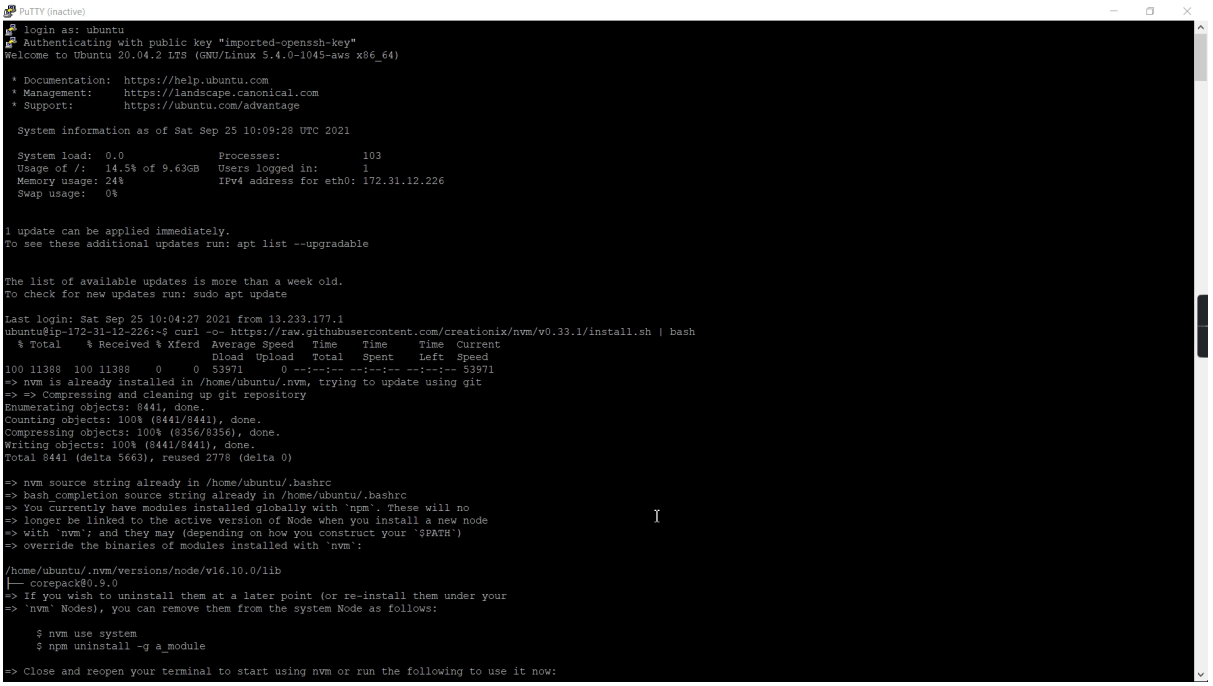
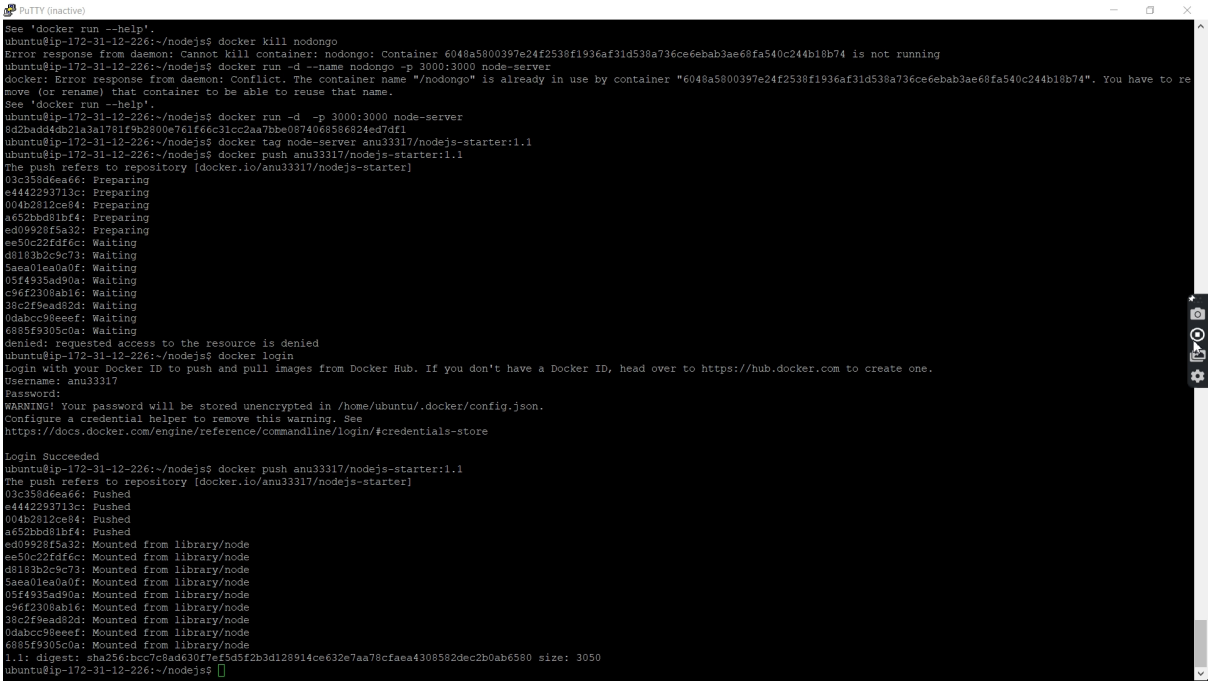
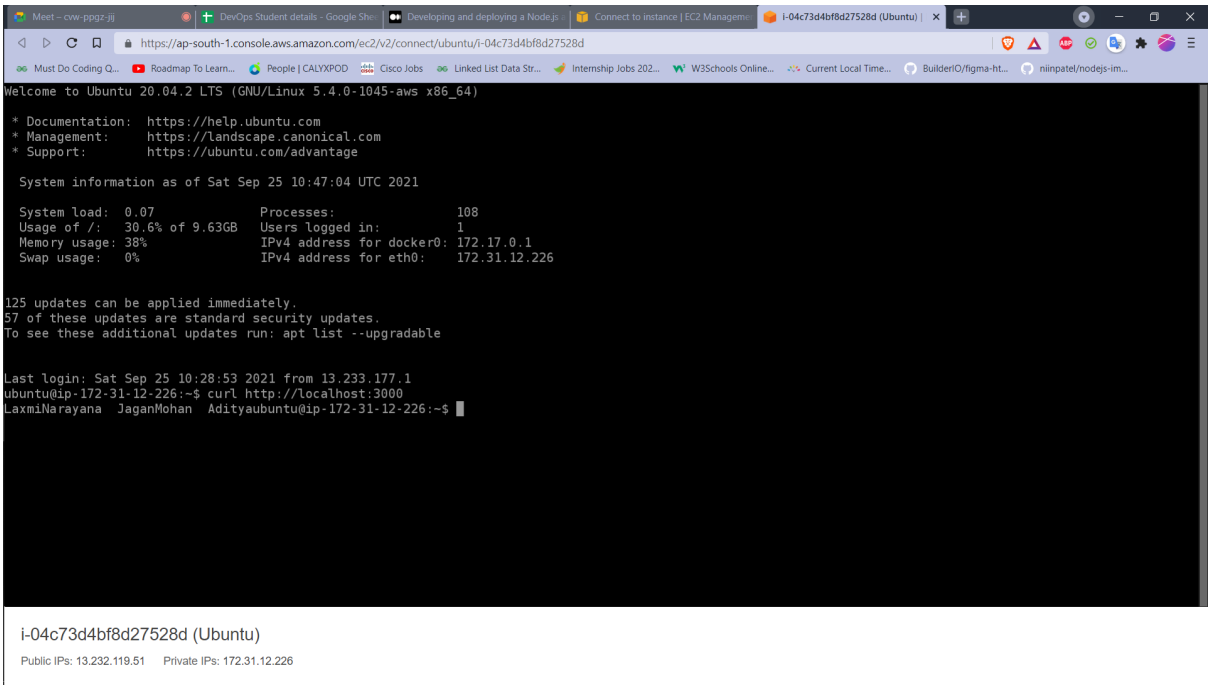
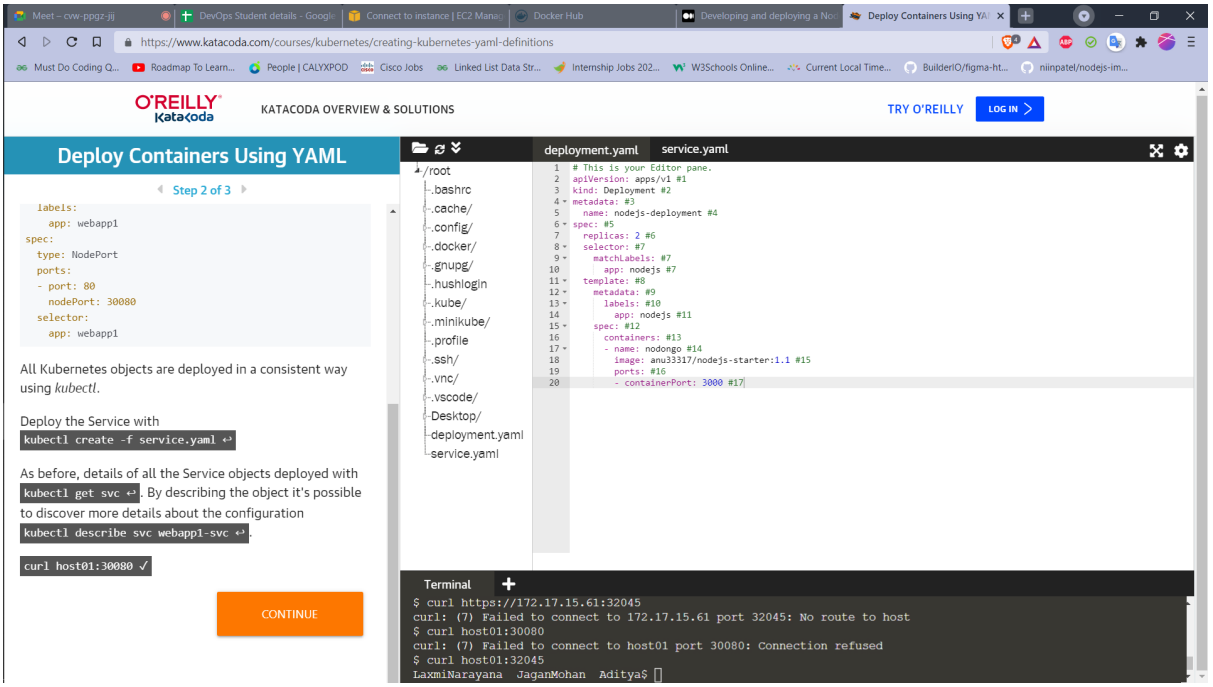
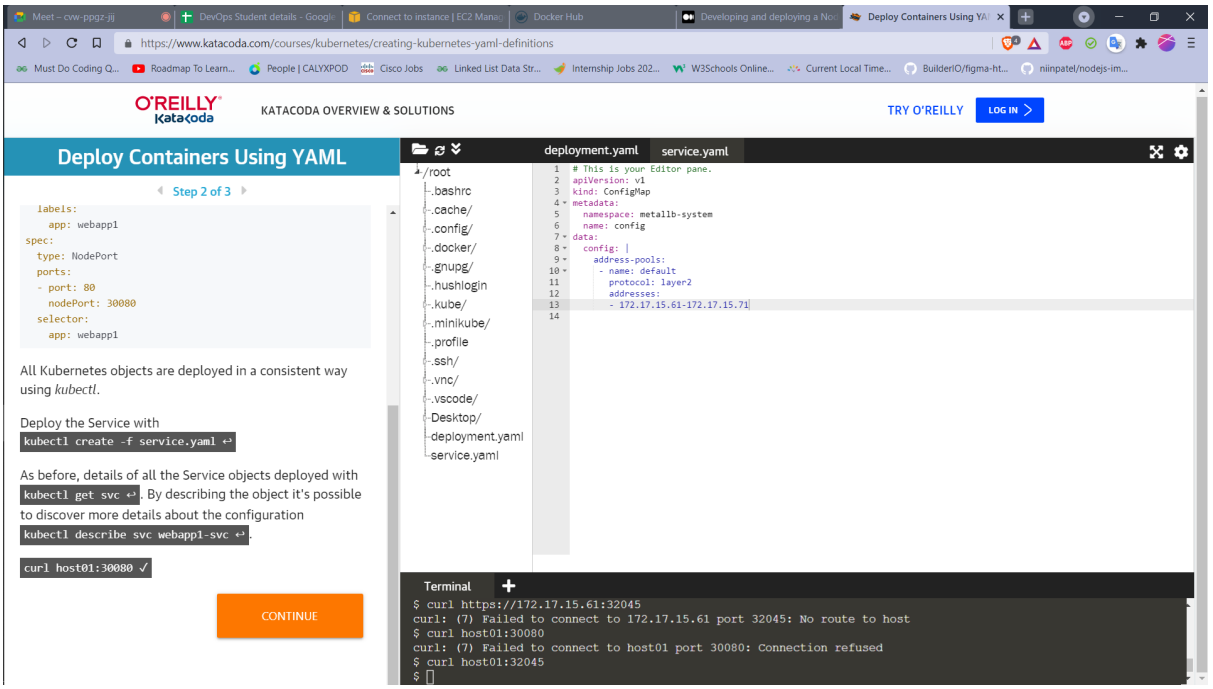
Developing and deploying a Node.js app from Docker to Kubernetes.

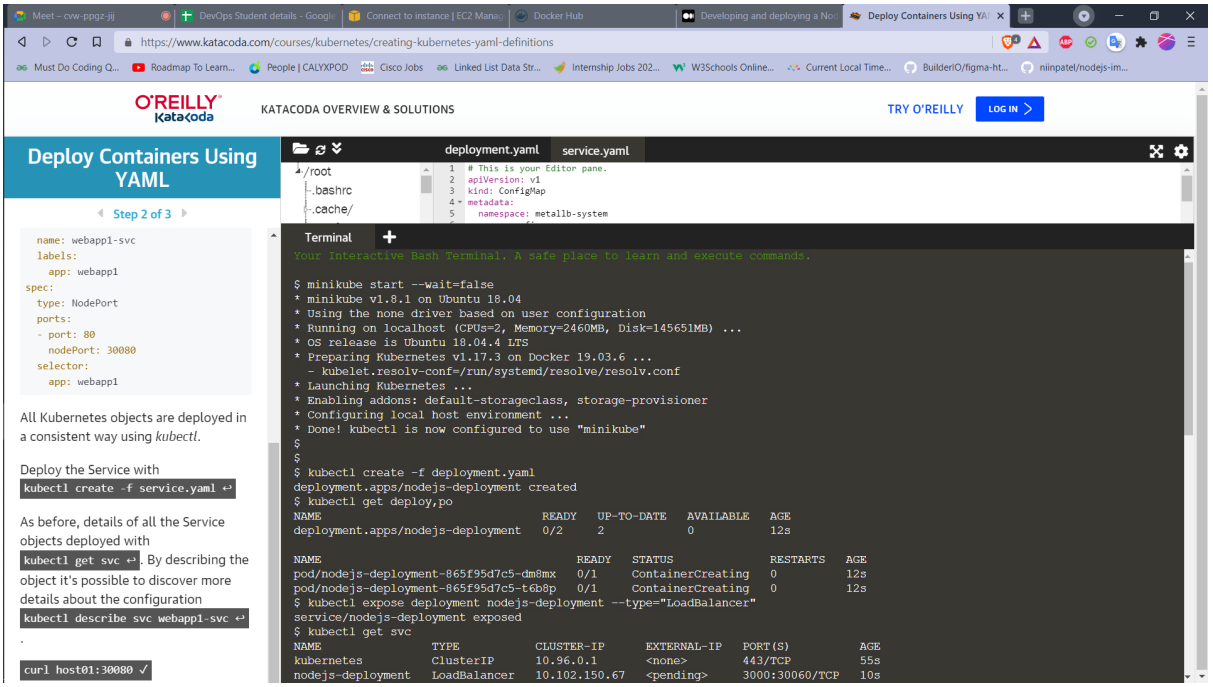


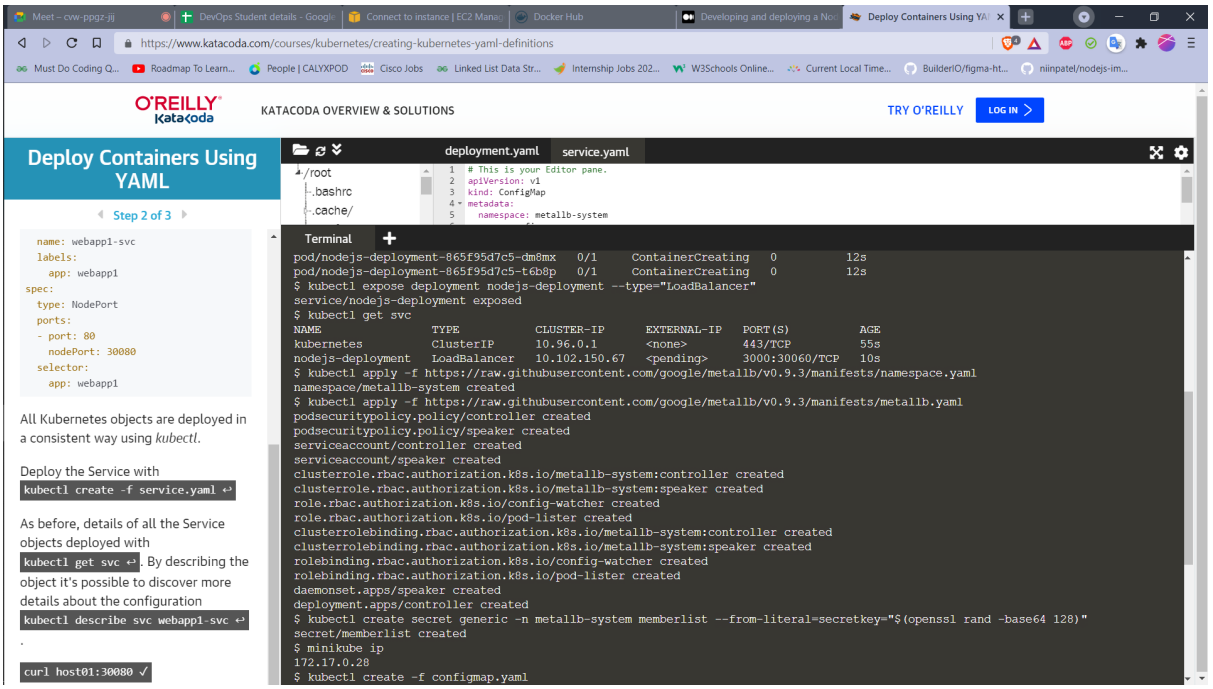

PuTTY (inactive)
See 'docker run --help'.
ubuntu@ip-172-31-12-226:~/nodejs$ docker kill nodongo
Error response from daemon: Cannot kill container: nodongo: Container 6048a5800397e24f2538f1936af31d538a736ce6ebab3ae68fa540c244b18b74 is not running
ubuntu@ip-172-31-12-226:~/nodejs$ docker run -d --name nodongo -p 3000:3000 node-server
docker: Error response from daemon: Conflict. The container name "/nodongo" is already in use by container "6048a5800397e24f2538f1936af31d538a736ce6ebab3ae68fa540c244b18b74". You have to re
move (or rename) that container to be able to reuse that name.
See 'docker run --help'.
ubuntu@ip-172-31-12-226:~/nodejs$ docker run -d -p 3000:3000 node-server
8d2badd4db21a3a1781f9b2800e761f66c31cc2aa7bbe0874068586824ed7df1
ubuntu@ip-172-31-12-226:~/nodejs$ docker tag node-server anu33317/nodejs-starter:1.1
ubuntu@ip-172-31-12-226:~/nodejs$ docker push anu33317/nodejs-starter:1.1
The push refers to repository [docker.io/anu33317/nodejs-starter]
03c358d6ea66: Preparing
e4442293713c: Preparing
004b2812ce84: Preparing
a652bbd81bf4: Preparing
ed09928f5a32: Preparing
ee50c22fdf6c: Waiting
d8183b2c9c73: Waiting
5aea01ea0a0f: Waiting
05f4935ad90a: Waiting
c96f2308ab16: Waiting
38c2f9ead82d: Waiting
0dabcc98eeef: Waiting
6885f9305c0a: Waiting
denied: requested access to the resource is denied
ubuntu@ip-172-31-12-226:~/nodejs$ docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.
Username: anu33317
Password:
WARNING! Your password will be stored unencrypted in /home/ubuntu/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store
Login Succeeded
ubuntu@ip-172-31-12-226:~/nodejs$ docker push anu33317/nodejs-starter:1.1
The push refers to repository [docker.io/anu33317/nodejs-starter]
03c358d6ea66: Pushed
e4442293713c: Pushed
004b2812ce84: Pushed
a652bbd81bf4: Pushed
ed09928f5a32: Mounted from library/node
ee50c22fdf6c: Mounted from library/node
d8183b2c9c73: Mounted from library/node
5aea01ea0a0f: Mounted from library/node
05f4935ad90a: Mounted from library/node
c96f2308ab16: Mounted from library/node
38c2f9ead82d: Mounted from library/node
0dabcc98eeef: Mounted from library/node
6885f9305c0a: Mounted from library/node
1.1: digest: sha256:bcc7c8ad630f7ef5d5f2b3d128914ce632e7aa78cfaea4308582dec2b0ab6580 size: 3050
ubuntu@ip-172-31-12-226:~/nodejs$

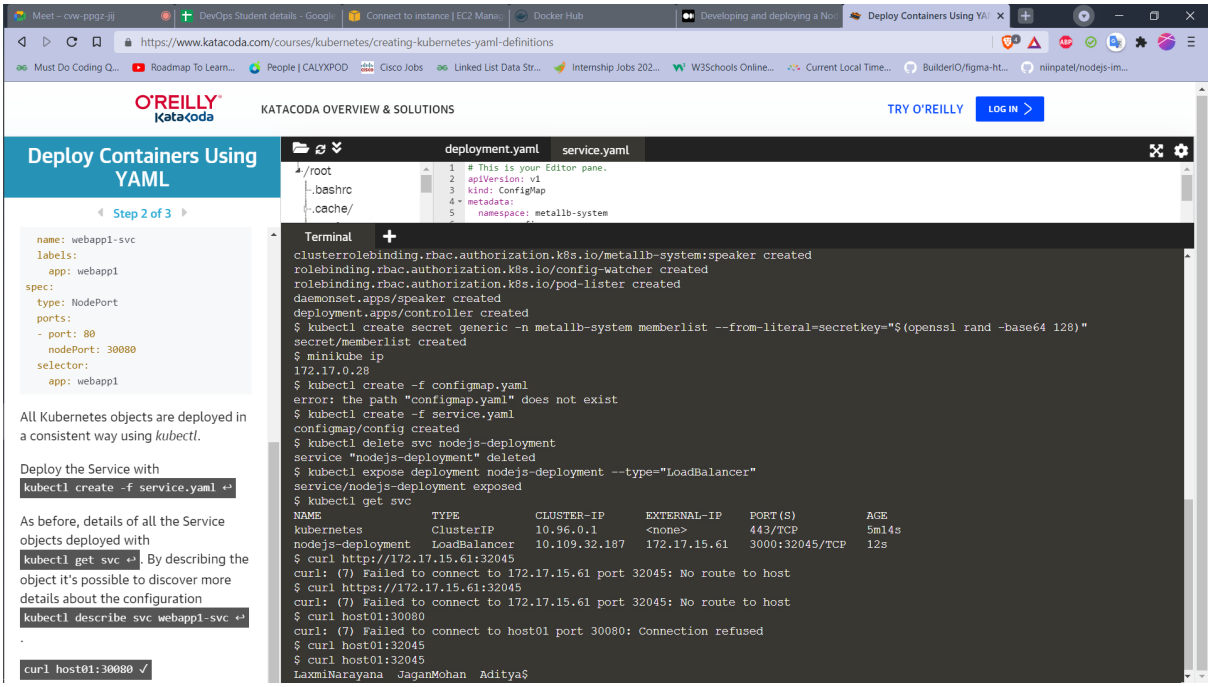

Meet – cvw-ppgz-jjj
DevOps Student details - Google She
Developing and deploying a Nodejs
Connect to instance | EC2 Manageme
i-04c73d4bf8d27528d (Ubuntu)
https://ap-south-1.console.aws.amazon.com/ec2/v2/connect/ubuntu/i-04c73d4bf8d27528d
Must Do Coding Q...
Roadmap To Learn...
People | CALYXPOD
Cisco Jobs
Linked List Data Str...
Internship Jobs 202...
W3Schools Online...
Current Local Time...
BuilderIO/figma-ht...
niinpatel/nodejs-im...
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-1045-aws x86_64)
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage
System information as of Sat Sep 25 10:47:04 UTC 2021
System load: 0.07 Processes: 108
Usage of /: 30.6% of 9.63GB Users logged in: 1
Memory usage: 38% IPv4 address for docker0: 172.17.0.1
Swap usage: 0% IPv4 address for eth0: 172.31.12.226
125 updates can be applied immediately.
57 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable
Last login: Sat Sep 25 10:28:53 2021 from 13.233.177.1
ubuntu@ip-172-31-12-226:~$ curl http://localhost:3000
LaxmiNarayana JaganMohan Adityaubuntu@ip-172-31-12-226:~$
i-04c73d4bf8d27528d (Ubuntu)
Public IPs: 13.232.119.51 Private IPs: 172.31.12.226


https://www.katacoda.com/courses/kubernetes/creating-kubernetes-yaml-definitions
O'REILLY Katacoda
KATACODA OVERVIEW & SOLUTIONS
TRY O'REILLY
LOG IN
Deploy Containers Using YAML
Step 2 of 3
labels:
app: webapp1
spec:
type: NodePort
ports:
- port: 80
nodePort: 30080
selector:
app: webapp1
All Kubernetes objects are deployed in a consistent way using kubectl.
Deploy the Service with
kubectl create -f service.yaml
As before, details of all the Service objects deployed with kubectl get svc. By describing the object it's possible to discover more details about the configuration
kubectl describe svc webapp1-svc.
curl host01:30080
CONTINUE
/root
.bashrc
.cache/
.config/
.docker/
.gnupg/
.hushlogin
.kube/
.minikube/
.profile
.ssh/
.vnc/
.vscode/
Desktop/
deployment.yaml
service.yaml
deployment.yaml
service.yaml
This is your Editor pane.
apiVersion: apps/v1 #1
kind: Deployment #2
metadata: #3
name: nodejs-deployment #4
spec: #5
replicas: 2 #6
selector: #7
matchLabels: #7
app: nodejs #7
template: #8
metadata: #9
labels: #10
app: nodejs #11
spec: #12
containers: #13
- name: nodongo #14
image: anu33317/nodejs-starter:1.1 #15
ports: #16
- containerPort: 3000 #17
Terminal
$ curl https://172.17.15.61:32045
curl: (7) Failed to connect to 172.17.15.61 port 32045: No route to host
$ curl host01:30080
curl: (7) Failed to connect to host01 port 30080: Connection refused
$ curl host01:32045
LaxmiNarayana JaganMohan Aditya$


https://www.katacoda.com/courses/kubernetes/creating-kubernetes-yaml-definitions
O'REILLY Katacoda
KATACODA OVERVIEW & SOLUTIONS
TRY O'REILLY
LOG IN
Deploy Containers Using YAML
Step 2 of 3
labels:
app: webapp1
spec:
type: NodePort
ports:
- port: 80
nodePort: 30080
selector:
app: webapp1
All Kubernetes objects are deployed in a consistent way using kubectl.
Deploy the Service with
kubectl create -f service.yaml
As before, details of all the Service objects deployed with kubectl get svc. By describing the object it's possible to discover more details about the configuration
kubectl describe svc webapp1-svc.
curl host01:30080
CONTINUE
/root
.bashrc
.cache/
.config/
.docker/
.gnupg/
.hushlogin
.kube/
.minikube/
.profile
.ssh/
.vnc/
.vscode/
Desktop/
deployment.yaml
service.yaml
deployment.yaml
service.yaml
This is your Editor pane.
apiVersion: v1
kind: ConfigMap
metadata:
namespace: metallb-system
name: config
data:
config: |
address-pools:
- name: default
protocol: layer2
addresses:
- 172.17.15.61-172.17.15.71
Terminal
$ curl https://172.17.15.61:32045
curl: (7) Failed to connect to 172.17.15.61 port 32045: No route to host
$ curl host01:30080
curl: (7) Failed to connect to host01 port 30080: Connection refused
$ curl host01:32045
$


https://www.katacoda.com/courses/kubernetes/creating-kubernetes-yaml-definitions
O'REILLY Katacoda
KATACODA OVERVIEW & SOLUTIONS
TRY O'REILLY
LOG IN
Deploy Containers Using YAML
Step 2 of 3
name: webapp1-svc
labels:
app: webapp1
spec:
type: NodePort
ports:
- port: 80
nodePort: 30080
selector:
app: webapp1
All Kubernetes objects are deployed in a consistent way using kubectl.
Deploy the Service with
kubectl create -f service.yaml
As before, details of all the Service objects deployed with
kubectl get svc
. By describing the object it's possible to discover more details about the configuration
kubectl describe svc webapp1-svc
.
curl host01:30080
deployment.yaml
service.yaml
/root
.bashrc
.cache/
This is your Editor pane.
apiVersion: v1
kind: ConfigMap
metadata:
namespace: metallb-system
Terminal
Your Interactive Bash Terminal. A safe place to learn and execute commands.
$ minikube start --wait=false
* minikube v1.8.1 on Ubuntu 18.04
* Using the none driver based on user configuration
* Running on localhost (CPUs=2, Memory=2460MB, Disk=145651MB) ...
* OS release is Ubuntu 18.04.4 LTS
* Preparing Kubernetes v1.17.3 on Docker 19.03.6 ...
- kubelet.resolv-conf=/run/systemd/resolve/resolv.conf
* Launching Kubernetes ...
* Enabling addons: default-storageclass, storage-provisioner
* Configuring local host environment ...
* Done! kubectl is now configured to use "minikube"
$
$
$ kubectl create -f deployment.yaml
deployment.apps/nodejs-deployment created
$ kubectl get deploy,po
NAME READY UP-TO-DATE AVAILABLE AGE
deployment.apps/nodejs-deployment 0/2 2 0 12s
NAME READY STATUS RESTARTS AGE
pod/nodejs-deployment-865f95d7c5-dm8mx 0/1 ContainerCreating 0 12s
pod/nodejs-deployment-865f95d7c5-t6b8p 0/1 ContainerCreating 0 12s
$ kubectl expose deployment nodejs-deployment --type="LoadBalancer"
service/nodejs-deployment exposed
$ kubectl get svc
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 55s
nodejs-deployment LoadBalancer 10.102.150.67 <pending> 3000:30060/TCP 10s


https://www.katacoda.com/courses/kubernetes/creating-kubernetes-yaml-definitions
O'REILLY Katacoda
KATACODA OVERVIEW & SOLUTIONS
TRY O'REILLY
LOG IN
Deploy Containers Using YAML
Step 2 of 3
name: webapp1-svc
labels:
app: webapp1
spec:
type: NodePort
ports:
- port: 80
nodePort: 30080
selector:
app: webapp1
All Kubernetes objects are deployed in a consistent way using kubectl.
Deploy the Service with
kubectl create -f service.yaml
As before, details of all the Service objects deployed with
kubectl get svc
. By describing the object it's possible to discover more details about the configuration
kubectl describe svc webapp1-svc
.
curl host01:30080
deployment.yaml
service.yaml
/root
.bashrc
.cache/
This is your Editor pane.
apiVersion: v1
kind: ConfigMap
metadata:
namespace: metallb-system
Terminal
pod/nodejs-deployment-865f95d7c5-dm8mx 0/1 ContainerCreating 0 12s
pod/nodejs-deployment-865f95d7c5-t6b8p 0/1 ContainerCreating 0 12s
$ kubectl expose deployment nodejs-deployment --type="LoadBalancer"
service/nodejs-deployment exposed
$ kubectl get svc
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 55s
nodejs-deployment LoadBalancer 10.102.150.67 <pending> 3000:30060/TCP 10s
$ kubectl apply -f https://raw.githubusercontent.com/google/metallb/v0.9.3/manifests/namespace.yaml
namespace/metallb-system created
$ kubectl apply -f https://raw.githubusercontent.com/google/metallb/v0.9.3/manifests/metallb.yaml
podsecuritypolicy.policy/controller created
podsecuritypolicy.policy/speaker created
serviceaccount/controller created
serviceaccount/speaker created
clusterrole.rbac.authorization.k8s.io/metallb-system:controller created
clusterrole.rbac.authorization.k8s.io/metallb-system:speaker created
role.rbac.authorization.k8s.io/config-watcher created
role.rbac.authorization.k8s.io/pod-lister created
clusterrolebinding.rbac.authorization.k8s.io/metallb-system:controller created
clusterrolebinding.rbac.authorization.k8s.io/metallb-system:speaker created
rolebinding.rbac.authorization.k8s.io/config-watcher created
rolebinding.rbac.authorization.k8s.io/pod-lister created
daemonset.apps/speaker created
deployment.apps/controller created
$ kubectl create secret generic -n metallb-system memberlist --from-literal=secretkey="$(openssl rand -base64 128)"
secret/memberlist created
$ minikube ip
172.17.0.28
$ kubectl create -f configmap.yaml


Deploy Containers Using YAML
Step 2 of 3
name: webapp1-svc
labels:
app: webapp1
spec:
type: NodePort
ports:
- port: 80
nodePort: 30080
selector:
app: webapp1
All Kubernetes objects are deployed in a consistent way using kubectl.
Deploy the Service with
kubectl create -f service.yaml
As before, details of all the Service objects deployed with
kubectl get svc
. By describing the object it's possible to discover more details about the configuration
kubectl describe svc webapp1-svc
.
curl host01:30080
KATACODA OVERVIEW & SOLUTIONS
TRY O'REILLY
LOG IN
deployment.yaml
service.yaml
/root
.bashrc
.cache/
This is your Editor pane.
apiVersion: v1
kind: ConfigMap
metadata:
namespace: metallb-system
Terminal
clusterrolebinding.rbac.authorization.k8s.io/metallb-system:speaker created
rolebinding.rbac.authorization.k8s.io/config-watcher created
rolebinding.rbac.authorization.k8s.io/pod-lister created
daemonset.apps/speaker created
deployment.apps/controller created
$ kubectl create secret generic -n metallb-system memberlist --from-literal=secretkey="$(openssl rand -base64 128)"
secret/memberlist created
$ minikube ip
172.17.0.28
$ kubectl create -f configmap.yaml
error: the path "configmap.yaml" does not exist
$ kubectl create -f service.yaml
configmap/config created
$ kubectl delete svc nodejs-deployment
service "nodejs-deployment" deleted
$ kubectl expose deployment nodejs-deployment --type="LoadBalancer"
service/nodejs-deployment exposed
$ kubectl get svc
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 5m14s
nodejs-deployment LoadBalancer 10.109.32.187 172.17.15.61 3000:32045/TCP 12s
$ curl http://172.17.15.61:32045
curl: (7) Failed to connect to 172.17.15.61 port 32045: No route to host
$ curl https://172.17.15.61:32045
curl: (7) Failed to connect to 172.17.15.61 port 32045: No route to host
$ curl host01:30080
curl: (7) Failed to connect to host01 port 30080: Connection refused
$ curl host01:32045
$ curl host01:32045
LaxmiNarayana JaganMohan Aditya$