kubernetes

kubernetes

kubernetes

every cloud supports kubernetes

amazon web services™

Microsoft Azure

Google Cloud Platform

What has Docker Done for Us?

Continuous delivery

- Deliver software more often and with less errors
- No time spent on dev-to-ops handoffs

Improved Security

- Containers help isolate each part of your system and provides better control of each component of your system

Run anything, anywhere

- All languages, all databases, all operating systems
- Any distribution, any cloud, any machine

Reproducibility

- Reduces the times we say “it worked on my machine”

kubernetes

What Does Kubernetes do?

Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications.

Improves reliability

- Continuously monitors and manages your containers
- Will scale your application to handle changes in load

Better use of infrastructure resources

- Helps reduce infrastructure requirements by gracefully scaling up and down your entire platform
- Coordinates what containers run where and when across your system
- How do all the different types of containers in a system talk to each other?

Easily coordinate deployments of your system

- Which containers need to be deployed
- Where should the containers be deployed

what is kubernetes?

kubernetes (n.) - greek word for pilot or helm

“A declarative language for launching containers”

What is Kubernetes?

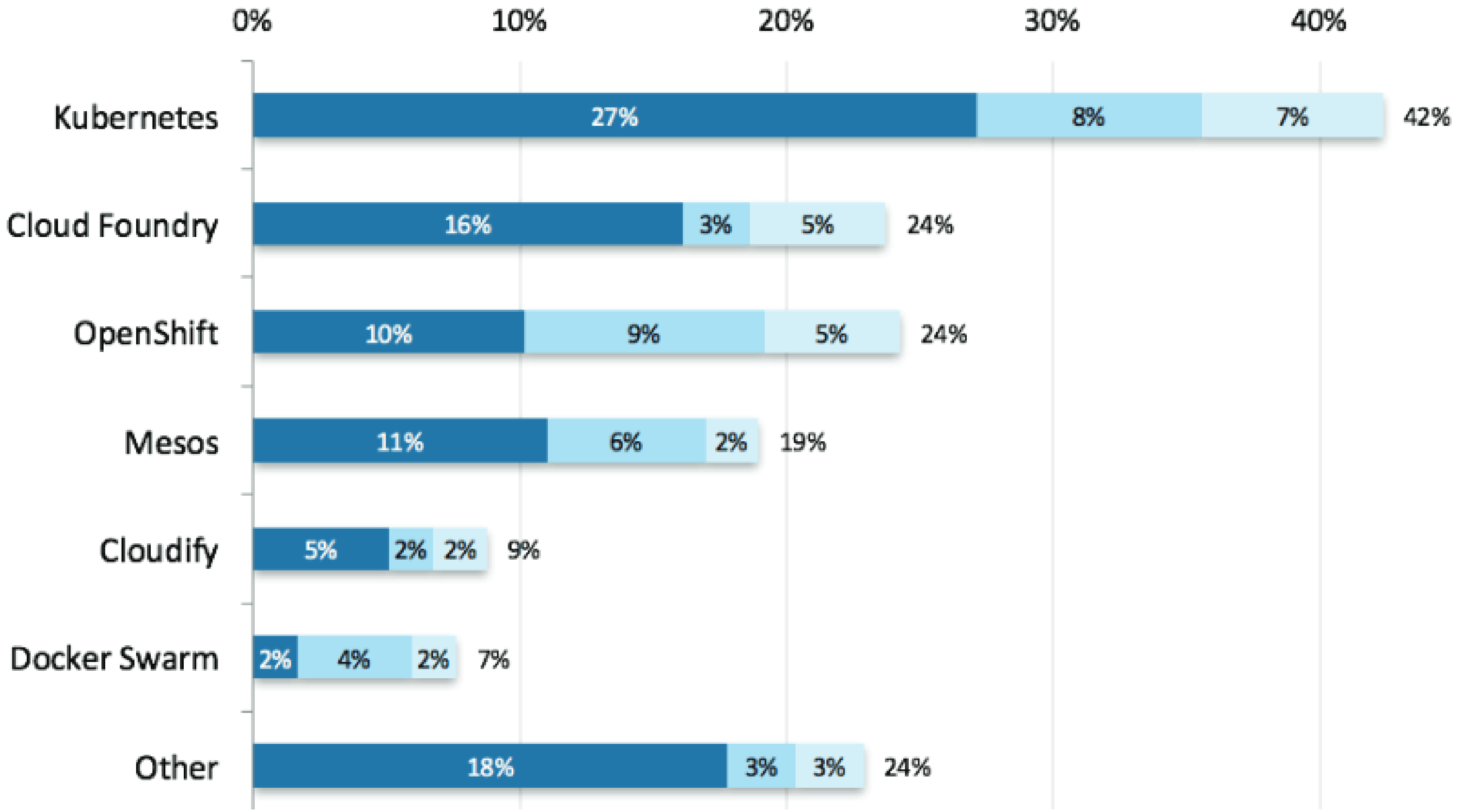
A highly collaborative open source project originally conceived by Google

- ❑Google has 10+ years experience w/ containerized apps
- ❑Red Hat has been a member since day 0.
- ❑Red Hat is the second largest contributing member with many ideas coming from geard Sometimes called: kube or k8s (that's 'k' + 8 letters + 's')
- ❑Start, stop, update, and manage a cluster of machines running containers in a consistent and maintainable way.
- ❑Kubernetes does NOT and will not expose all of the 'features' of the docker command line.

kubernetes

kubernetes is an ecosystem...

kubernetes won the container orchestration war...



what is kubernetes?

it runs containers

what is a container?

not a vm

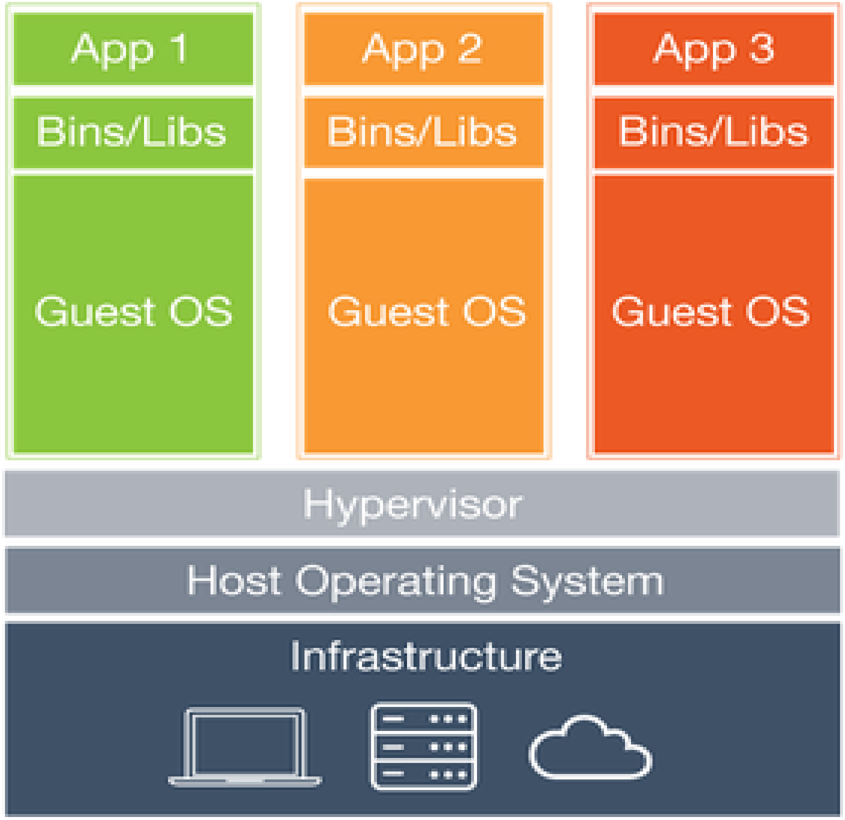
Features

Some of the platform features which K8S offers are:

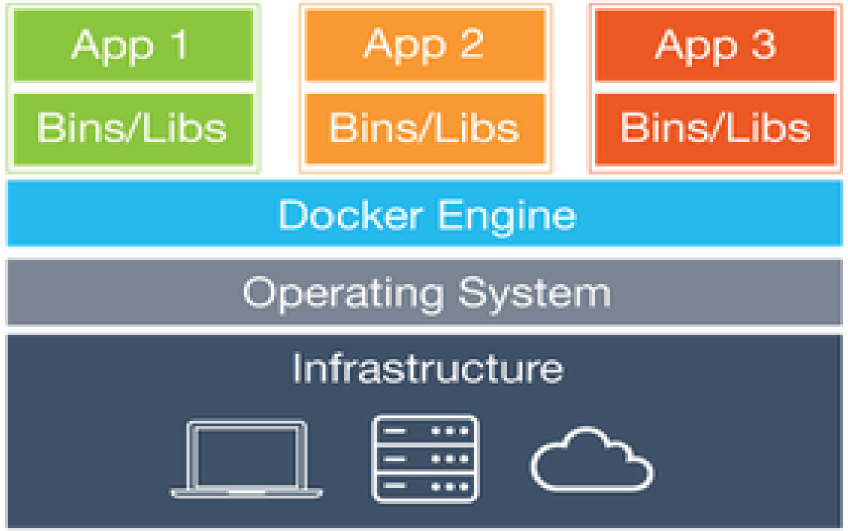
- ❑Container grouping using pod
- ❑Self-healing
- ❑Auto- Scalablility
- ❑DNS management
- ❑Load balancing
- ❑Rolling update or rollback
- ❑Resource monitoring and logging

kubernetes

vm vs container



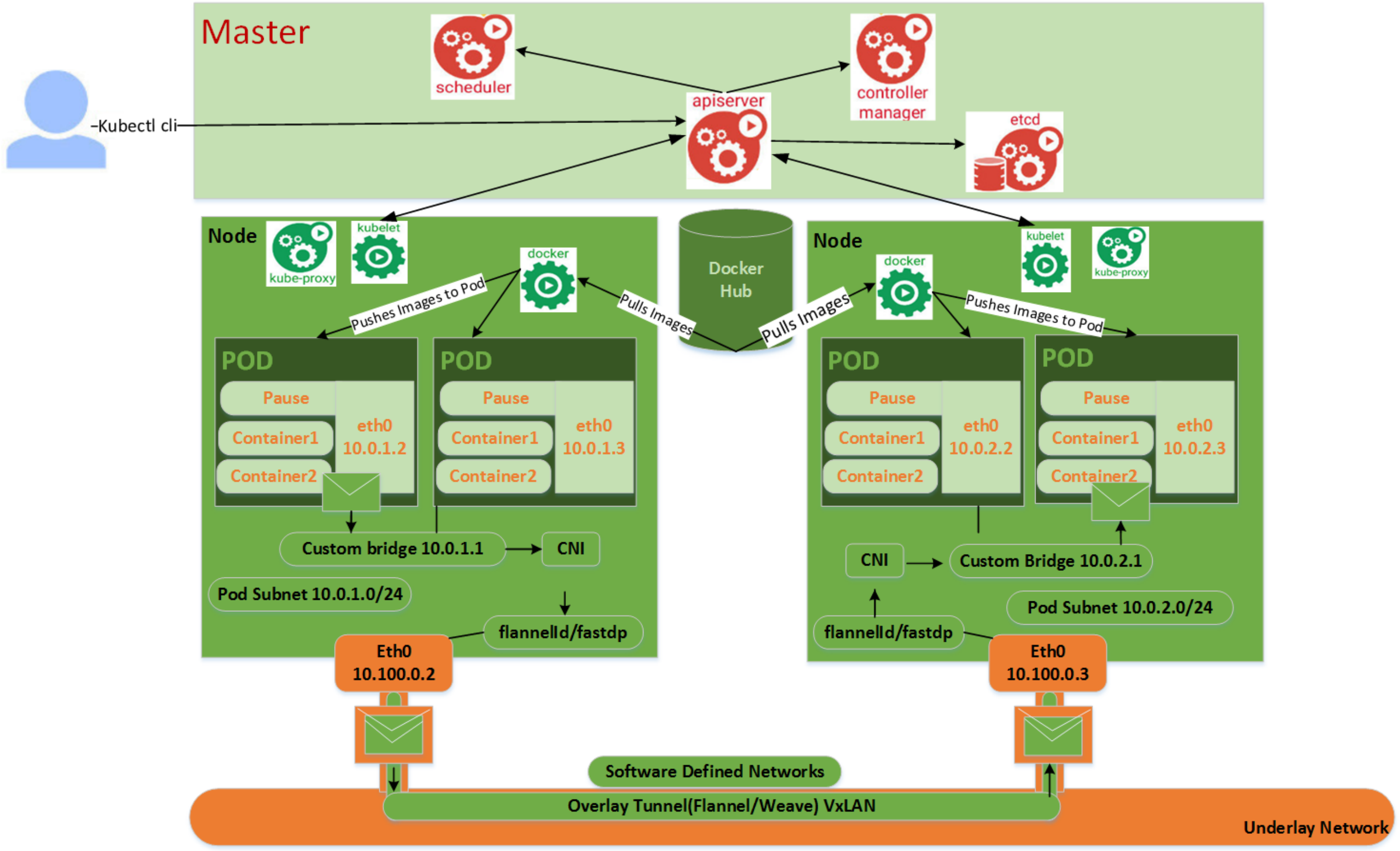
Virtualization



Containers

kubernetes

Kubernetes Architecture



Kubernetes Master

The Kubernetes master components are:

kube-API Server: Users use this to interact with the manifest yaml, via Rest operations or kubectl cli. It is employed for every operation related to API Objects, like pod creation, and it is the only component which stores the desired state in etcd.

Scheduler: Users use this to issue a command to create pod as per manifest yaml to the API Server using kubectl cli. After this action is performed, it is the scheduler's responsibility to allocate pods to available nodes based on the resource requirement.

Kubernetes Master

Controller Manager: The Controller Manager performs operations on the resources based on the cluster state and makes changes to bring the current state application to the desired state as per the manifest yaml. In other words, the Controller Manager reconciles the actual state with the desired state. There are multiple specialized controllers inside a Controller Manager in order to simplify cluster management. For example, the Node Controller checks to see if any currently running nodes are down and takes the corrective measures, whereas the Replication Controller ensures that the desired number of pods are actually running in the nodes.

etcd: All configuration information about cluster states is stored in the etcd in the form of key/value pairs, and this component is implemented by CoreOS. These states show the nodes that are included in the cluster and the pods that are needed to be running in it.

Kubernetes Master

Addons: In order for a server DNS record to be added to Kubernetes, we need a Cluster DNS addon. Addons help in extending the functionality related to Kubernetes clusters or nodes. There are many other addons available like fluntd for logging, rbac for role based access and so on.

Kubernetes Nodes

Docker: A Docker Daemon is running in each node. If the container image is not present, then it will pull an image from docker registry and run it.

Kubelet: A Kubelet node agent periodically checks the health of the containers in a pod. In addition, it ensures that the volume is mounted as per manifest, and it downloads the sensitive information required to run the container. It also links the node to the API server

Kube-proxy: Kube-proxy runs in each node for load distribution among the pods and makes services available to the external host. It uses iptable rules or round robin to forward requests to the correct containers.

Kubernetes Nodes

The components that are installed in a Kubernetes node are:

For Highly Available and Fault Tolerant Kubernetes production and deployment, multiple master nodes and a separate etcd cluster is required. If three API servers are run, a network load balancer is required to properly distribute the load into the servers. The only remaining problem is needing three actors for the Controller Manager and Scheduler to maintain cluster states and allocating nodes. In order to do it more efficiently yet reliably, only one actor should perform the actual change, but other instances are still needed in case a machine is down. In order to fix this, we can use a lease-lock in API to perform a master election and the flag used for it is **–leader-elect**.

Kubernetes Nodes

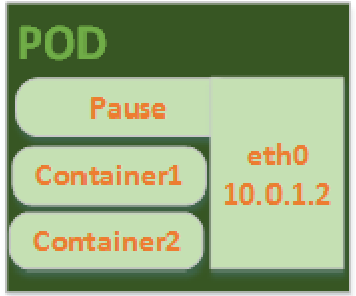
Kubernetes achieves networking from Pod to Pod through either:

- Layer 2 (switching solution)
- Layer 3 (bridging solution)
- Overlay solutions (weave and flannel)

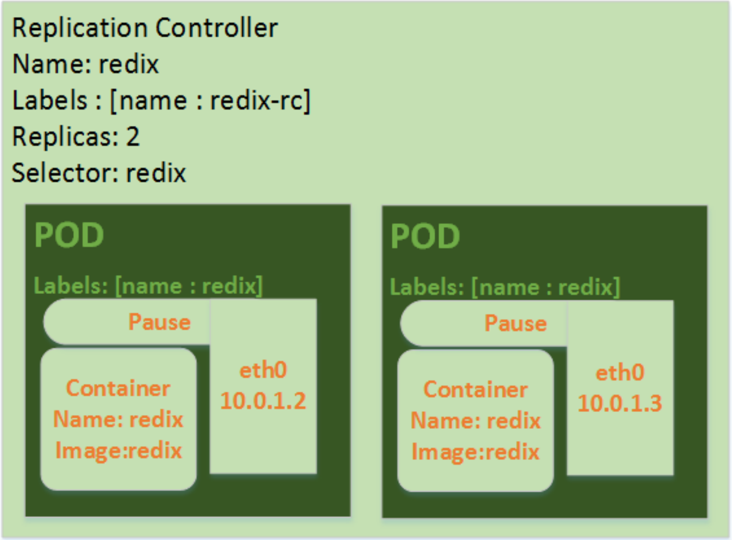
These allow pod-to-pod communication throughout the cluster and provide unique IP addresses for each Pod.

Kubernetes Key Features

1. Pod — collection of containers-- A pod is a deployment unit in the K8S with a single IP address. Inside it, the Pause container handles networking by holding a network's namespace, port and ip address, which in turn is used by all containers within the pod.



2. Replication Controller-- A replication controller ensures that the desired number of containers are up and running at any given time. Pod templates are used to define the container image identifiers, ports, and labels. Using liveness probes, it auto-heals pods and maintains the number of pods as per desired state. It can also be manually controlled by manipulating the replica count using kubectl.



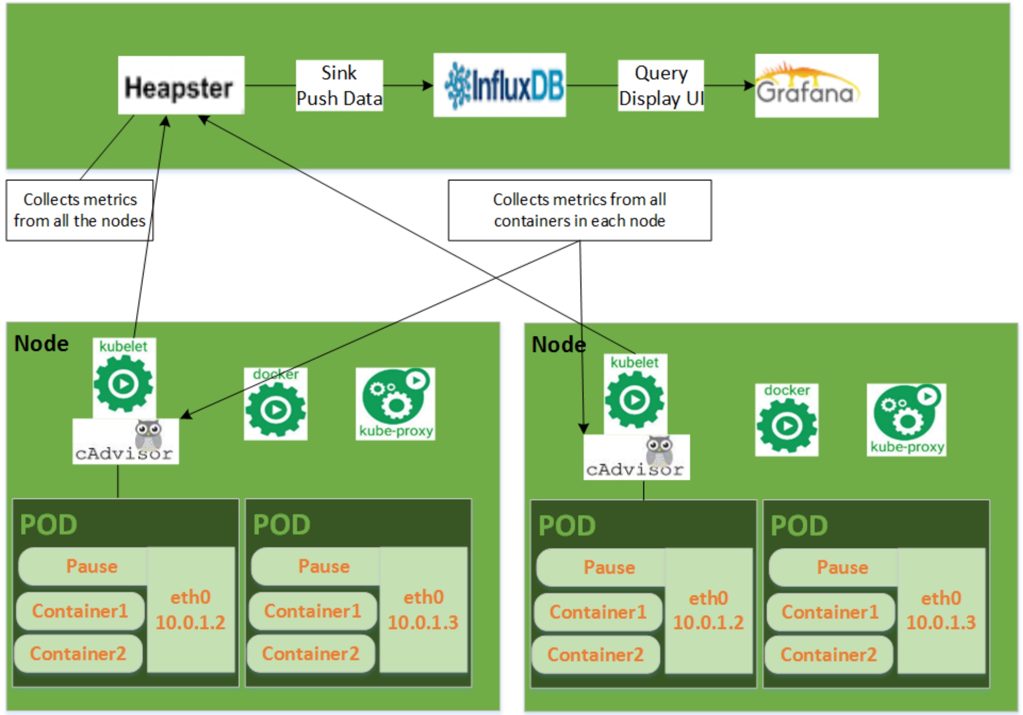
Kubernetes Key Features

3. Storage Management-- Pods are ephemeral in nature — any information stored in a pod or container will be lost once the pod is killed or rescheduled. In order to avoid data loss, a persistent system — like Amazon Elastic Block Storage (EBS) or Google Compute Engine's Persistent Disks (GCE PD) — or a distributed file system — such as the Network File System (NFS) or the Gluster File System (GFS) — is needed.

Kubernetes Key Features

4. Resource Monitoring

Monitoring is one of the key aspects to run infrastructure successfully. It is the base of hierarchy of reliability. Heapster is an addon used to collect metrics from kubelet, which is integrated with a cAdvisor. cAdvisor is used to collect metrics related to CPU, memory, I/O, and network stats of the running containers. Data collected by Heapster is stored in an influx DB and is displayed in the UI using Grafana. There are also other sinks available like Kafka or Elastic Search, which can be use for storing data and displaying it in the UI.



Kubernetes Key Features

5. Health Checking

Health checking in kubernetes is done by a kubelet agent. It is divided into two liveness and readiness probes

There are mainly three types of handlers:

ExecAction: Shell command is executed, and if the resulting exit code is 0, it means that the instance is healthy. Under any other circumstances, the instance is not healthy.

TCPAction: Kubelet will try to connect to a specified port, and if it establishes a connection to the given socket, the diagnostic is successful.

Kubernetes Key Features

There are mainly three types of handlers:

HTTPGetAction: Based on the HTTP endpoint that the application exposes, kubelet performs an HTTP GET request against the container IP address on a specified path, and if it returns with a 200 to 300 response code, the diagnostic is successful.

Each probe usually has three results:

Success: The Container has passed the diagnostic.

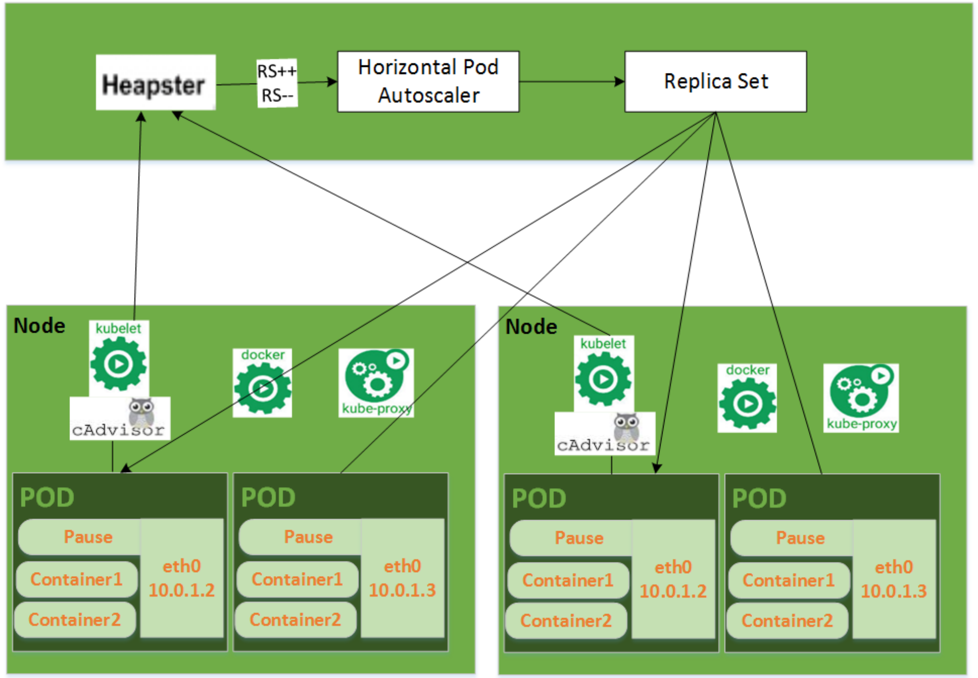
Failure: The Container has failed the diagnostic.

Unknown: The diagnostic has failed, so no action should be taken.

Kubernetes Key Features

6. Horizontal Auto Scaling

Autoscaling utilizes computational resources based on the load. K8S scale pod automatically uses a HorizontalPodAutoscaler object, which gets metrics data from Heapster, and it decreases or increases the number of pods accordingly. For example, if auto-scaling is based on memory utilization, then the controller starts observing memory usage in the pod and scales the replica count based on it.



Kubernetes Key Features

7. Service Discovery

Kubernetes pods are ephemeral, and the Replication Controller creates them dynamically on any node, so it is a challenge to discover services in the cluster. A service needs to discover an IP address and ports dynamically related to each other to communicate within a cluster.

There are two primary ways of finding it — Environment variables and DNS

DNS based service discovery is preferable, and it is available as a cluster add-on. It keeps track of new services in cluster and creates a set of DNS records for each.

Kubernetes Key Features

8. Networking

To manage a cluster fully, a network has to be setup properly, and there are three distinct networking problems to solve:

1. Container-to-Container communications: pods solve this problem through localhost communications and by using the Pause container network namespace

2. Pod-to-Pod communications: this problem is solved by the software defined networking as shown in the Architecture diagram above

3. External-to-Pod communications: this is covered by services.

Kubernetes Key Features

Kubernetes provides a wide range of networking options. Furthermore, there is now support for the **Container Networking Interface** (**CNI**) plugins, which is common plugin architecture for containers. It’s currently supported by several orchestration tools such as Kubernetes, Mesos, and CloudFoundry.

There are various overlay plugins, some of which are discussed below:
Flannel is a very simple etcd backed overlay network that comes from CoreOS. It creates another virtual, routable IP Per Pod network, which runs above the underlay network; ergo, it is called an *overlay* network. Each Pod will be assigned one ip address in this overlay network, and they communicate with each other using their IP directly.

Weave provides an overlay network that is compatible with Kubernetes through a CNI plugin.

Kubernetes Key Features

9. ***Services***

Kubernetes services are abstractions which route traffic to a set of pods to provide a microservice. Kube-proxy runs on each node and manages services by setting up a bunch of iptable rules.

There are three modes of setting up services:

1. **ClusterIP** (only provides access internally)
2. **NodePort** (needed to open firewall on a port; not recommended for public access)
3. **LoadBalancer** (owned by public cloud providers like AWS or GKE)

Kubernetes Key Features

10. ConfigMap and Secret

ConfigMap makes it possible to inject a configuration based on an environment while keeping the container image identical across multiple environments. These can be injected by mounting volumes or environment variables, and it stores these values in the key/value format.

Secrets are used to store sensitive data such as passwords, OAuth tokens, etc.

Kubernetes Key Features

11. Rolling Deployment and Rollback

A Deployment object holds one or more replica sets to support the rollback mechanism. In other words, it creates a new replica set every time the deployment configuration is changed and keeps the previous version in order to have the option of rollback. Only one replica set will be in active state at a certain time.

For rolling deployment, the strategy type required is “**RollingUpdate**” and **“minReadySecs,”** which specifies the time that the application takes to serve traffic. It will be unavailable if we leave it on default in the case that the application pods are not ready.

Kubernetes Key Features

This action can be done by running the command below:

$kubectl set image deployment <deploy> <container>=<image> — record
OR By replacing content in deployment yaml file and running the command below:

$ kubectl replace -f <yaml> — record
If the new version is not behaving as expected, then it is possible to rollback to the previous version by running the below command:

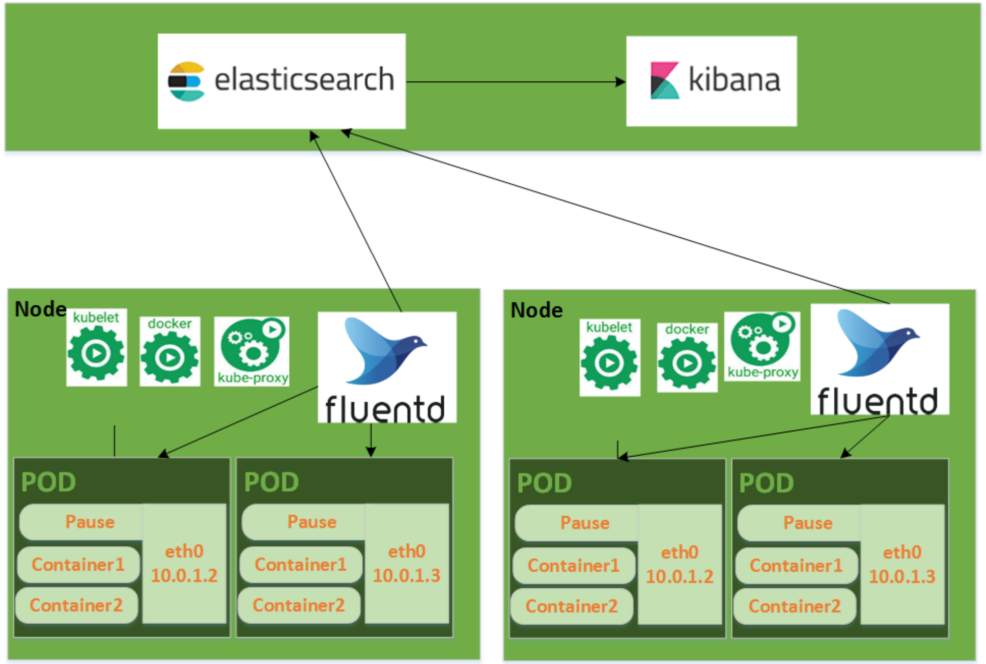
$ kubectl rollout undo deployment <deployment>
If the desired version is any revision other than the previous one, then run:

$ kubectl rollout undo deployment <deployment> — to-revision=<revision>

Kubernetes Key Features

12. Logging

To oversee application behavior, one has to check logs — multiple are generated in each pod. To start searching logs in the Dashboard UI, there has to be some mechanism that collects and aggregates them into one log viewer. To illustrate, Fluentd, an open source tool and part of **Cloud Native Computing Foundation (CNCF)**, combines perfectly with ElasticSearch and Kibana.



References

1. **Large-scale cluster management at Google with Borg, Google Research:** https://research.google.com/pubs/pub43438.html

2. **Pods, Kubernetes Docs:** http://kubernetes.io/docs/user-guide/pods

3. **Ingress Controllers, Kubernetes:** https://github.com/kubernetes/contrib/tree/master/ingress/controllers

4. **Service Load Balancer, Kubernetes:** https://github.com/kubernetes/contrib/tree/master/service-loadbalancer

References

6. **cAdvisor, Google:**
   https://github.com/google/cadvisor

7. **Heapster, Kubernetes:**
   https://github.com/kubernetes/heapster

8. **Monitoring, Kubernetes:**
   http://kubernetes.io/docs/user-guide/monitoring

9. **Docker, Manage data in containers:**
   https://docs.docker.com/engine/userguide/containers/dockervolumes/