

Kubernetes Task

Task Description:

Setup minikube at your local and explore creating namespaces (Go through official documentation).

Techstacks needs to be used :

- Vbox, WSL
- Docker
- Minikube
- Kubectl

If the Local system has any issues you can use AWS.

- AWS EC2 (t2.medium)

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> choco install minikube
chocolatey v2.6.0
Installing the following packages:
minikube
By installing, you accept licenses for the packages.
Downloading package from source 'https://community.chocolatey.org/api/v2/'
Progress: Downloading kubernetes-cli 1.34.2... 100%

kubernetes-cli v1.34.2 [Approved]
kubernetes-cli package files install completed. Performing other installation steps.
The package kubernetes-cli wants to run 'chocolateyInstall.ps1'.
Note: If you don't run this script, the installation will fail.
Note: To confirm automatically next time, use '-y' or consider:
choco feature enable -n allowGlobalConfirmation
Do you want to run the script?([Y]es/[A]ll scripts/[N]o/[P]rint): Y

Extracting 64-bit C:\ProgramData\chocolatey\lib\kubernetes-cli\tools\kubernetes-client-windows-amd64.tar.gz to C:\ProgramData\chocolatey\lib\kubernetes-cli\tools...
C:\ProgramData\chocolatey\lib\kubernetes-cli\tools
Extracting 64-bit C:\ProgramData\chocolatey\lib\kubernetes-cli\tools\kubernetes-client-windows-amd64.tar to C:\ProgramData\chocolatey\lib\kubernetes-cli\tools...
C:\ProgramData\chocolatey\lib\kubernetes-cli\tools
Environment Vars (like PATH) have changed. Close/reopen your shell to
see the changes (or in powershell/cmd.exe just type 'refreshenv').
ShimGen has successfully created a shim for kubectl-convert.exe
ShimGen has successfully created a shim for kubectl.exe
The install of kubernetes-cli was successful.
  Deployed to 'C:\ProgramData\chocolatey\lib\kubernetes-cli\tools'
Downloading package from source 'https://community.chocolatey.org/api/v2/'
Progress: Downloading Minikube 1.37.0... 100%

Minikube v1.37.0 [Approved]
Minikube package files install completed. Performing other installation steps.
ShimGen has successfully created a shim for minikube.exe
The install of Minikube was successful.
  Deployed to 'C:\ProgramData\chocolatey\lib\Minikube'

Chocolatey installed 2/2 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).

Enjoy using Chocolatey? Explore more amazing features to take your
experience to the next level at
https://chocolatey.org/compare
PS C:\WINDOWS\system32> minikube --version
Error: unknown flag: --version
See 'minikube --help' for usage.
PS C:\WINDOWS\system32> minikube version
minikube version: v1.37.0
commit: 65318f4cffff9c12cc87ec9eb8f4cdd57b25047f3
```

```

PS C:\Users\sheer\Documents\kubernetes\Kubernetes> minikube status
🔍 Profile "minikube" not found. Run "minikube profile list" to view all profiles.
👉 To start a cluster, run: "minikube start"
PS C:\Users\sheer\Documents\kubernetes\Kubernetes> minikube start
🚀 minikube v1.37.0 on Microsoft Windows 11 Home Single Language 10.0.26200.7171 Build 26200.7171
🔧 Automatically selected the docker driver
🔧 Using Docker Desktop driver with root privileges
👉 Starting "minikube" primary control-plane node in "minikube" cluster
📡 Pulling base image v0.0.48 ...
📦 Downloading Kubernetes v1.34.0 preload ...
> preloaded-images-k8s-v18-v1...: 337.07 MiB / 337.07 MiB 100.00% 2.70 Mi
> gcr.io/k8s-minikube/kicbase...: 488.52 MiB / 488.52 MiB 100.00% 1.85 Mi
🔥 Creating docker container (CPUs=2, Memory=3072MB) ...
❗ Failing to connect to https://registry.k8s.io/ from inside the minikube container
💡 To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
📦 Preparing Kubernetes v1.34.0 on Docker 28.4.0 ...
🔗 Configuring bridge CNI (Container Networking Interface) ...
🔍 Verifying Kubernetes components...
  ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟 Enabled addons: storage-provisioner, default-storageclass
👉 Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
PS C:\Users\sheer\Documents\kubernetes\Kubernetes> minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured

```

```

Kubernetes > networkpol > backend.yml > {} metadata > {} labels > abc app
io.k8s.api.core.v1.Pod (v1@pod.json)
1  apiVersion: v1
2  kind: Pod
3  metadata:
4    name: backend-pod
5    namespace: net-demo
6    labels:
7      app: backend-pod
8  spec:
9    containers:
10     - name: backend-pod
11       image: nginx:alpine
12       ports:
13         - containerPort: 80
14

```

```

Kubernetes > networkpol > frontend.yml > {} spec > {} containers > {} 0 > {} ports > {} 0
io.k8s.api.core.v1.Pod (v1@pod.json)
1  apiVersion: v1
2  kind: Pod
3  metadata:
4    name: frontend-service
5    namespace: net-demo
6    labels:
7      app.kubernetes.io/name: frontend-service
8  spec:
9    containers:
10     - name: frontend-service
11       image: httpd:alpine
12       ports:
13         - containerPort: 81
14

```

```

PS C:\Users\sheer\Documents\kubernetes\Kubernetes\networkpol> kubectl config get-context --current --namespace=net-demo
error: unknown flag: --current
See 'kubectl config --help' for usage.
PS C:\Users\sheer\Documents\kubernetes\Kubernetes\networkpol> kubectl create namespace net-demo
namespace/net-demo created
PS C:\Users\sheer\Documents\kubernetes\Kubernetes\networkpol> kubectl apply -f frontend.yml -n net-demo
pod/frontend-service created
PS C:\Users\sheer\Documents\kubernetes\Kubernetes\networkpol> kubectl apply -f backend.yml -n net-demo
pod/backend-pod created
PS C:\Users\sheer\Documents\kubernetes\Kubernetes\networkpol> kubectl get pods -n kube-system
NAME                                READY   STATUS    RESTARTS   AGE
coredns-66bc5c9577-8vw66            1/1     Running   0           5m9s
etcd-minikube                       1/1     Running   0           5m16s
kube-apiserver-minikube             1/1     Running   0           5m17s
kube-controller-manager-minikube    1/1     Running   0           5m16s
kube-proxy-nw6v7                    1/1     Running   0           5m9s
kube-scheduler-minikube             1/1     Running   0           5m15s
storage-provisioner                 1/1     Running   1 (5m6s ago) 5m8s
PS C:\Users\sheer\Documents\kubernetes\Kubernetes\networkpol> kubectl get pods -n net-demo
NAME            READY   STATUS    RESTARTS   AGE
backend-pod     1/1     Running   0           28s
frontend-service 1/1     Running   0           46s
PS C:\Users\sheer\Documents\kubernetes\Kubernetes\networkpol> kubectl get svc
NAME      TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
kubernetes ClusterIP  10.96.0.1     <none>        443/TCP      6m10s
PS C:\Users\sheer\Documents\kubernetes\Kubernetes\networkpol> kubectl expose pod frontend-service --type=ClusterIP --port=81
Error from server (NotFound): pods "frontend-service" not found
PS C:\Users\sheer\Documents\kubernetes\Kubernetes\networkpol> kubectl expose pod frontend-service --type=ClusterIP --port=81 -n net-demo
service/frontend-service exposed
PS C:\Users\sheer\Documents\kubernetes\Kubernetes\networkpol> kubectl expose pod backend-pod --type=ClusterIP --port=80 -n net-demo
service/backend-pod exposed
PS C:\Users\sheer\Documents\kubernetes\Kubernetes\networkpol> kubectl get svc -n net-demo
NAME            TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
backend-pod     ClusterIP   10.97.247.142 <none>        80/TCP      13s
frontend-service ClusterIP   10.104.118.129 <none>        81/TCP      75s

```