How to make the best use of Live Sessions

- Please login on time
- Please do a check on your network connection and audio before the class to have a smooth session
- All participants will be on mute, by default. You will be unmuted when requested or as needed
- Please use the “Questions” panel on your webinar tool to interact with the instructor at any point during the class
- Ask and answer questions to make your learning interactive
- Please have the support phone number (US : 1855 818 0063 (toll free), India : +91 90191 17772) and raise tickets from LMS in case of any issues with the tool
- Most often logging off or rejoining will help solve the tool related issues

edureka!

COURSE OUTLINE

Module 02

Introduction to Kubernetes

Kubernetes Architecture

Deploy app to Kubernetes Cluster

Expose App, Scale App And Update App

Managing State with Deployments

Federations, Auditing and Debugging Kubernetes, Security best practices

edureka!

Kubernetes Architecture

Objectives

After completing this module, you should be able to understand:

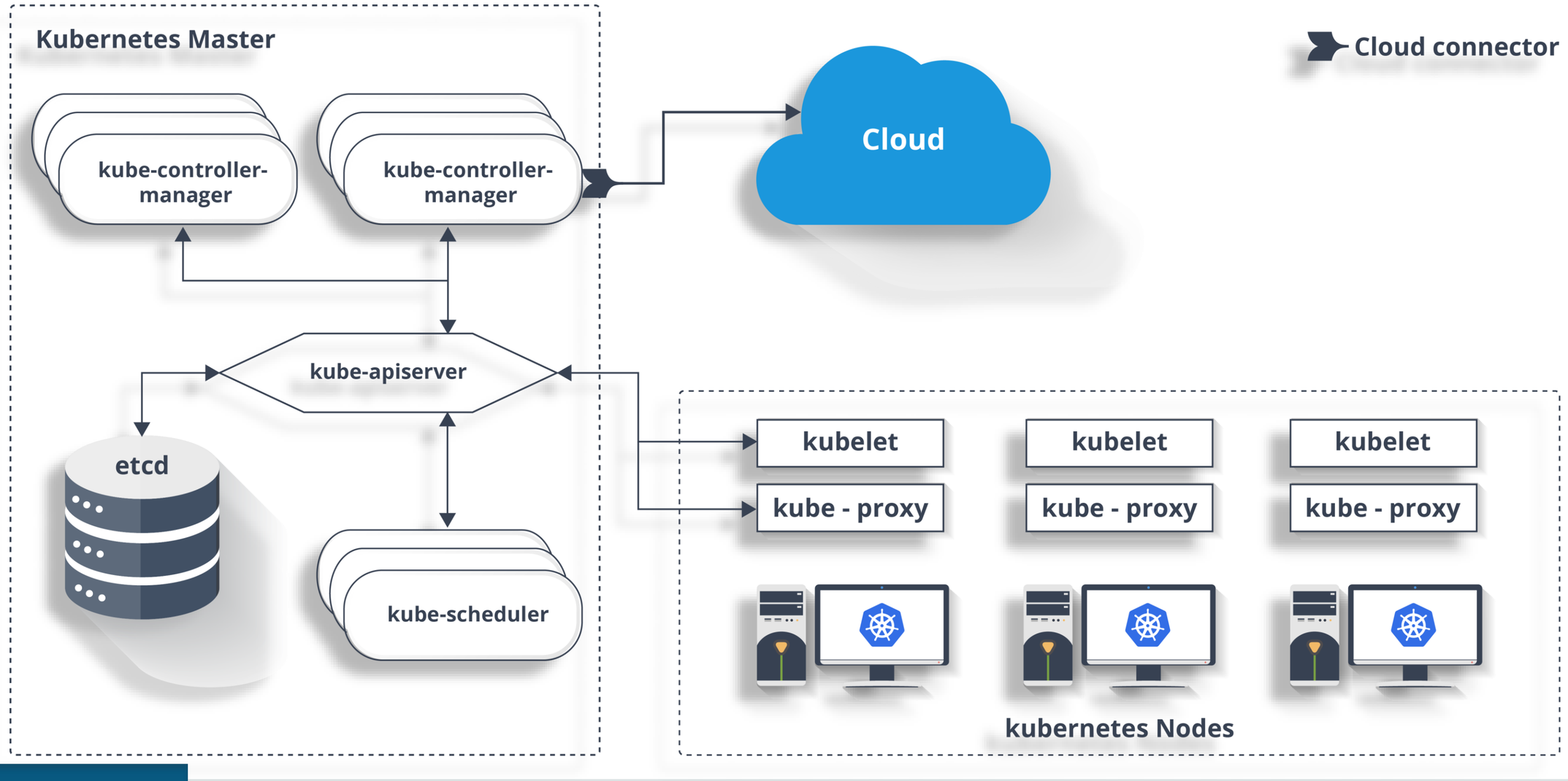
- Master Components of Kubernetes
- Node components of Kubernetes
  - Docker, kubelet, kube-proxy, kubectl
- Addons
  - Cluster DNS
  - Kubernetes Dashboard
  - Container Resource Monitoring
  - Cluster level logging

edureka!

Kubernetes Architecture

edureka!

Architecture Diagram



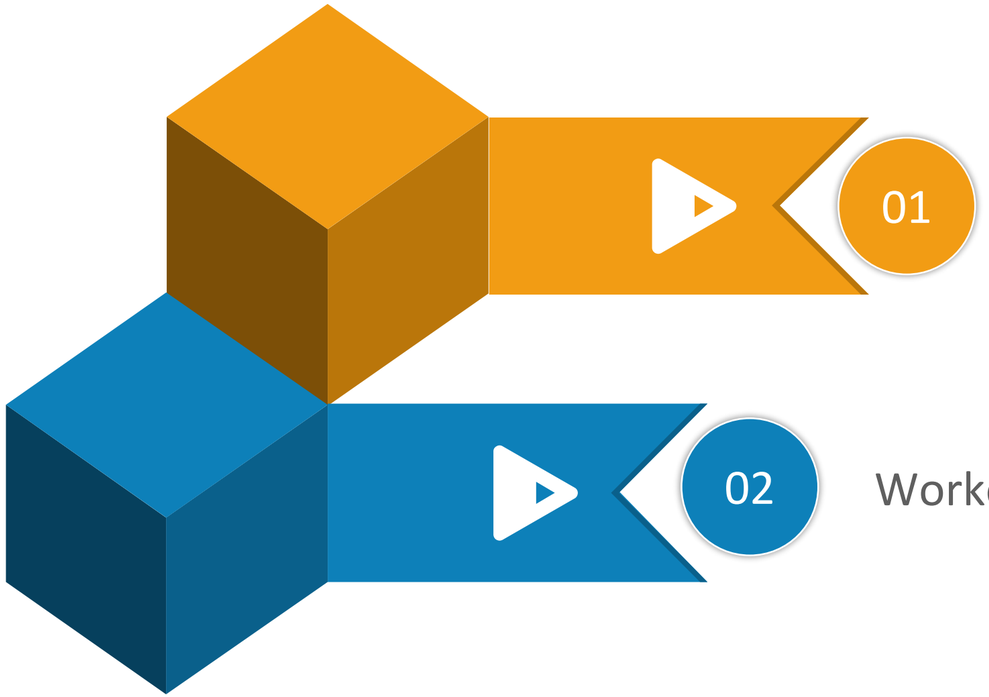
edureka!

Master Components of Kubernetes

edureka!

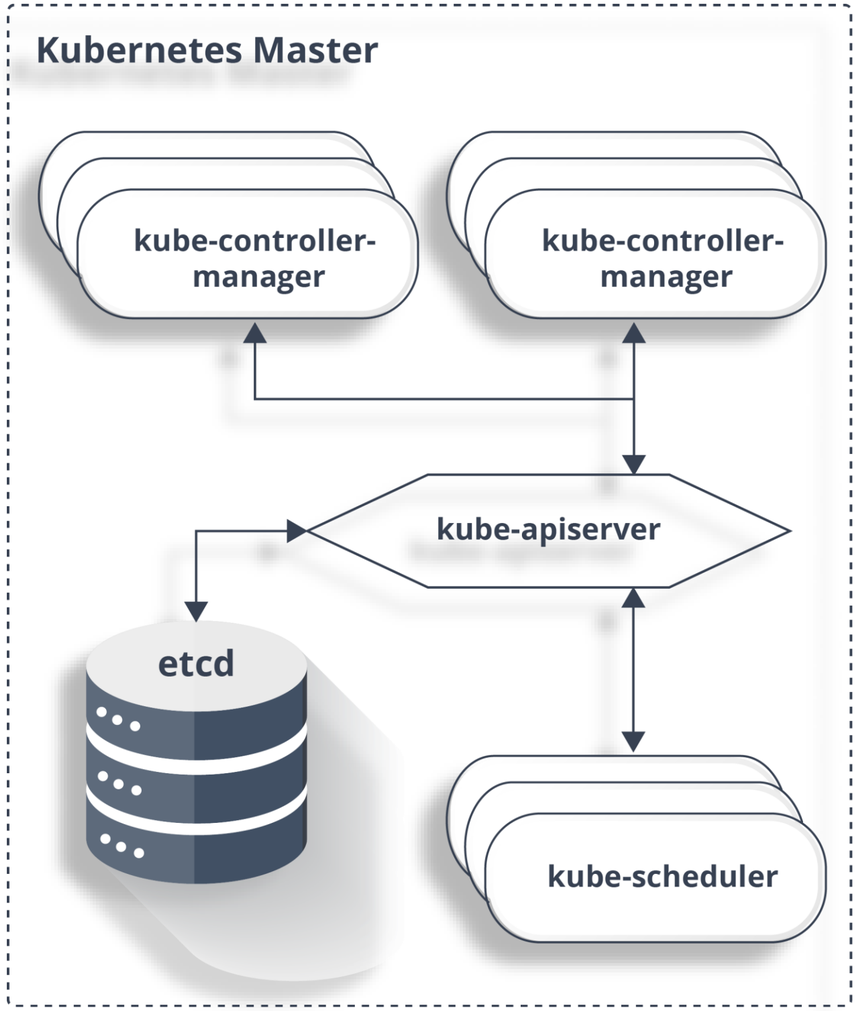
Master Components of Kubernetes

Kubernetes solution is based on two main types of nodes:



Master node

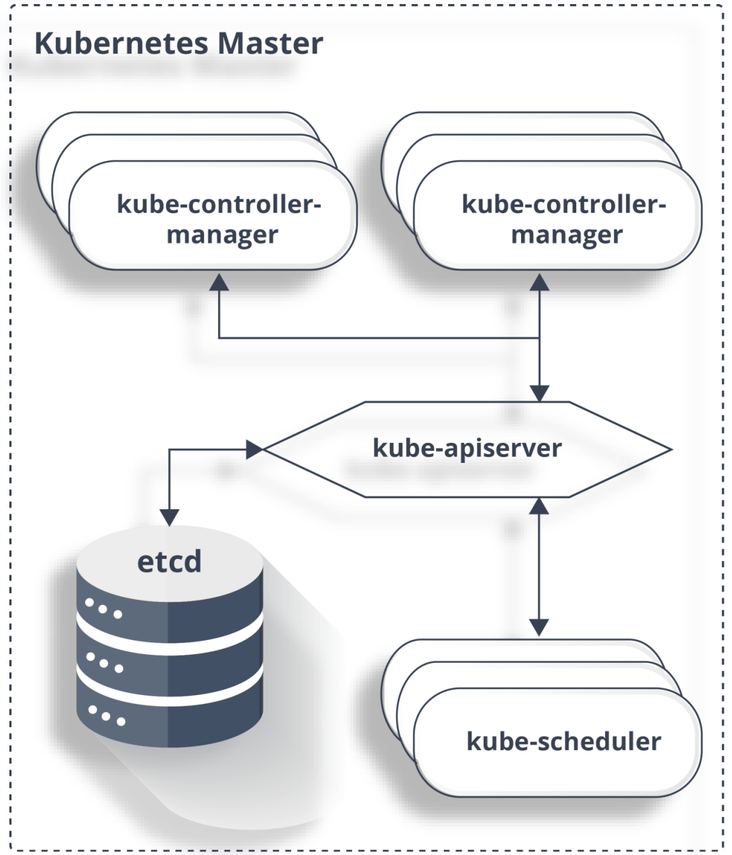
Worker node / minions.



edureka!

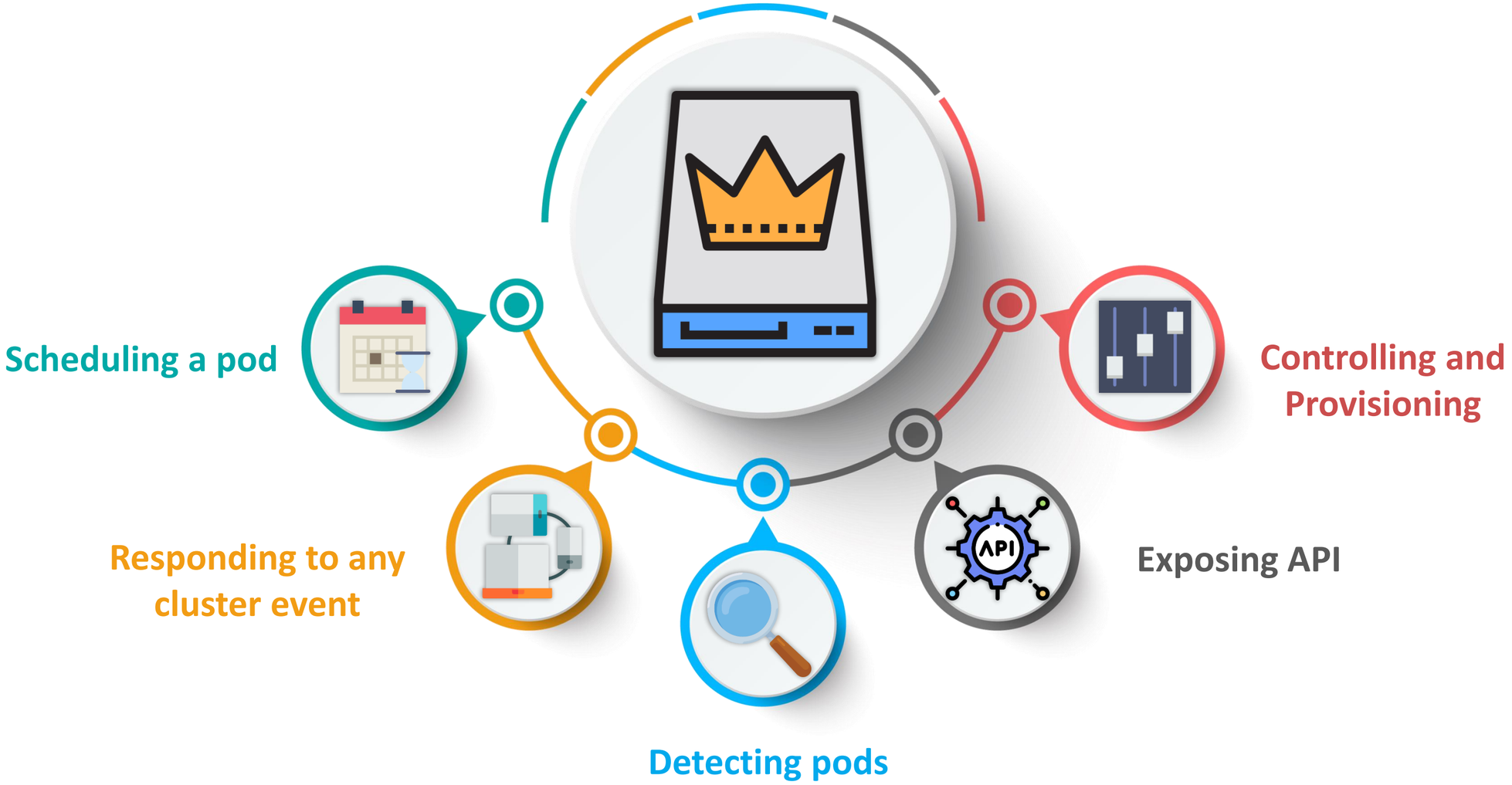
Master Components of Kubernetes

- This is the main part of the Kubernetes cluster
- Takes all the decision related to the cluster
- Can be configured on standalone machine
- Multiple master are configured in cluster for highly available environment.
- As a best practise, don’t run user-container on Master node.



edureka!

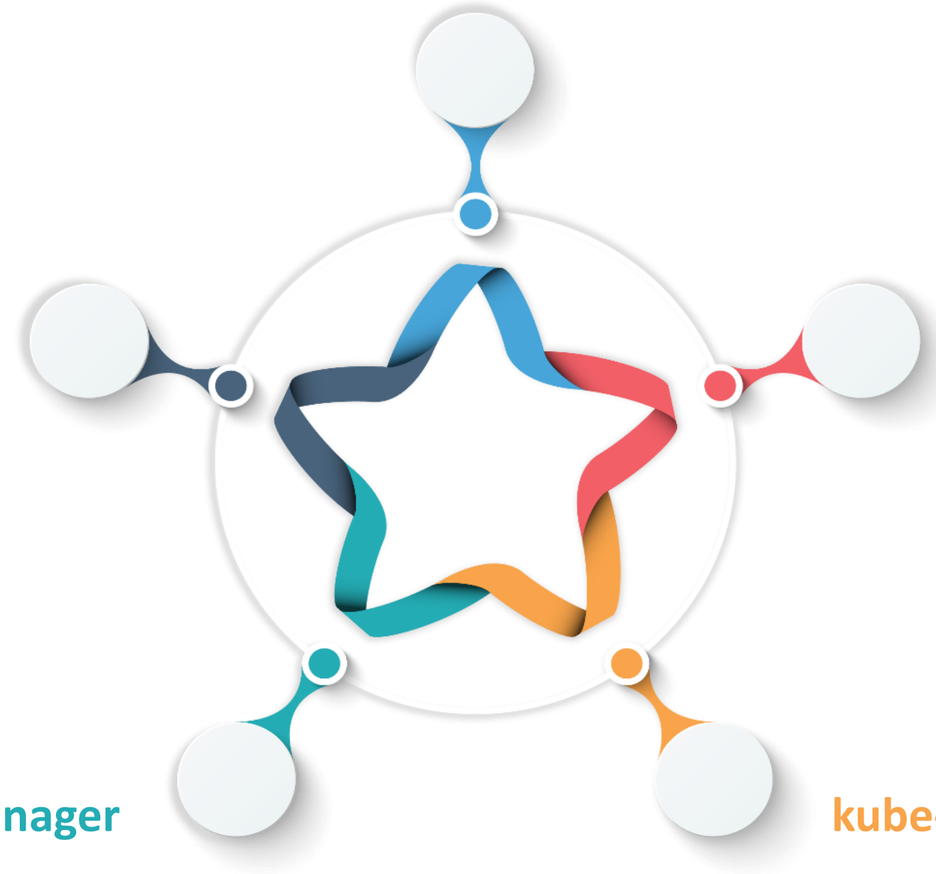
Master Node Functionality



Master Node Components

Kube-apiserver

Cloud-controller-manager



etcd key-value store

kube-controller-manager

kube-scheduler

edureka!

Master Node Components

1. **Kube apiserver**
2. ETCD
3. Kube Scheduler
4. Kube Controller Manager
5. Cloud Controller Manager

- This is the front-end of the master node control plane
- It exposes all the APIs of the kubernetes Master node components
- It is responsible for establishing communication between Kubernetes Node and the Kubernetes Master components
- It follows the scale-out architecture i.e. it can expand horizontally by adding more instances

edureka!

Master Node Components

1. Kube apiserver
2. **ETCD**
3. Kube Scheduler
4. Kube Controller Manager
5. Cloud Controller Manager

- Kubernetes master node uses key-value store for backing all data of cluster
- “etcd key-value” is distributed datastore which holds all the configuration data of the cluster
- Etcd Key-value is the reference point for all the components in the cluster

edureka!

Master Node Components

1. Kube apiserver
2. **ETCD**
3. Kube Scheduler
4. Kube Controller Manager
5. Cloud Controller Manager

- Provides reliable datastore through distributed locking mechanism, write barriers and ensure that leader is sending heartbeats periodically
- It is intended to provide a highly available and permanent data storage and retrieval
- It is recommended to have a backup of etcd file

edureka!

Master Node Components

1. Kube apiserver
2. ETCD
3. **Kube Scheduler**
4. Kube Controller Manager
5. Cloud Controller Manager

- Responsible for distribution and management of workload on the worker nodes
- It selects the most suitable node to run the unscheduled pod based on resource requirement
- It keeps track of resource utilization and makes sure that workload is not scheduled on nodes which are already full

edureka!

Kube-Scheduler

For effective resource scheduling, scheduler knows :

- What is the resource requirement for new workload
- What is the resource available on each of the worker node in the cluster
- Different policies in place like affinity/anti-affinity, data locality, quality-of-service.

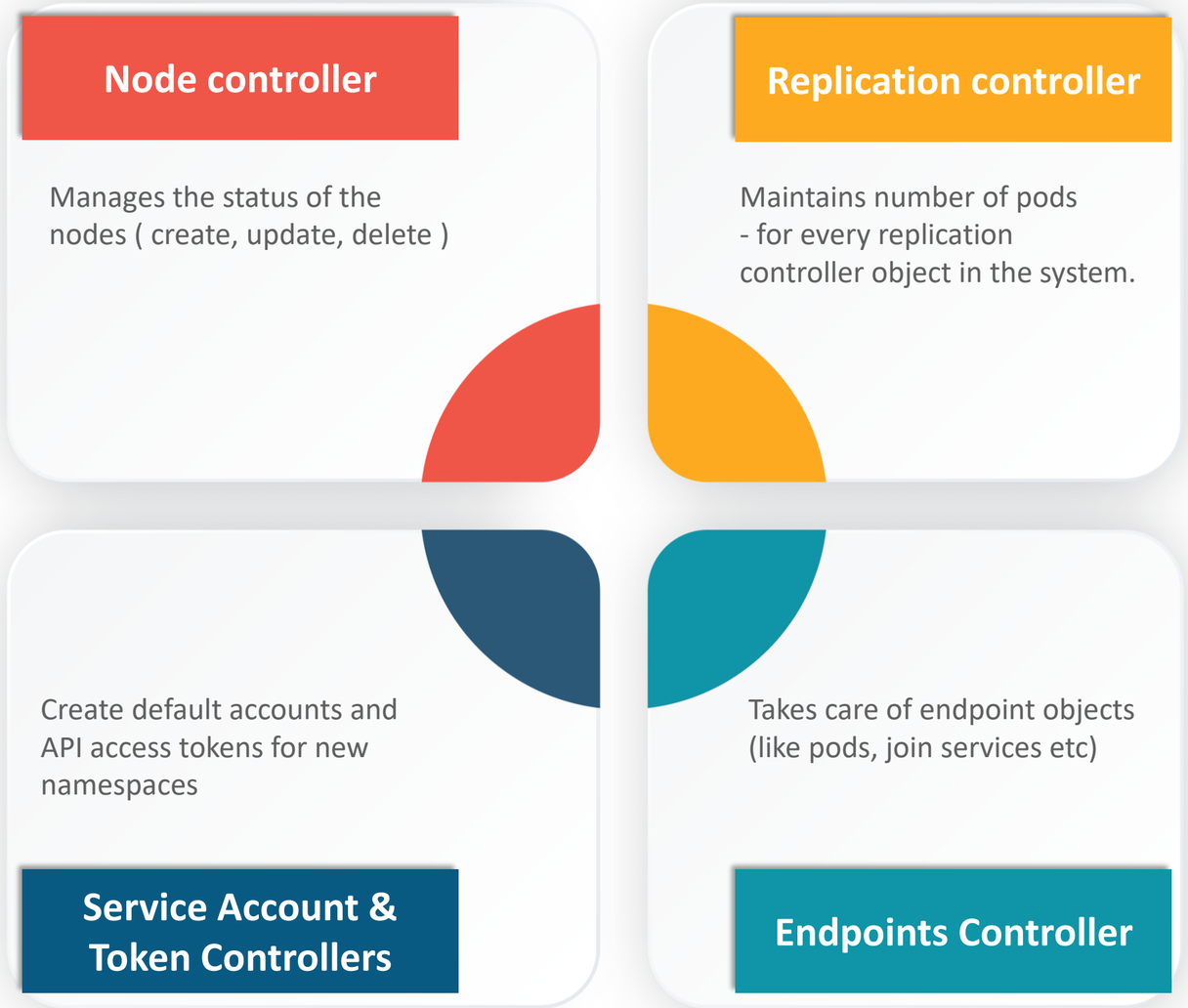
Master Node Components

1. Kube apiserver
2. ETCD
3. Kube Scheduler
4. **Kube Controller Manager**
5. Cloud Controller Manager

- There are multiple controller processes running on the master node, but are compiled together to run as a single process
- Each controllers has it’s own responsibility and communicate with the API server to manage the end-points
- Controller Manager is daemon that embed controllers
- Controller manager does namespace creation and garbage collection

edureka!

Controller runnings on master



Master Node Components

1. Kube apiserver
2. ETCD
3. Kube Scheduler
4. Kube Controller Manager
5. **Cloud Controller Manager**

- Manages communication with the underlying cloud services
- It might be split out into several different container depending on which cloud platform you are running on
- It is responsible for persistent storage and network routing
- Cloud controller manager helps abstracting the cloud specific code from the core kubernetes specific code
- It enables cloud vendors and kubernetes code to be developed without any inter-dependency

edureka!

Master Components of Kubernetes

1. Kube apiserver
2. ETCD
3. Kube Schedular
4. Kube Controller Manager
5. **Cloud Controller Manager**

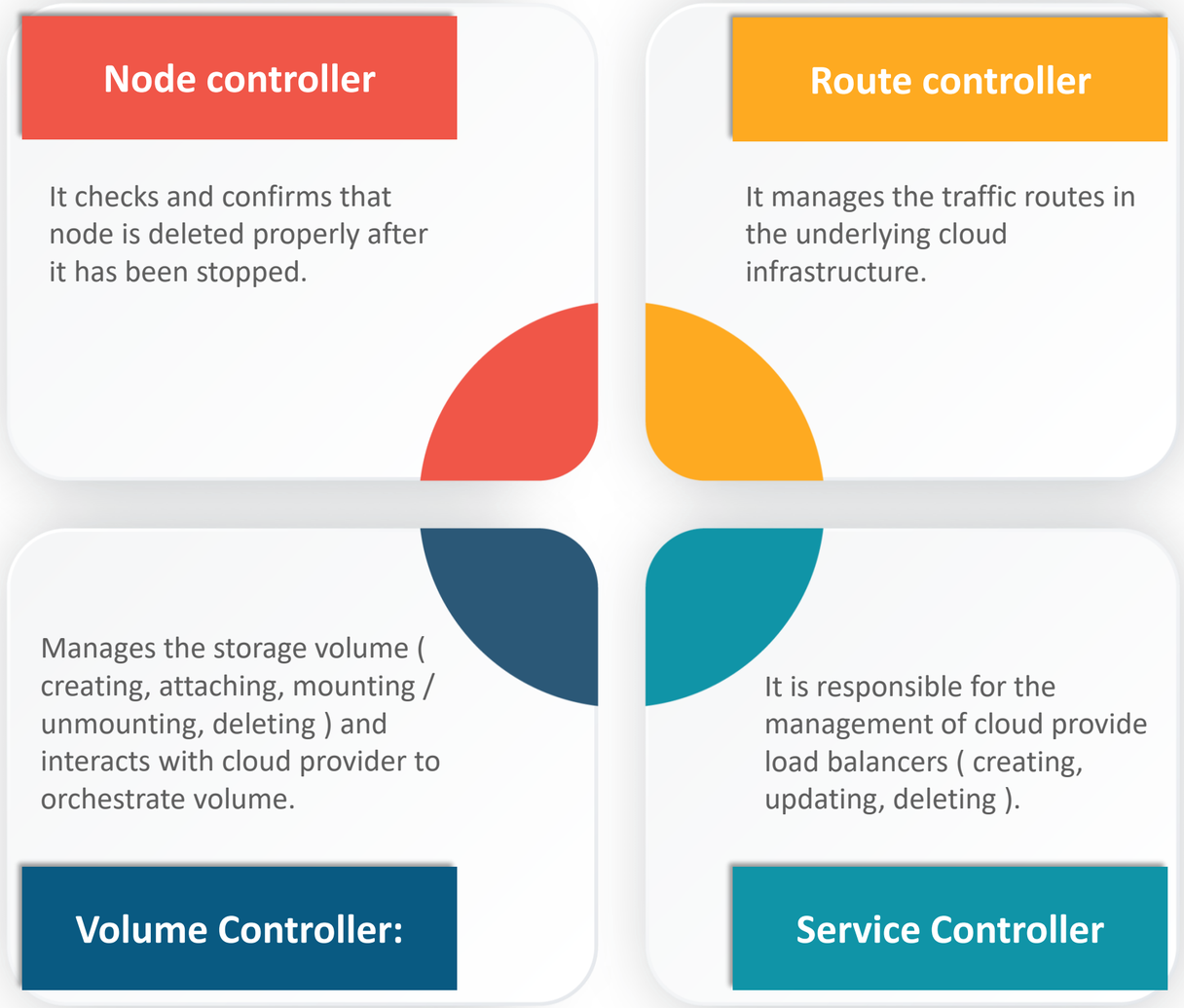
- Your cloud blocks the respective controller depending upon the use-case

- Cloud vendor develops their code and connects with kubernetes cloud-controller-manager while running the kubernetes

edureka!

Cloud-controller-manager Cont....:

- The cloud-controller has dependency on cloud service provider

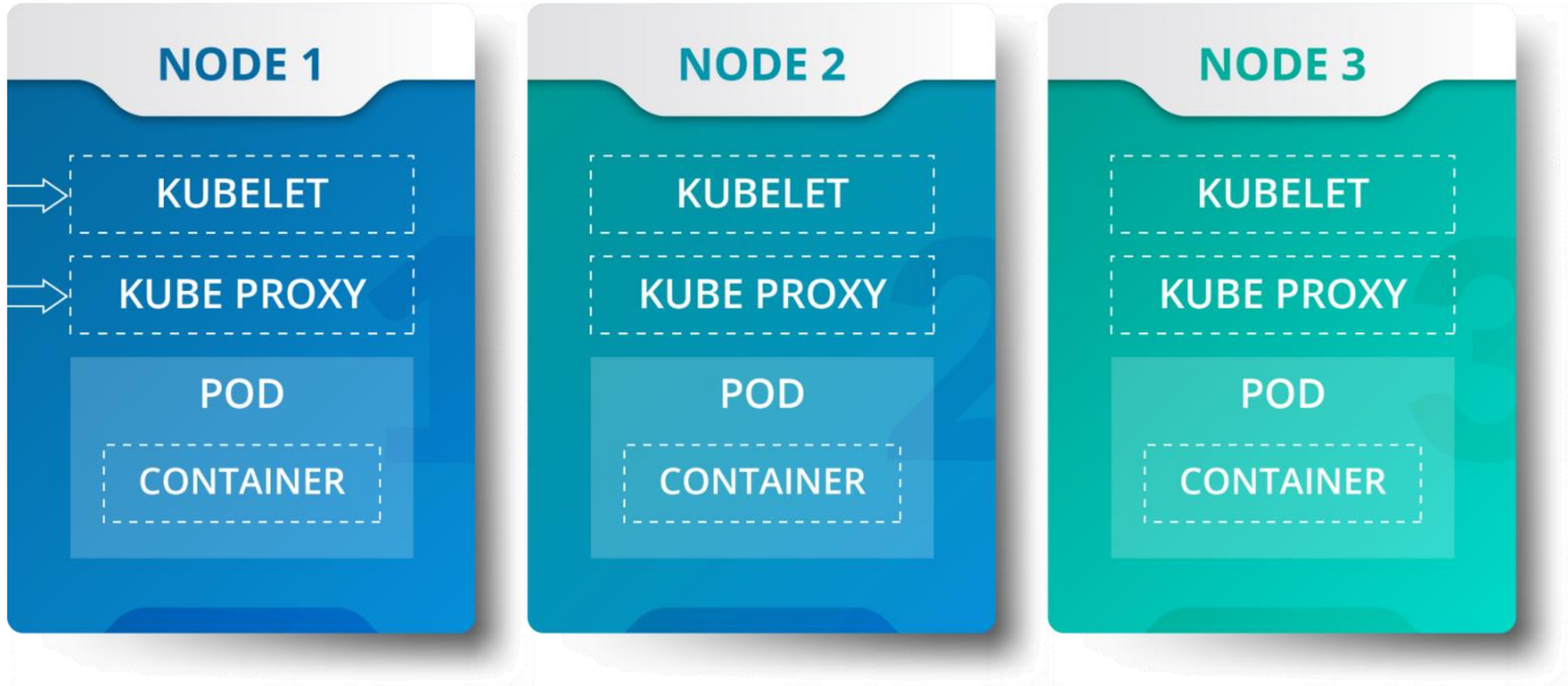


Node components

edureka!

Node Components of Kubernetes

- kubelet
- kube-proxy
- kubectl
- Container (Docker)



Node Components of Kubernetes

- Kubernetes nodes are the workhorses of the kubernetes cluster
- They were previously referred as Minions
- Customer workload or application runs on these nodes
- Multiple components run on each node:
  - Container runtime (Docker)
  - kubelet
  - kube-proxy
  - kubectl

Node Components of Kubernetes

1. **Container Runtime**
2. Kubelet
3. Kube-proxy
4. Kubectl

- **Container runtime** is the main software within a node which is responsible for running container

- **Docker:** Docker is just one but most popular container runtime which kubernetes supports
  - Other commonly used container runtime :
    - rkt
    - runc

edureka!

Node Components of Kubernetes

1. Container Runtime
2. **Kubelet**
3. Kube-proxy
4. Kubectl

- Kubelet is an agent service which runs on each node and enables slave to communicate with master
- Kubelet works on PodSpec that are provided to it . PodSpec contains description of containers
- Kubelet ensures that containers described in PodSpec are Healthy and running
- It only manages the container that are created by Kubernetes master to which it belongs

edureka!

Node Components of Kubernetes

1. Container Runtime
2. Kubelet
3. **Kube-proxy**
4. Kubectl

- Kubernetes kube-proxy runs on each node
- It can do simple TCP/UDP packet forwarding across backend network service
- It is a network proxy which reflects the services as configured in kubernetes API on each node
- Docker-link-compatible environment variables provides the cluster IPs and ports which are opened by proxy

edureka!

Node Components of Kubernetes

1. Container Runtime
2. Kubelet
3. Kube-proxy
4. **Kubectl**

- Kubectl is the interface/platform using which you can pass commands to the cluster
- It provides the CLI to run commands against the kubernetes cluster
- Kubectl command-line tool supports several different ways to create and manage kubernetes component

edureka!

DEMO : Create a Kubernetes Cluster

edureka!

Add-ons

edureka!

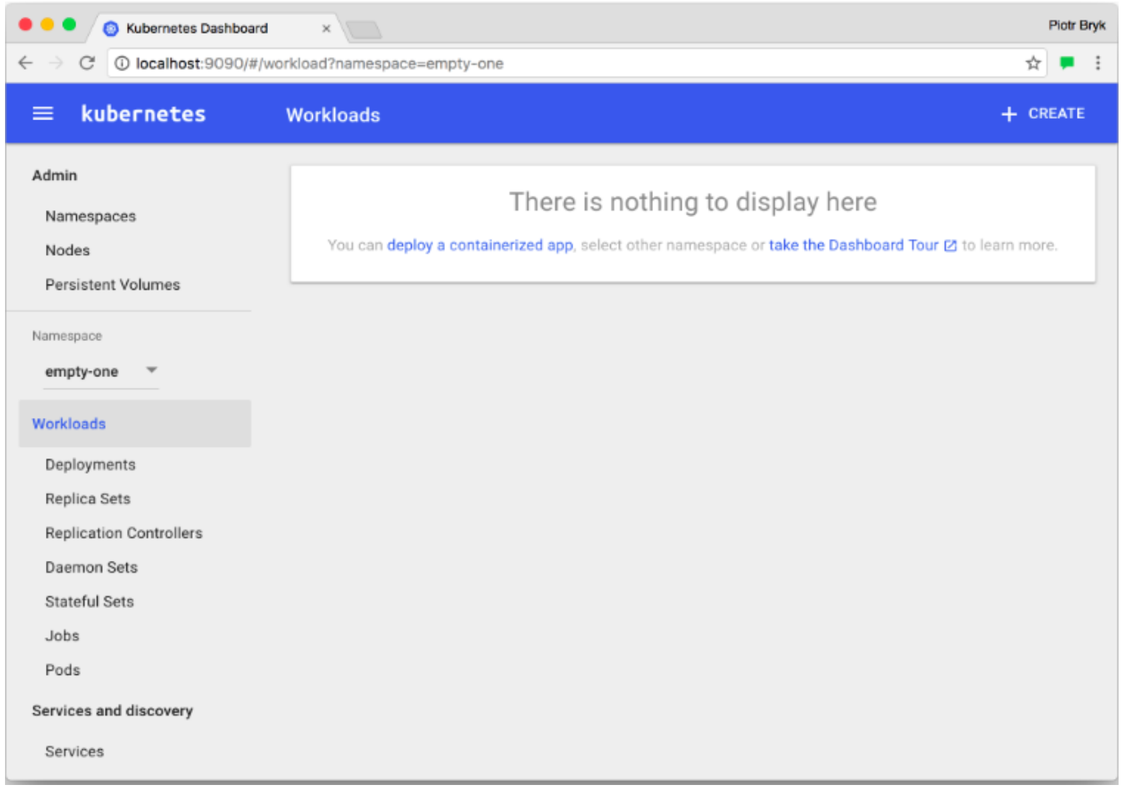
Addons: Cluster DNS

- This is an optional item with regards to the functionality of the kubernetes cluster
- For DNS, Kubernetes will create a DNS Pod and required services on the cluster
- Through kubelets, it will update rest of the containers on the cluster with the DNS Service IP and do the name resolution
- Every service in the kubernetes cluster is assigned a DNS name (including the DNS server itself)

Addons: Dashboard

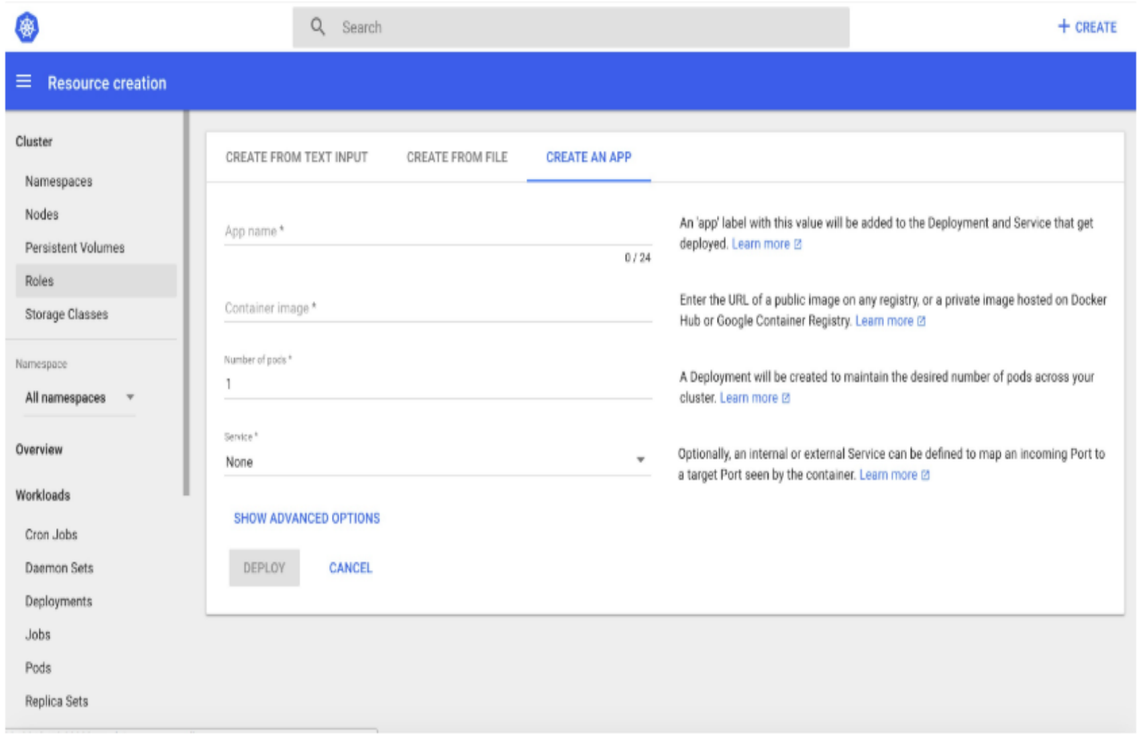
- Kubernetes Dashboard provides the GUI capabilities to the cluster to operate and manage clusters
- It eases the job of admin / developer / IT manager
    - Helps to containerize application
    - Troubleshooting containerized application
    - Overview of all the applications running on the containerized application

Addons: Kubernetes Dashboard



For Developers :

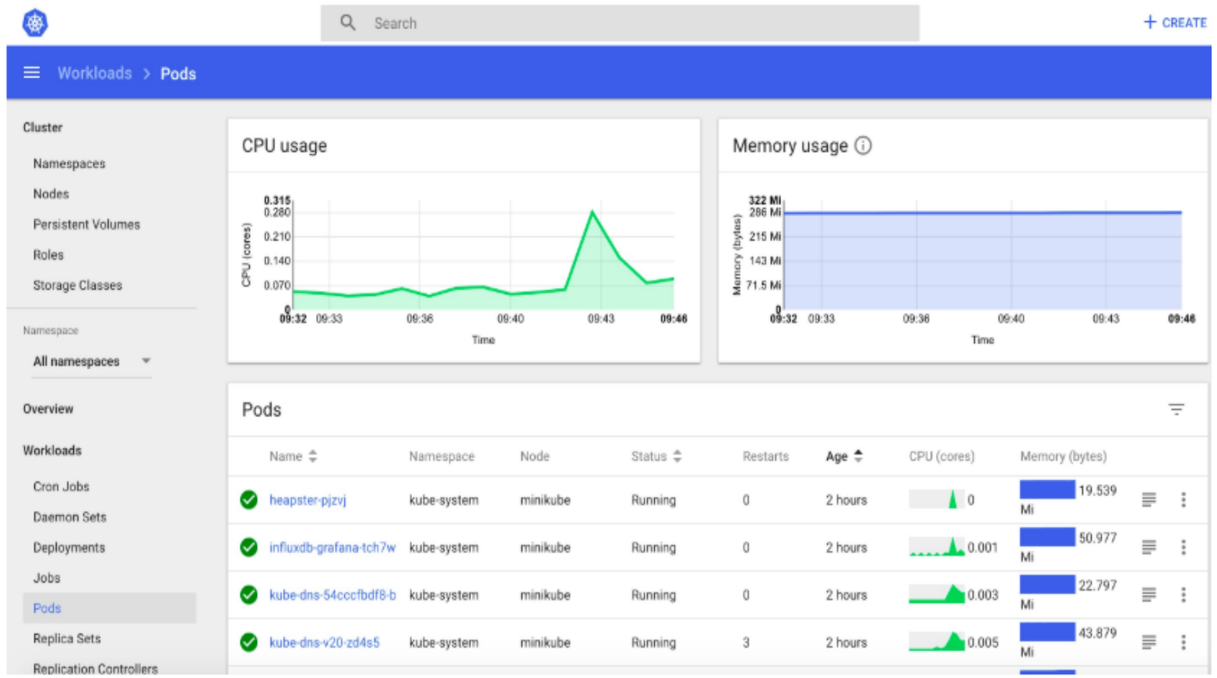
- View to containerize application



Ref : [Kubernetes Dashboard]

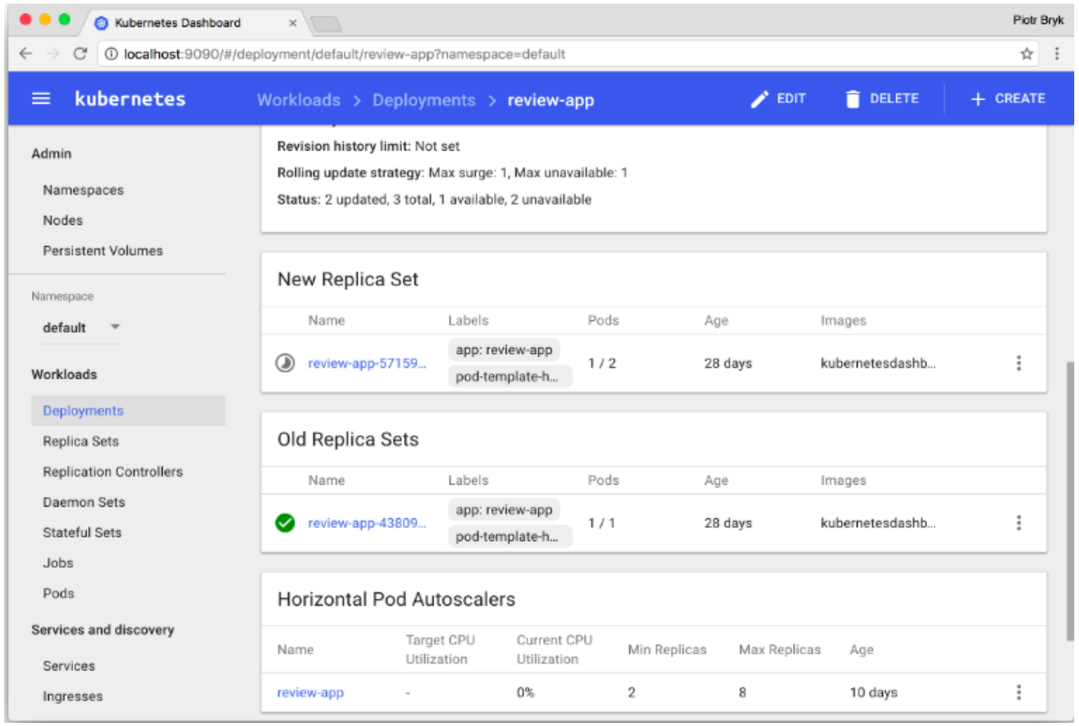
edureka!

Addons: Kubernetes Dashboard



Admin View

- Understand the usage of the cluster
- Workload overview



Ref : [Kubernetes Dashboard]

edureka!

Addons: Container Resource Monitoring

- Monitoring of kubernetes cluster environment is much different from regular client-server architecture
- Kubernetes factored the management of the cluster by creating abstraction at different level like container, pods, services and whole cluster
- For users, it becomes very important to understand the performance of the application and resource utilization at all the different abstraction layer
- This approach is very important for large cluster and distributed resources

Addons: Container Resource Monitoring Tools

Heapster

- Gathers data and events
- From the containers and pods within the cluster

InfluxDB

- Used along with Heapster for visualizing data within your Kubernetes environment

Grafana

- A time-series database
- Stores data captured by all the Heapster pods

CAdvisor

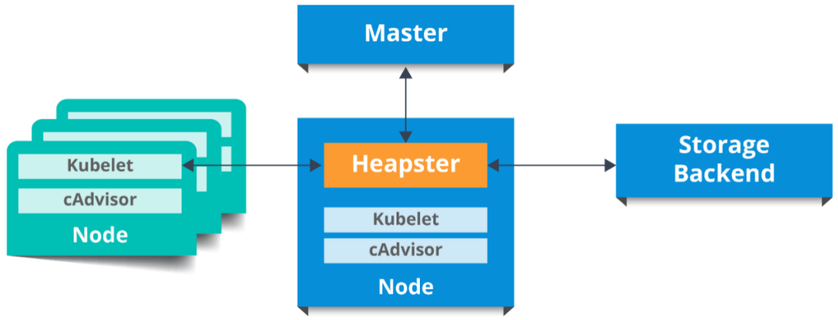
- Is a built-in tool in a kubelet that automatically discovers all the active containers monitors their performance and resource usage.

Prometheus

- Project of CNCF (Cloud Native Computing Foundation), which provides a powerful querying, alerting and visualization capabilities

Addons: Container Resource Monitoring Tools

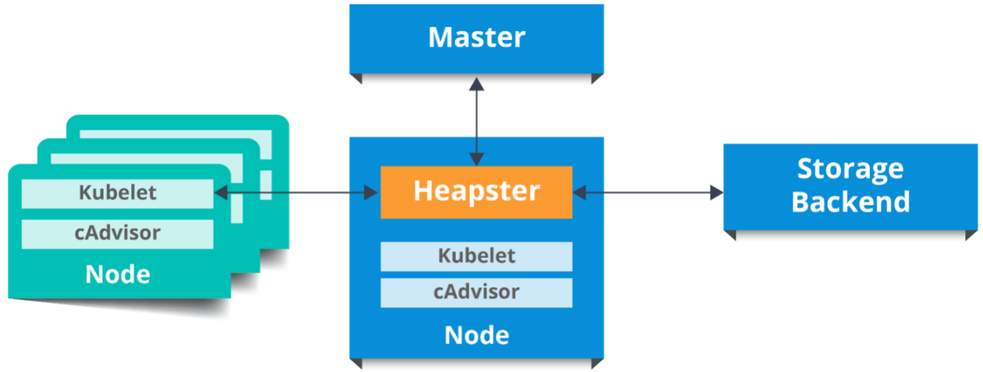
- On each node kubelet runs
- Kubelet fetches the data from cAdvisor
- This data is been passed on to Heapster
- Heapster classifies the data provided by kubelet and puts proper label to it
- This data is then pushed to the backend storage
- Visualization tools fetches the data from the backend storage and present it on the graphical dashboard



Addons: Process Flow(Heapster)

Let’s understand each components in more details:

- **Heapster:** It is a cluster wide aggregator of data provided by kubelet running on each node
- It is supported natively on kubernetes cluster
- It runs as a pod, just like any other pod in the kubernetes cluster
- It discovers all nodes in the cluster and queries usage information from the kubernetes nodes in the cluster, via on-machine kubernetes agent



Addons: Process Flow(cAdvisor)

cAdvisor:

- It is a resource usage and performance analysis agent
- It runs natively on docker containers and is a part of kubernetes binaries
- It runs on all the containers and collects :
  - CPU stats,
  - Memory stats,
  - Filesystem stats,
  - network usage statistics

Image ref link : Here

Installation of Tools – cAdvisor

```
$sudo docker run \
--volume=/:/rootfs:ro \
--volume=/var/run:/var/run:rw \
--volume=/sys:/sys:ro \
--volume=/var/lib/docker/:/var/lib/docker:ro \
--volume=/dev/disk/:/dev/disk:ro \
--publish=8080:8080 \
--detach=true \
--name=cadvisor \
google/cadvisor:latest
```

Addons: Process Flow(InfluxDB)

InfluxDB

- InfluxDB is used as a time-series database
- It is optimized for fast, highly available time-series queries used in operations, maintains and real-time analytics
- In kubernetes it runs as pod
- InfluxDB has no external dependencies
- It is accessed through sql-like language for querying built-in time-centric functions for data structures

Installation of Tools – Heapster using InfluxDB

```
$ kubectl create -f deploy/kube-config/influxdb/
$ kubectl create -f deploy/kube-config/rbac/heapster-rbac.yaml
```

YAML file can be taken from:
https://github.com/kubernetes/heapster/tree/master/deploy/kube-config/influxdb

Addons: Process Flow(Grafana)

Grafana

- InfluxDB works with Grafana
- Grafana provides graphical representation of Data processed by heapster and stored in influxDB
- Grafana is also an opensource tool, known for time-series analytics and visualization
- Grafana dashboard can easily be optimised and expanded

Installation of Tools - Grafana

- Installation

```
$ docker run -i -p 3000:3000 grafana/grafana
```

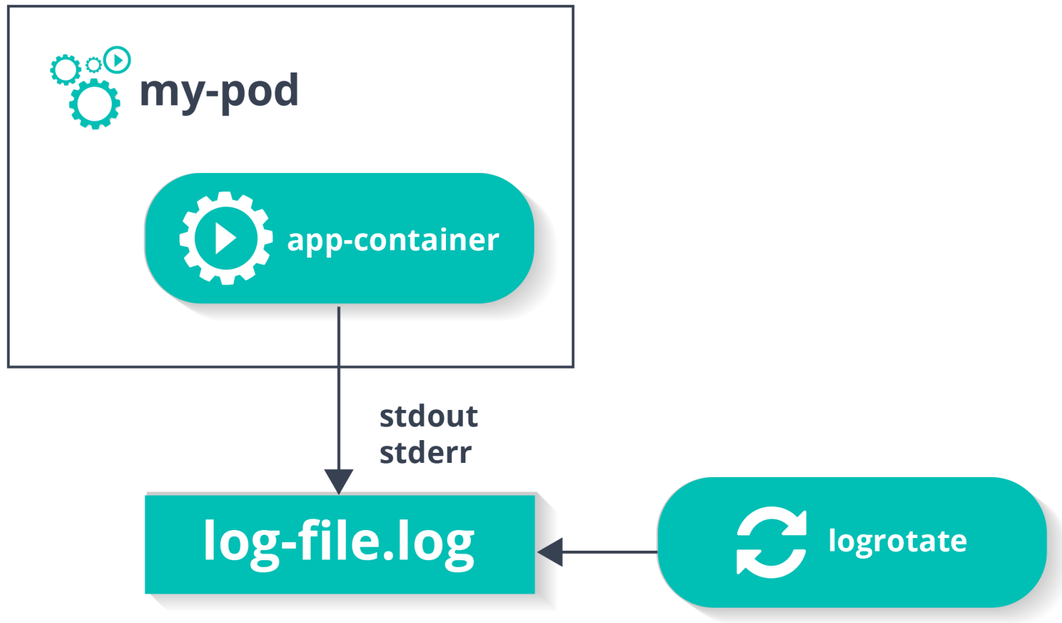
- Configuration:
  - sqlite3 database - /var/lib/grafana
  - configuration files - /etc/grafana/

Addons: Cluster level logging

- Cluster level logging requires additional backend to store
- By default, kubernetes does not have native storage for log data
- Cluster level logging assumes that there is a data-store inside or outside of kubernetes cluster
- All the containerized application writes to stdout
- Docker includes multiple logging mechanism
- Multiple logging mechanism helps to pull out the logs from each containers - These mechanisms are called logging driver

Addons: Cluster level logging *(Cont...)*

- Kubenetes does not provide log rotation by default
- Additional log-rotation tools needs to be implement to do so, avoiding logs to consume all the storage space

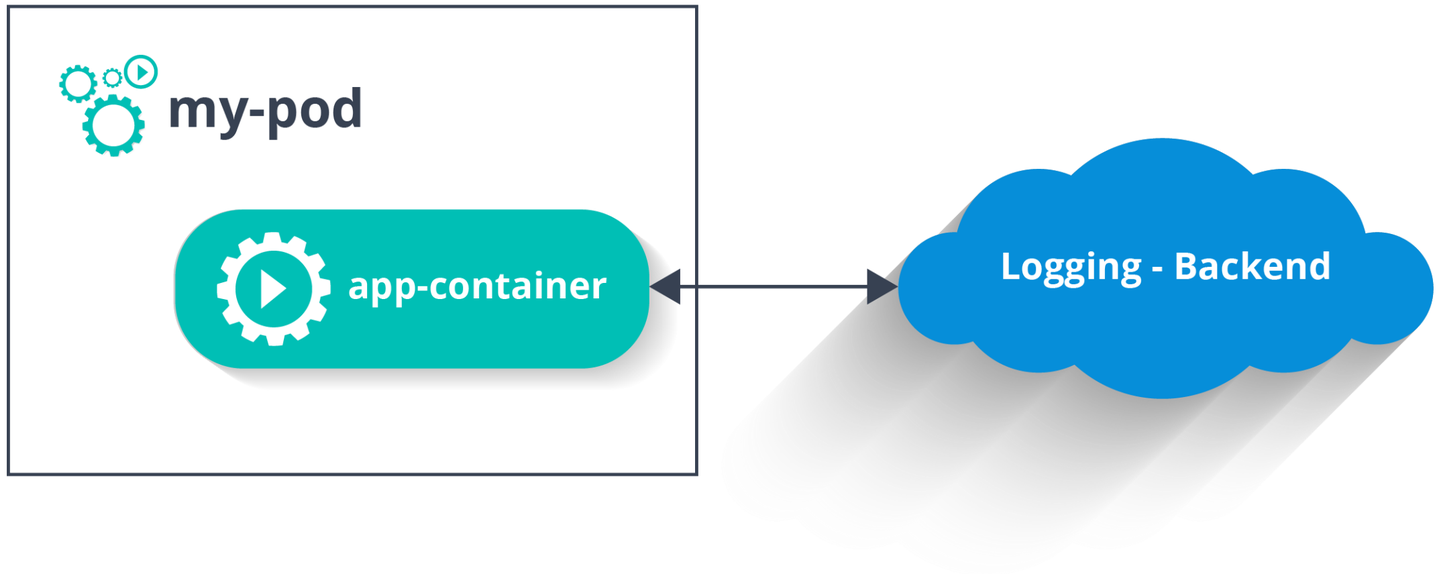


Addons: Cluster level logging *(Cont...)*

- As mentioned earlier, kubernetes does not provide native solution for cluster-level logging. To handle this, here are few approaches. :
  - Configure mechanism to move the logs directly from application to a designated place like backend storage
  - Create another container next to main container (also called sidecar) for logging purpose
  - Use an individual node-level agent to push the logs from individual to a designated place

Let’s discuss each one of them in detail…

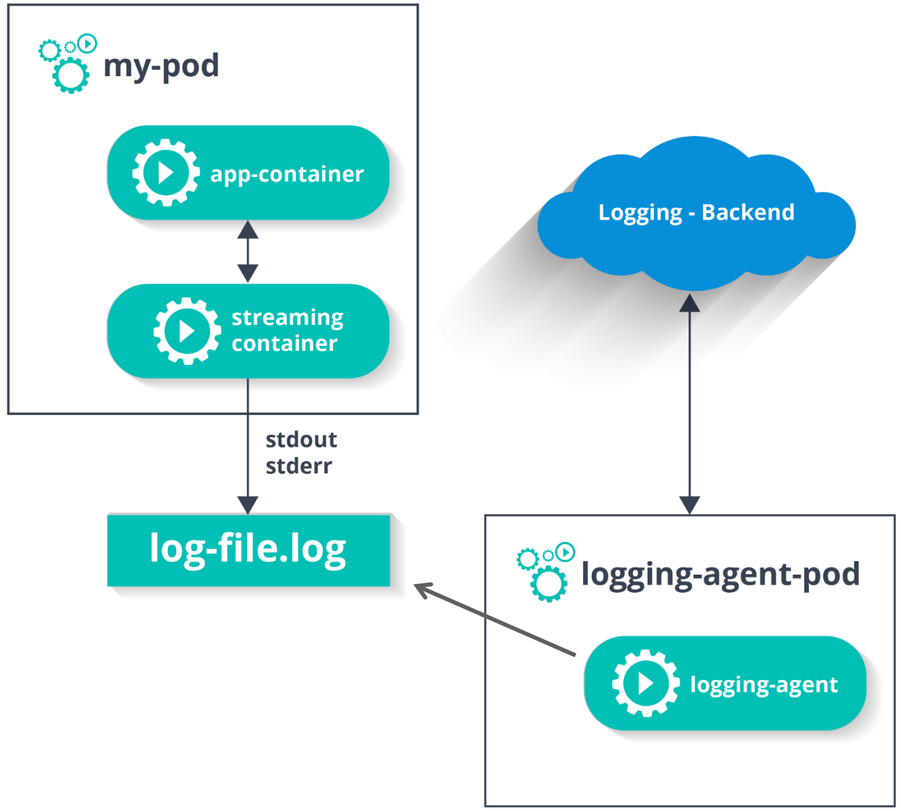
Addons: Cluster level logging *(Cont...)*



- Configure mechanism to move the logs directly from application to designated place like backend storage
- Configuration of this kind of logging is outside the scope of kubernetes and needs to be taken care by the application vendor / team

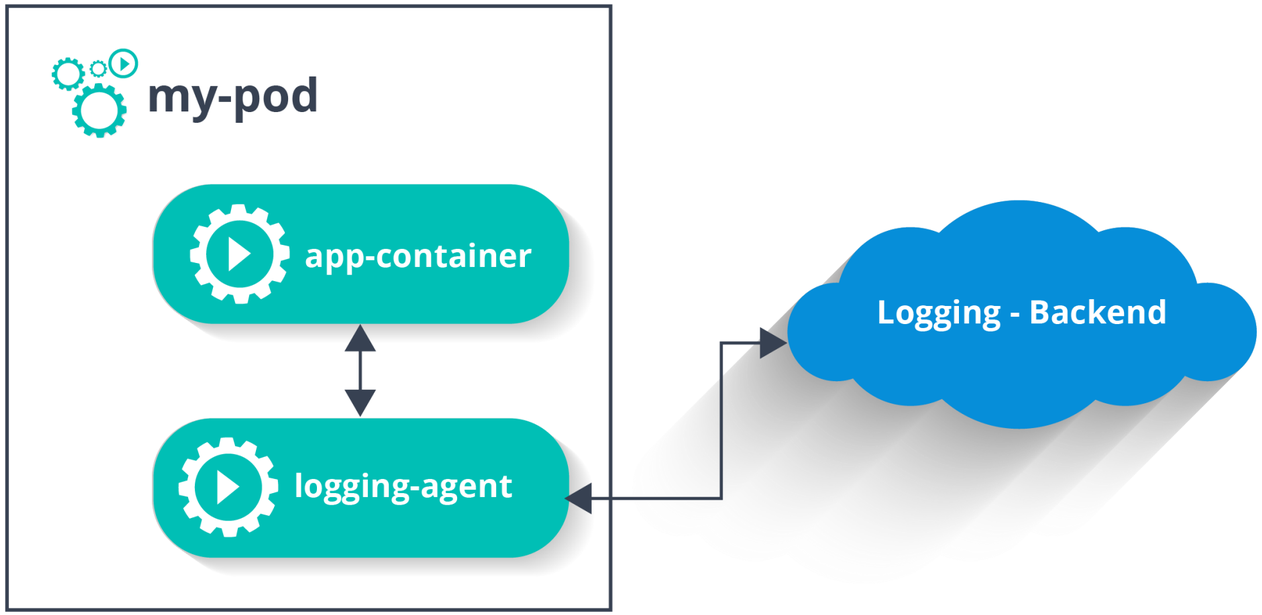
Addons: Cluster level logging *(Cont...)*

- Create another container next to main container (also called sidecar) for logging purpose
- In this approach, you can take the advantage of kubelet and logging agent
- Sidecar reads the logs from std-err and std-out from file and prints to its stream
- Another pods running with the logging agents on the cluster picks up the log
- This approach does not consume significant resource from the application container



edureka!

Addons: Cluster level logging (Cont...)



- This container runs the side-car container with logging agent running to it
- Drawback with this approach:
  - Side-car container consumes lot of resources
  - Logs can not be accessed using kubctl

Kubeadm

- For test and practise point of view, minikube was the first project which made the installation of kubernetes locally hasslefree
- Later, ubuntu introduced juju which allowed to deploy multi node kubernetes cluster in ubuntu
- Now, there is a new project from Mirantis which gives users the abiility to create multi-node kubernetes clusters on any operation system where you can run docker
- This helps to understand how kubernetes clusters works , how they are created in production environment
- Kubeadm - kubeadm is a tool to bootstrap kubernetes cluster on existing infrastructure
- In this lab, we will setup the cluster on node-2

Pre-requisite

- Install VM (which you have done.)
- Install docker as you did in hands-on in Module-1
- In this lab, we will setup the cluster on node-2

DEMO : Managing Cluster

edureka!

Get Cluster Details

- Check docker containers, pods

```
sudo kubectl get nodes
sudo docker ps
sudo kubectl get pods --all-
namespaces
Sudo kubectl cluster-info
```

```
[sachin@node-2:~/kubeadm-dind-cluster$ sudo kubectl cluster-info
Kubernetes master is running at http://localhost:8080
KubeDNS is running at http://localhost:8080/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
```

```
[sachin@node-2:~/kubeadm-dind-cluster/fixed$
[sachin@node-2:~/kubeadm-dind-cluster/fixed$ sudo kubectl get nodes
NAME          STATUS     ROLES     AGE       VERSION
kube-master   Ready      master    1h        v1.8.11
kube-node-1   NotReady   <none>    1h        v1.8.11
kube-node-2   NotReady   <none>    1h        v1.8.11
[sachin@node-2:~/kubeadm-dind-cluster/fixed$ sudo docker ps
CONTAINER ID        IMAGE                                COMMAND                  CREATED             STATUS              PORTS                    NAMES
c35666f97646        mirantis/kubeadm-dind-cluster:v1.8   "/sbin/dind_init sys…"   24 minutes ago      Up 24 minutes       8080/tcp                 kube-node-2
a874fd72008a        mirantis/kubeadm-dind-cluster:v1.8   "/sbin/dind_init sys…"   24 minutes ago      Up 24 minutes       8080/tcp                 kube-node-1
a3e076e14a84        mirantis/kubeadm-dind-cluster:v1.8   "/sbin/dind_init sys…"   24 minutes ago      Up 24 minutes       127.0.0.1:8080->8080/tcp   kube-master
[sachin@node-2:~/kubeadm-dind-cluster/fixed$ sudo kubectl get pods --all-namespaces
NAMESPACE     NAME                                    READY     STATUS    RESTARTS   AGE
kube-system   etcd-kube-master                        1/1       Running   2          1h
kube-system   kube-apiserver-kube-master              1/1       Running   2          1h
kube-system   kube-controller-manager-kube-master     1/1       Running   2          23m
kube-system   kube-dns-855bdc94cb-52dq5               3/3       Running   0          24m
kube-system   kube-proxy-kxrc4                        1/1       Running   0          24m
kube-system   kube-proxy-mxcrf                        1/1       Running   0          24m
kube-system   kube-proxy-svr8p                        1/1       Running   0          24m
kube-system   kube-scheduler-kube-master              1/1       Running   2          1h
kube-system   kubernetes-dashboard-6b5bdcfbc6-xc8w9   1/1       Running   0          24m
sachin@node-2:~/kubeadm-dind-cluster/fixed$
```

edureka!

Install KubeCtl

- Run these commands as root

```
Install kubelet kubeadm kubectl
#apt-get update && apt-get install -y apt-transport-https
curl
#curl -s https://packages.cloud.google.com/apt/doc/apt-
key.gpg | apt-key add -
#cat <<EOF >/etc/apt/sources.list.d/kubernetes.list
 deb http://apt.kubernetes.io/ kubernetes-xenial main
 EOF
#apt-get update
#apt-get install -y kubelet kubeadm kubectl
```

Install KubeCtl

- Turn-off the swap

```
sachin@node-2:~/kubeadm-dind-cluster$ sudo swapoff -a
sachin@node-2:~/kubeadm-dind-cluster$ sudo vi /etc/fstab
sachin@node-2:~/kubeadm-dind-cluster$
```

- Set the permission

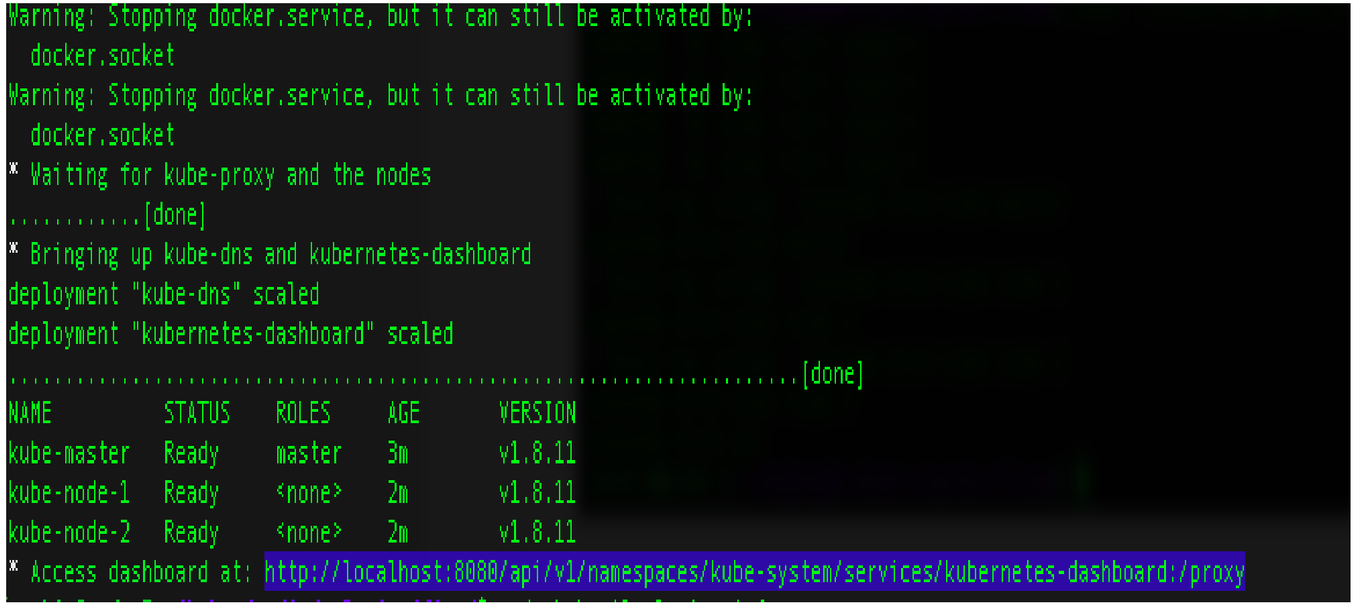
```
sachin@node-2:~/kubeadm-dind-cluster$ ls
architecture.md  code-of-conduct.md  dind-cluster.sh  gce-setup.sh  OWNERS          SECURITY_CONTACTS
AUTHORS.md       config.sh           Dockerfile       image         OWNERS_ALIASES  test.sh
build            CONTRIBUTING.md     fixed            LICENSE       README.md
sachin@node-2:~/kubeadm-dind-cluster$ cd fixed/
sachin@node-2:~/kubeadm-dind-cluster/fixed$ ls
dind-cluster-stable.sh  dind-cluster-v1.10.sh  dind-cluster-v1.8.sh  dind-cluster-v1.9.sh
sachin@node-2:~/kubeadm-dind-cluster/fixed$ chmod +x dind-cluster-v1.8.sh
```

DEMO: Create Kubernetes Dashboard

edureka!

Access The Dashboard

- Dashboard access weblink would be provided at the time of creation of the cluster



Quiz

1. What is ETCD in kubernetes cluster ?
    a. Key-value pair to store information for configuration
    b. It is an English term to say “Et cetera” , meaning other things.
    c. Selects the most suitable node to run the unscheduled pod
    d. None of the above.

Answers

A

1. What is ETCD in kubernetes cluster ?
   a. Key-value pair to store information for con
   b. It is an English term to say “Et cetera” , meaning other things.
   c. Selects the most suitable node to run the unscheduled pod
   d. None of the above.

Answer A

edureka!

Quiz

2. What are minions kubernetes cluster ?

    a. They are cartoon character

    b. They are work-horse / worker node of the kubernetes cluster

    c. They are monitoring engine used widely in kubernetes

    d. They are docker container service.

Answers

A

2. What are minions kubernetes cluster ?

   a. They are cartoon character

   b. They are work-horse / worker node of the kubernetes cluster

   c. They are monitoring engine used widely in kubernetes

   d. They are docker container service.

Answer B

edureka!

Quiz

3. Name controller-manager which has got dependency on cloud provider ?

    a. Node Controller : It checks and confirms that node is deleted properly after it has been stopped.

    b. Route controller: It manager the traffic routes in the underlying cloud infrastructure.

    c. Service Controller: It is responsible for the management of cloud provide load balancers (creating, updating, deleting).

    d. Volume Controller: Manages the storage volume (creating, attaching, mounting / unmounting, deleting) and interacts with cloud provider to orchestrate volume

    e. All of the above

Answers

A

3. Name controller-manager which has got dependency on cloud provider ?

    a. Node Controller : It checks and confirms that node is deleted properly after it has been stopped.

    b. Route controller: It manager the traffic routes in the underlying cloud infrastructure.

    c. Service Controller: It is responsible for the management of cloud provide load balancers (creating, updating, deleting).

    d. Volume Controller: Manages the storage volume (creating, attaching, mounting / unmounting, deleting) and interacts with cloud provider to orchestrate volume

    e. All of the above

Answer E: All of the above

Quiz

4. Kube-scheduler keeps track of resource utilization on all the compute nodes and ensure that workload is not scheduled on node which is already having scarcity of resources.

   a. True

   b. False

Answers

A

4. Kube-scheduler keeps track of resource utilization on all the compute nodes and ensure that workload is not scheduled on node which is already having scarcity of resources.

   a. True

   b. False

Answer A

Quiz

5. Map the definition correctly

   a. Heapster: 1) Comes as a built-in tool into kubelet and automatically discover all the active containers. It focuses on container level performance and resource usage.

   b. InfluxDB : 2) A time-series database that stores the data captured by all the Heapster pods

   c. Grafana: 3) Used along with Heapster for visualizing data within your Kubernetes environment

   d. CAdvisor: 4) It gathers data and events from the containers and pods within the cluster

Answers

A

5. Map the definition correctly

   a. Heapster: 1) Comes as a built-in tool into kubelet and automatically discover all the active containers. It focuses on container level performance and resource usage.

   b. InfluxDB : 2) A time-series database that stores the data captured by all the Heapster pods

   c. Grafana: 3) Used along with Heapster for visualizing data within your Kubernetes environment

   d. CAdvisor: 4) It gathers data and events from the containers and pods within the cluster

Answer :
A-4 , b-2, c-3, d-1

Quiz

6. What is prometheus ?

Answers

A

6. What is prometheus ?

Answer : Project of CNCF (Cloud Native Computing Foundation), which provides a powerful querying, alerting and visualization capabilities

Quiz

7. Is heapster mandatory component for docker engine to work properly?

   a. Yes

   b. No

Answers

A

7. Is heapster mandatory component for docker engine to work properly?

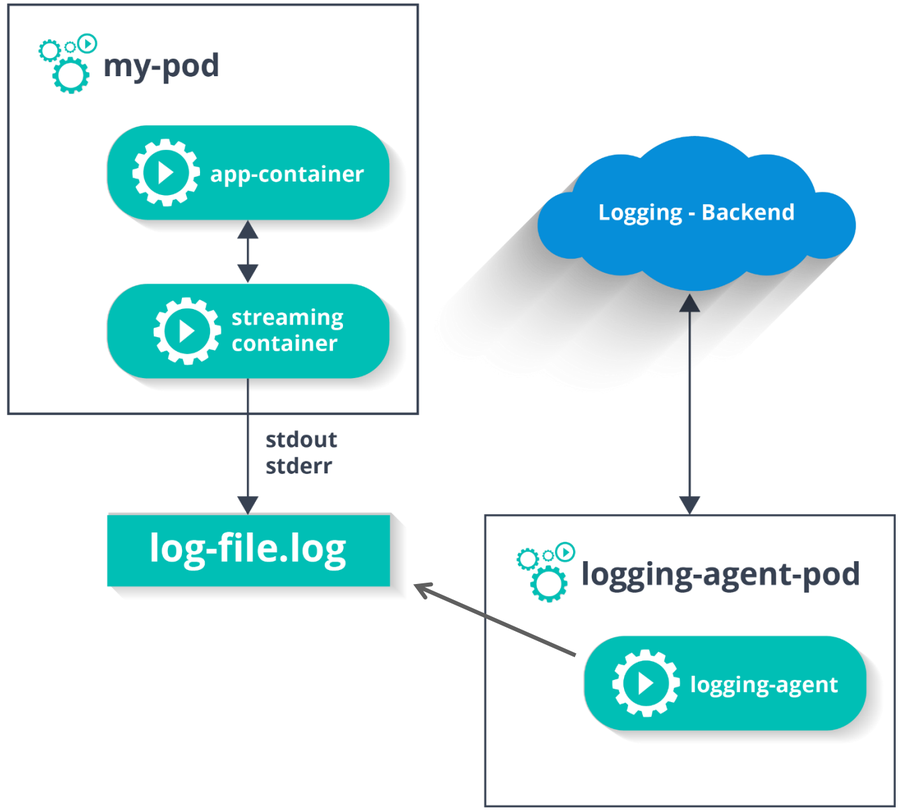
   a. Yes

   b. No

Answer B: No, It is a cluster wide aggregator of data provided by kubelet running on each node.
It is used for resource monitoring. Nothing to do with basic functionality of kubernetes or docker engine.

Quiz

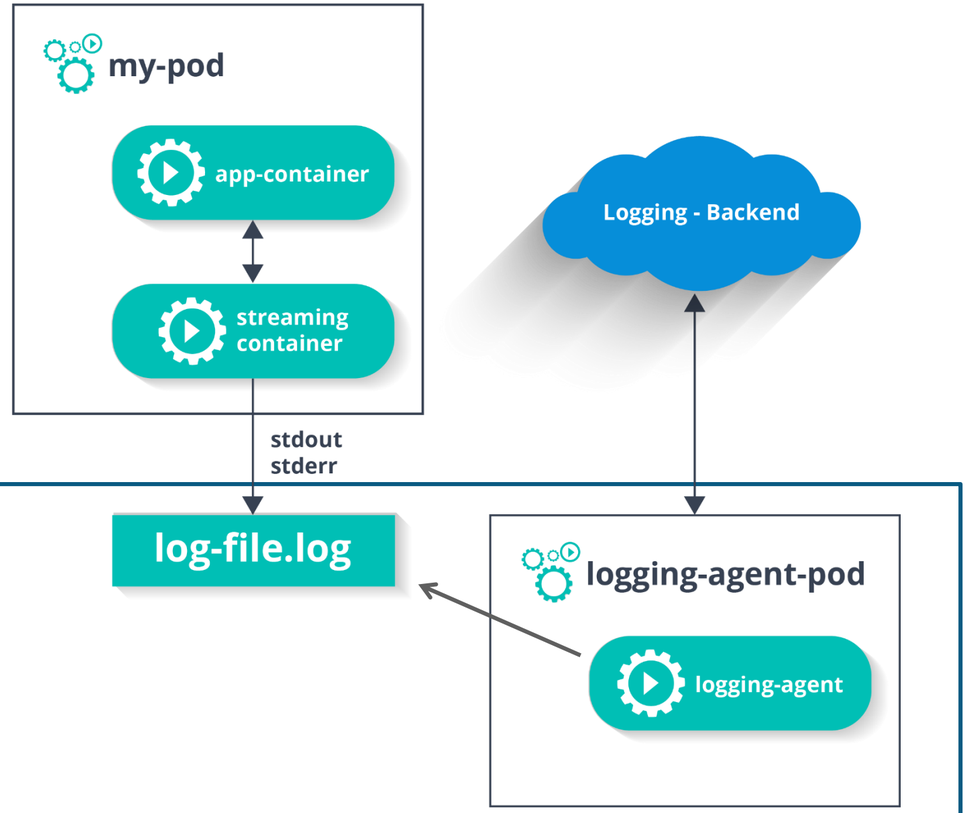
8. From the picture on side, explain the type of architecture it is ?



Answers

A

8. From the picture on side, explain the type of architecture it is ?



Answer : Side-car container with logging-agent outside the application container

Summary

- In this module, you should have learnt:
  - Master Components of Kubernetes
  - Node components of Kubernetes
    - Docker, kubelet, kube-proxy, kubectl
  - Addons
    - Cluster DNS
    - Kubernetes Dashboard
    - Container Resource Monitoring
    - Cluster level logging

edureka!

edureka!

Thank You

For more information please visit our website
www.edureka.co