**KPLABS Course**

**Certified Kubernetes Application Developer**

**Observability**

## **ISSUED BY**

Zeal Vora

## **REPRESENTATIVE**

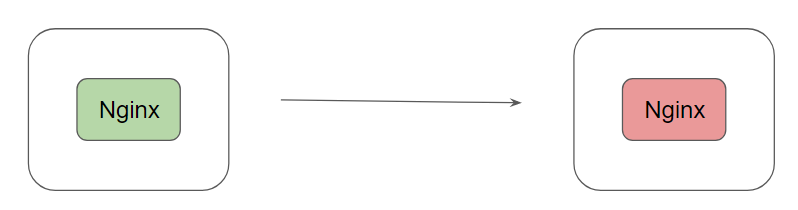
[instructors@kplabs.in](mailto:instructors@kplabs.in)

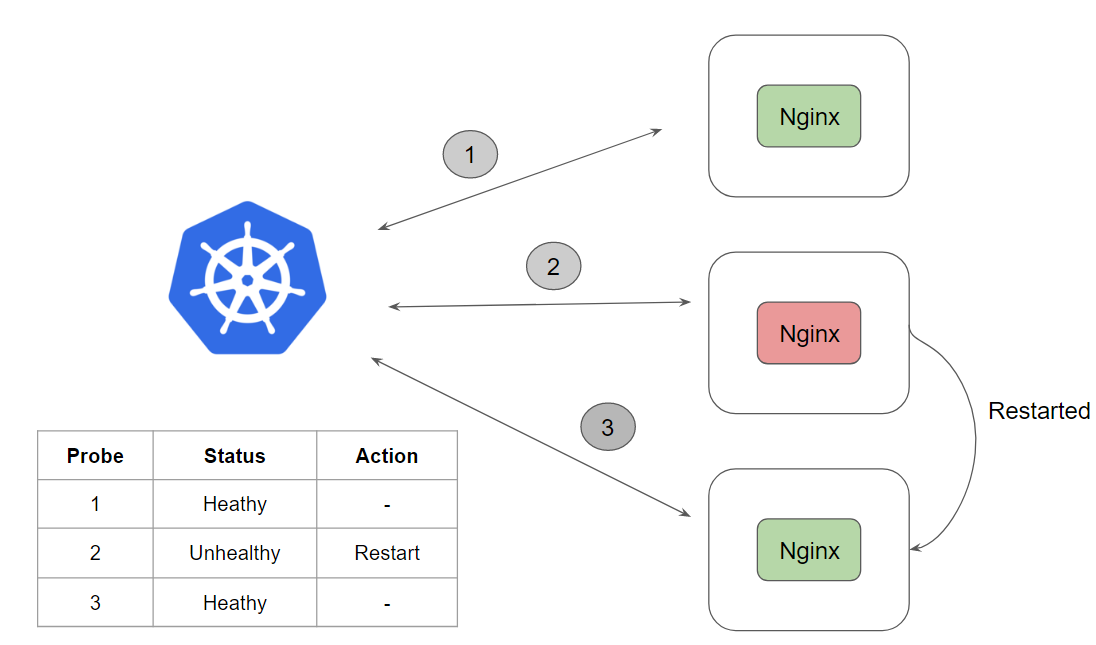
# 

**Module 1: Understanding Liveness Probe**

Many applications running for long periods of time eventually transition to broken states, and cannot recover except by being restarted.

Kubernetes provides liveness probes to detect and remedy such situations.





There are 3 types of probes which can be used with Liveness

* HTTP
* Command
* TCP

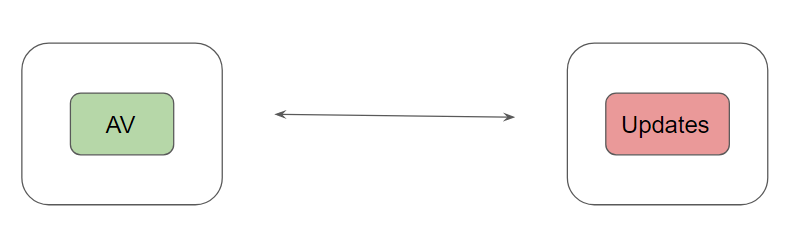
In this demo we had taken an example based on command.

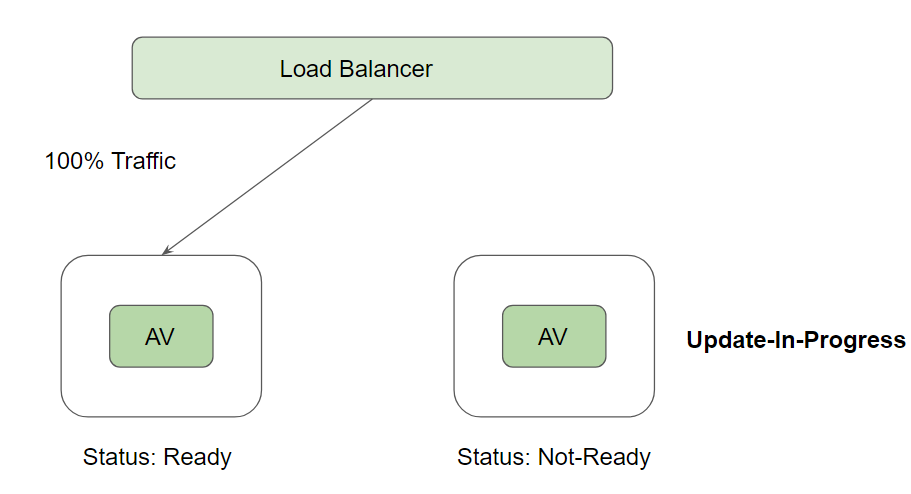
**Module 2: Understanding Readiness Probe**

It can happen that an application is running but temporarily unavailable to serve traffic.

For example, an application is running but it is still loading it’s large configuration files from an external vendor.

In such-case, we don’t want to kill the container however we also do not want it to serve the traffic.





Syntax of Readiness Probe:

readinessProbe:

exec:

command:

- cat

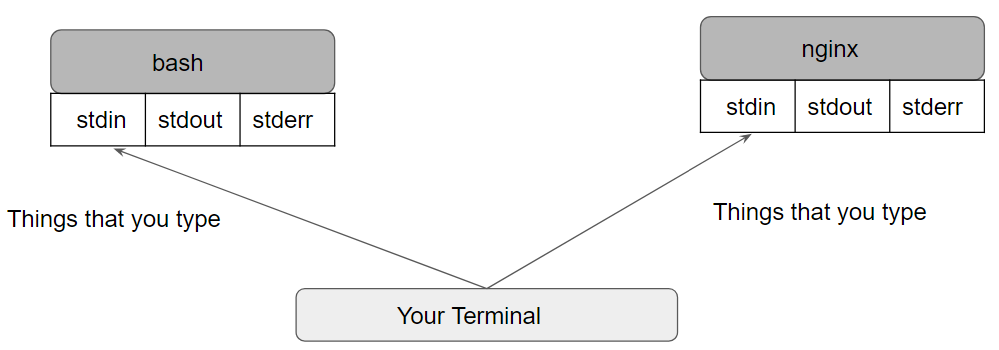
- /tmp/healthy

initialDelaySeconds: 5

periodSeconds: 5

**Module 3: Understanding Docker Logging Drivers**

UNIX and Linux commands typically open three I/O streams when they run, called STDIN, STDOUT, and STDERR

****

There are a lot of logging driver options available in Docker, some of these include:

* json-file
* none
* syslog
* local
* journald
* splunk
* awslogs

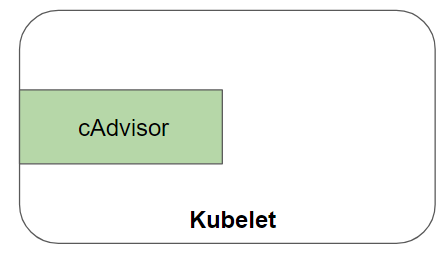
The docker logs command is not available for drivers other than json-file and journald.

**Module 4: Monitoring Cluster Components**

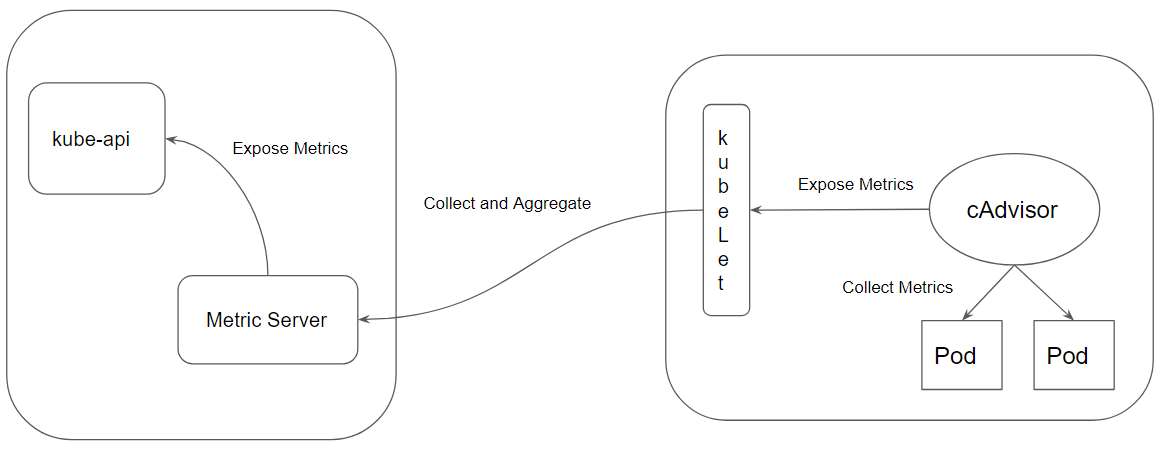
4.1 Overview about cAdvisor

One of the important functionalities of a Kubelet is to retrieve metrics aggregate and expose them through the Kubelet Summary API.

This is achieved with cAdvisor.



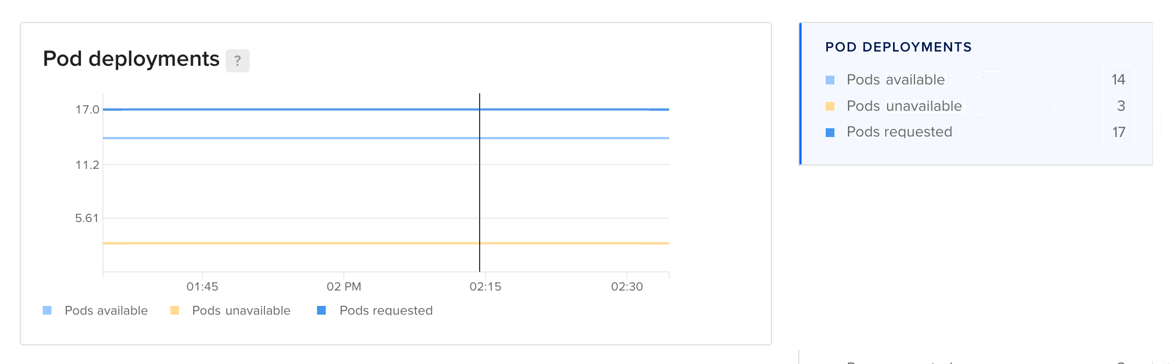
The following diagram illustrates the high-level workflow



4.2 Overview about kube-state-metrics

kube-state-metrics is a simple service that listens to the Kubernetes API server and generates metrics about the state of the objects.

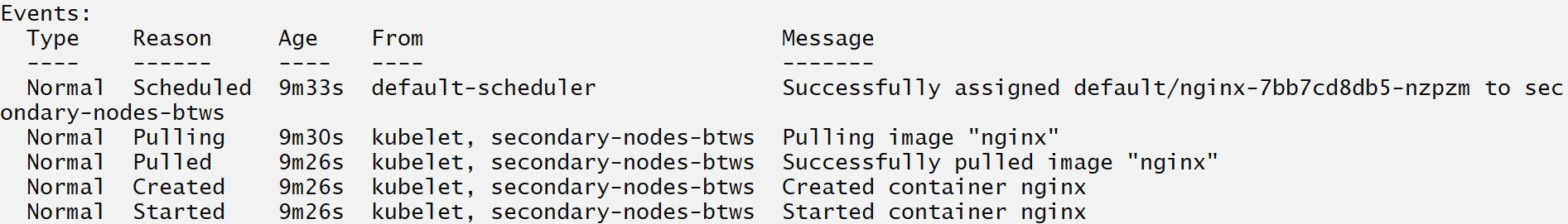
It is not focused on the health of the individual Kubernetes components, but rather on the health of the various objects inside, such as deployments, nodes and pods.



**Module 5: Kubernetes Events**

Kubernetes Events are created when other resources have state changes, errors, or other messages that should be broadcast to the system.

It provides insight into what is happening inside a cluster, such as what decisions were made by scheduler or why some pods were evicted from the node.



Events and Namespaces

Events are namespaced.

Hence if you want event of a pod in “kplabs-namespace” then you will have to explicitly specify the --namespace kplabs-namespace.

To see events from all namespaces, you can use the --all-namespaces argument.

Important Pointer

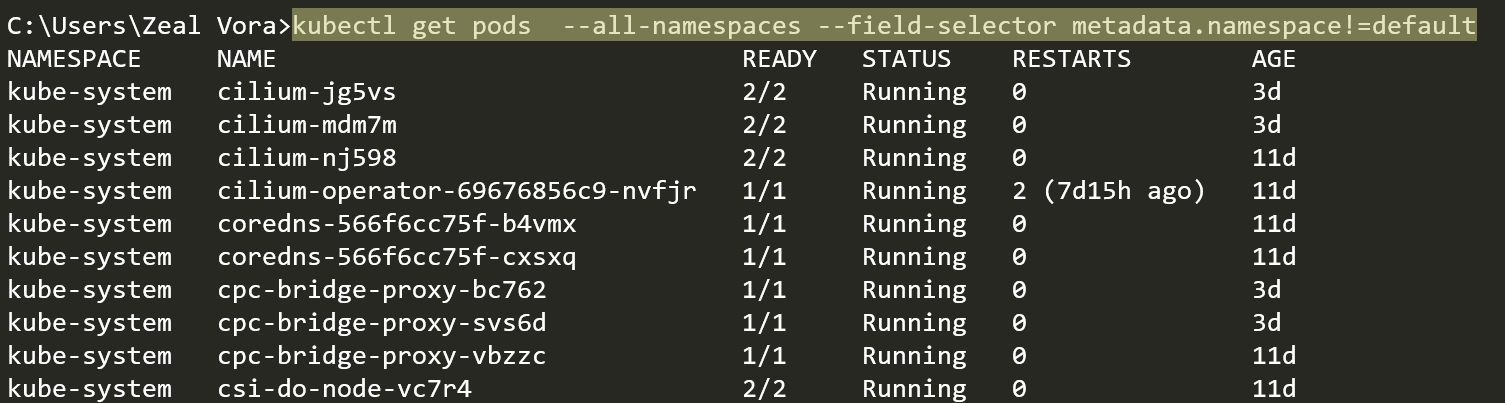
All the events are stored in the master server.

To avoid filling up master’s disk, a retention policy is enforced: events are removed one hour after the last occurrence.

To provide longer history and aggregation capabilities, a third party solution should be installed to capture events.

**Module 6: Field Selectors**

Field selectors let you select Kubernetes resources based on the value of one or more resource fields



By default, no selectors/filters are applied, meaning that all resources of the specified type are selected.

This makes the kubectl queries kubectl get pods and kubectl get pods --field-selector "" equivalent.

## **Join Our Discord Community**

We invite you to join our Discord community, where you can interact with our support team for any course-based technical queries and connect with other students who are doing the same course.

Joining URL:

<http://kplabs.in/chat>

