**Create VM:**

gcloud compute zones **list**

gcloud config **set** compute/zone <zone>

**Cloud shell - launch a new VM instance**

gcloud compute instances **create** ubuntu \

*--image-project ubuntu-os-cloud \*

*--image ubuntu-1604-xenial-v20160420c*

### Cloud shell - log into the VM instance

**gcloud** compute ssh ubuntu

**Install Docker CE (**[**https://docs.docker.com/install/linux/docker-ce/ubuntu/#set-up-the-repository**](https://docs.docker.com/install/linux/docker-ce/ubuntu/#set-up-the-repository)**)**

sudo apt-get update

sudo apt-get install \

apt-transport-https \

ca-certificates \

curl \

software-properties-common

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add –

sudo add-apt-repository \

"deb [arch=amd64] https://download.docker.com/linux/ubuntu \

$(lsb\_release -cs) \

stable"

sudo apt-get update

sudo apt-get install docker-ce

**Compile Samples:**

git clone <https://github.com/dotnet/dotnet-docker-samples>

cd dotnet-docker-samples/aspnetapp/

sudo docker build -t aspnetapp .

**Create image and run:**

docker run -d -p 8080:80 --name myapp aspnetapp

**ToTest example**

**curl** -H "Authorization: Bearer $TOKEN" http://127.0.0.1:10080/secure

**Create empty webapi**

dotnet new web

Link “how to build ASP container”: <https://docs.docker.com/engine/examples/dotnetcore/#build-and-run-the-docker-image>

**Docker**

sudo docker build -t monolith:1.0.0 .

sudo docker tag monolith:1.0.0 smolsky/ud615\_dotnetcore\_monolith:1.0.0

docker login

docker push smolsky/ud615\_dotnetcore\_monolith:1.0.0

**kubernets**

**Create k8s cluster:**

**gcloud** container clusters create k0 --zone europe-west1-d

**Create pod:**

kubectl **create** -**f** monolith.yaml

**Get pods / describe**:

kubectl **get** pods

**kubectl** describe pods monolith

kubectl delete pod monolith

**Interacting with pods (set up port-forwarding)**

kubectl port-forward monolith 10080:80

**Debug Pod**

**kubectl** logs monolith

**kubectl** logs -f monolith

kubectl logs -c nginx secure-monolith

kubectl exec monolith --stdin --tty -c monolith /bin/sh

ping -c 3 google.com

exit

**Secrets**

>ls tls

ca-key.pem ca.pem cert.pem key.pem ssl-extensions-x509.cnf **update-tls.sh**

>

to create the tls-certs secret from the TLS certificates stored under the tls directory:

kubectl create secret generic tls-certs --**from**-**file**=tls/

kubectl will create a key for each file in the tls directory under the tls-certs secret bucket. Use the kubectl describe command to verify that:

**kubectl** describe secrets tls-certs

**Configs**

Next we need to create a configmap entry for the proxy.conf nginx configuration file using the kubectl create configmap command:

proxy.conf

Use the kubectl describe configmap command to get more details about the nginx-proxy-conf configmap entry:

**kubectl** describe configmap nginx-proxy-conf

### Create the secure-monolith Pod using kubectl.

kubectl **create** -**f** pods/secure-monolith.yaml

kubectl **get** pods secure-monolith

kubectl port-forward secure-monolith 10443:443

curl *--cacert tls/ca.pem https://127.0.0.1:10443*

**Create Service**

kubectl create -f services/monolith.yaml

gcloud compute firewall-rules create allow-monolith-nodeport --allow=tcp:31000

kubectl describe service monolith | grep Endpoints