


# Assignment 5: Container Orchestration

## Initial setup:



Experiments ▾Storage ▾**News!**

Docs ▾bharu21 ▾

The all-day electrical shutdown of the CloudLab Utah and Apt clusters has been postponed, and will NOT happen the week of February 24-28. Our datacenter will give us at least 10 business days advance notice, we will keep you updated.

Current Usage: 0 Node Hours, Prev Week: 94.46, Prev Month: 290.57 (30 day rank: 613 of 1556 users)

1. Select a Profile

2. Parameterize

3. Finalize

4. Schedule

**Selected Profile:** small-lan-40

Variable number of nodes in a lan. You have the option of picking from one of several standard images we provide, or just use the default (typically a recent version of Ubuntu). You may also optionally pick the specific hardware type for all the nodes in the lan.


Copy Profile

Show Profile

Change Profile

Previous

Next



Experiments ▾Storage ▾**News!**

Docs ▾bharu21 ▾

The all-day electrical shutdown of the CloudLab Utah and Apt clusters has been postponed, and will NOT happen the week of February 24-28. Our datacenter will give us at least 10 business days advance notice, we will keep you updated.

Current Usage: 0 Node Hours, Prev Week: 94.46, Prev Month: 290.57 (30 day rank: 613 of 1556 users)

1. Select a Profile

2. Parameterize

3. Finalize

4. Schedule

**Selected Profile:** small-lan-40

Save/Load Parameters ▾Resource Availability

+ Show All Parameter Help

This profile is parameterized; please make your selections below, and then click **Next**.

Number of Nodes ⓘ2

Select OS image ⓘUBUNTU 22.04 ▾

Optional physical node type ⓘAny ▾

Use XEN VMs ⓘ☐

Start X11 VNC on your nodes ⓘ☐

▶ Advanced

Previous

Next

Current Usage: 0 Node Hours, Prev Week: 93.75, Prev Month: 290.57 (30 day rank: 612 of 1554 users)

1. Select a Profile 2. Parameterize 3. Finalize 4. Schedule

Selected Profile: small-ian:40

Please review the selections below and then click Next.

Name: Kubernetes

Project: Assignment

Cluster: Please Select

Advanced Options

Check Resource Availability

Click Node to Select

node1

node0

Previous Next

Powered by emulab Question or comment? Join the Help Forum Supported by NSF © 2025 The University of Utah

Experiments Storage News! Docs bharu21

The all-day electrical shutdown of the CloudLab Utah and Apt clusters has been postponed, and will NOT happen the week of February 24-28. Our datacenter will give us at least 10 business days advance notice, we will keep you updated.

Current Usage: 0 Node Hours, Prev Week: 93.75, Prev Month: 290.57 (30 day rank: 612 of 1554 users)

Please wait while we get your experiment ready

Name:	Kubernetes
State:	created
Profile:	small-ian
Creator:	bharu21
Project:	Assignment
Started:	Mar 1, 2025 8:52 PM
Expires:	Mar 2, 2025 12:52 PM (in 16 hours)
Portal Log	Request Help

Copy Extend Terminate

Not sure how to proceed or have further questions? Join the users group and ask a question. Be sure to include any error messages in your question and the URL of your experiment status page.

## Prepare the VM:

Before setting up Kubernetes, the virtual machines (VMs) need to be **updated** and **configured**.

Each node (master and worker) should have:

- The **container runtime (containerd)** installed, which allows running containers.
- **Essential Kubernetes components (kubeadm, kubelet, and kubectl)** installed to create and manage the cluster.

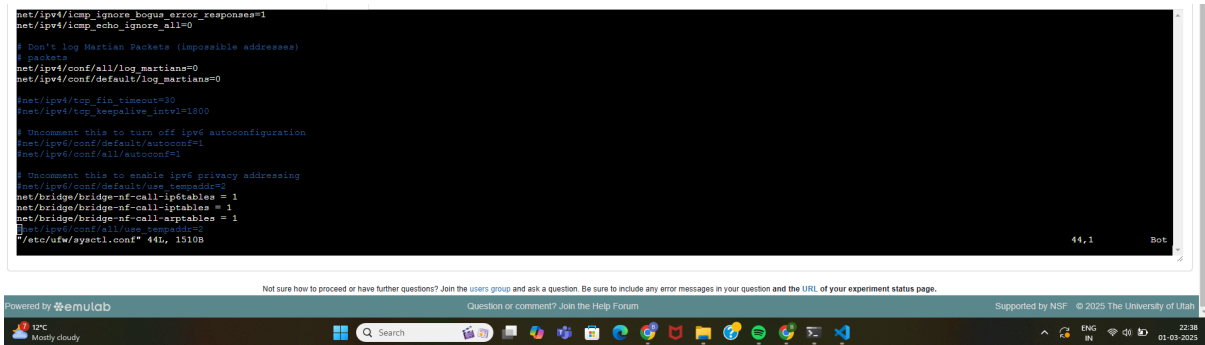
```
root@node0:/users/bharu21# vi /etc/fstab
```

```

# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
#<file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/sda3 during curtin installation
# /dev/disk/by-uuid/971337a8-0a6e-48bd-9cfc-6cb9719427c1 / ext3 defaults 0 1
# /boot/efi was on /dev/sda3 during curtin installation
# /dev/disk/by-partlabel/EFI\\x20System /boot/efi vfat defaults 0 1
# the following swap devices added by /etc/systemd/system-generator/systemd-fstab-generator
UUID=0b8a8a0b-9ec3-44f1-8163-f2aef4810d4b swap swap defaults 0 0

# /etc/fstab" 13D, 728B

```



## Create a cluster:

```
root@node0:/users/bharu21# sudo rm -rf /etc/apt/keyrings/kubernetes-archive-keyring.gpg
root@node0:/users/bharu21# sudo apt update && sudo apt upgrade -y
Hit:1 http://repos.emulab.net/emulab/ubuntu jammy InRelease
Hit:2 http://repos.emulab.net/grub-backports/ubuntu jammy InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu jammy InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:5 http://us.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:6 http://us.archive.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
1 package can be upgraded. Run 'apt list --upgradable' to see it.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
Get more security updates through Ubuntu Pro with 'esm-apps' enabled:

root@node0:/users/bharu21# sudo apt install -y apt-transport-https ca-certificates curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20240203-22.04.1).
curl is already the newest version (7.81.0-ubuntu1.20).
apt-transport-https is already the newest version (2.4.13).
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
root@node0:/users/bharu21# sudo mkdir -p /etc/apt/keyrings
root@node0:/users/bharu21# curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.28/deb/Release.key | sudo tee /etc/apt/keyrings/kubernetes-archive-keyring.gpg > /dev/null
root@node0:/users/bharu21# echo "deb [signed-by=/etc/apt/keyrings/kubernetes-archive-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.28/deb/ /" | sudo tee /etc/apt/sources.list.d/kuberne
deb [signed-by=/etc/apt/keyrings/kubernetes-archive-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.28/deb/ /
root@node0:/users/bharu21# sudo apt update

Processing triggers for man-db (2.10.2-1) ...
root@node0:/users/bharu21# kubeadm init --pod-network-cidr=192.168.0.0/16
I0302 13:11:11.304531 35922 version.go:256] remote version is much newer: v1.32.2; falling back to: stable-1.28
[init] Using Kubernetes version: v1.28.15
[preflight] Running pre-flight checks
[WARNING] swap is enabled; production deployments should disable swap unless testing the NodeSwap feature gate of the kubelet
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action in beforehand using 'kubeadm config images pull'
```

Not sure how to proceed or have further questions? Join the [users group](#) and ask a question. Be sure to include any error messages in your question and the URL of your experiment status page.

```
root@node0:/users/bharu21# sudo systemctl start docker
```

```
root@node1:/users/bharu21# sudo kubeadm init --pod-network-cidr=192.168.0.0/16
I0302 17:52:48.392512 9480 version.go:256] remote version is much newer: v1.32.2; falling back to: stable-1.28
[init] Using Kubernetes version: v1.28.15
[preflight] Running pre-flight checks
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action in beforehand using 'kubeadm config images pull'

Alternatively, if you are the root user, you can run:
export KUBECONFIG=/etc/kubernetes/admin.conf

You should now deploy a pod network to the cluster.
Run 'kubectl apply -f [podnetwork].yaml' with one of the options listed at:
https://kubernetes.io/docs/concepts/cluster-administration/addons/

Then you can join any number of worker nodes by running the following on each as root:

kubeadm join 128.110.217.154:6443 --token d24zxo.hprnqsnhy8bqaj \
--discovery-token-ca-cert-hash sha256:3badaa9723473e51ce981b6020e53bd74f895bc50905c3d94f88f5f96da2e5f5
root@master-node:~#
```

## Starting the cluster:

The master node generates a **join** command, which the worker node executes to become part of the cluster.

Once joined, the worker node must appear in **kubectl get nodes** as **Ready**.

Kubernetes requires a Container Network Interface (CNI) plugin for pod-to-pod communication.

Options include Calico (recommended) or Flannel.

The CNI plugin allows worker nodes to send and receive traffic between pods.

```
kubeadm join 128.110.217.154:6443 --token d24zxo.hprnhqsnhy8bgaqj \
--discovery-token-ca-cert-hash sha256:3badaa9723473e51ce981b6020e53bd74f895bc50905c3d94f88f5f96da2e5f5
root@master-node:~# mkdir -p $HOME/.kube
root@master-node:~# sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
root@master-node:~# sudo chown $(id -u):$(id -g) $HOME/.kube/config
root@master-node:~#
```

```
root@node0:~# kubectl get nodes
NAME          STATUS    ROLES    AGE   VERSION
master-node   Ready     control-plane  72m   v1.28.15
root@node0:~# kubectl get nodes --all-namespaces
```

```
root@node1:~# kubectl get nodes
NAME          STATUS    ROLES    AGE   VERSION
master-node   NotReady  control-plane  106m   v1.28.15
root@node0:~# kubectl get nodes
NAME          STATUS    ROLES    AGE   VERSION
master-node   Ready     control-plane  107m   v1.28.15
worker-node   NotReady  <none>     25s   v1.28.15
root@node0:~#
```

## Deploy test pod:

A simple test pod is created to verify if the worker node can schedule and run workloads.

If the pod starts on the worker node, it confirms that the cluster is functioning correctly.

```
kube-scheduler-master-node 0/1 CrashLoopBackOff 56 (33m ago) 106m
root@node0:~# kubectl run test-pod --image=busybox --restart=Never -- sleep 3600
pod/test-pod created
root@node0:~#
```

```
root@node0:~# kubectl get pods -o wide
NAME          READY   STATUS    RESTARTS   AGE   IP          NODE          NOMINATED NODE   READINESS GATES
test-pod      0/1     Pending   0           2m18s <none>      <none>        <none>           <none>
root@node0:~#
```

## 2. Deployment.yml of NGINX

A Deployment is created to manage two replicas of Nginx.

The Deployment ensures high availability, meaning if a pod fails, Kubernetes automatically replaces it.

The **deployment.yml** file defines:

- The number of replicas (2).
- The Nginx image to be used.
- The port (80) on which Nginx runs.

### **Deployment.yml:**

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:latest
          ports:
            - containerPort: 80
```

### **Service.yml**

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
spec:
  selector:
    app: nginx
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
  type: LoadBalancer
```

```
root@node0:~# nano deployment.yml
root@node0:~# nano deployment.yml
root@node0:~# kubectl apply -f deployment.yml
deployment.apps/nginx-deployment created

root@node0:~/assignment-5# nano service.yml
root@node0:~/assignment-5# kubectl apply -f service.yml
service/nginx-service created
root@node0:~/assignment-5#
```

```
service/nginx-service created
root@node0:~/assignment-5# kubectl get svc
NAME          TYPE          CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
kubernetes    ClusterIP     10.96.0.1       <none>           443/TCP          152m
nginx-service  LoadBalancer 10.99.116.65    <pending>        80:31536/TCP     32s
root@node0:~/assignment-5#
```