Kubernetes schedules applications dynamically based on scheduling policy, so you don’t always know where applications are running. But they still need to be monitored. That’s why using a monitoring system or tool with service discovery is a must. It will automatically adapt metric collection to moving containers so applications can be continuously monitored without interruption.

Kubernetes has the ability to distribute containerized applications across multiple data centers and potentially different cloud providers. That means metrics must be collected and aggregated among all these different sources.

1. **Metrics to monitor**

A monitoring tool integrating with Kubernetes and its different APIs, there are several key types of metrics that need to be closely tracked:

Running pods and their deployments

Usual resource metrics such as CPU, memory usage, and disk I/O

Container-native metrics

Application metrics for which a service discovery feature in your monitoring tool is essential

All these metrics should be aggregated using Kubernetes labels and correlated with events from Kubernetes and container technologies.

1. **Collecting these metrics**

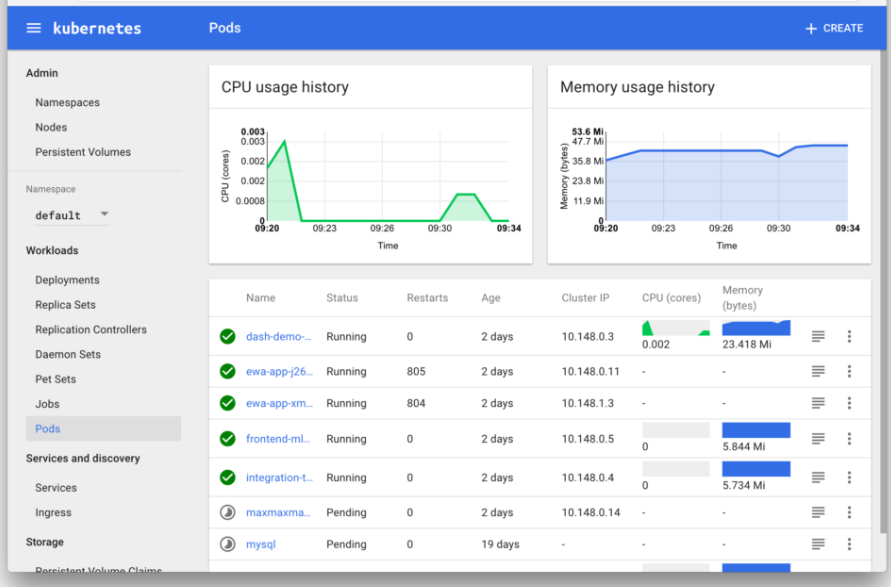
Whether you want to track these key performance metrics by combining Heapster, a storage backend, and a graphing tool, or by integrating a monitoring tool with the different components of your infrastructure.

1. Configure public REST API endpoint monitoring alert.
2. Alerting on Services Running on Kubernetes:

Tools :

* Probes
* cAdvisor
* Heapster
* Kubernetes Dashboard
* Kube-state-metrics

1. Liveness and Readiness Probes
2. Dashboard:



**Cluster Monitoring:** For cluster monitoring, the objective is to monitor the health of the entire Kubernetes cluster. As an administrator, we are interested in discovering if all the nodes in the cluster are working properly and at what capacity, how many applications are running on each node, and the resource utilization of the entire cluster.

Node resource utilization – there are many metrics in this area, all related to resource utilization. Network bandwidth, disk utilization, CPU, and memory utilization are examples of this. Using these metrics, one can find out whether or not to increase or decrease the number and size of nodes in the cluster.

The number of nodes – the number of nodes available is an important metric to follow. This allows you to figure out what you are paying for (if you are using cloud providers), and to discover what the cluster is being used for.

Running pods – the number of pods running will show you if the number of nodes available is sufficient and if they will be able to handle the entire workload in case a node fails.

**Pod Monitoring**

The act of monitoring a pod can be separated into three categories: (1) Kubernetes metrics, (2) container metrics, and (3) application metrics.

**Monitoring Kubernetes with Datadog:**

Monitoring the Dockerized infrastructure orchestrated with Kubernetes requires a tool capable of:

Ingesting metrics from all the different layers of your infrastructure, even if your clusters are distributed across multiple data centers or cloud providers

Aggregating metrics around Kubernetes labels for better context

Tracking your running applications via Autodiscovery as they move across hosts

All the advanced graphing and alerting features you need for production-ready infrastructure

Datadog is built to monitor modern infrastructure and offers all these essential functionalities:

* Autodiscovery: check
* Install the Datadog Agent : kubectl create -f /path/to/the/manifest/.yaml
* Configure the Agent

instances:

host: localhost

port: 4194

method: http

* Check that the Agent is running:

kubectl get daemonset