

Assisted Practice Project4-

Problem 1

Instances (1/2)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 D
JenkinsInstance	i-04360505948a652bb	Terminated	t2.micro	—	No alarms	us-east-1a	—
kubernetesinst...	i-03c27fd08eeae5287	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-54-242-46

Instance: i-03c27fd08eeae5287 (kubernetesinstance)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

Instance summary

Instance ID: i-03c27fd08eeae5287 (kubernetesinstance)

Public IPv4 address: 54.242.48.221 | [open address](#)

Private IPv4 addresses: 172.31.24.97

Instance state: **Running**

Public IPv4 DNS: ec2-54-242-48-221.compute-1.amazonaws.com | [open address](#)

Private IP DNS name (IPv4 only): ip-172-31-24-97.ec2.internal

```
ubuntu@ip-172-31-24-97:~$ sudo apt-get install socat kubelet kubeadm kubectl
Preparing to unpack .../4-socat_1.7.4.1-3ubuntu4_amd64.deb ...
Unpacking socat (1.7.4.1-3ubuntu4) ...
Selecting previously unselected package kubelet.
Preparing to unpack .../5-kubelet_1.22.8-00_amd64.deb ...
Unpacking kubelet (1.22.8-00) ...
Selecting previously unselected package kubectl.
Preparing to unpack .../6-kubectl_1.22.8-00_amd64.deb ...
Unpacking kubectl (1.22.8-00) ...
Selecting previously unselected package kubeadm.
Preparing to unpack .../7-kubeadm_1.22.8-00_amd64.deb ...
Unpacking kubeadm (1.22.8-00) ...
Setting up socat (1.7.4.1-3ubuntu4) ...
Setting up kubelet (1.22.8-00) ...
Setting up kubectl (1.22.8-00) ...
Setting up kubeadm (1.22.8-00) ...
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service → /lib/systemd/system/kubelet.service.
Processing triggers for man-db (2.10.2-1) ...
Scanning processes...
Scanning Linux images...

Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-24-97:~$ sudo apt-mark hold kubelet kubeadm kubectl
kubelet set on hold.
kubeadm set on hold.
kubectl set on hold.
ubuntu@ip-172-31-24-97:~$ kubectl version
Client Version: version.Info{Major:"1", Minor:"22", GitVersion:"v1.22.8", GitCommit:"7061dbb75f982e8ab21f9be7e8ffcaae8e0d44", GitTreeState:"clean", BuildDate:"2022-03-16T14:10:06Z", GoVersion:"go1.16.15", Compiler:"gc", Platform:"linux/amd64"}
The connection to the server localhost:8080 was refused - did you specify the right host or port?
ubuntu@ip-172-31-24-97:~$
```

Problem 2

The screenshot shows the AWS Management Console interface. On the left is a navigation menu with options like 'New EC2 Experience', 'EC2 Dashboard', 'Events', 'Tags', 'Limits', 'Instances', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', 'Scheduled Instances', 'Capacity Reservations', and 'Images'. The main area displays a table of EC2 instances. Two instances are visible: 'JenkinsInstance' (Terminated) and 'kubernetainstance...' (Running). Below the table, the details for instance 'i-03c27fd08eeae5287' are shown, including its ID, state (Running), IP addresses, and DNS information.

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNs
<input type="checkbox"/>	JenkinsInstance	i-04360505948a652bb	Terminated	t2.micro	-	No alarms	us-east-1a	-
<input checked="" type="checkbox"/>	kubernetainstance...	i-03c27fd08eeae5287	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-54-242-48-221.compute-1.amazonaws.com

Instance: i-03c27fd08eeae5287 (kubernetainstance)

- Details:** Instance summary info
 - Instance ID: i-03c27fd08eeae5287 (kubernetainstance)
 - IPv4 address: 54.242.48.221 | open address
 - Private IPv4 addresses: 172.31.24.97
 - Instance state: Running
 - Public IPv4 DNS: ec2-54-242-48-221.compute-1.amazonaws.com | open address
 - Hostname type: ip-172-31-24-97.ec2.internal
 - Private IP DNS name (IPv4 only): ip-172-31-24-97.ec2.internal

```
ubuntu@ip-172-31-85-68:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:6 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://packages.cloud.google.com/apt kubernetes-xenial InRelease [938 B]
Fetched 120 kB in 1 (192 Kbps)
Reading package lists... Done
ubuntu@ip-172-31-85-68:~$ sudo apt-install maven -y
sudo apt-install: command not found
ubuntu@ip-172-31-85-68:~$
```

```
ubuntu@ip-172-31-26-218: ~  
Terminal Sessions View X server Tools Games Settings Macros Help  
Session Servers Tools Games Sessions View Split Multitex Tunneling Packages Settings Help  
Quick connect...  
[home/ubuntu/]  
Name  
cache  
ssh  
bash_logout  
bashrc  
profile  
xauthrc  
[control-plane] Creating static Pod manifest for "kube-scheduler"  
[etcd] Creating static Pod manifest for local etcd in "/etc/kubernetes/manifests"  
[wait-control-plane] Waiting for the kubelet to boot up the control plane as static Pods from directory "/etc/kubernetes/manifests". This can take up to 4m0s  
[api] All control plane components are healthy after 15.505138 seconds  
[upload-config] Storing the configuration used in ConfigMap "kubeadm-config" in the "kube-system" Namespace  
[kubelet] Creating a ConfigMap "kubelet-config-1.22" in namespace kube-system with the configuration for the kubelets in the cluster  
[load-certs] Starting phase: Please see https://kubernetes.io/docs/setup/independent/install-kubernetes/#load-certs  
[mark-control-plane] Marking the node ip-172-31-26-218 as control-plane by adding the labels: [node-role.kubernetes.io/master(deprecated) node-role.kubernetes.io/control-plane node.kubernetes.io/exclude-from-external-load-balancers]  
[mark-control-plane] Marking the node ip-172-31-26-218 as control-plane by adding the taints [node-role.kubernetes.io/master:NoSchedule]  
[bootstrap-token] Using token: 225331.hnf4z4swg5b04h  
[bootstrap-token] Configuring RBAC rules: Cluster-info ConfigMap, RBAC Roles  
[bootstrap-token] configured RBAC rules to allow Node Bootstrap tokens to get nodes  
[bootstrap-token] configured RBAC rules to allow Node Bootstrap tokens to post CSRs in order for nodes to get long term certificate credentials  
[bootstrap-token] configured RBAC rules to allow the csrapprover controller automatically approve CSRs from a Node Bootstrap Token  
[bootstrap-token] configured RBAC rules to allow the certificate rotation for all node client certificates in the cluster  
[bootstrap-token] Creating the "cluster-info" ConfigMap in the "kube-public" namespace  
[kubelet-finalize] Updating "/etc/kubernetes/kubelet.conf" to point to a rotatable kubelet client certificate and key  
[addons] Applied essential addon: CoreDNS  
[addons] Applied essential addon: kube-proxy  
  
Your Kubernetes control-plane has initialized successfully!  
  
To start using your cluster, you need to run the following as a regular user:  
  
mkdir -p $HOME/.kube  
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config  
sudo chown $(id -u):$(id -g) $HOME/.kube/config  
  
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 172.31.26.218:6443 --token 225331.hnf4z4swg5b04h \\  
--discovery-token-ca-cert-hash sha256:86b112f9b0b5988662121aabe5ec0ad889d356e0c6370dd7cd224f4791d7b  
ubuntu@ip-172-31-26-218:~$ export kubever=$(kubectl version | base64 | tr -d '\n')  
Follow terminal folder
```

Problem 3

Instances (1/2) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 D
JenkinsInstance	i-04360505948a652bb	Terminated	t2.micro	-	No alarms	us-east-1a	-
kubernetesinst...	i-03c27fd08eeae5287	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-54-242-48

Instance: i-03c27fd08eeae5287 (kubernetesinstance)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

Instance summary Info

Instance ID	i-03c27fd08eeae5287 (kubernetesinstance)	Public IPv4 address	54.242.48.221 open address	Private IPv4 addresses	172.31.24.97
IPv6 address	-	Instance state	Running	Public IPv4 DNS	ec2-54-242-48-221.compute-1.amazonaws.com open address
Hostname type	IP name: ip-172-31-24-97.ec2.internal	Private IP DNS name (IPv4 only)	ip-172-31-24-97.ec2.internal		

```
[INFO] No sources to compile
[INFO] --- maven-surefire-plugin:2.22.2:test (default-test) @ demo-docker ---
[INFO] No tests to run.
[INFO] --- maven-jar-plugin:3.2.0:jar (default-jar) @ demo-docker ---
[INFO] Building jar: /home/ubuntu/SpringBootDockerApp/target/demo-docker-0.0.1-SNAPSHOT.jar
[INFO] --- spring-boot-maven-plugin:2.5.6:repackage (repackage) @ demo-docker ---
[INFO] Replacing main artifact with repackaged archive
[INFO] --- maven-install-plugin:2.5.2:install (default-install) @ demo-docker ---
[INFO] Installing /home/ubuntu/SpringBootDockerApp/target/demo-docker-0.0.1-SNAPSHOT.jar to /home/ubuntu/.m2/repository/com/example/demo-docker/0.0.1-SNAPSHOT/demo-docker-0.0.1-SNAPSHOT.jar
[INFO] Installing /home/ubuntu/SpringBootDockerApp/pom.xml to /home/ubuntu/.m2/repository/com/example/demo-docker/0.0.1-SNAPSHOT/pom.xml
[INFO] BUILD SUCCESS
[INFO] Total time: 11.367 s
[INFO] Finished at: 2022-10-07T06:05:39Z
[INFO]
Step 1/5 : FROM java
manifest for java:latest not found: manifest unknown: manifest unknown
ubuntu@ip-172-31-85-68:~/SpringBootDockerApp$ sudo docker build -t springboot .
Sending build context to Docker daemon 17.62MB
Step 1/5 : FROM java
manifest for java:latest not found: manifest unknown: manifest unknown
ubuntu@ip-172-31-85-68:~/SpringBootDockerApp$ sudo docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
k8s.gcr.io/kube-apiserver   v1.22.15       bd16c7ea581a   2 weeks ago    129MB
k8s.gcr.io/kube-controller-manager   v1.22.15       c9f8999a4422   2 weeks ago    122MB
k8s.gcr.io/kube-scheduler   v1.22.15       b4999e7b5215   2 weeks ago    52.7MB
k8s.gcr.io/kube-proxy       v1.22.15       8f9e316d565d   2 weeks ago    104MB
k8s.gcr.io/pause            3.5-pa0        004811815384   15 months ago  295MB
k8s.gcr.io/coredns/coredns v1.8.4         8d147537fb7d   16 months ago  47.6MB
k8s.gcr.io/pause            3.5            ed210e3e45b    18 months ago  683kB
ubuntu@ip-172-31-85-68:~/SpringBootDockerApp$ pod/springbootapp created
ubuntu@ip-172-31-85-68:~/SpringBootDockerApp$ kubectl run springbootapp --image=springboot --port=8082
```

Problem 4

Instances (1/2) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 D
JenkinsInstance	i-04360505948a652bb	Terminated	t2.micro	-	No alarms	us-east-1a	-
kubernetesinst...	i-03c27fd08eeae5287	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-54-242-48

Instance: i-03c27fd08eeae5287 (kubernetesinstance)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

Instance summary Info

Instance ID	i-03c27fd08eeae5287 (kubernetesinstance)	Public IPv4 address	54.242.48.221 open address	Private IPv4 addresses	172.31.24.97
IPv6 address	-	Instance state	Running	Public IPv4 DNS	ec2-54-242-48-221.compute-1.amazonaws.com open address
Hostname type	IP name: ip-172-31-24-97.ec2.internal	Private IP DNS name (IPv4 only)	ip-172-31-24-97.ec2.internal		

