Kubernetes: The Simple Way?

Introduction

As microservices architecture becomes more and more common in IT, enterprise companies are now beginning to look at the benefits of this. This challenge raises important strategic questions around how to do it securely, with the right amount of investment from an infrastructure and people perspective.

The early adopters have been using containerisation (mainly docker) for a while now, and it soon became clear they required something to orchestrate and manage these containers. For a short period of time there were several competing orchestration engines, for example MesOS, Docker Swarm, Kubernetes. As of now, it's clear Kubernetes won the race for the title of most used: this technology is rapidly becoming an Industry standard.

Kubernetes container orchestration is fascinating, but as with every fascinating new cutting-edge technology, comes the steep learning curve and heavy investment for IT organisations. Operations, developer teams, and security teams all need to educate themselves in the topic, a keen understanding of the technology is vital. This is a huge investment from an enterprise company's point of view. Usually this is one of the reasons that large companies have technical debts, and can find themselves playing to catch up with leaner, more agile, startup companies. IT leaders need to be sure that a technology is mature enough, has official training, the right amount of community-based knowledge and other enterprise companies have adopted the technology.

Kubernetes helps companies to provide better service for their end users. It allows IT to react faster to changes in business focus, implement new features, and have higher availability and reliability of services. To achieve all this, Kubernetes comes with solutions such as self-healing, node-pools, readiness and liveness probes, autoscaling etc. To be able to implement and understand all these, as alluded to earlier, the learning curve is steep, as this technology differs greatly from traditional IT.

Operation teams need to understand :

- how to design/deploy/operate Kubernetes clusters;
- the basic components including master nodes with scheduler, etcd, kube controller, kubectl, API server and worker nodes with kubectl, kube proxy.
- general concepts such as Kubernetes networking, Pods, or objects such as deployments, replicas, ingress/egress
- how to implement better observability with dashboards, how to monitor, implement logging, etc.

Developer teams need to understand:

- how to separate their monoliths into microservices.
- how to rewrite applications to communicate through APIs.
- how to use SaaS solutions from external providers as components of their application.
- how to containerize their application code with all its required libraries and dependencies, and how to write dockerfiles to achieve this.
- last bot not least the obvious one not mentioned yet: the use of container registries.

Security teams need to understand:

- how to secure Kubernetes clusters.
- how to secure the containers running on those.

- how to keep containers, the code and all the components updated to have a secure environment.
- how to separate access, implement RBAC.
- how to secure environments to adhere to regulations, etc.

Unfortunately, with vanilla Kubernetes, many of these are not included out of the box. Most of the previously mentioned challenges need to be tackled with additional, third party components or require manual configuration. For many enterprise companies, this will require additional agreements, with (perhaps) multiple vendors, to achieve the desired state of the infrastructure. As a result, support processes can be made more complex when issues arise, which can create additional frustration when a quick resolution is required.

Managed Kubernetes

So how is it possible to make this simpler, how can Kubernetes be "simple"?

Hyperscalers such as Azure, AWS and GCP are already offering "managed" Kubernetes solutions, AKS in Azure, EKS in AWS and GKE in GCP. These products provide a quick and easy solution for Operations to deploy and manage Kubernetes infrastructures. The master-node components are managed and operated by the Hyperscalers, in Azure AKS these are provided free of charge (correct at the time of writing this article 29.04.2020), and Operations only need to focus on the management of the worker-nodes, the deployments and network access. All these solutions are packaged with a cloud-centric monitoring solution and can rely on other PaaS/SaaS solutions from the cloud vendor to implement CI/CD, logging, better security. Unfortunately, certain other components, for example cluster external access with ingress, better observability with dashboards and autoscaling, all require a greater level of understanding around Kubernetes concepts and third-party solutions.

From a development team perspective, managed Kubernetes has the same challenges, developers still need to understand how to create and secure containers, how to use container registries, or how to write Kubernetes configuration YAML(s) for their deployments.

At many enterprises, security teams have already been looking into how to secure cloud-based workloads, Kubernetes infrastructure is no different and should be considered just another service. From a container security perspective, managed Kubernetes services bring the same challenges as vanilla Kubernetes (base images coming from untrusted sources, developer code and securing related libraries, etc.). All these vulnerabilities can be remediated with the same third-party tools and proper governance models, but can still require the involvement of additional third-parties.

As highlighted, managed Kubernetes is certainly one of the ways forward, it allows companies to have a certain level of freedom to choose certain components, providing they have a good base understanding of Kubernetes. Managed Kubernetes Service takes over some of the operational burdens, allowing companies to focus on delivering more value to their customers, rather than spending time with operational challenges.

What if a company has no Kubernetes knowledge? What if their main focus is on development, and they don't want to deal with complex support processes involving many third-parties? What can an enterprise company do if they want a turn-key solution, which allows their staff to easily and quickly build infrastructure for containerised workloads, on-premises or in the cloud?

OpenShift

Let's make it even more simple!

If there is a demand in IT, then there must be a solution somewhere! The solution we are talking about in this context is called OpenShift from Red Hat. Red Hat needs little introduction, a large enterprise, recently acquired by IBM and since 1993 have been a prominent member of the open source community. From their inception, Red Hat have focused on the Linux/Unix operating systems and grew into a multinational company. They are able to offer enterprise ready solutions across the whole IT landscape, middleware, database, operating system, container orchestration and others.

OpenShift was released in 2011, originally with custom developed technologies for the container and container orchestration technologies. From version 3, they have adopted Docker as the container technology, and Kubernetes as the container orchestration technology.

OpenShift is a turn-key solution provided by Red Hat. It is a platform-as-a-service product, built around a core of application containers powered by Docker, with orchestration and management provided by Kubernetes, on a foundation of Red Hat Enterprise Linux. OpenShift comes with built in components such as dashboards, container registry, Red Hat service mesh, templates etc.

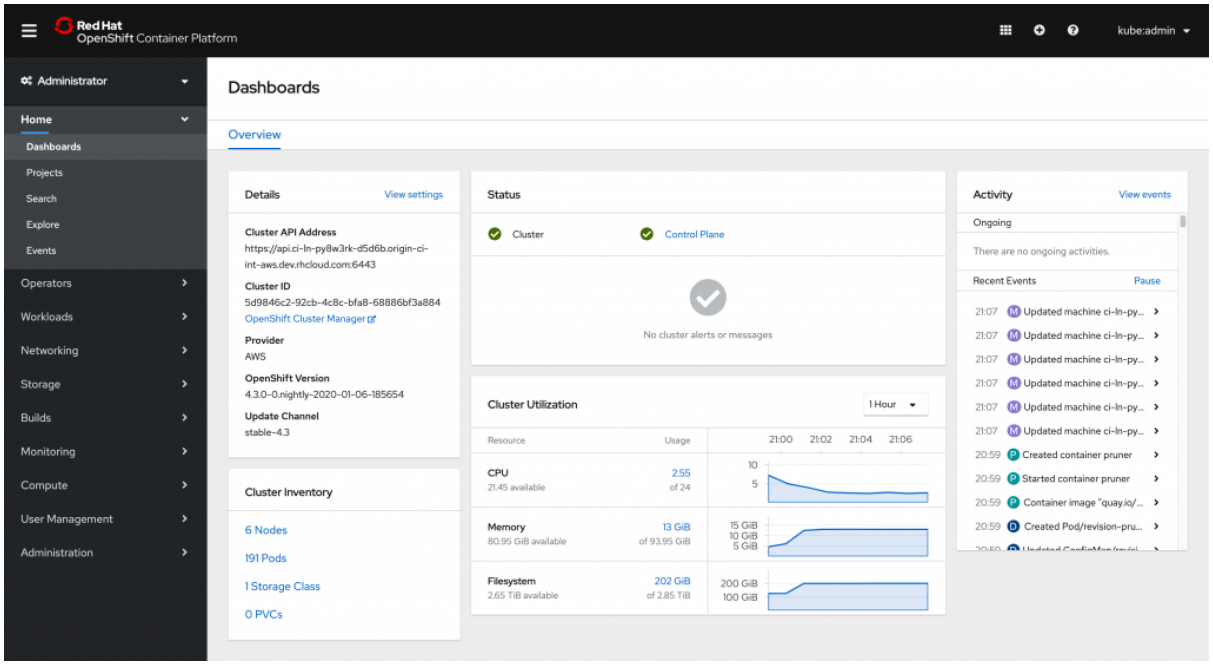

Figure 1 – Red Hat OpenShift dashboards (source: openshift.com)

It allows developer teams to concentrate on their primary task of developing code, by providing capabilities such as “source to image” and preset templates, negating the need for the developers to write any Kubernetes or docker related code to deploy their applications.

As a well-tested and supported product from Red Hat, it provides the key assurance of security for enterprises.

Architecture:

OpenShift uses the upstream Kubernetes as a basis and modifies some of its basic components to provide an enterprise grade service. Using the same master/worker concept to provide HA for each component.

Application architecture:

OpenShift uses the notion of "projects" to provide isolation and distinction between admin and user access. Apps run in containers, which are built by the platform after commits to repositories.

Kubernetes vs OpenShift

To choose between Kubernetes and OpenShift a CTO/CIO should consider the following aspects.

Should consider Kubernetes if;

- Their company or companies are already using a mature Kubernetes platform or have existing knowledge of the Kubernetes product portfolio.
- There is a requirement for Middleware, which is better suited to Kubernetes.
- Utilising the latest open source technology is valued at the company.
- Their company or companies have a preference or requirement, to keep CI/CD outside of the cluster.

Should consider OpenShift if;

- Their company or companies have existing Red Hat subscriptions and investment in OpenShift.
- Red Hat based middleware is used or preferred.
- Their company or companies value security hardened, pre integrated and/or tested open source solutions.
- A user-friendly turn-key solution with limited admin overhead is preferred.
- Kubernetes environments in multi/hybrid cloud scenarios need to be managed.
- Built-in CI/CD features are expected.

Developers and Operations are always looking at IT solutions from different, sometimes closed-minded, points of view.

Developers within any Kubernetes environment need to learn how to containerize applications, work with image registries and deploy applications onto Kubernetes platforms. Operations are often more focused on observability, monitoring and logging capabilities.

These processes can be complex with a vanilla Kubernetes implementation. In comparison, OpenShift comes with solutions such as Source-to-image, templates and built-in CI/CD to help developers to focus on business goals. It also comes with integrated logging, monitoring and dashboard solutions with automated installation.

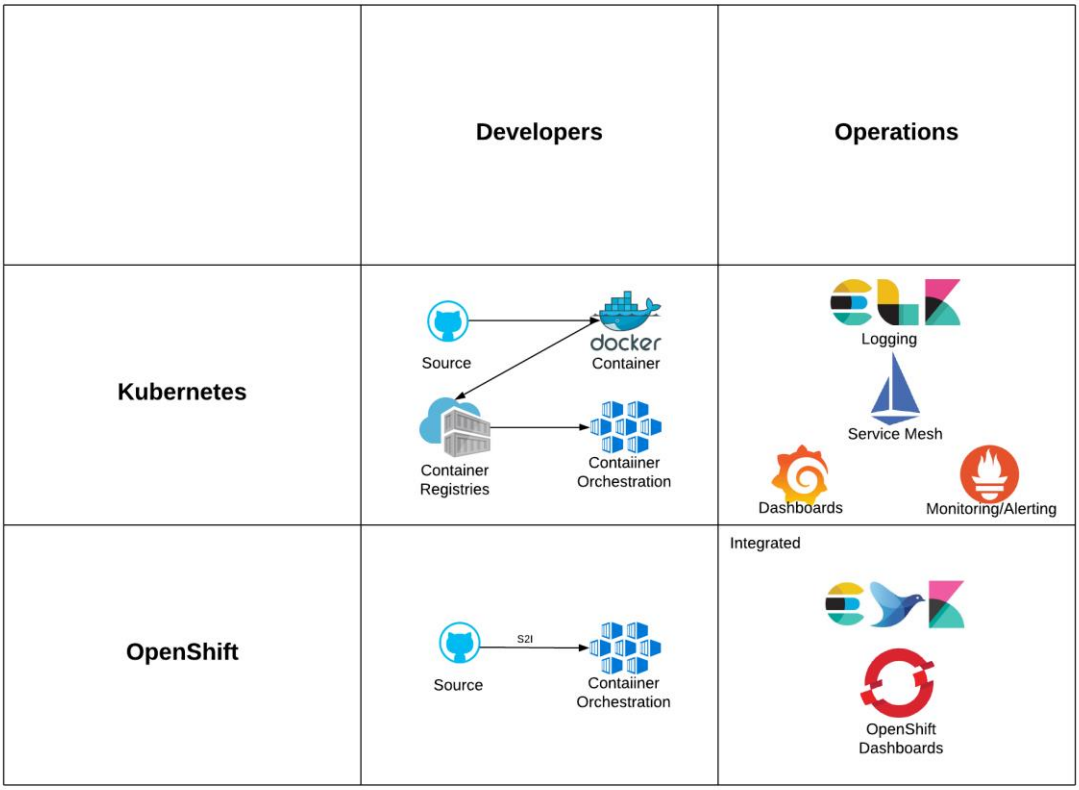

Figure 2 - Kubernetes vs Red Hat OpenShift

Even though a turn-key solution sounds impressive, they shouldn't always be considered more straightforward and easy to start your journey. It's because of this that Red Hat and Azure have partnered to support and provide OpenShift in Azure as a service. There are two options, OpenShift in Azure and Azure Red Hat OpenShift managed containerization.

OpenShift on Azure:

There is a choice when it comes to choosing a container platform and a cloud solution to run the mission critical systems that power a business. With Red Hat OpenShift and Microsoft Azure, companies can quickly deploy a containerized, hybrid environment, to meet digital business needs.

Primary Capabilities:

- Supported, integrated, and automated architecture, with a validated cluster deployment.
- Seamless Kubernetes deployment on the Azure public cloud.
- Fully scalable, global and enterprise-grade public cloud with access to Azure Marketplace.

Key benefits:

- Accelerated time to market on a best-of-breed platform.
- Consistent experience across your hybrid cloud.
- Scalable, reliable and supported hybrid environment, with a certified ecosystem of proven ISV solutions.

Challenges Addressed:

- Keeping up with the ever-changing set of open source projects.
- Servicing the increased needs of your app development teams.
- Managing a diverse, complex non-compliant development, security and operations environment.

Differentiators:

- Joint development and engineering.
- Quick issue resolution via co-located support.
- Enhanced security for containers, network, storage, users, APIs and the cluster.
- Containers with cloud-based consumption model and integrated billing.

Having a unified solution is critical when working seamlessly across on-premises and cloud deployments, OpenShift can be easily deployed to any location. Red Hat and Microsoft are building on shared open source Linux technologies to ensure OpenShift success.

The co-located support engineers can resolve issues faster and more easily than a disjointed model where you do not know who owns the issue. Integrated support goes beyond break/fix and provides a set of best practices.

Customers want reliability, dependability, and flexibility in a supported, sustained engineering lifecycle, this can be easily achieved with Microsoft Azure and Red Hat's tested and trusted subscription model.

Azure Red Hat OpenShift (ARO)

When resources are constrained and skilled talent is scarce, businesses are looking to run containers in the cloud with minimal efforts on maintenance. Azure Red Hat OpenShift (ARO) lets customers gain all the benefits of a container platform, without the need to deploy and manage the environments. This changes the focus from infrastructure management to application development that results in business outcomes.

Primary Capabilities:

- Fully managed Red Hat OpenShift on Azure.
- Jointly engineered, developed and supported by Microsoft and Red Hat.
- Access to hundreds of managed Azure services, like Azure Database for MySQL, Azure Cosmos DB, and Azure Cache for Redis, to develop apps.

Key Benefits

- The ability to focus on application development, not on container platform management.
- The value of containers without deploying and managing the environment and platform yourself.
- Reduced operational overhead.

Challenges Addressed

- Finding the expertise and resources to build custom solutions.
- Maintaining data sovereignty in hybrid environments.
- Ensuring security and compliance across complex infrastructure environments.

Differentiators

- Fully managed container offering.
- Jointly engineered and supported with a 99.9% uptime SLA.
- Containers with cloud-based consumption and the built-in ability to scale as needed.
- The ability to leverage existing Azure commitments.

Companies choosing ARO can implement a container orchestration platform with OpenShift on Azure without the need to hire and retain talent or maintain the budget to hire new operations people to manage new platforms. Allowing developers to concentrate on business innovation versus running infrastructure.

Eliminate the operational complexity of deploying and managing an enterprise container platform at scale and ensuring guaranteed uptime and availability with a defined SLA, security and compliance.

Reduce operational costs by only paying for what you need, when you need it.

Maintain a single agreement with Red Hat and Microsoft, requiring no separate contract or subscription while gaining joint support and security from Microsoft and Red Hat.

Customer stories OpenShift on Azure

Multinational Airline Technical Support Company

CHALLENGE

A Multinational Airline Technical Support company developed its digital software as a service (SaaS) platform for maintenance, repair and overhaul operations using Red Hat Linux and other open-source technologies. The company wanted to move the solution to the cloud.

SOLUTION

The company chose to migrate the solution to Microsoft Azure for its robust and flexible infrastructure capabilities, its network of global data centers and its support for open-source solutions.

RESULTS

The company can run its open-source technology stack easily on Azure, helping the company provide airlines with solutions that cut costs, optimize operations and improve safety. The stage is set for more exciting future developments.

Customer stories ARO

Midmarket Insurance Company

CHALLENGE

A Midmarket Insurance Company was lacking in-house skills to effectively run or manage OpenShift themselves.

SOLUTION

ARO gave the company the ability to quickly deploy an OpenShift cluster on Azure, in specific regions, for quick consumption, lowering the time to drive value.

RESULTS

The Insurance Company was able to focus its efforts on business outcomes and end user benefits through application development and integration.