



kubernetes

CS777 Big Data Analytics

Aditya Maheshwari



Fall 2023

WHAT & WHY?

Kubernetes (K8s): Open-source container management by Google in 2014.

Function: Manages containerized apps efficiently.

CNCF Donation: Google donated it to CNCF in 2015.

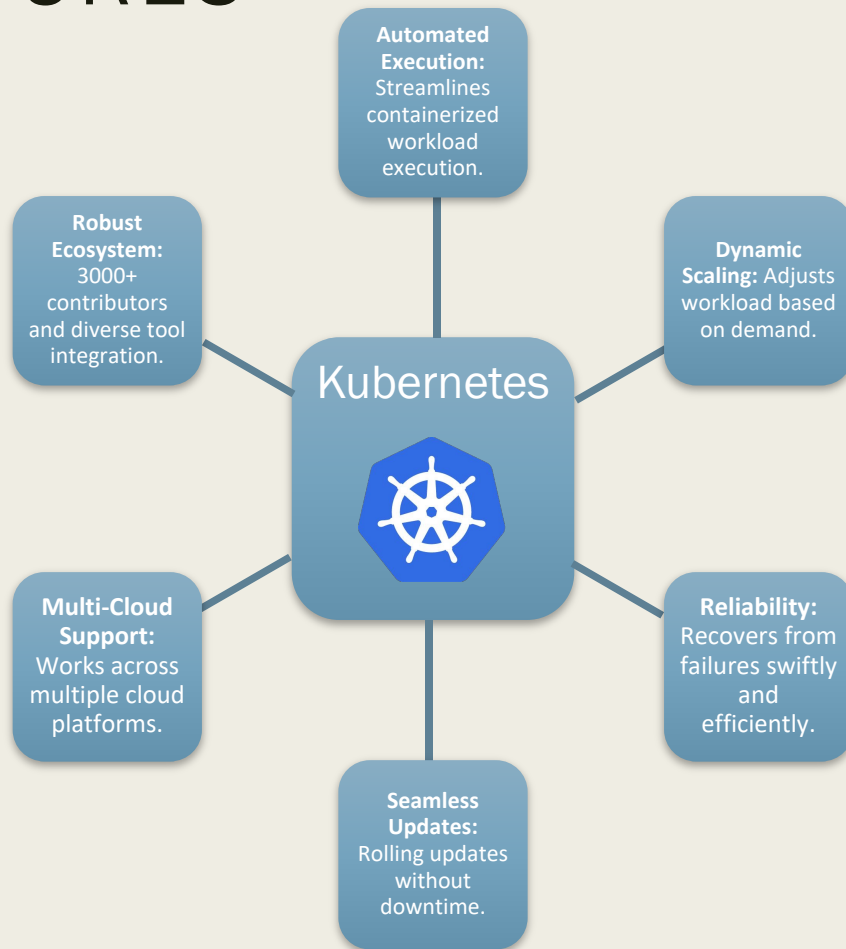
Cloud Usage: Compatible with AWS, Google Cloud, and private clouds.

Containerization Advantages: Efficient, consistent, and portable.

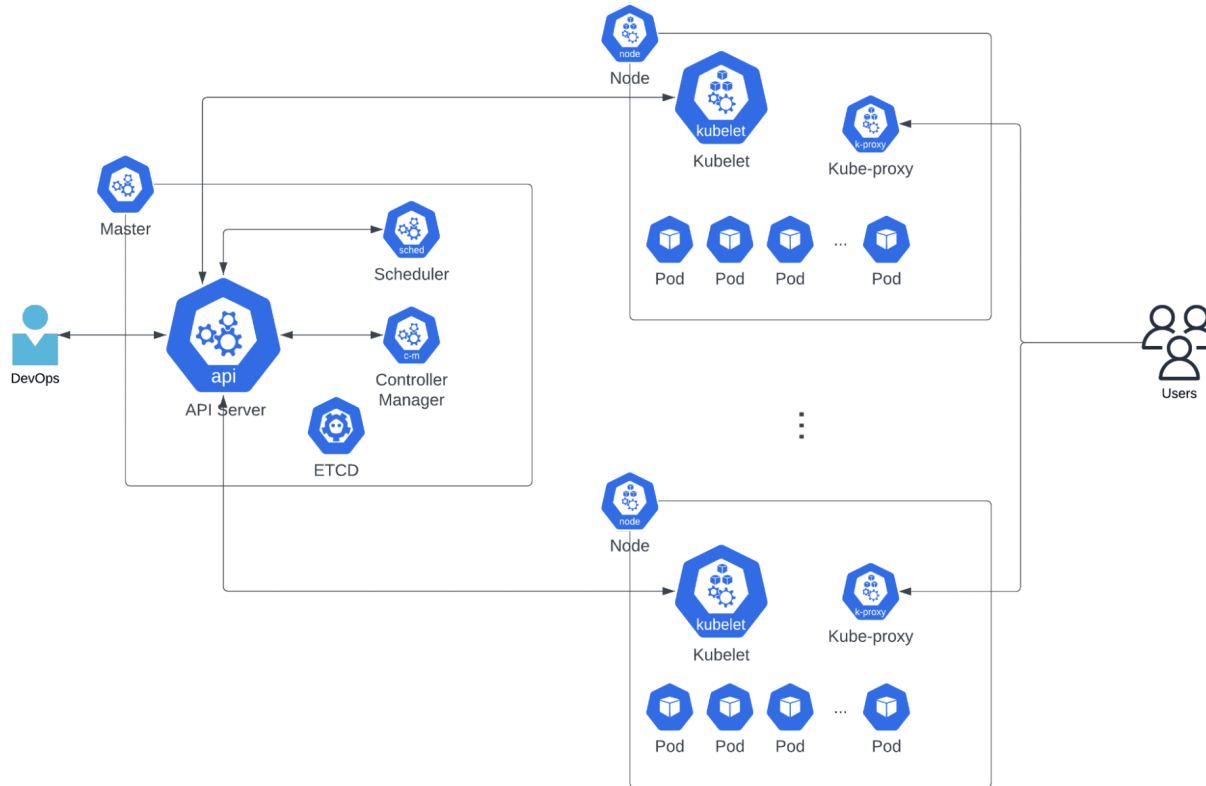
Complexity Challenge: Containerization complexity addressed by Kubernetes.

Facilitating Features: Streamlines management and deployment.

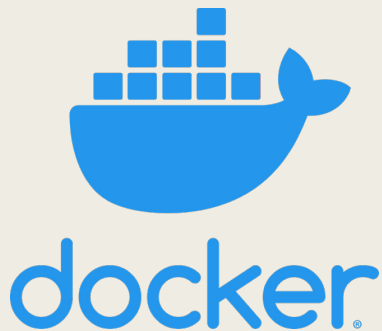
KEY FEATURES



ARCHITECTURE



SOFTWARES & INSTALLATIONS



Install Docker

<https://docs.docker.com>



>> brew install kubectl

>> kubectl version --client



>> brew install minikube

>> minikube start

CREATION OF KUBERNETES CLUSTER

```
1 Terminal commands to create a minikube cluster
2
3 minikube status
4 minikube start --driver=docker
5 minikube ip
6 minikube ssh
7 docker ps
8 kubectl cluster-info
9 kubectl get nodes
10 kubectl get pods
11 kubectl get namespaces
12 kubectl get pods --namespace = kube-system
13 kubectl run nginx --image = nginx
14 kubectl describe pod nginx
15 minikube ssh
16 kubectl get pods -o wide
17 alias k='kubectl'
18 k create deployments
19 k get pods
20 k describe deployment nginx-deployment
21 k scale deployment nginx-deployment --replicas=5
22 k get deployments
23 k expose deployment nginx-deployment --port = 8080 --target-port = 80
24 k get services
25 k describe service nginx-deployment
```

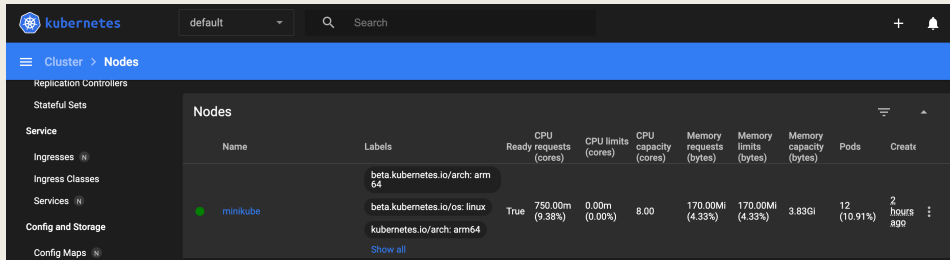
Creating a cluster using terminal commands (manually)

```
kubernetes-dashboard.yaml x
1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4   name: nginx-deployment
5 spec:
6   replicas: 3
7   selector:
8     matchLabels:
9       app: nginx
10  template:
11    metadata:
12      labels:
13        app: nginx
14    spec:
15      containers:
16        - name: nginx
17          image: nginx
18          ports:
19            - containerPort: 80
20  ---
21 apiVersion: v1
22 kind: Service
23 metadata:
24   name: nginx-service
25 spec:
26   selector:
27     app: nginx
28   ports:
29     - protocol: TCP
30       port: 8080
31       targetPort: 80
32   type: NodePort
```

Creating a cluster using .yaml file

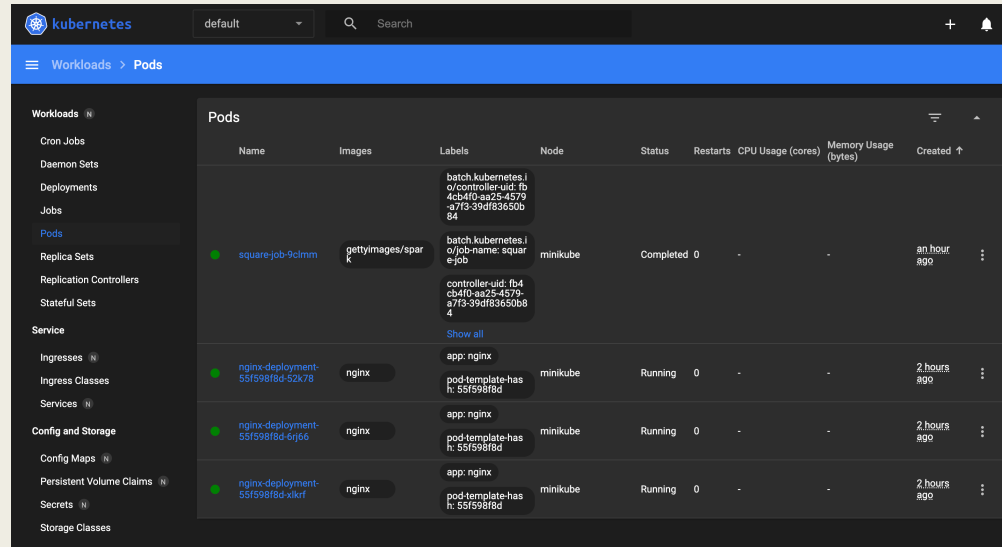


KUBERNETES DASHBOARD



This screenshot shows the 'Nodes' page in the Kubernetes Dashboard. The left sidebar contains navigation links for Cluster, Nodes, Replication Controllers, Stateful Sets, Service, Ingresses, Ingress Classes, Services, Config and Storage, and Config Maps. The main content area displays a table of nodes. A tooltip is visible over the 'Labels' column for the 'minikube' node, showing its labels: 'beta.kubernetes.io/arch: arm64', 'beta.kubernetes.io/os: linux', and 'kubernetes.io/arch: arm64'. The table columns include Name, Labels, CPU Ready requests (cores), CPU limits (cores), CPU capacity (cores), Memory requests (bytes), Memory limits (bytes), Memory capacity (bytes), Pods, and Create.

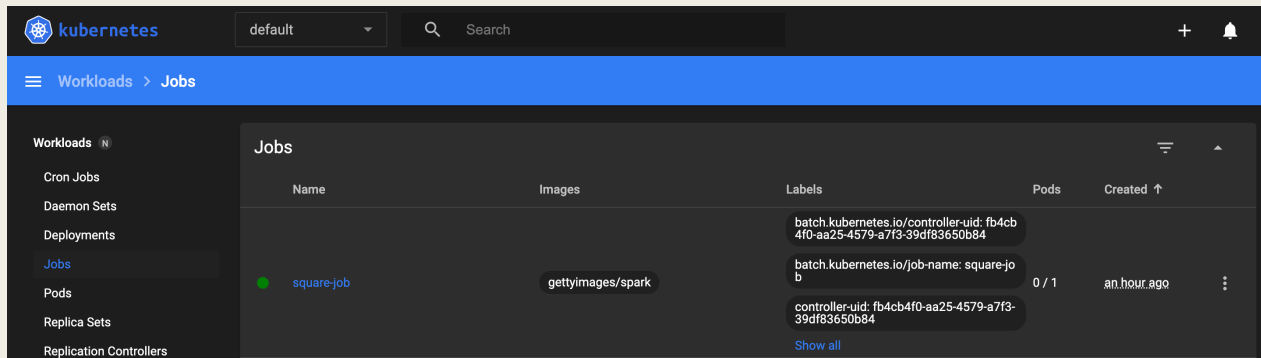
Name	Labels	CPU Ready requests (cores)	CPU limits (cores)	CPU capacity (cores)	Memory requests (bytes)	Memory limits (bytes)	Memory capacity (bytes)	Pods	Create
minikube	beta.kubernetes.io/arch: arm64 beta.kubernetes.io/os: linux kubernetes.io/arch: arm64	True	750.00m (9.38%)	0.00m (0.00%)	8.00	170.00Mi (4.33%)	170.00Mi (4.33%)	3.83Gi (10.91%)	12 2 hours ago



This screenshot shows the 'Pods' page in the Kubernetes Dashboard. The left sidebar contains navigation links for Workloads, Cron Jobs, Daemon Sets, Deployments, Jobs, Pods, Replica Sets, Replication Controllers, Stateful Sets, Service, Ingresses, Ingress Classes, Services, Config and Storage, Config Maps, Persistent Volume Claims, Secrets, and Storage Classes. The main content area displays a table of pods. A tooltip is visible over the 'Labels' column for the 'square-job-9c1mm' pod, showing its labels: 'batch.kubernetes.io/controller-uid: fb4cb4f0-a7f3-39df83650b84' and 'batch.kubernetes.io/job-name: square-job'. The table columns include Name, Images, Labels, Node, Status, Restarts, CPU Usage (cores), Memory Usage (bytes), and Created.

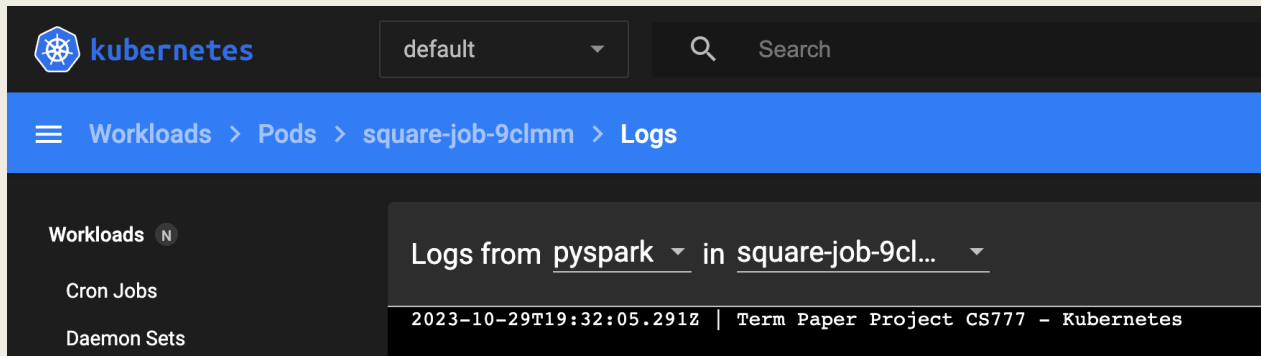
Name	Images	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created
square-job-9c1mm	gettyimages/spar...	batch.kubernetes.io/controller-uid: fb4cb4f0-a7f3-39df83650b84 batch.kubernetes.io/job-name: square-job	minikube	Completed	0	-	-	an hour ago
nginx-deployment-55f598f8d-52k78	nginx	app: nginx pod-template-hash: 55f598f8d	minikube	Running	0	-	-	2 hours ago
nginx-deployment-55f598f8d-6rj66	nginx	app: nginx pod-template-hash: 55f598f8d	minikube	Running	0	-	-	2 hours ago
nginx-deployment-55f598f8d-x1cxf	nginx	app: nginx pod-template-hash: 55f598f8d	minikube	Running	0	-	-	2 hours ago

KUBERNETES JOB OUTPUT



The screenshot shows the Kubernetes dashboard's 'Jobs' page. The left sidebar lists various workload types, with 'Jobs' selected. The main area displays a table of jobs. One job, 'square-job', is shown with a green status icon. The table columns are Name, Images, Labels, Pods, and Created. The 'square-job' is using the 'gettyimages/spark' image. The labels include 'batch.kubernetes.io/controller-uid: fb4cb4f0-aa25-4579-a7f3-39df83650b84', 'batch.kubernetes.io/job-name: square-job', and 'controller-uid: fb4cb4f0-aa25-4579-a7f3-39df83650b84'. It has 0 pods out of 1 requested and was created 'an hour ago'. A 'Show all' link is at the bottom right of the table.

Name	Images	Labels	Pods	Created ↑
square-job	gettyimages/spark	batch.kubernetes.io/controller-uid: fb4cb4f0-aa25-4579-a7f3-39df83650b84 batch.kubernetes.io/job-name: square-job controller-uid: fb4cb4f0-aa25-4579-a7f3-39df83650b84	0 / 1	an hour ago



The screenshot shows the Kubernetes dashboard's 'Logs' page for a specific job. The breadcrumb trail is 'Workloads > Pods > square-job-9clmm > Logs'. The left sidebar shows 'Workloads' selected. The main area has a dropdown menu set to 'Logs from pyspark' in 'square-job-9clmm'. Below this, a log entry is displayed: '2023-10-29T19:32:05.291Z | Term Paper Project CS777 - Kubernetes'.

Logs from pyspark in square-job-9clmm

2023-10-29T19:32:05.291Z | Term Paper Project CS777 - Kubernetes

USE CASES

- 1.Distributed Processing:** Manages Hadoop and Spark frameworks.
- 2.Data Storage Management:** Efficient handling of data storage systems.
- 3.Distributed Systems Management:** Deploys MongoDB, Kafka, and more.

ADVANTAGES	DISADVANTAGES
Scalability	Complex configuration and integration
Portability	Steep learning curve
Stability	

CONCLUSION

- **Kubernetes for Big Data:** Empowers efficient Big Data management.
- **Feature-Rich Support:** Scalability, stability, and portability for Big Data.
- **Ongoing Advancements:** Continuous improvement via contributions.
- **Learning Curve:** Demands familiarity but offers substantial benefits.

THANK YOU