Q1) Deploy a complex Scenario for Taint and Node maintenance.

Ans:

Step:1 Deploy scenario

```
kubectl create namespace sock-shop
kubectl create -f https://raw.githubusercontent.com/k21academyuk/docker/master/Nodemaintenance.yaml
```

```
root@master:~# kubectl create ns sock-shop
namespace/sock-shop created
root@master:~# kubectl create -f https://raw.githubusercontent.com/k21academyuk/docker/master/Nodemaintenance.yaml
deployment.apps/carts-db created
service/carts-db created
deployment.apps/carts created
service/carts created
deployment.apps/catalogue-db created
service/catalogue-db created
deployment.apps/catalogue created
service/catalogue created
deployment.apps/front-end created
service/front-end created
deployment.apps/orders-db created
service/orders-db created
deployment.apps/orders created
service/orders created
deployment.apps/payment created
service/payment created
deployment.apps/queue-master created
service/queue-master created
deployment.apps/rabbitmq created
service/rabbitmq created
deployment.apps/shipping created
service/shipping created
deployment.apps/user-db created
service/user-db created
deployment.apps/user created
service/user created
root@master:~#
```

With the following command, we can explore the status of all resources in the sock-shop namespace:

```
kubectl get all -n sock-shop
```

```
root@master:~# kubectl get all -n sock-shop
NAME                                READY   STATUS              RESTARTS   AGE
pod/carts-7c9df6fdb4-qvh5j          0/1     ContainerCreating   0          58s
pod/carts-db-6c6c68b747-xlr2r       0/1     ContainerCreating   0          58s
pod/catalogue-7c6dcb64f7-9j9ck      1/1     Running             0          58s
pod/catalogue-db-96f6f6b4c-z4v96    0/1     ContainerCreating   0          58s
pod/front-end-7b8bcd59cb-5kvkw      1/1     Running             0          57s
pod/orders-c9994cff9-67779          0/1     ContainerCreating   0          57s
pod/orders-db-659949975f-7vjld      0/1     ContainerCreating   0          57s
pod/payment-8576977df5-wstpp        0/1     ContainerCreating   0          57s
pod/queue-master-bbb6c4b9d-s57b7    0/1     ContainerCreating   0          56s
pod/rabbitmq-6d77f74dc-ccwf8        0/1     ContainerCreating   0          56s
pod/shipping-5d7c4f8bbf-58ft7       0/1     ContainerCreating   0          56s
pod/user-846f474c46-kcxgg           0/1     ContainerCreating   0          56s
pod/user-db-5f68d7b558-p9dss        0/1     ContainerCreating   0          56s

NAME                   TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
service/carts          ClusterIP   10.103.215.157   <none>        80/TCP           58s
service/carts-db       ClusterIP   10.110.240.84    <none>        27017/TCP        58s
service/catalogue      ClusterIP   10.110.189.192   <none>        80/TCP           57s
service/catalogue-db   ClusterIP   10.109.235.170   <none>        3306/TCP         58s
service/front-end      NodePort    10.106.188.178   <none>        80:30001/TCP     57s
service/orders         ClusterIP   10.102.255.116   <none>        80/TCP           57s
service/orders-db      ClusterIP   10.109.202.140   <none>        27017/TCP        57s
service/payment        ClusterIP   10.98.64.127     <none>        80/TCP           57s
service/queue-master   ClusterIP   10.109.178.108   <none>        80/TCP           56s
service/rabbitmq       ClusterIP   10.103.161.223   <none>        5672/TCP         56s
service/shipping       ClusterIP   10.105.94.185    <none>        80/TCP           56s
service/user           ClusterIP   10.99.19.42      <none>        80/TCP           56s
service/user-db        ClusterIP   10.107.205.178   <none>        27017/TCP        56s

NAME                              READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/carts             0/1     1            0           58s
deployment.apps/carts-db          0/1     1            0           58s
deployment.apps/catalogue         1/1     1            1           58s
deployment.apps/catalogue-db      0/1     1            0           58s
deployment.apps/front-end         1/1     1            1           57s
deployment.apps/orders            0/1     1            0           57s
deployment.apps/orders-db         0/1     1            0           57s
deployment.apps/payment           0/1     1            0           57s
deployment.apps/queue-master      0/1     1            0           57s
deployment.apps/rabbitmq          0/1     1            0           56s
deployment.apps/shipping          0/1     1            0           56s
```

Step 2: Drain worker node

```
kubectl get nodes
```

```
root@master:/home/ubuntu/microservices-demo/deploy/kubernetes# kubectl get nodes
NAME      STATUS      ROLES    AGE    VERSION
master    Ready       master   6d2h   v1.19.2
worker    NotReady    <none>   6d2h   v1.19.2
root@master:/home/ubuntu/microservices-demo/deploy/kubernetes#
```

```
kubectl drain worker --ignore-daemonsets --delete-local-data
```

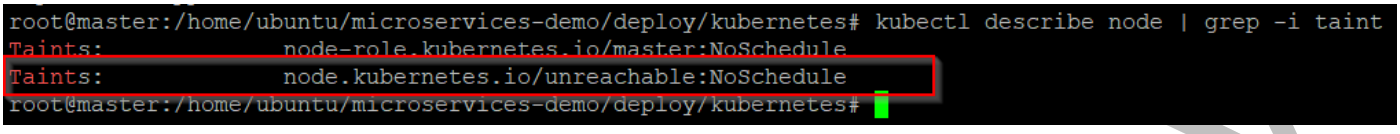
Note that all PODs are in Pending state now, since we have drained the only worker node available in our cluster:

```
kubectl get all -n sock-shop
```

Step 3: Perform Maintenance Actions

The worker node and the master node has the following taints:

```
kubectl describe node | grep -i taint
```



Step 4: Take Node into Service again

```
kubectl uncordon worker
```

The taint is removed from the worker node again:

```
kubectl describe node | grep -i taint
# output:
Taints:             node-role.kubernetes.io/master:NoSchedule
Taints:             <none>
```

After some time, all PODs are up and running again:

```
kubectl get pods -n sock-shop
# output:
NAME                           READY   STATUS    RESTARTS   AGE
carts-d646975bf-qdh7h          1/1     Running   0          20m
carts-db-f5677f464-dtmhm       1/1     Running   0          20m
catalogue-7db746bc5b-njrz7     1/1     Running   0          20m
catalogue-db-55965799b9-lp6h6  1/1     Running   0          20m
front-end-55b99f8c59-7bv8p     1/1     Running   0          20m
orders-986494c98-fvjh6         1/1     Running   0          20m
orders-db-78568b65bb-d7lpt     1/1     Running   0          20m
payment-7d4d4bf9b4-c8f7v       1/1     Running   0          20m
queue-master-6b5b5c7658-k5bkr  1/1     Running   0          20m
rabbitmq-f984b75c4-qtv7l       1/1     Running   0          20m
shipping-7b8865d964-pphhf      1/1     Running   0          20m
user-54555fbb7-cjzw8           1/1     Running   0          20m
user-db-7b9846c559-264xt       1/1     Running   0          20m
```

Step 5: Delete the Deployment

Since we have chosen to deploy the sock-shop into its own namespace, we can easily delete all of its resources with a single command:

```
kubectl delete namespace sock-shop
```

```
# output:
namespace "sock-shop" deleted
```