

EXPLORER ...

> OPEN EDITORS

< LAB ASSESSMENT

- > ansible
- > docker
- > grafana and Promet...
- > jenkins
- ✓ kubernetes
 - > .terraform
 - main.tf
 - \$ master-userdata.sh
 - { } terraform.tfstate
 - terraform.tfstate.b...
 - \$ worker-userdata.sh
- > terraform
- { } terraform.tfstate

main.tf docker \$ docker-userdata.sh docker-key.pem main.tf kubernetes X main.tf grafana and Prometheus \$ ec2-server-userdata.sh \$ prom-graf-userdata.sh

kubernetes > main.tf > resource "aws_instance" "worker"

```
26 resource "aws_security_group" "kube_sg" {  
27   name      = "kube-cluster-sg"  
28   description = "Allow inbound traffic for Kubernetes Cluster"  
29  
30   ingress {  
31     from_port  = 0  
32     to_port    = 65535  
33     protocol   = "tcp"  
34     cidr_blocks = ["0.0.0.0/0"]  
35   }  
36  
37   egress {  
38     from_port  = 0  
39     to_port    = 0  
40     protocol   = "-1"  
41     cidr_blocks = ["0.0.0.0/0"]  
42   }  
43  
44   tags = {  
45     Name = "kube-cluster-sg"  
46   }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

zsh - kubernetes + ×

```
● mac@SirNicks-MBP Lab Assessemnt % cd kubernetes  
● mac@SirNicks-MBP kubernetes % terraform init  
  Initializing the backend...  
  Initializing provider plugins...  
    - Reusing previous version of hashicorp/local from the dependency lock file  
    - Reusing previous version of hashicorp/aws from the dependency lock file  
    - Reusing previous version of hashicorp/tls from the dependency lock file  
    - Using previously-installed hashicorp/local v2.5.2  
    - Using previously-installed hashicorp/aws v5.76.0  
    - Using previously-installed hashicorp/tls v4.0.6  
  
  Terraform has been successfully initialized!  
  
  You may now begin working with Terraform. Try running "terraform plan" to see  
  any changes that are required for your infrastructure. All Terraform commands  
  should now work.  
  
  If you ever set or change modules or backend configuration for Terraform,  
  rerun this command to reinitialize your working directory. If you forget, other  
  commands will detect it and remind you to do so if necessary.  
○ mac@SirNicks-MBP kubernetes % terraform validate
```

> OUTLINE

> TIMELINE

× 0 ▲ 0 ⌂ 0

Ln 66, Col 70 Spaces: 2 UTF-8 LF {} Terraform

EXPLORER ...

> OPEN EDITORS

< LAB ASSESSMENT

- > ansible
- > docker
- > grafana and Promet...
- > jenkins
- < kubernetes
 - > .terraform
 - .terraform.lock.hcl**
 - .terraform.tfstate.l...**
 - main.tf**
 - \$ master-userdata.sh
 - {} terraform.tfstate
 - terraform.tfstate.b...**
 - \$ worker-userdata.sh
- > terraform
- {} terraform.tfstate

main.tf docker \$ docker-userdata.sh docker-key.pem main.tf kubernetes X main.tf grafana and Prometheus \$ ec2-server-userdata.sh \$ prom-graf-userdata.sh

kubernetes > main.tf > resource "aws_instance" "worker"

```

26  resource "aws_security_group" "kube_sg" {
27    name      = "kube-cluster-sg"
28    description = "Allow inbound traffic for Kubernetes Cluster"
29
30    ingress {
31      from_port  = 0
32      to_port    = 65535
33      protocol   = "tcp"
34      cidr_blocks = ["0.0.0.0/0"]
35    }
36
37    egress {
38      from_port  = 0
39      to_port    = 0
40      protocol   = "-1"
41      cidr_blocks = ["0.0.0.0/0"]
42    }
43
44    tags = {
45      Name = "kube-cluster-sg"
46    }

```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

terraform - kubernetes

```

● mac@SirNicks-MBP Lab Assessemnt % cd kubernetes
● mac@SirNicks-MBP kubernetes % terraform init
  Initializing the backend...
  Initializing provider plugins...
    - Reusing previous version of hashicorp/local from the dependency lock file
    - Reusing previous version of hashicorp/aws from the dependency lock file
    - Reusing previous version of hashicorp/tls from the dependency lock file
    - Using previously-installed hashicorp/local v2.5.2
    - Using previously-installed hashicorp/aws v5.76.0
    - Using previously-installed hashicorp/tls v4.0.6

  Terraform has been successfully initialized!

  You may now begin working with Terraform. Try running "terraform plan" to see
  any changes that are required for your infrastructure. All Terraform commands
  should now work.

  If you ever set or change modules or backend configuration for Terraform,
  rerun this command to reinitialize your working directory. If you forget, other
  commands will detect it and remind you to do so if necessary.
● mac@SirNicks-MBP kubernetes % terraform validate
  Success! The configuration is valid.

○ mac@SirNicks-MBP kubernetes % terraform plan

```

> OUTLINE

> TIMELINE

Ln 66, Col 70 Spaces: 2 UTF-8 LF {} Terraform

EXPLORER ...

> OPEN EDITORS

< LAB ASSESSMENT

- > ansible
- > docker
- > grafana and Promet...
- > jenkins
- < kubernetes
- > .terraform
- main.tf lock.hcl
- .terraform.tfstate.l...

main.tf

\$ master-userdata.sh

{} terraform.tfstate

≡ terraform.tfstate.b...

\$ worker-userdata.sh

> terraform

{} terraform.tfstate

main.tf docker

\$ docker-userdata.sh

docker-key.pem

main.tf kubernetes X

main.tf grafana and Prometheus

\$ ec2-server-userdata.sh

\$ prom-graf-userdata.sh

kubernetes > main.tf > resource "aws_instance" "worker"

```
26 resource "aws_security_group" "kube_sg" {  
27   name      = "kube-cluster-sg"  
28   description = "Allow inbound traffic for Kubernetes Cluster"  
29  
30   ingress {  
31     from_port  = 0  
32     to_port    = 65535  
33     protocol   = "tcp"  
34     cidr_blocks = ["0.0.0.0/0"]  
35   }  
36  
37   egress {  
38     from_port  = 0  
39     to_port    = 0  
40     protocol   = "-1"  
41     cidr_blocks = ["0.0.0.0/0"]  
42   }  
43  
44   tags = {  
45     Name = "kube-cluster-sg"  
46   }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

terraform - kubernetes

```
+ content_sha256      = (known after apply)  
+ content_sha512      = (known after apply)  
+ directory_permission = "0777"  
+ file_permission     = "600"  
+ filename             = "kube-key.pem"  
+ id                  = (known after apply)  
}  
  
# tls_private_key.keypair will be created  
+ resource "tls_private_key" "keypair" {  
  + algorithm          = "RSA"  
  + ecdsa_curve        = "P224"  
  + id                 = (known after apply)  
  + private_key.openssh = (sensitive value)  
  + private_key_pem    = (sensitive value)  
  + private_key_pkcs8  = (sensitive value)  
  + public_key_fingerprint_md5 = (known after apply)  
  + public_key_fingerprint_sha256 = (known after apply)  
  + public_key.openssh  = (known after apply)  
  + public_key_pem     = (known after apply)  
  + rsa_bits           = 4096  
}  
  
Plan: 7 to add, 0 to change, 0 to destroy.  
  
Changes to Outputs:  
+ master      = (known after apply)  
+ worker_nodes = [  
  + (known after apply),  
  + (known after apply),  
]
```

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.

O mac@SirNicks-MBP kubernetes % terraform apply -auto-approve

Ln 66, Col 70 Spaces: 2 UTF-8 LF {} Terraform

The screenshot shows a Visual Studio Code (VS Code) window with the following details:

- Explorer View:** Shows a tree structure of files and folders under the "LAB AS..." category, with "kubernetes" selected.
- Editor View:** Displays a Terraform configuration file (`main.tf`) for a Kubernetes cluster. The code defines an AWS Security Group ("kube_sg") with ingress and egress rules, and an AWS Instance ("aws_instance") resource.
- Terminal View:** Shows the output of a Terraform plan command. It includes:
 - Plan:** 7 to add, 0 to change, 0 to destroy.
 - Changes to Outputs:** Lists new outputs: `master` and `worker_nodes`.
 - Resource Creation Log:** Shows the creation of various resources like `tls_private_key.keypair`, `aws_security_group.kube_sg`, and multiple `aws_instance` resources.
 - Completion Message:** `Apply complete! Resources: 7 added, 0 changed, 0 destroyed.`
 - Outputs:** Prints the values of `master` and `worker_nodes`.
- Status Bar:** Shows file status (e.g., `④ 0 ▲ 0 ↵ 0`), current file path (`mac@SirNicks-MBP kubernetes %`), and system information (Ln 66, Col 70, Spaces: 2, UTF-8, LF, {}, Terraform).

EXPLORER ...

> OPEN EDITORS

< LAB ASSESSMENT

- > ansible
- > docker
- > grafana and Promet...
- > jenkins
- ✓ kubernetes
 - > .terraform
 - ✓ .terraform.lock.hcl
 - ✓ kube-key.pem
 - ✓ main.tf
 - \$ master-userdata.sh
 - { } terraform.tfstate
 - ✗ terraform.tfstate.b...
 - \$ worker-userdata.sh
 - > terraform
 - { } terraform.tfstate

26 resource "aws_security_group" "kube_sg" {
 27 name = "kube-cluster-sg"
 28 description = "Allow inbound traffic for Kubernetes Cluster"
 29
 30 ingress {
 31 from_port = 0
 32 to_port = 65535
 33 protocol = "tcp"
 34 cidr_blocks = ["0.0.0.0/0"]
 35 }
 36
 37 egress {
 38 from_port = 0
 39 to_port = 0
 40 protocol = "-1"
 41 cidr_blocks = ["0.0.0.0/0"]
 42 }
 43
 44 tags = {
 45 Name = "kube-cluster-sg"
 46 }

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

ssh - kubernetes + ×

```
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '35.179.163.112' (ED25519) to the list of known hosts.
ubuntu@35.179.163.112: Permission denied (publickey).
○ mac@SirNicks-MBP kubernetes % ssh -i kube-key.pem ubuntu@35.179.163.112
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1016-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Wed Nov 27 14:52:37 UTC 2024

System load: 0.15           Processes:          146
Usage of /: 49.9% of 6.71GB  Users logged in:      0
Memory usage: 20%           IPv4 address for enX0: 172.31.24.101
Swap usage:  0%             

Expanded Security Maintenance for Applications is not enabled.

64 updates can be applied immediately.
33 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
```

outline timeline

ubuntu@master172:~\$

Ln 66, Col 70 Spaces: 2 UTF-8 LF {} Terraform

EXPLORER ...

> OPEN EDITORS

< LAB ASSESSMENT

> ansible

> docker

> grafana and Prometheus

> jenkins

> kubernetes

> .terraform

Y .terraform.lock.hcl

🔒 kube-key.pem

Y main.tf

\$ master-userdata.sh

{ } terraform.tfstate

≡ terraform.tfstate.b...

\$ worker-userdata.sh

> terraform

{ } terraform.tfstate

main.tf docker

\$ docker-userdata.sh

🔒 docker-key.pem

Y main.tf kubernetes X

Y main.tf grafana and Prometheus

\$ ec2-server-userdata.sh

\$ prom-graf-userdata.sh

kubernetes > Y main.tf > 📁 resource "aws_instance" "worker"

```
26   resource "aws_security_group" "kube_sg" {
27     name      = "kube-cluster-sg"
28     description = "Allow inbound traffic for Kubernetes Cluster"
29
30     ingress {
31       from_port  = 0
32       to_port    = 65535
33       protocol   = "tcp"
34       cidr_blocks = ["0.0.0.0/0"]
35     }
36
37     egress {
38       from_port  = 0
39       to_port    = 0
40       protocol   = "-1"
41       cidr_blocks = ["0.0.0.0/0"]
42     }
43
44     tags = {
45       Name = "kube-cluster-sg"
46     }
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

ssh - kubernetes + ×

```
Memory usage: 20%          IPv4 address for enX0: 172.31.24.101
Swap usage:  0%

Expanded Security Maintenance for Applications is not enabled.

64 updates can be applied immediately.
33 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

ubuntu@master172:~$ kubectl get nodes
NAME           STATUS   ROLES      AGE   VERSION
master172.31.24.101   Ready   control-plane   6m54s   v1.29.11
ubuntu@master172:~$ ls -al
total 40
drwxr-x--- 5 ubuntu  ubuntu 4096 Nov 27 14:52 .
drwxr-xr-x 3 root   root  4096 Nov 27 14:45 ..
-rw-r--r-- 1 ubuntu  ubuntu 220 Mar 31 2024 .bash_logout
-rw-r--r-- 1 ubuntu  ubuntu 3771 Mar 31 2024 .bashrc
drwx----- 2 ubuntu  ubuntu 4096 Nov 27 14:52 .cache
drwxrwxr-x 3 ubuntu  ubuntu 4096 Nov 27 14:47 .kube
-rw-r--r-- 1 ubuntu  ubuntu  807 Mar 31 2024 .profile
drwx----- 2 ubuntu  ubuntu 4096 Nov 27 14:45 .ssh
-rw-r--r-- 1 ubuntu  ubuntu  0 Nov 27 14:47 .sudo_as_admin_successful
-rw-r--r-- 1 root   root  6121 Nov 27 14:47 output.log
ubuntu@master172:~$ cat output.log
```

EXPLORER ...

> OPEN EDITORS

< LAB ASSESSMENT

- > ansible
- > docker
- > grafana and Promet...
- > jenkins
- < kubernetes
- > .terraform
- ↳ .terraform.lock.hcl
- ↳ kube-key.pem
- ↳ main.tf
- \$ master-userdata.sh
- { } terraform.tfstate
- ≡ terraform.tfstate.b...
- \$ worker-userdata.sh
- > terraform
- { } terraform.tfstate

main.tf docker \$ docker-userdata.sh 🔒 docker-key.pem main.tf kubernetes X main.tf grafana and Prometheus \$ ec2-server-userdata.sh \$ prom-graf-userdata.sh

kubernetes > main.tf > resource "aws_instance" "worker"

```

26   resource "aws_security_group" "kube_sg" {
27     name      = "kube-cluster-sg"
28     description = "Allow inbound traffic for Kubernetes Cluster"
29
30     ingress {
31       from_port  = 0
32       to_port    = 65535
33       protocol   = "tcp"
34       cidr_blocks = ["0.0.0.0/0"]
35     }
36
37     egress {
38       from_port  = 0
39       to_port    = 0
40       protocol   = "-1"
41       cidr_blocks = ["0.0.0.0/0"]
42     }
43
44     tags = {
45       Name = "kube-cluster-sg"
46     }

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

[bootstrap-token] Configured RBAC rules to allow 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/>

You can now join any number of the control-plane node running the following command on each as root:

```

kubeadm join 172.31.24.101:6443 --token ubk1gl.vy4iswhliolodkqo \
--discovery-token-ca-cert-hash sha256:3f2014e41a40eda3eeecb09de68eeaf9c9a38fe28e21f58010d21c2c275d37f12 \
--control-plane --certificate-key ce2f85900781b7bdaa1b1ce58e2de7f63d70287c013457626e024e85708a7da7

```

Please note that the certificate-key gives access to cluster sensitive data, keep it secret!
As a safeguard, uploaded-certs will be deleted in two hours; If necessary, you can use
"\"kubeadm init phase upload-certs --upload-certs\" to reload certs afterward.

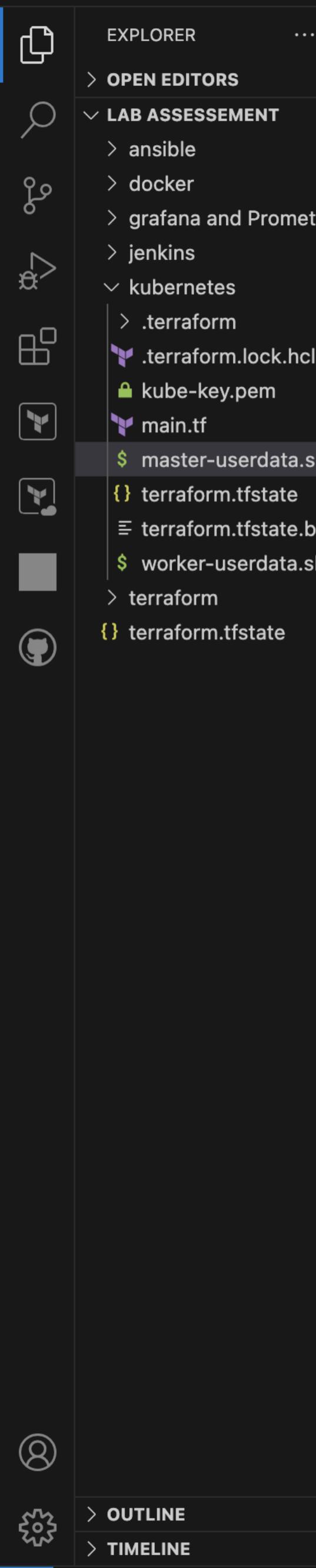
Then you can join any number of worker nodes by running the following on each as root:

```

kubeadm join 172.31.24.101:6443 --token ubk1gl.vy4iswhliolodkqo \
--discovery-token-ca-cert-hash sha256:3f2014e41a40eda3eeecb09de68eeaf9c9a38fe28e21f58010d21c2c275d37f12
ubuntu@master172:~$ 

```

Ln 66, Col 70 Spaces: 2 UTF-8 LF {} Terraform



ubuntu@master172:~\$ cat output.log

```
I1127 14:46:45.438659    3275 version.go:256] remote version is much newer: v1.31.3; falling back to: stable-1.29
[init] Using Kubernetes version: v1.29.11
[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'
W1127 14:46:58.375658    3275 checks.go:835] detected that the sandbox image "registry.k8s.io/pause:3.8" of the container runtime is inconsistent with that used by kubeadm. It is recommended that using "registry.k8s.io/pause:3.9" as the CRI sandbox image.
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster.local master172.31.24.101] and IPs [10.96.0.1 172.31.24.101]
[certs] Generating "apiserver-kubelet-client" certificate and key
[certs] Generating "front-proxy-ca" certificate and key
[certs] Generating "front-proxy-client" certificate and key
[certs] Generating "etcd/ca" certificate and key
[certs] Generating "etcd/server" certificate and key
[certs] etcd/server serving cert is signed for DNS names [localhost master172.31.24.101] and IPs [172.31.24.101 127.0.0.1 ::1]
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [localhost master172.31.24.101] and IPs [172.31.24.101 127.0.0.1 ::1]
[certs] Generating "etcd/healthcheck-client" certificate and key
[certs] Generating "apiserver-etcd-client" certificate and key
[certs] Generating "sa" key and public key
[kubeconfig] Using kubeconfig folder "/etc/kubernetes"
[kubeconfig] Writing "admin.conf" kubeconfig file
[kubeconfig] Writing "super-admin.conf" kubeconfig file
[kubeconfig] Writing "kubelet.conf" kubeconfig file
[kubeconfig] Writing "controller-manager.conf" kubeconfig file
[kubeconfig] Writing "scheduler.conf" kubeconfig file
[etcd] Creating static Pod manifest for local etcd in "/etc/kubernetes/manifests"
[control-plane] Using manifest folder "/etc/kubernetes/manifests"
[control-plane] Creating static Pod manifest for "kube-apiserver"
[control-plane] Creating static Pod manifest for "kube-controller-manager"
[control-plane] Creating static Pod manifest for "kube-scheduler"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Starting the kubelet
[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
[apiclient] All control plane components are healthy after 6.001378 seconds
[upload-config] Storing the configuration used in ConfigMap "kubeadm-config" in the "kube-system" Namespace
[kubelet] Creating a ConfigMap "kubelet-config" in namespace kube-system with the configuration for the kubelets in the cluster
[upload-certs] Storing the certificates in Secret "kubeadm-certs" in the "kube-system" Namespace
[upload-certs] Using certificate key:
ce2f85900781b7bdaa1b1ce58e2de7f63d70287c013457626e024e85708a7da7
[mark-control-plane] Marking the node master172.31.24.101 as control-plane by adding the labels: [node-role.kubernetes.io/control-plane node.kubernetes.io/exclude-from-external-load-balancers]
[mark-control-plane] Marking the node master172.31.24.101 as control-plane by adding the taints [node-role.kubernetes.io/control-plane:NoSchedule]
[bootstrap-token] Using token: ubk1gl.vy4iswhliolodkqo
[bootstrap-token] Configuring bootstrap tokens, 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 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:

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
kubeadm join 172.31.24.101:6443 --token ubk1gl.vy4iswhliolodkqo \
    --discovery-token-ca-cert-hash sha256:3f2014e41a40eda3eeecb09de68eeaf9c9a38fe28e21f58010d21c2c275d37f12
ubuntu@master172:~$ exit
```

```
logout
Connection to 35.179.163.112 closed.
● mac@SirNicks-MBP kubernetes % terraform output
master = "35.179.163.112"
worker_nodes = [
    "13.40.173.93",
    "3.8.137.207",
]

```

```
mac@SirNicks-MBP kubernetes % ssh -i kube-key.pem ubuntu@13.40.173.93
The authenticity of host '13.40.173.93 (13.40.173.93)' can't be established.
ED25519 key fingerprint is SHA256:tt4HecYxPLsbL0LAh/OIr/ISOHvajuMEnpQN/B4GpPs.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '13.40.173.93' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1016-aws x86_64)
```

```
* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/pro
```

System information as of Wed Nov 27 15:11:53 UTC 2024

System load: 0.0 Processes: 117
Usage of /: 34.4% of 6.71GB Users logged in: 0
Memory usage: 7% IPv4 address for enX0: 172.31.31.243
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled

64 updates can be applied immediately.
33 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See <https://ubuntu.com/esm> or run: sudo pro status

The programs included with the Ubuntu system are free software:

Col 1 Spaces: 2 UTF-8 LF Shell Script

EXPLORER ...

> OPEN EDITORS

LAB ASSESSMENT

- > ansible
- > docker
- > grafana and Promet...
- > jenkins
- ✓ kubernetes
 - > .terraform
 - ✓ .terraform.lock.hcl
 - ✓ kube-key.pem
 - ✓ main.tf
- ✓ \$ master-userdata.sh
- {} terraform.tfstate
- ✗ terraform.tfstate.b...
- ✓ \$ worker-userdata.sh
- > terraform
- {} terraform.tfstate

main.tf docker docker-key.pem main.tf kubernetes \$ master-userdata.sh × main.tf grafana and Prometheus \$ ec2-server-userdata.sh \$ prom-graf-userdata.sh

kubernetes > \$ master-userdata.sh

```
64 apt-get install -y kubelet kubeadm kubectl kubernetes-cni nfs-common
65
66 # Set the hostname to indicate master node
67 sudo hostnamectl set-hostname master$(hostname -i)
68
69 # Initialize Kubernetes cluster with specified CIDR and save output log
70 sudo kubeadm init --pod-network-cidr=192.168.0.0/16 --upload-certs --control-plane-endpoint $(hostname -I | awk '{print $1}'):6443 > /home/ubuntu/output.log 2>&1
71
72 # Give 'ubuntu' user access to Kubernetes configuration file
73 sudo su -c 'mkdir -p $HOME/.kube' ubuntu
74 sudo su -c 'sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config' ubuntu
75 sudo su -c 'sudo chown $(id -u):$(id -g) $HOME/.kube/config' ubuntu
76
77 # Using the ubuntu user install Weave Net as the pod network
78 sudo su -c 'kubectl apply -f https://github.com/weaveworks/weave/releases/download/v2.8.1/weave-daemonset-k8s.yaml' ubuntu
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS ssh - kubernetes + ▾

Memory usage: 7% IPv4 address for enX0: 172.31.31.243
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.
64 updates can be applied immediately.
33 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable
Enable ESM Apps to receive additional future security updates.
See <https://ubuntu.com/esm> or run: sudo pro status

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

```
ubuntu@ip-172-31-31-243:~$ sudo su
root@ip-172-31-31-243:/home/ubuntu# kubeadm join 172.31.24.101:6443 --token ubk1gl.vy4iswhliolodkqo \
--discovery-token-ca-cert-hash sha256:3f2014e41a40eda3eebc09de68eeaf9c9a38fe28e21f58010d21c2c275d37f12
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap...
```

This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.

root@ip-172-31-31-243:/home/ubuntu#

Ln 1, Col 1 Spaces: 2 UTF-8 LF Shell Script

EXPLORER ...

> OPEN EDITORS

< LAB ASSESSMENT

- > ansible
- > docker
- > grafana and Promet...
- > jenkins
- < kubernetes
- > .terraform
- .terraform.lock.hcl**
- kube-key.pem**
- main.tf**
- \$ master-userdata.sh
- {} terraform.tfstate
- ≡ terraform.tfstate.b...
- \$ worker-userdata.sh

> terraform

{} terraform.tfstate

main.tf docker

docker-key.pem

main.tf kubernetes

kube-key.pem

main.tf grafana and Prometheus

\$ ec2-server-userdata.sh

\$ prom-graf-userdata.sh

kubernetes > main.tf > resource "aws_instance" "worker"

```

26  resource "aws_security_group" "kube_sg" {
27    name      = "kube-cluster-sg"
28    description = "Allow inbound traffic for Kubernetes Cluster"
29
30    ingress {
31      from_port  = 0
32      to_port    = 65535
33      protocol   = "tcp"
34      cidr_blocks = ["0.0.0.0/0"]
35    }
36
37    egress {
38      from_port  = 0
39      to_port    = 0
40      protocol   = "-1"
41      cidr_blocks = ["0.0.0.0/0"]
42    }

```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

ssh - kubernetes + ⚙️ 🗑️ ⌂ ⌄ ⌅ ⌈ ⌉

```

ubuntu@master172:~$ exit
logout
Connection to 35.179.163.112 closed.
mac@SirNicks-MBP kubernetes % ssh -i kube-key.pem ubuntu@3.8.137.207
The authenticity of host '3.8.137.207 (3.8.137.207)' can't be established.
ED25519 key fingerprint is SHA256:UuVIhI7nDIrw45IwCPq1jJ1aX2e+/KFtDCGmw3uYQfs.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.8.137.207' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1016-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

System information as of Wed Nov 27 15:33:28 UTC 2024

System load: 0.0          Processes:      117
Usage of /: 34.4% of 6.71GB  Users logged in:  0
Memory usage: 7%           IPv4 address for enX0: 172.31.26.145
Swap usage: 0% 

Expanded Security Maintenance for Applications is not enabled.

64 updates can be applied immediately.
33 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

Ln 66, Col 70 Spaces: 2 UTF-8 LF {} Terraform

The screenshot shows a Visual Studio Code interface with multiple tabs open at the top, including 'main.tf docker', 'docker-key.pem', 'main.tf kubernetes X', 'kube-key.pem', 'main.tf grafana and Prometheus', '\$ ec2-server-userdata.sh', and '\$ prom-graf-userdata.sh'. The left sidebar has sections for 'OPEN EDITORS' and 'LAB ASSESSMENT' containing files like 'main.tf', '.terraform.lock.hcl', 'kube-key.pem', and 'worker-userdata.sh'. The main area is a terminal window titled 'zsh - kubernetes' showing the following output:

```
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-26-145:~$ sudo su
root@ip-172-31-26-145:/home/ubuntu# kubeadm join 172.31.24.101:6443 --token ubk1gl.vy4iswhliolodkqo \
--discovery-token-ca-cert-hash sha256:3f2014e41a40eda3eecb09de68eeaf9c9a38fe28e21f58010d21c2c275d37f12
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap...

This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.

root@ip-172-31-26-145:/home/ubuntu# exit
exit
ubuntu@ip-172-31-26-145:~$ exit
logout
Connection to 3.8.137.207 closed.
● mac@SirNicks-MBP kubernetes % terraform output
master = "35.179.163.112"
worker_nodes = [
    "13.40.173.93",
    "3.8.137.207",
]
mac@SirNicks-MBP kubernetes %
```

← → 🔍 Lab Assesment

EXPLORE
...
main.tf docker
🔒 docker-key.pem
main.tf kubernetes X
🔒 kube-key.pem
main.tf grafana and Prometheus
\$ ec2-server-userdata.sh
\$ prom-graf-userdata.sh
...

> OPEN EDITORS
kubernetes > main.tf > resource "aws_instance" "worker"

LAB ASSESSMENT
26 resource "aws_security_group" "kube_sg" {
27 name = "kube-cluster-sg"
28 description = "Allow inbound traffic for Kubernetes Cluster"
29
30 ingress {
31 from_port = 0
32 to_port = 65535
33 protocol = "tcp"
34 cidr_blocks = ["0.0.0.0/0"]
35 }
36
37 egress {
38 from_port = 0
39 to_port = 0
40 protocol = "-1"
41 cidr_blocks = ["0.0.0.0/0"]
42 }

> ansible
> docker
> grafana and Promet...
> jenkins
> kubernetes
> .terraform
.terraform.lock.hcl
🔒 kube-key.pem

main.tf
\$ master-userdata.sh
{} terraform.tfstate
= terraform.tfstate.b...
\$ worker-userdata.sh

> terraform
{} terraform.tfstate

PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS

ssh - kubernetes
+
...
^
X

* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.

```
root@ip-172-31-26-145:/home/ubuntu# exit
exit
ubuntu@ip-172-31-26-145:~$ exit
logout
Connection to 3.8.137.207 closed.
```

- mac@SirNicks-MBP kubernetes % terraform output


```
master = "35.179.163.112"
worker_nodes = [
  "13.40.173.93",
  "3.8.137.207",
]
```
- mac@SirNicks-MBP kubernetes % ssh -i kube-key.pem ubuntu@35.179.163.112


```
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1016-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Wed Nov 27 15:42:00 UTC 2024

System load: 0.08      Processes:          147
Usage of /: 50.1% of 6.71GB  Users logged in:  0
Memory usage: 20%        IPv4 address for enX0: 172.31.24.101
Swap usage:  0%
```

Expanded Security Maintenance for Applications is not enabled.

64 updates can be applied immediately.
 33 of these updates are standard security updates.
 To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
 See <https://ubuntu.com/esm> or run: sudo pro status

Last login: Wed Nov 27 15:32:37 2024 from 105.112.126.99
 ubuntu@master172:~\$

Ln 66, Col 70
Spaces: 2
UTF-8
LF
{}
Terraform
Q

eu-west-2.console.aws.amazon.com/ec2/home?region=eu-west-2#Instances:

S3 | IAM | EC2 | Route 53 | Lambda | CloudWatch | Amazon Transcribe | VPC

Search [Option+S]

Last updated less than a minute ago

Instances (3) Info

Find Instance by attribute or tag (case-sensitive)

All states ▾

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
<input type="checkbox"/>	worker-node-1	i-0d4272794ab8df7ad	Running	t2.medium	2/2 checks passed	View alarms +	eu-west-2a	ec2-3-8-137-207.eu-we...
<input type="checkbox"/>	worker-node-0	i-03982581f36fe057c	Running	t2.medium	2/2 checks passed	View alarms +	eu-west-2a	ec2-13-40-173-93.eu-w...
<input type="checkbox"/>	master-node	i-09d1a958bb68df6c2	Running	t2.medium	2/2 checks passed	View alarms +	eu-west-2a	ec2-35-179-163-112.eu...

Select an instance

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

```
PROBLEMS      OUTPUT      DEBUG CONSOLE      TERMINAL      PORT

apiVersion: apps/v1
kind: Deployment
metadata:
  name: pac-man-deployment
  labels:
    app: pac-man
spec:
  replicas: 1
  selector:
    matchLabels:
      app: pac-man
  template:
    metadata:
      labels:
        app: pac-man
  spec:
    containers:
      - name: pac-man
        image: jesseechoch/pacman-nodejs-app:latest
        ports:
          - containerPort: 8080
```

The screenshot shows a terminal window with the following content:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: pac-man-deployment
  labels:
    app: pac-man
spec:
  replicas: 1
  selector:
    matchLabels:
      app: pac-man
  template:
    metadata:
      labels:
        app: pac-man
    spec:
      containers:
```

* Documentation: <https://help.ubuntu.com>
* Management: <https://landscape.canonical.com>
* Support: <https://ubuntu.com/pro>

System information as of Wed Nov 27 15:42:00 UTC 2024

System load	Processes
0.08	147

Usage of /: 50.1% of 6.71GB
Memory usage: 20%
Swap usage: 0%

IPv4 address for enX0: 172.31.24.101

Expanded Security Maintenance for Applications is not enabled.

64 updates can be applied immediately.
33 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See <https://ubuntu.com/esm> or run: sudo pro status

```
Last login: Wed Nov 27 15:32:37 2024 from 105.112.126.99
ubuntu@master172:~$ kubectl get nodes
NAME           STATUS   ROLES      AGE   VERSION
ip-172-31-26-145   Ready    <none>   5m6s  v1.29.11
ip-172-31-31-243   Ready    <none>   27m   v1.29.11
master172.31.24.101   Ready    control-plane  55m   v1.29.11
ubuntu@master172:~$ vi deployment.yml
ubuntu@master172:~$ kubectl get nodes
NAME           STATUS   ROLES      AGE   VERSION
ip-172-31-26-145   Ready    <none>   21m   v1.29.11
ip-172-31-31-243   Ready    <none>   44m   v1.29.11
master172.31.24.101   Ready    control-plane  71m   v1.29.11
ubuntu@master172:~$ kubectl get pods
No resources found in default namespace.
ubuntu@master172:~$ kubectl apply -f deployment.yml
deployment.apps/pac-man-deployment created
ubuntu@master172:~$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
pac-man-deployment-6c5bff75df-2fvcz  0/1     ErrImagePull  0          9s
ubuntu@master172:~$
```

The screenshot shows a Microsoft Visual Studio Code interface with the following details:

- Left Sidebar:** Includes icons for Explorer, Open Editors, Lab Assessment, and various project files like ansible, docker, grafana and Prometheus, jenkins, kubernetes, .terraform, .terraform.lock.hcl, deployment.yml, kube-key.pem, main.tf, master-userdata.sh, terraform.tfstate, worker-userdata.sh, and terraform.tfstate.
- Top Bar:** Shows tabs for main.tf (docker), docker-key.pem, main.tf (kubernetes), master-userdata.sh, deployment.yml (active), main.tf (grafana and Prometheus), ec2-server-userdata.sh, and prom-graf-userdata.sh. A search bar "Lab Assesment" is also present.
- Editor Area:** Displays a YAML file for a Kubernetes deployment named "deployment.yml". The file defines a pod template with a container named "pac-man" running "pacman-nodejs-app:latest" image on port 8080. It also defines a service named "pac-man-service" with a selector for the pod, type "NodePort", and ports 80 (TCP) and 8080 (targetPort).
- Bottom Status Bar:** Shows file statistics (Ln 23, Col 1 (208 selected)), code analysis (Spaces: 2, UTF-8, LF), and the file extension (YAML).