Install terraform on ubuntu

```
sudo apt-get update && sudo apt-get install -y gnupg software-properties-common

wget -O- https://apt.releases.hashicorp.com/gpg | \
gpg --dearmor | \
sudo tee /usr/share/keyrings/hashicorp-archive-keyring.gpg


echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] \
https://apt.releases.hashicorp.com $(lsb_release -cs) main" | \
sudo tee /etc/apt/sources.list.d/hashicorp.list

sudo apt update

sudo apt-get install terraform -y
```

Terraform file for launching argocd

- **vim unati.tf**
- Save and exit

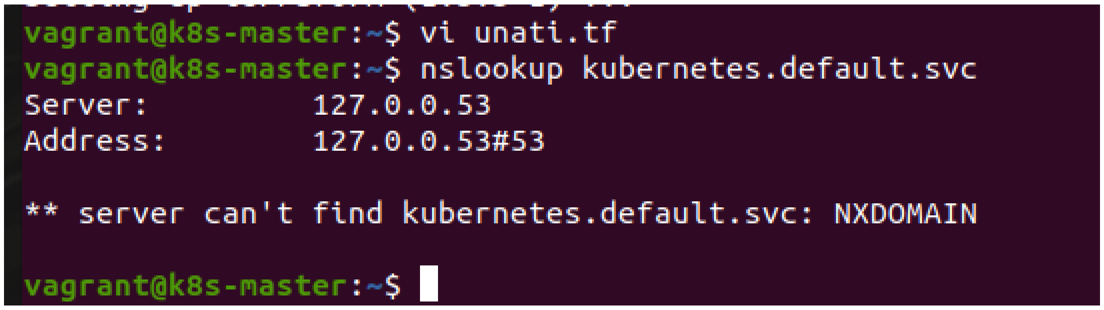
```
terraform {
 required_providers {
   helm = {
     source  = "hashicorp/helm"
     version = ">= 2.0.0, <= 3.0.0" # Update this to the appropriate version range
   }
   kubernetes = {
     source  = "hashicorp/kubernetes"
     version = ">= 1.0.0, <= 1.17.3" # Update this to the appropriate version range
   }
 }
}


provider "helm" {
 kubernetes {
   host = "https://kubernetes.default.svc"
   token = var.token
   cluster_ca_certificate = file(var.cluster_ca_certificate)
 }
}

variable "token" {
 type    = string
}

variable "cluster_ca_certificate" {
 type    = string
}

resource "helm_release" "argocd" {
 name  = "argocd"

 repository       = "https://argoproj.github.io/argo-helm"
 chart            = "argo-cd"
 namespace        = "argocd"
 version          = "4.9.7"
 create_namespace = true

 values = [
   <<-EOT
   data:
     secretToken: ${var.token}

   users:
     - name: admin
       password: password
   EOT
 ]
}
```

In above file look for below section code:

```
provider "helm" {
  kubernetes {
    host = "https://kubernetes.default.svc"
    token = var.token
    cluster_ca_certificate = file(var.cluster_ca_certificate)
  }
}
```

It is going to connect to the host https://kubernetes.default.svc

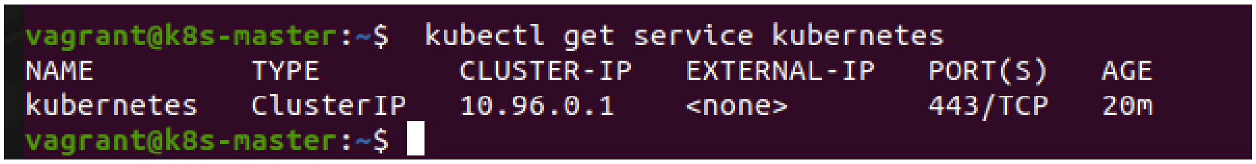
- Check whether that host reachable or not
- nslookup kubernetes.default.svc



```
vagrant@k8s-master:~$ vi unati.tf
vagrant@k8s-master:~$ nslookup kubernetes.default.svc
Server:         127.0.0.53
Address:        127.0.0.53#53

** server can't find kubernetes.default.svc: NXDOMAIN

vagrant@k8s-master:~$
```

No domain existed >? Lets find out cluster ip

- kubectl get service kubernetes



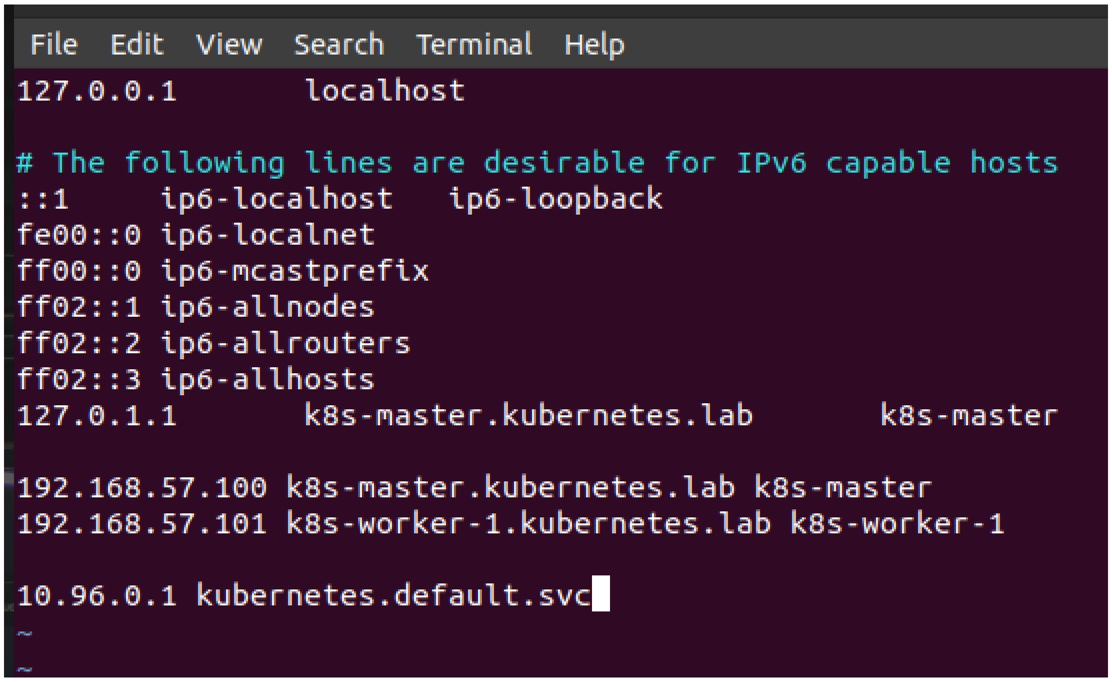
```
vagrant@k8s-master:~$  kubectl get service kubernetes
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   20m
vagrant@k8s-master:~$
```

– ok now we have two option either we change host in the code with this ip or add this domain with ip in /etc/hosts file

Lets go with 2nd method:

- Cluster ip is : 10.96.0.1
- Add below line in /etc/hosts
- vi /etc/hosts

10.96.0.1 kubernetes.default.svc

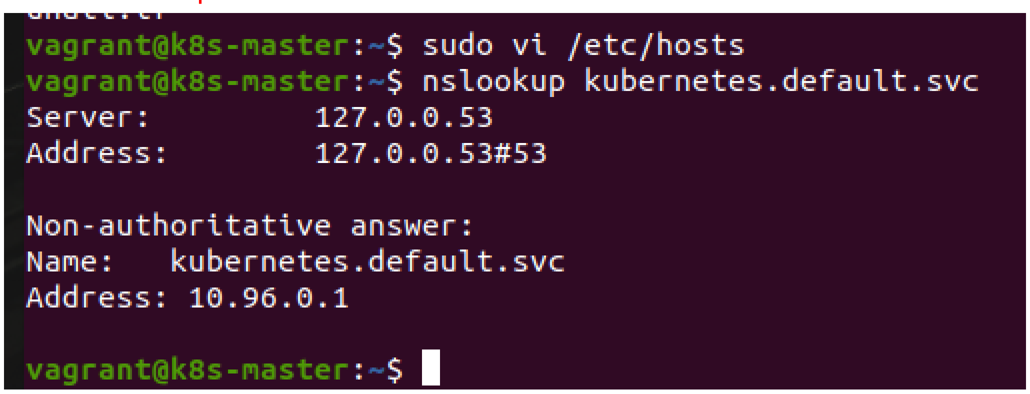


```
127.0.0.1       localhost

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost   ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
127.0.1.1       k8s-master.kubernetes.lab       k8s-master

192.168.57.100 k8s-master.kubernetes.lab k8s-master
192.168.57.101 k8s-worker-1.kubernetes.lab k8s-worker-1

10.96.0.1 kubernetes.default.svc
~
~
```

- Save , exit

Now check with nslookup :

- nslookup kubernetes.default.svc



```
vagrant@k8s-master:~$ sudo vi /etc/hosts
vagrant@k8s-master:~$ nslookup kubernetes.default.svc
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   kubernetes.default.svc
Address: 10.96.0.1

vagrant@k8s-master:~$
```

Looks good 🙂

Next , we have fresh installation of k8s cluster, by default we dont have service acc.. So lets create for us

- vi service-unati.yml

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: unati
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: unati-cluster-role-binding
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: unati
  namespace: default
---
apiVersion: v1
kind: Secret
metadata:
  name: unati-token
  annotations:
    kubernetes.io/service-account.name: unati
type: kubernetes.io/service-account-token
```

- Save ,exit

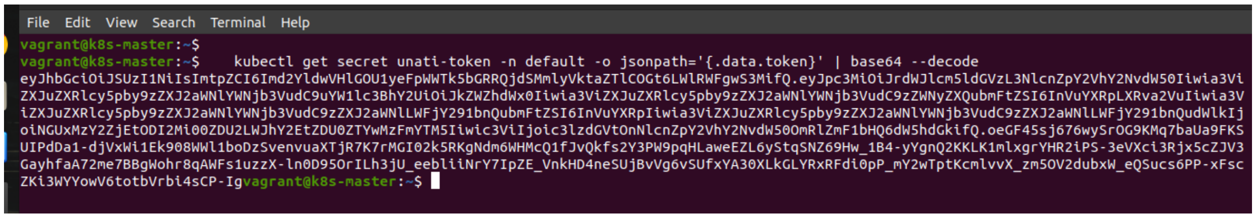
Apply this file to create service account

- kubectl apply -f service-unati.yml

Retrieve secret token and cluster certificate :

For token type below command :

- kubectl get secret unati-token -n default -o jsonpath='{.data.token}' | base64 --decode



```
vagrant@k8s-master:~$
vagrant@k8s-master:~$    kubectl get secret unati-token -n default -o jsonpath='{.data.token}' | base64 --decode
eyJhbGciOiJSUzI1NiIsImtpZCI6Imd2YldwVHlGOU1yeFpWWTk5bGRRQjdSMmlyVktaZTlCOGt6LWlRWFgwS3MifQ.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3Vi
ZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9uYW1lc3BhY2UiOiJkZWZhdWx0Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZWNyZXQubmFtZSI6InVuYXRpLXRva2VuIiwia3V
iZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZXJ2aWNlLWFjY291bnQubmFtZSI6InVuYXRpIiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZXJ2aWNlLWFjY291bnQudWlkIj
oiNGUxMzY2ZjEtODI2Mi00ZDU2LWJhY2EtZDU0ZTYwMzFmYTM5Iiwic3ViIjoic3lzdGVtOnNlcnZpY2VhY2NvdW50OmRlZmF1bHQ6dW5hdGkifQ.oeGF45sj676wySrOG9KMq7baUa9FKS
UIPdDa1-djVxWi1Ek908WWl1boDzSvenvuaXTjR7K7rMGI02k5RKgNdm6WHMcQ1fJvQkfs2Y3PW9pqHLaweEZL6yStqSNZ69Hw_1B4-yYgnQ2KKLK1mlxgrYHR2iPS-3eVXci3Rjx5cZJV3
GayhfaA72me7BBgWohr8qAWFs1uzzX-ln0D95OrILh3jU_eebliiNrY7IpZE_VnkHD4neSUjBvVg6vSUfxYA30XLkGLYRxRFdi0pP_mY2wTptKcmlvvX_zm5OV2dubxW_eQSucs6PP-xFsc
ZKi3WYYowV6totbVrbi4sCP-Igvagrant@k8s-master:~$
```

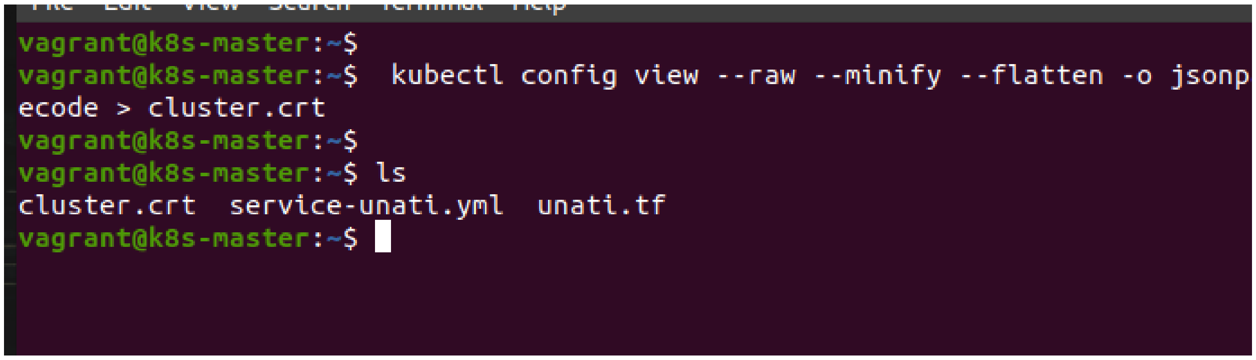
Copy token obtained and save it somewhere , we need it later

My token is:

eyJhbGciOiJSUzI1NiIsImtpZCI6Imd2YldwVHlGOU1yeFpWWTk5bGRRQjdSMmlyVktaZTlCOGt6LWlRWFgwS3MifQ.eyJ
pc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9uYW1lc3BhY2
UiOiJkZWZhdWx0Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZWNyZXQubmFtZSI6InVuYXRpLXRva2VuIiwia
3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZXJ2aWNlLWFjY291bnQubmFtZSI6InVuYXRpIiwia3ViZXJuZXRlcy5pby
9zZXJ2aWNlYWNjb3VudC9zZXJ2aWNlLWFjY291bnQudWlkIjoiNGUxMzY2ZjEtODI2Mi00ZDU2LWJhY2EtZDU0ZTYwM
zFmYTM5Iiwic3ViIjoic3lzdGVtOnNlcnZpY2VhY2NvdW50OmRlZmF1bHQ6dW5hdGkifQ.oeGF45sj676wySrOG9KMq7
baUa9FKSUIPdDa1-djVxWi1Ek908WWl1boDzSvenvuaXTjR7K7rMGI02k5RKgNdm6WHMcQ1fJvQkfs2Y3PW9pqHLaw
eEZL6yStqSNZ69Hw_1B4-yYgnQ2KKLK1mlxgrYHR2iPS-3eVXci3Rjx5cZJV3GayhfaA72me7BBgWohr8qAWFs1uzzX-ln0
D95OrILh3jU_eebliiNrY7IpZE_VnkHD4neSUjBvVg6vSUfxYA30XLkGLYRxRFdi0pP_mY2wTptKcmlvvX_zm5OV2dubxW_
eQSucs6PP-xFscZKi3WYYowV6totbVrbi4sCP-Ig

For cert type below command :

- kubectl config view --raw --minify --flatten -o jsonpath='{.clusters[].cluster.certificate-authority-data}' | base64 --decode > cluster.crt



```
vagrant@k8s-master:~$
vagrant@k8s-master:~$  kubectl config view --raw --minify --flatten -o jsonp
ecode > cluster.crt
vagrant@k8s-master:~$
vagrant@k8s-master:~$ ls
cluster.crt  service-unati.yml  unati.tf
vagrant@k8s-master:~$
```

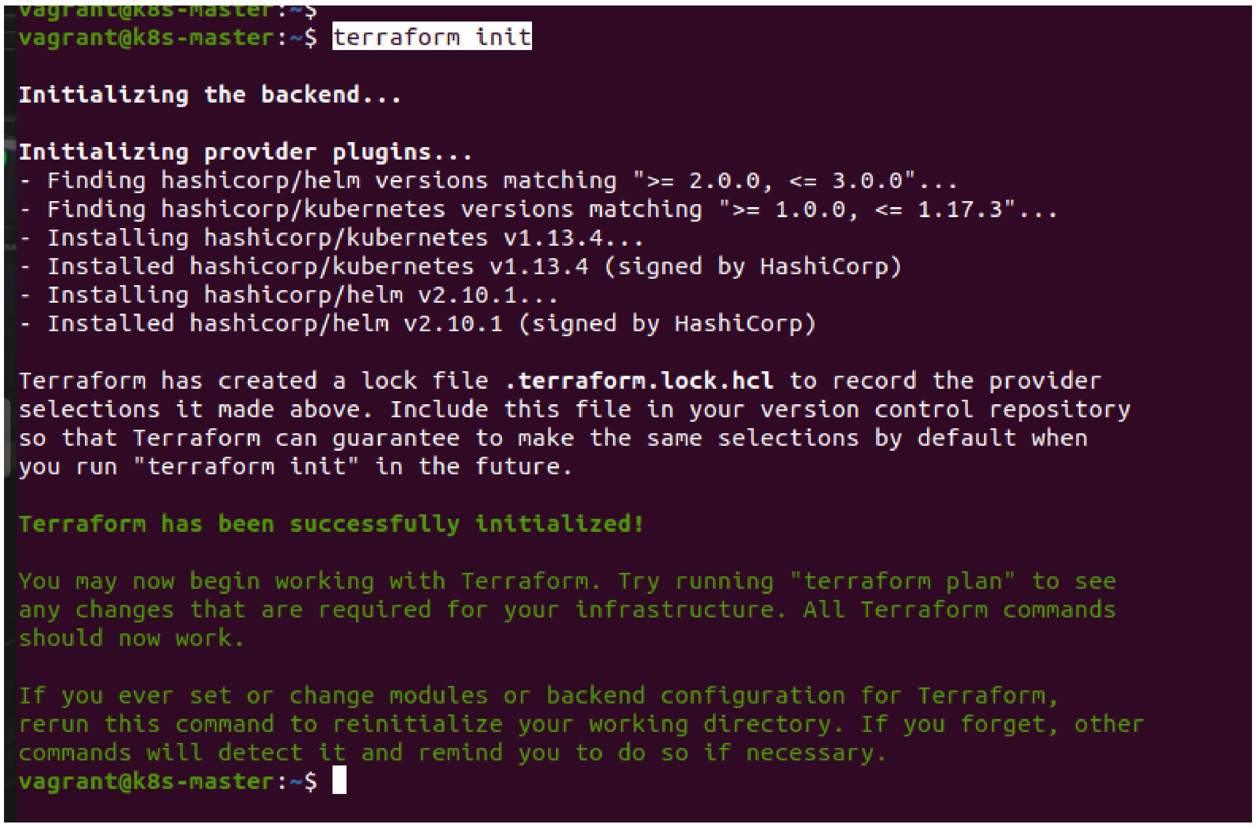
- It will create cert file name : cluster.crt

We have enough things to deploy argocd

- Cluster certificate
- Token

Lets go further,

- terraform init

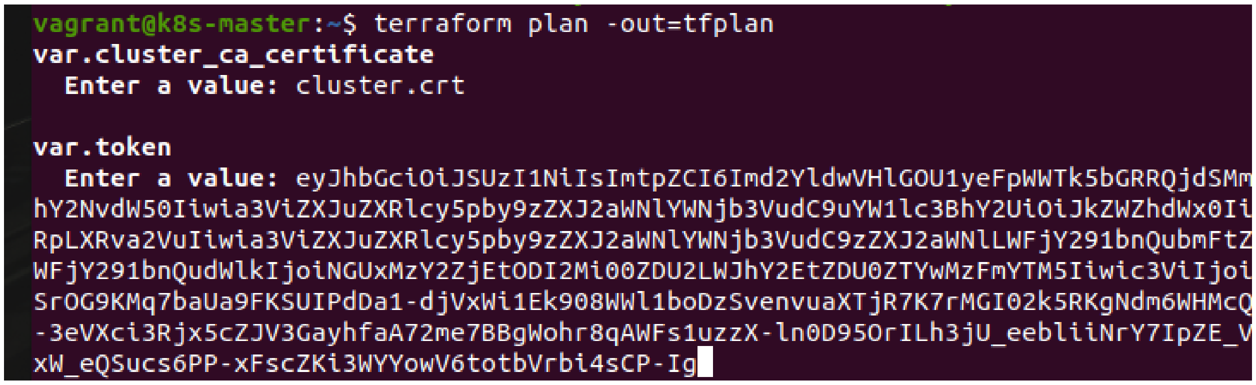


```
vagrant@k8s-master:~$
vagrant@k8s-master:~$ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding hashicorp/helm versions matching ">= 2.0.0, <= 3.0.0"...
- Finding hashicorp/kubernetes versions matching ">= 1.0.0, <= 1.17.3"...
- Installing hashicorp/kubernetes v1.13.4...
- Installed hashicorp/kubernetes v1.13.4 (signed by HashiCorp)
- Installing hashicorp/helm v2.10.1...
- Installed hashicorp/helm v2.10.1 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

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.
vagrant@k8s-master:~$
```

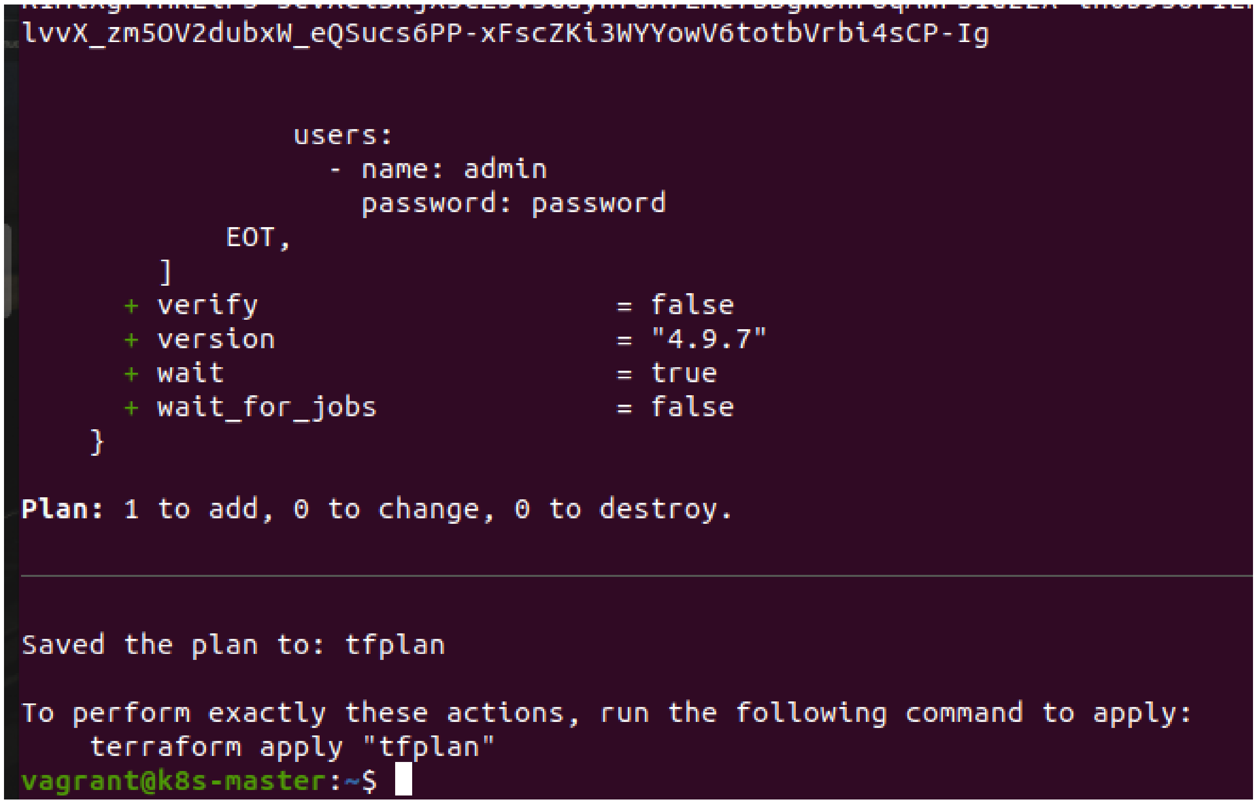
- terraform plan -out=tfplan

The `terraform plan -out=tfplan` command generates a plan of the changes that Terraform will make to your infrastructure. The plan is saved in a file named `tfplan`.

- It will ask you for cluster cert , provide path of cert, in my case it is in same dir
    - Previously Cert name is : cluster.crt
    - Provide token : copy the contents and paste there



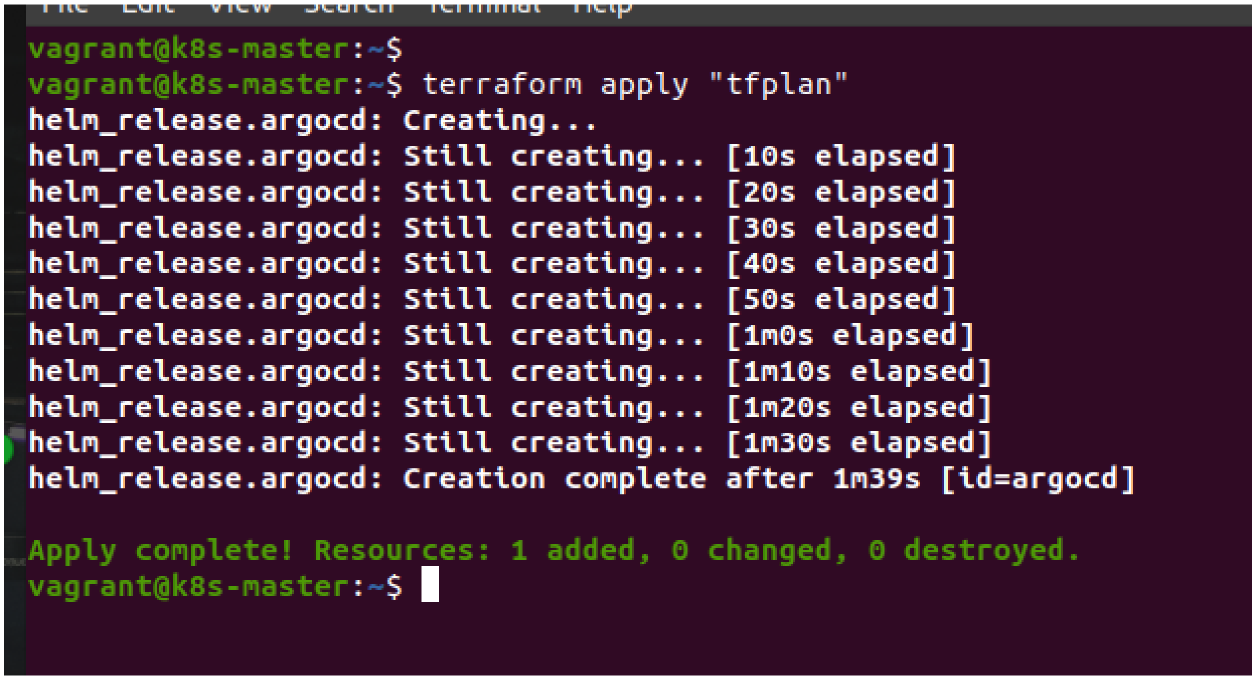
```
vagrant@k8s-master:~$ terraform plan -out=tfplan
var.cluster_ca_certificate
  Enter a value: cluster.crt

var.token
  Enter a value: eyJhbGciOiJSUzI1NiIsImtpZCI6Imd2YldwVHlGOU1yeFpWWTk5bGRRQjdSMm
hY2NvdW50Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9uYW1lc3BhY2UiOiJkZWZhdWx0Ii
RpLXRva2VuIiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZXJ2aWNlLWFjY291bnQubmFtZ
WFjY291bnQudWlkIjoiNGUxMzY2ZjEtODI2Mi00ZDU2LWJhY2EtZDU0ZTYwMzFmYTM5Iiwic3ViIjoi
SrOG9KMq7baUa9FKSUIPdDa1-djVxWi1Ek908WWl1boDzSvenvuaXTjR7K7rMGI02k5RKgNdm6WHMcQ
-3eVXci3Rjx5cZJV3GayhfaA72me7BBgWohr8qAWFs1uzzX-ln0D95OrILh3jU_eebliiNrY7IpZE_V
xW_eQSucs6PP-xFscZKi3WYYowV6totbVrbi4sCP-Ig
```



```
lvvX_zm5OV2dubxW_eQSucs6PP-xFscZKi3WYYowV6totbVrbi4sCP-Ig


                users:
                  - name: admin
                    password: password
            EOT,
        ]
      + verify                          = false
      + version                         = "4.9.7"
      + wait                            = true
      + wait_for_jobs                   = false
    }

Plan: 1 to add, 0 to change, 0 to destroy.

─────────────────────────────────────────────────────────────────────────

Saved the plan to: tfplan

To perform exactly these actions, run the following command to apply:
    terraform apply "tfplan"
vagrant@k8s-master:~$
```

- terraform apply "tfplan"

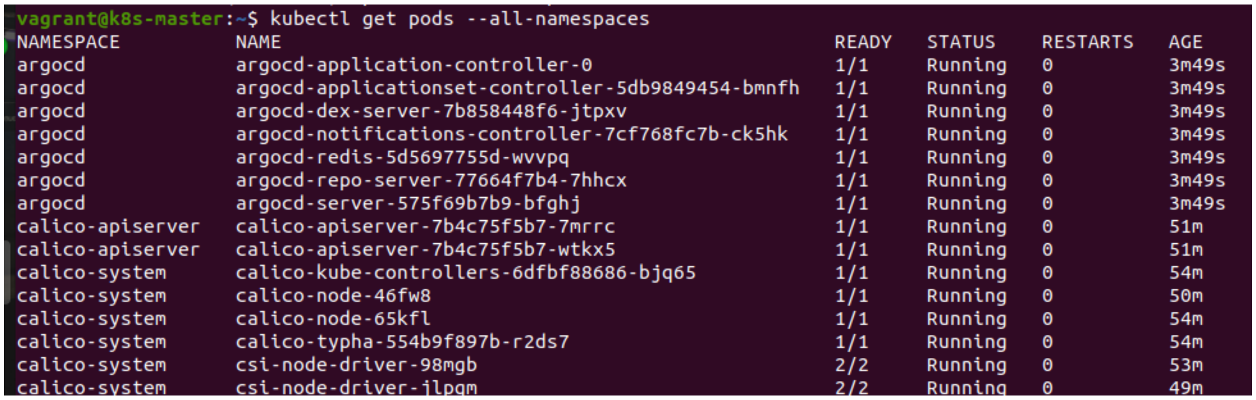
The `terraform apply "tfplan"` command applies the plan that was generated by the `terraform plan -out=tfplan` command. The plan is read from the file named `tfplan`.



```
vagrant@k8s-master:~$
vagrant@k8s-master:~$ terraform apply "tfplan"
helm_release.argocd: Creating...
helm_release.argocd: Still creating... [10s elapsed]
helm_release.argocd: Still creating... [20s elapsed]
helm_release.argocd: Still creating... [30s elapsed]
helm_release.argocd: Still creating... [40s elapsed]
helm_release.argocd: Still creating... [50s elapsed]
helm_release.argocd: Still creating... [1m0s elapsed]
helm_release.argocd: Still creating... [1m10s elapsed]
helm_release.argocd: Still creating... [1m20s elapsed]
helm_release.argocd: Still creating... [1m30s elapsed]
helm_release.argocd: Creation complete after 1m39s [id=argocd]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
vagrant@k8s-master:~$
```

- Check arogocd pods

- kubectl get pods --all-namespaces



```
vagrant@k8s-master:~$ kubectl get pods --all-namespaces
NAMESPACE          NAME                                                READY   STATUS    RESTARTS   AGE
argocd             argocd-application-controller-0                     1/1     Running   0          3m49s
argocd             argocd-applicationset-controller-5db9849454-bmnfh   1/1     Running   0          3m49s
argocd             argocd-dex-server-7b858448f6-jtpxv                  1/1     Running   0          3m49s
argocd             argocd-notifications-controller-7cf768fc7b-ck5hk    1/1     Running   0          3m49s
argocd             argocd-redis-5d5697755d-wvvpq                       1/1     Running   0          3m49s
argocd             argocd-repo-server-77664f7b4-7hhcx                  1/1     Running   0          3m49s
argocd             argocd-server-575f69b7b9-bfghj                      1/1     Running   0          3m49s
calico-apiserver   calico-apiserver-7b4c75f5b7-7mrrc                   1/1     Running   0          51m
calico-apiserver   calico-apiserver-7b4c75f5b7-wtkx5                   1/1     Running   0          51m
calico-system      calico-kube-controllers-6dfbf88686-bjq65            1/1     Running   0          54m
calico-system      calico-node-46fw8                                   1/1     Running   0          50m
calico-system      calico-node-65kfl                                   1/1     Running   0          54m
calico-system      calico-typha-554b9f897b-r2ds7                       1/1     Running   0          54m
calico-system      csi-node-driver-98mgb                               2/2     Running   0          53m
calico-system      csi-node-driver-jlpqm                               2/2     Running   0          49m
```