hat Advanced Cluster Security aims to help with that – through it, you can easily *scan container images to find known vulnerabilities, audit workloads, and clusters using industry standards such as NIST, PCI, and others, analyze network traffic, and create policies accordingly, among other great features*. You can apply all of these features to multiple different clusters, which helps you keep all your environments secure, no matter where they are.

We will look at Red Hat Advanced Cluster Security using practical examples in the last part of this book.

### Red Hat Quay – storing and managing container images in a central repository

A central container image registry isn't usually a required tool. However, deploying applications on several clusters without it makes the build and deployment activity a bit challenging. Red Hat Quay is a container image registry that provides not only the usual capabilities of an image registry (storing your container images) but also provides image vulnerability scans, a time machine, replication, garbage collection, automated builds, authentication, authorization, and more.

We will learn how to use Red Hat Quay in the last part of this book.

### OpenShift Plus – the whole package

Red Hat OpenShift, Advanced Cluster Management, Advanced Cluster Security, and Quay are different products. However, with the OpenShift Plus package, you can have all of them in one subscription only, which is probably the best way to go if you are planning to adopt a hybrid or multi-cloud Kubernetes strategy.

We will cover OpenShift Plus in more detail with practical examples in the last part of this book.

## Summary

In this chapter, we looked at the main challenges of public cloud usage and how the hybrid cloud helps mitigate some of them. You now understand how containers, Kubernetes, and OpenShift can help you implement a successful hybrid cloud strategy. Finally, we learned about the different types of OpenShift offerings and additional tools that support hybrid cloud adoption.

In the next chapter, you will learn about the architectural aspects of an OpenShift cluster.

# Further reading

If you want to find out more about the concepts that were covered in this chapter, check out the following references:

- The following are the public studies and surveys that were mentioned in this chapter related to hybrid cloud and container adoption:
  - *IDG 2020 Cloud Computing Study:* **https://www.idg.com/tools-for-marketers/2020-cloud-computing-study/**
  - *HashiCorp State of Cloud Strategy Survey:* **https://www.hashicorp.com/state-of-the-cloud**
  - *Cloud-Native Survey 2020: Containers in production jump up 300% from our first survey:* **https://www.cncf.io/blog/2020/11/17/cloud-native-survey-2020-containers-in-production-jump-300-from-our-first-survey/**
- *Red Hat OpenShift landing page:* **https://www.redhat.com/en/technologies/cloud-computing/openshift**
- *Red Hat Managed cloud services landing page:* **https://www.redhat.com/en/technologies/cloud-computing/openshift/managed-cloud-services**
- *Red Hat Advanced Cluster Security for Kubernetes landing page:* **https://www.redhat.com/en/technologies/cloud-computing/openshift/advanced-cluster-security-kubernetes**
- *Red Hat Advanced Cluster Management for Kubernetes landing page:* **https://www.redhat.com/en/technologies/management/advanced-cluster-management**
- *Red Hat Quay landing page:* **https://www.redhat.com/en/technologies/cloud-computing/quay**
- *Red Hat OpenShift Platform Plus landing page:* **https://www.redhat.com/en/technologies/cloud-computing/openshift/platform-**

**plus**

- *OpenShift Container Platform 4.x Tested Integrations and Supportability Matrix:* **https://access.redhat.com/articles/4128421**
- *OpenShift Container Platform installation overview (from official documentation):* **https://docs.openshift.com/container-platform/latest/installing/index.html**
- *Supported installation methods for different platforms (from official documentation):* **https://docs.openshift.com/container-platform/latest/installing/installing-preparing.html#supported-installation-methods-for-different-platforms**
- *Kubernetes statistics page:* **https://k8s.devstats.cncf.io/**

| Feature | OpenShift Container Platform (OCP) Functionality |
|---|---|
| Built-in CI/CD Pipelines, Application Console | OpenShift pipelines(*), OpenShift GitOps(*), Developer Console. |
| Integrated Development Environment | OpenShift CodeReady Workspaces(*) and IDE extensions (VS Code and IntelliJ). |
| Serverless Middleware | OpenShift Serverless(*). |
| Service Mesh | OpenShift Service Mesh(*). |
| Automated Container Builds | S2I, BuildConfig. |
| Dashboard | Administrator and Developer dash-boards are available. |

**Previous chapter**

‹ [Part 1 – Design Architectures for Red Hat OpenShift](#)

**Next chapter**

[Chapter 2: Architecture Overview and Definitions](#) ›