Project

Install dependencies on master and worker nodes

```
TASK [Install k8s] *****************************************************************************************************************************************************
changed: [master] => (item=kubelet)
changed: [node1] => (item=kubelet)
changed: [node2] => (item=kubelet)
changed: [master] => (item=kubeadm)
changed: [node1] => (item=kubeadm)
changed: [node2] => (item=kubeadm)
ok: [master] => (item=kubectl)
ok: [node1] => (item=kubectl)
ok: [node2] => (item=kubectl)

TASK [enable kubelet] **************************************************************************************************************************************************
changed: [master]
changed: [node1]
changed: [node2]

PLAY RECAP *************************************************************************************************************************************************************
master                     : ok=8    changed=7    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
node1                      : ok=8    changed=7    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
node2                      : ok=8    changed=7    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[ec2-user@ip-10-0-1-20 ~]$
```

Create a single-host Kubernetes cluster

Initialize the master

```
[ec2-user@ip-10-0-1-20 ~]$ ssh -i "justakey.pem" ec2-user@10.0.2.60
Last login: Wed Aug 24 04:20:24 2022 from ip-10-0-1-20.ap-southeast-1.compute.internal

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-2-60 ~]$ sudo kubeadm init --pod-network-cidr=192.168.0.0/16
[init] Using Kubernetes version: v1.25.0
[preflight] Running pre-flight checks
        [WARNING FileExisting-tc]: tc not found in system path
[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'
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
```

Configure kubectl

```
kubeadm join 10.0.2.60:6443 --token 08b7r6.9l0hc9qs6tfq51tc \
        --discovery-token-ca-cert-hash sha256:7760720c6887bd331a499c4088099925e0cf5df8ae841
[ec2-user@ip-10-0-2-60 ~]$ mkdir -p $HOME/.kube
[ec2-user@ip-10-0-2-60 ~]$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
[ec2-user@ip-10-0-2-60 ~]$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
[ec2-user@ip-10-0-2-60 ~]$
```

Install Calico in the master node

Install the Tigera Calico operator and custom resource definitions

```
[ec2-user@ip-10-0-2-60 ~]$ kubectl create -f https://raw.githubusercontent.com/projectcalico/calico/v3.24.0/manifests/tigera-operator.yaml
namespace/tigera-operator created
customresourcedefinition.apiextensions.k8s.io/bgpconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/bgppeers.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/blockaffinities.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/caliconodestatuses.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/clusterinformations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/felixconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/globalnetworkpolicies.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/globalnetworksets.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/hostendpoints.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamblocks.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamconfigs.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamhandles.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ippools.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipreservations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/kubecontrollersconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/networkpolicies.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/networksets.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/apiservers.operator.tigera.io created
customresourcedefinition.apiextensions.k8s.io/imagesets.operator.tigera.io created
customresourcedefinition.apiextensions.k8s.io/installations.operator.tigera.io created
customresourcedefinition.apiextensions.k8s.io/tigerastatuses.operator.tigera.io created
serviceaccount/tigera-operator created
clusterrole.rbac.authorization.k8s.io/tigera-operator created
clusterrolebinding.rbac.authorization.k8s.io/tigera-operator created
deployment.apps/tigera-operator created
[ec2-user@ip-10-0-2-60 ~]$
```

Install Calico by creating the necessary custom resource

```
[ec2-user@ip-10-0-2-60 ~]$ kubectl create -f https://raw.githubusercontent.com/projectcalico/calico/v3.24.0/manifests/custom-resources.yaml

installation.operator.tigera.io/default created
apiserver.operator.tigera.io/default created
[ec2-user@ip-10-0-2-60 ~]$
```

Confirm the pods are running

```
Every 2,0s: kubectl get pods -n calico-system

NAME                                       READY   STATUS    RESTARTS   AGE
calico-kube-controllers-795d7b8659-pmdxl   1/1     Running   0          79s
calico-node-w7nxs                          1/1     Running   0          79s
calico-typha-86cb4877b6-q4p4r              1/1     Running   0          79s
csi-node-driver-tpjvh                      2/2     Running   0          37s
```

Join the nodes into the cluster

- Node1

```
[ec2-user@ip-10-0-4-44 ~]$ sudo kubeadm join 10.0.2.60:6443 --token 08b7r6.9l0hc9qs6tfq51tc --discovery-token-ca-cert-hash sha256:7760720c6887bd331a499c4088099925e0cf5df8ae841fe0734df7d3026e043b
[preflight] Running pre-flight checks
        [WARNING FileExisting-tc]: tc not found in system path
[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.

[ec2-user@ip-10-0-4-44 ~]$
```

- Node2

```
[ec2-user@ip-10-0-1-20 ~]$ ssh -i "justakey.pem" ec2-user@10.0.5.11
Last login: Wed Aug 24 04:20:24 2022 from ip-10-0-1-20.ap-southeast-1.compute.internal

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-5-11 ~]$ sudo kubeadm join 10.0.2.60:6443 --token 08b7r6.9l0hc9qs6tfq51tc --discovery-token-ca-cert-hash sha256:7760720c6887bd331a499c4088099925e0cf5df8ae841fe0734df7d3026e043b
[preflight] Running pre-flight checks
        [WARNING FileExisting-tc]: tc not found in system path
[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.

[ec2-user@ip-10-0-5-11 ~]$
```

Connect to master node to check if all the nodes are connected

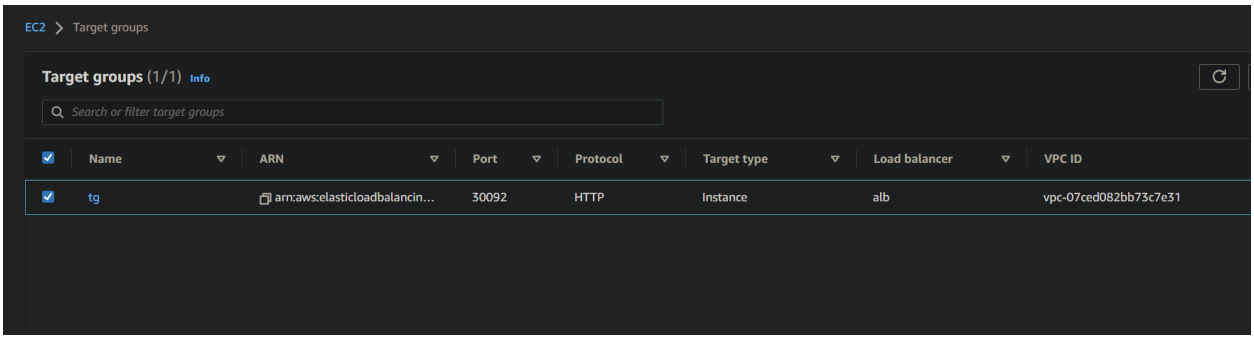
```
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-2-60 ~]$ kubectl get nodes
NAME                                           STATUS   ROLES           AGE    VERSION
ip-10-0-2-60.ap-southeast-1.compute.internal   Ready    control-plane   18m    v1.25.0
ip-10-0-4-44.ap-southeast-1.compute.internal   Ready    <none>          3m9s   v1.25.0
ip-10-0-5-11.ap-southeast-1.compute.internal   Ready    <none>          88s    v1.25.0
[ec2-user@ip-10-0-2-60 ~]$
```

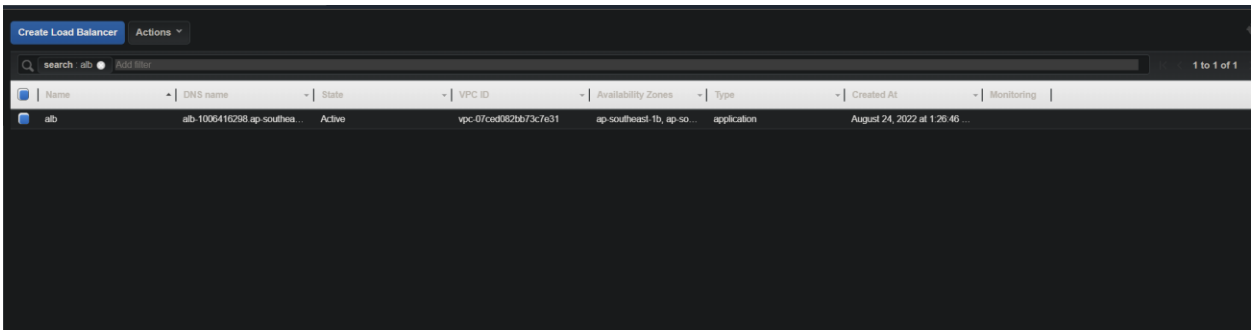
Deploy WordPress on the cluster

```
[ec2-user@ip-10-0-2-60 ~]$ nano kustomization.yaml
[ec2-user@ip-10-0-2-60 ~]$ nano mysql-deployment.yaml
[ec2-user@ip-10-0-2-60 ~]$ nano wordpress-deployment.yaml
[ec2-user@ip-10-0-2-60 ~]$ kubectl apply -k ./
secret/mysql-pass-72mh6dg77t created
service/wordpress created
service/wordpress-mysql created
persistentvolumeclaim/mysql-pv-claim created
persistentvolumeclaim/wp-pv-claim created
deployment.apps/wordpress created
deployment.apps/wordpress-mysql created
[ec2-user@ip-10-0-2-60 ~]$
```

```
[ec2-user@ip-10-0-2-60 ~]$ kubectl get all
NAME                                 READY   STATUS    RESTARTS   AGE
pod/wordpress-76b9d4cc49-bhvnj       1/1     Running   0          32s
pod/wordpress-mysql-6c8bfcc555-lx9tv   1/1     Running   0          32s

NAME                    TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
service/kubernetes      ClusterIP   10.96.0.1       <none>        443/TCP        106m
service/wordpress       NodePort    10.108.130.63   <none>        80:30092/TCP   33s
service/wordpress-mysql   ClusterIP   None            <none>        3306/TCP       33s

NAME                              READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/wordpress         1/1     1            1           33s
deployment.apps/wordpress-mysql   1/1     1            1           33s

NAME                                         DESIRED   CURRENT   READY   AGE
replicaset.apps/wordpress-76b9d4cc49         1         1         1       32s
replicaset.apps/wordpress-mysql-6c8bfcc555   1         1         1       32s
[ec2-user@ip-10-0-2-60 ~]$
```

Create loadbalancer





Final result