SUPPLYDRIVE
Gecontroleerd data delen in de toeleverketen

Research Document

Aron Hemmes

Semester 5 – 2021 - 2022

Research Questions

What is Kubernetes?

Kubernetes, is an open-source system for automating deployment, scaling, and management of containerized applications. It groups containers that make up an application into logical units for easy management and discovery. Kubernetes is designed to be scalable, it can run many containers without issue and load balanced.

What are the components?

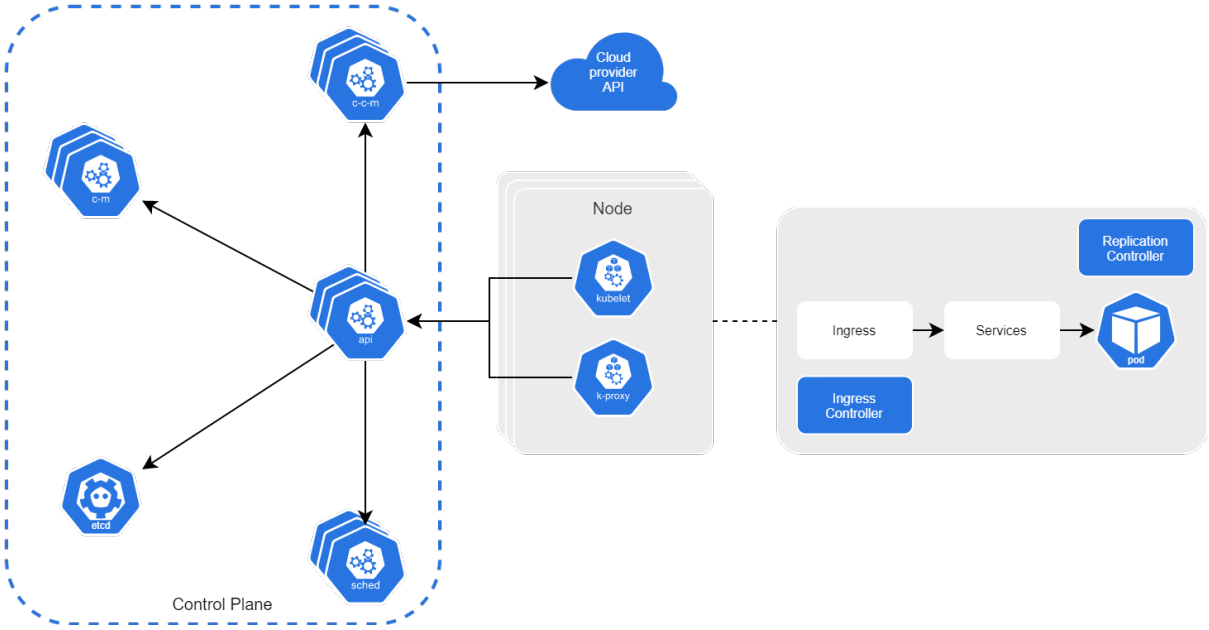

Figure 1: Kubernetes Cluster

Pods

pods usually are created automatically using workload resources and are used in kubernetes to manage containers rather than managing them directly. Each Pod is meant to run a single instance of a given application. If you want to scale your application horizontally (to provide more overall resources by running more instances), you should use multiple Pods, one for each instance. In Kubernetes, this is typically referred to as replication. Replicated Pods are usually created and managed as a group by a workload resource and its controller.

Nodes

Kubernetes runs your workload by placing containers into Pods to run on Nodes. Each node is managed by the control plane and contains the services necessary to run Pods. Now we know what a node is we can talk about the components that run on every node.

Kubelet
A kubelet takes the PodSpecs that describe the pod and container and ensures they are healthy and that the containers are running in a pod.

k-proxy
K-proxy maintains network rules on a node, allowing network communication to your pod from inside or outside of the cluster.

Container runtime
The container runtime is the software that is responsible for running containers. Kubernetes supports

several container runtimes: Docker, Containerd, CRI-O and other and any other implementation of the CRI (Container Runtime Interface). Since Docker is deprecated because it doesn't support CRI, Containerd is a good alternative.

Cluster

When you deploy Kubernetes, you get a cluster. A Kubernetes cluster consists of a set of nodes that run containerized applications. Containerizing applications packages an app with its dependences and some necessary services.

Control Plane

The container orchestration layer in the cluster that exposes the API and interfaces to define, deploy, and manage the lifecycle of containers.

Deployments

A Deployment is a workload resource and provides declarative updates for Pods and ReplicaSets.

You describe a desired state in a Deployment, and the Deployment Controller changes the actual state to the desired state at a controlled rate. You can define Deployments to create new ReplicaSets, or to remove existing Deployments and adopt all their resources with new Deployments.

StatefulSets

StatefulSet is the workload API object used to manage stateful applications.

Manages the deployment and scaling of a set of Pods, and provides guarantees about the ordering and uniqueness of these Pods.

Like a Deployment, a StatefulSet manages Pods that are based on an identical container spec. Unlike a Deployment, a StatefulSet maintains a sticky identity for each of their Pods. These pods are created from the same spec, but are not interchangeable: each has a persistent identifier that it maintains across any rescheduling.

Service

An abstract way to expose an application running on a set of Pods as a network service. With Kubernetes you don't need to modify your application to use an unfamiliar service discovery mechanism. Kubernetes gives Pods their own IP addresses and a single DNS name for a set of Pods, and can load-balance across them.

Ingress

Ingress exposes HTTP and HTTPS routes from outside the cluster to services within the cluster. Traffic routing is controlled by rules defined on the Ingress resource.

An Ingress may be configured to give Services externally-reachable URLs, load balance traffic, terminate SSL / TLS, and offer name-based virtual hosting. An Ingress controller is responsible for fulfilling the Ingress, usually with a load balancer, though it may also configure your edge router or additional frontends to help handle the traffic.

Ingress Controller

In order for the Ingress resource to work, the cluster must have an ingress controller running.

Config Map

A ConfigMap is an API object used to store non-confidential data in key-value pairs. Pods can consume ConfigMaps as environment variables, command-line arguments, or as configuration files in a volume.

A ConfigMap allows you to decouple environment-specific configuration from your container images, so that your applications are easily portable.

Sources:

- Kubernetes components
- Kubernetes Ingress

What are the top Kubernetes security vulnerabilities and how do you address them?

Kubernetes has a lot of potential customisation, this means however that if Kubernetes is configured wrong, you possibly have security vulnerabilities. To avoid these vulnerabilities I will describe various of them to get a starting point for securing a Kubernetes-cluster.

Securing Kubernetes hosts

Kubernetes is an active company constantly trying to improve their services and patch security vulnerabilities, thus one of the best defenses is to make sure you are running the latest version of Kubernetes. Other security measurements involve implementing specific firewall rules, implementing a patch management and configuration management system and taking security measurements depending on the datacenter environment. Securing the Kubernetes Dashboard is also of high importance, if installed wrong by the owners this could create huge security gaps.

Securing Kubernetes components

To secure a cluster and cluster nodes the best measurement is to limit access to them. This can be done by setting up authentication and authorization on the cluster and cluster nodes and making sure your network blocks access to ports and limit access to the server expect from trusted networks. You should also limit direct SSH access to nodes instead users should use "kubectl exec", which provides direct access to the containers without the ability to access the host. Kubelets expose HTTPS endpoints, these grant control over the node and containers. Clusters should enable Kubelet authentication and authorization to protect them.

Securing the control plane

It is also of high importance to limit access and control to who are allowed to perform actions in your API. Kubernetes expects that all API communication in the cluster is encrypted with TLS to prevent traffic sniffing, verify the identity of the server, and (for mutual TLS) verify the identity of the client. Another important measurement is to restrict access to etcd and protect it differently from the rest of the cluster, gaining write access to the etcd is equivalent to gaining root on the entire cluster, even read access can be exploited fairly easily. It is recommended to isolate the etcd servers behind a firewall that ONLY the API servers may access.

Sources:

- Kubernetes OWASP cheatsheet
- Kubernetes 11 ways not to get hacked

What to test in- and how to test a Kubernetes-cluster?

E2E testing

Sources:

- Kubernetes end-to-end testing

How do you setup a Kubernetes-cluster?

What is IDS Connector Deployment?

To use the IDS-network SCSN designed, I have to create a kubernetes cluster using IDS connector deployment. An indept guide can be found here: smart-connected-supplier-network.gitbook.io/processmanual/deployment/ids-connector-deployment.

How is it different from a regular Kubernetes deployment?

What is required to get it working?

Kubernetes POC

Now that we know the basics about kubernetes we should practice the use of it in a mini project and test it thoroughly. I made a git repository here: github.com/Alpacron/Kubernetes-POC.

Supplydrive - The Assignment

Now that we know how kubernetes works and even made a mini project to confirm the library research, we need to research the best use case for Supplydrive. We will be doing this in form of library research.