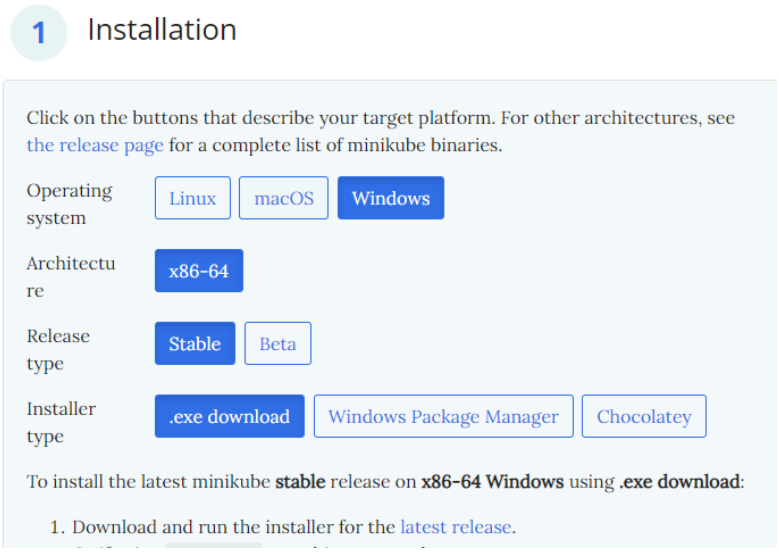
Cloud Computing

Introduction to Kubernetes

Exercise One - Create a Cluster (2%)

Task 1: I installed Minikube by downloading the setup file.exe. You can use the website [minikube website] to install, scroll down and click on latest release.



I then run the command: **minikube start** in my terminal to start my cluster which also created a container.

```
C:\Users\waleed>minikube start
* minikube v1.27.0 on Microsoft Windows 10 Pro 10.0.19043 Build 19043
! Kubernetes 1.25.0 has a known issue with resolv.conf. minikube is using a workaround that should work for most use cas
es.
! For more information, see: https://github.com/kubernetes/kubernetes/issues/112135
* Automatically selected the docker driver
* Using Docker Desktop driver with root privileges
* Starting control plane node minikube in cluster minikube
* Pulling base image ...
* Downloading Kubernetes v1.25.0 preload ...
    > preloaded-images-k8s-v18-v1...:  385.37 MiB / 385.37 MiB  100.00% 5.35 Mi
    > gcr.io/k8s-minikube/kicbase:  386.76 MiB / 386.76 MiB  100.00% 4.60 MiB p
    > gcr.io/k8s-minikube/kicbase:  0 B [_____________________] ?% ? p/s 1m1s
* Creating docker container (CPUs=2, Memory=4000MB) ...
* Preparing Kubernetes v1.25.0 on Docker 20.10.17 ...
  - Generating certificates and keys ...
  - Booting up control plane ...
  - Configuring RBAC rules ...
* 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
```

Task 2: I then ran the command **kubectl create deployment hello-minikube –image=docker.io/nginx:1.23** which created a deployment inside the cluster. Deploy the nginx:1.23 Docker image on a Kubernetes cluster, by creating a Deployment using the kubectl command:

```
C:\Users\waleed\Desktop\PRACTICAL3_waleed_wazir_19396951\ex1>kubectl create deployment hello-minikube --image=docker.io/nginx:1.23
deployment.apps/hello-minikube created
```

I then used the command **kubectl get pods,** this command allows us to see any pods inside the cluster, here we can see that there is one pod running in the container.

```
C:\Users\waleed>kubectl get pods
NAME                              READY   STATUS    RESTARTS   AGE
hello-minikube-65dc654df9-djsz4   1/1     Running   0          23s
```

I then used the command **Kubectl get deployments** which allowed me to see the deployments created inside the container.

```
C:\Users\waleed>kubectl get deployments
NAME             READY   UP-TO-DATE   AVAILABLE   AGE
hello-minikube   1/1     1            1           3m19s
```

I then ran the command **kubectl get nodes –o wide** which gave us the clusters runtime.

```
C:\Users\waleed>kubectl get nodes -o wide
NAME       STATUS   ROLES           AGE   VERSION   INTERNAL-IP    EXTERNAL-IP   OS-IMAGE             KERNEL-VERSION                     CONTAINER-RUNTIME
minikube   Ready    control-plane   12m   v1.25.0   192.168.49.2   <none>        Ubuntu 20.04.5 LTS   5.10.16.3-microsoft-standard-WSL2   docker://20.10.17
```

```
CONTAINER-RUNTIME
docker://20.10.17
```