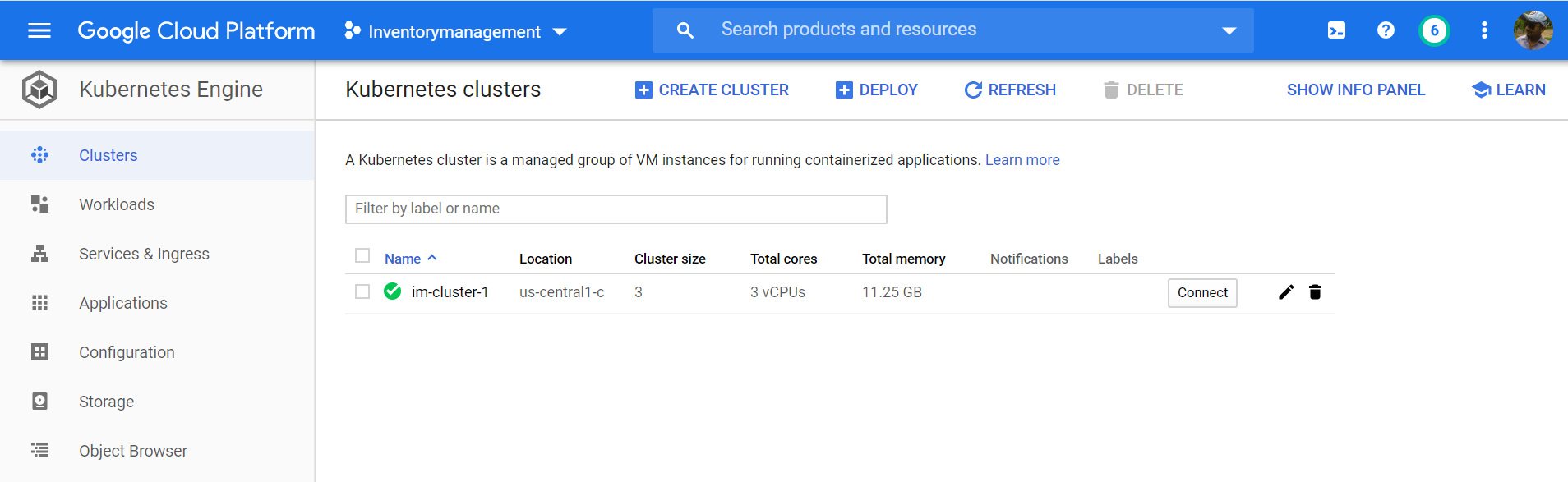
**1. Created a project with name InventoryManagementApp**

**2 .Created a GKE cluster with name im-cluster-1**



Connect google cloud CLI with cluster:

>gcloud container clusters get-credentials im-cluster-1 --zone us-central1-c --project inventorymanagement-281211

**3. OrderAPI (Service type ClusterIP)**

* Created a docker image for OrderAPI using the below command from the project directory

docker build -t ashwanitaneja88/order\_api\_dimg:latest .

* docker login -u <user> -p <pwd>
* Push docker image to Container Registry

docker push ashwanitaneja88/order\_api\_dimg:latest

* Create pods using orders-api-deployment.yaml

kubectl apply -f orders-api-deployment.yaml

kubectl get pods (used to verify the pods)

* Created Service with type ClusterIP using orders-api-service.yaml

kubectl apply -f orders-api-service.yaml

kubectl get services (used to check the services)

**4. Deployed single instance SQL Server 2017 on GKE**

* Created a Kubernetes secret

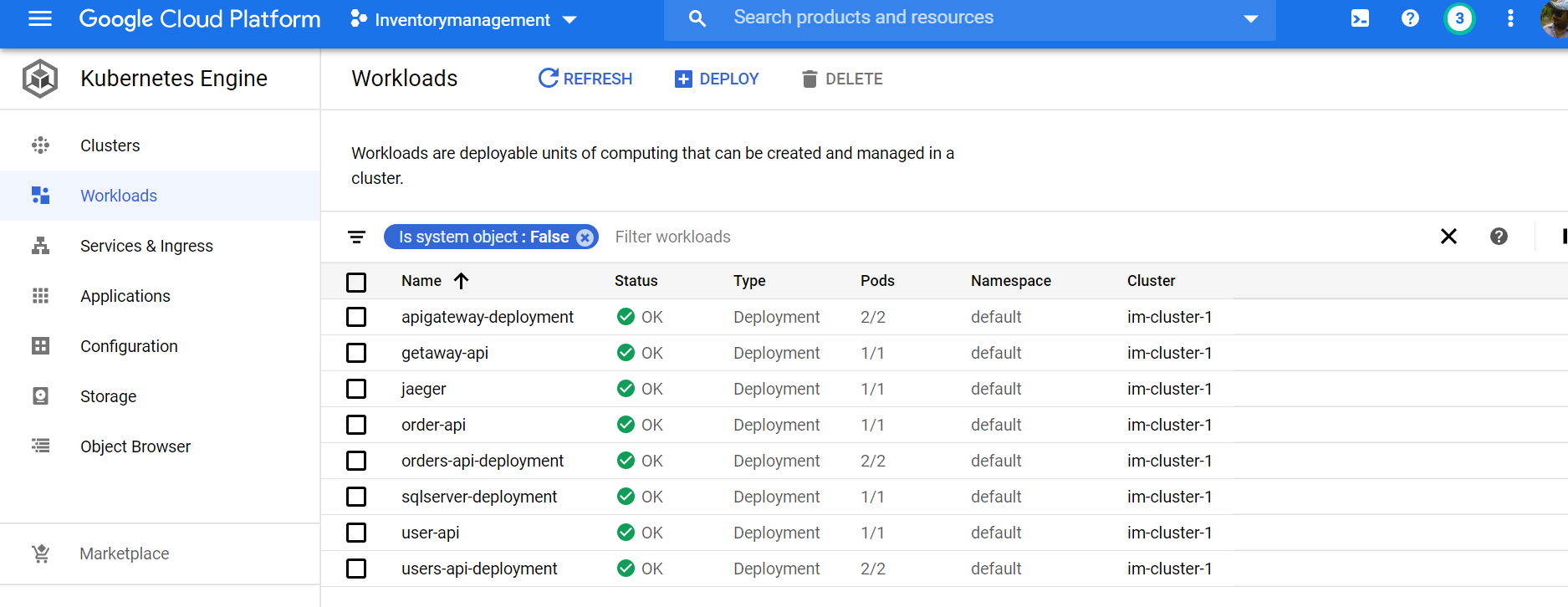
kubectl create secret generic sqlserver-secret   --from-literal=SA\_PASSWORD="password@321”

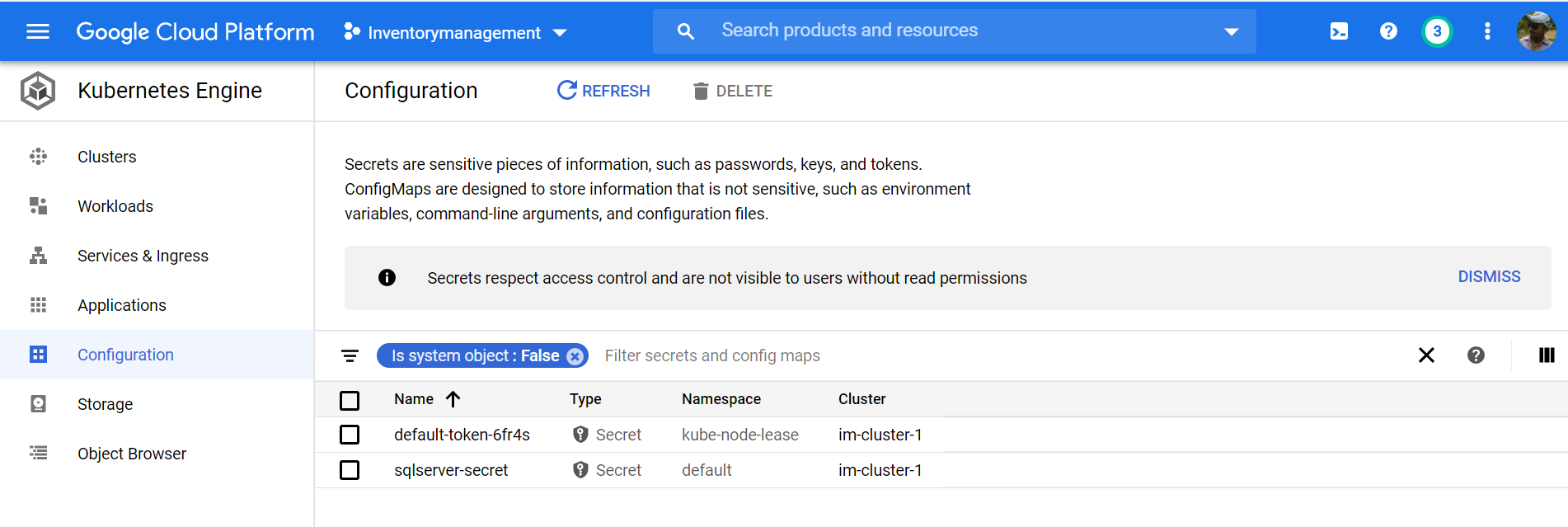
* Created the StorageClass and PersistentVolumeClaim

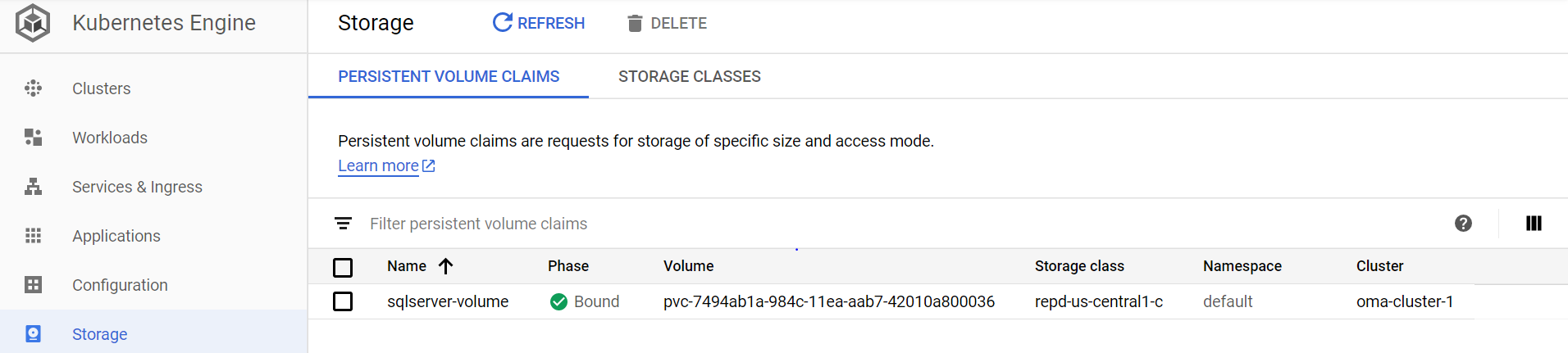
kubectl create -f sqlserver-volume.yaml

* Setting up and deploying SQL Server

kubectl create -f sqlserver-deployment.yaml







**5. UserAPI (Service type ClusterIP)**

* Created a docker image for UserAPI using the below command

docker build -t ashwanitaneja88/user\_api\_dimg:latest .

* docker login -u <user> -p <pwd>
* Push docker image to Container Registry

docker push ashwanitaneja88/user\_api\_dimg:latest

* Created pods using users-api-deployment.yaml

kubectl apply -f users-api-deployment.yaml

kubectl get pods (used to verify the pods)

* Created Service with type ClusterIP using users-api-service.yaml

kubectl apply -f users-api-service.yaml

* kubectl get services (used to check the services)

**6. APIGateway as Aggregates Service (Service type LoadBalancer)**

* Created a docker image for APIGateway using the below command

docker build -t ashwanitaneja88/apigateway\_dimg:latest .

* Push docker image to Container Registry

docker push ashwanitaneja88/apigateway\_dimg:latest

* Created pods using apigateway-deployment.yaml

kubectl apply -f apigateway-deployment.yaml

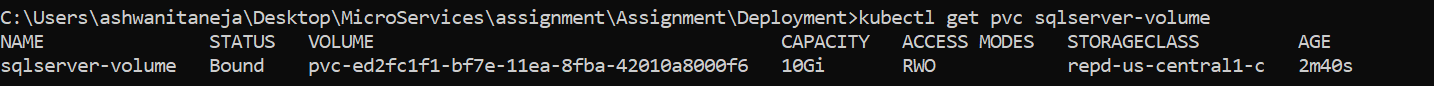
kubectl get pods (used to verify the pods)

* Created Service with type LoadBalancer using apigateway -service.yaml

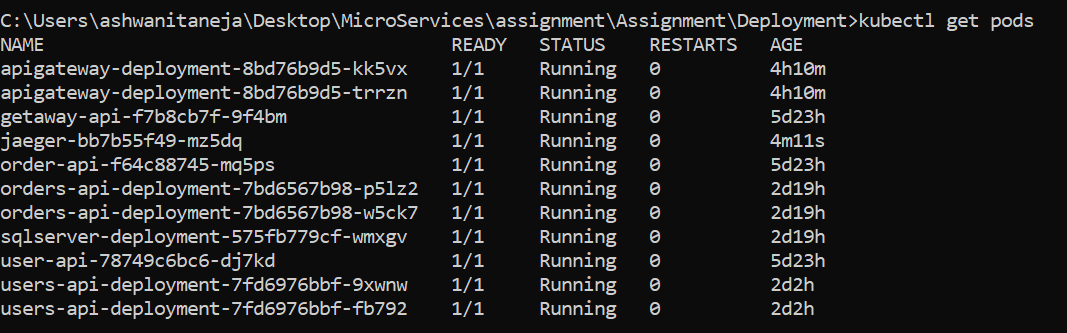
kubectl apply -f apigateway-service.yaml

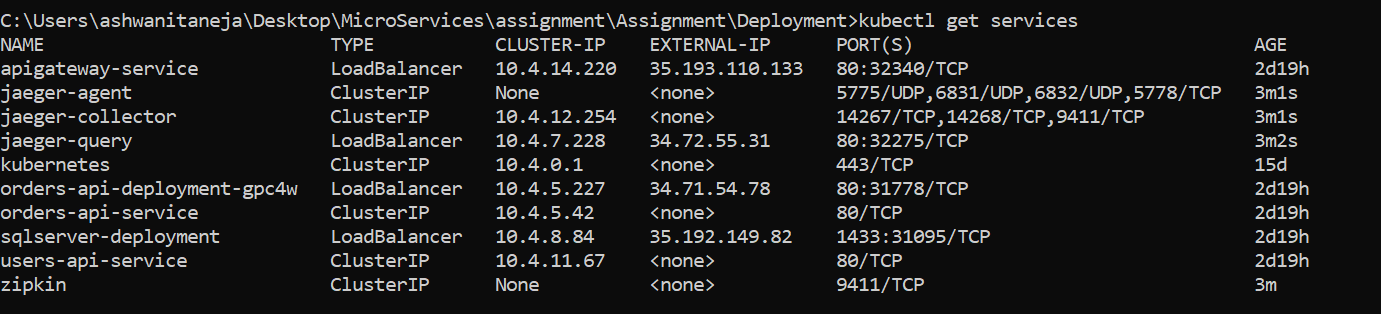
* kubectl get services (used to check the services)

**Persistent Volume**



**Running Pods**



**Running Services**

**Note:-**

**All Yaml files mentioned in the doc and images are attached to attached in deliverable.**

GIT-Hub Link:

<https://github.com/AshwaniNagarro/InventoryManagementApp>

Jaeger service Deployment

**kubectl create -f https://raw.githubusercontent.com/jaegertracing/jaeger-kubernetes/master/all-in-one/jaeger-all-in-one-template.yml**