1. **Pod**
2. How many pods exist on the system? In the current(default) namespace.

kubectl get pods

1. Create a new pod with the nginx image.

kubectl run nginx --image=nginx

1. What is the image used for creating the pod?

kubectl describe pod <podname>

1. Which nodes are these pods placed on?

kubectl get pods -o wide

1. How many containers are part of the pod webapp?

kubectl get pods

1. What images are used in the new webapp pod?

Kubectl describe pod <podname>

1. What is the state of the container agentx in the pod webapp?

Kubectl describe pod <podname>

1. What is reason of error in pod?

Kubectl describe pod <podname>

1. What does the READY column in the output of the kubectl get pods command indicate?

Running container in the pods/ Total number of pods

1. Delete the webapp Pod.

Kubectl delete pod <podname>

1. Create a new pod with the name redis and with the image redis123.

kubectl run redis --image=redis123

1. Now change the image on this pod to redis.

kubectl edit pod redis (Edit the image section)

1. **Relica Set**
2. How many ReplicaSets exist on the system?

kubectl get rs

1. How many PODs are DESIRED in the new-replica-set?

Select the desired count

1. What is the image used to create the pods in the new-replica-set?

kubectl describe pod <podname> BUSYBOX777

1. How many pods are reday in relicaset?

Kubectl get pods

1. Why do you think the PODs are not ready?

Image not found – BUSYBOX777

1. Why are there still 4 PODs, even after you deleted one?

Replicaset ensures that the desired number of pods are avaialabe everytime

1. Create a ReplicaSet using the replicaset-definition-1.yaml file located at /root/ (There is an issue with the file, so try to fix it.)

vim replicaset-definition-1.yaml

kubectl create -f /root/replicaset-definition-1.yaml

1. Delete the two newly created ReplicaSets - replicaset-1 and replicaset-2

kubectl delete rs replicaset-1

1. Fix the original replica set new-replica-set to use the correct busybox image. (Either delete and recreate the ReplicaSet or Update the existing ReplicaSet and then delete all PODs, so new ones with the correct image will be created.)

kubectl edit pod <podname>

1. Scale the ReplicaSet to 5 PODs. (Use kubectl scale command or edit the replicaset using kubectl edit replicaset.)

kubectl edit replicaset

1. Now scale the ReplicaSet down to 2 PODs. (Use the kubectl scale command or edit the replicaset using kubectl edit replicaset)

kubectl edit replicaset

1. **Deployments**
2. How many Deployments exist on the systems

kubectl get deployments

1. How many Deployments exist on the system now?

kubectl get deployments

1. How many ReplicaSets exist on the system now?

kubectl get rs

1. Out of all the existing PODs, how many are ready?

kubectl get rs

1. What is the image used to create the pods in the new deployment?

Kubectl describe pod <podname>

6. Create a new Deployment using the deployment-definition-1.yaml file located at /root/.

vim deployment-definition-1.yaml

kubectl apply -f deployment-definition-1.yaml

1. Create a new Deployment with the below attributes using your own deployment definition file.

[Name: httpd-frontend;   
Replicas: 3;   
Image: httpd:2.4-alpine]

Copy d deployment-definition-1.yaml

vim deployment-definition-2.yaml

kubectl apply -f deployment-definition-2.yaml

1. **Namespaces:**
2. How many Namespaces exist on the system?

kubectl get namespaces

1. How many pods exist in the research namespace?

kubectl get pods --namespace=research

kubectl -n research get pods --no-headers | wc -l

1. Create a POD in the finance namespace.

kubectl run redis --image=redis -n finance

1. What pods are there in finance?

Kubectl get pods –namespace=finance

1. Which namespace has the blue pod in it?

kubectl get pods --all-namespaces

1. What DNS name should the Blue application use to access the database 'db-service' in the 'dev' namespace

(Since the blue application and the db-service are in different namespaces in this case, we need to use the service name along with the namespace to access the database. The FQDN (fully Qualified Domain Name) for the db-service in this example would be db-service.dev.svc.cluster.local.  
  
Note: You can also access it using the service name and namespace like this: db-service.dev)

1. **Services**
2. How many Services exist on the system?

kubectl get service

1. What is the type of the default kubernetes service?

ClusterIP

1. What is the targetPort configured on the kubernetes service?

kubectl describe service

1. How many labels are configured on the kubernetes service?

kubectl describe service

1. How many Endpoints are attached on the kubernetes service?

kubectl describe service

1. What is the image used to create the pods in the deployment?

kubectl get pods

kubectl describe pods <podname>

1. Create a new service to access the web application using the service-definition-1.yaml file

Name: webapp-service  
Type: NodePort  
targetPort: 8080  
port: 8080  
nodePort: 30080  
selector: simple-webapp

vim service-definition-1.yaml

kubectl apply -f service-definition-1.yaml

Imperative Command

1. Deploy a pod named nginx-pod using the nginx:alpine image.

kubectl run nginx-pod --image=nginx:alpine

1. Deploy a redis pod using the redis:alpine image with the labels set to tier=db.

kubectl run redis --image=redis:alpine --dry-run=client -oyaml > redis-pod.yaml

kubectl create -f redis-pod.yaml

1. Create a service redis-service to expose the redis application within the cluster on port 6379.

kubectl expose pod redis --port=6379 --name redis-service

1. Create a deployment named webapp using the image kodekloud/webapp-color with 3 replicas.

kubectl create deployment webapp --image=kodekloud/webapp-color --replicas=3

1. Create a new pod called custom-nginx using the nginx image and expose it on container port 8080.

kubectl run custom-nginx --image=nginx --port=8080

1. Create a new namespace called dev-ns.

kubectl create namespace dev-ns

1. Create a new deployment called redis-deploy in the dev-ns namespace with the redis image. It should have 2 replicas.

kubectl create deployment redis-deploy --image=redis --replicas=2 -n dev-ns

or

Step 1: Create the deployment YAML file  
kubectl create deployment redis-deploy --image redis --namespace=dev-ns --dry-run=client -o yaml > deploy.yaml  
Step 2: Edit the YAML file and add update the replicas to 2  
Step 3: Run kubectl apply -f deploy.yaml to create the deployment in the dev-ns namespace.  
You can also use kubectl scale deployment or kubectl edit deployment to change the number of replicas once the object has been created.

1. Create a pod called httpd using the image httpd:alpine in the default namespace. Next, create a service of type ClusterIP by the same name (httpd). The target port for the service should be 80.

 'httpd' pod created with the correct image?

 'httpd' service is of type 'ClusterIP'?

 'httpd' service uses correct target port 80?

 'httpd' service exposes the 'httpd' pod?

kubectl run httpd --image=httpd:alpine --port=80 –expose

1. **Scheduling**
2. A pod definition file nginx.yaml is given. Create a pod using the file.

kubectl create -f nginx.yaml

1. Why is the POD in a pending state? (Inspect the environment for various kubernetes control plane components.)

kubectl get pods --namespace kube-system

1. Manually schedule the pod on node01.

Edit the nginx file – add node name

kubectl get pods -o wide

1. Now schedule the same pod on the master/controlplane node.

Edit the nginx file – change node name

kubectl get pods -o wide

1. **Label and Selectors**
2. We have deployed a number of PODs. They are labelled with tier, env and bu. How many PODs exist in the dev environment? (Use selectors to filter the output)

kubectl get pods --selector env=dev

1. How many PODs are in the finance business unit (bu)?

kubectl get pods --selector bu=finance

1. How many objects are in the prod environment including PODs, ReplicaSets and any other objects?

kubectl get all --selector env=prod

1. Identify the POD which is part of the prod environment, the finance BU and of frontend tier?

kubectl get all --selector env=prod,bu=finance,tier=frontend

1. A ReplicaSet definition file is given replicaset-definition-1.yaml. Try to create the replicaset. There is an issue with the file. Try to fix it.

Edit the given file

kubectl apply -f replicaset-definition-1.yaml

1. Taints and Tolerance
2. How many Nodes exist on the system?

kubectl get nodes

1. Do any taints exist on node01?

kubectl describe node node01

or

kubectl describe node node01 | grep -i taints

1. Create a taint on node01 with key of spray, value of mortein and effect of NoSchedule

kubectl taint nodes node01 spray=mortein:NoSchedule

1. Create a new pod with the NGINX image, and Pod name as mosquito

kubectl run mosquito --image=nginx

1. Create another pod named bee with the NGINX image, which has a toleration set to the taint mortein

kubectl run bee --image=nginx --dry-run=client -oyaml > bee-pod.yaml

kubectl create -f bee.yaml

1. Remove the taint on master/controlplane, which currently has the taint effect of NoSchedule

Kubectl describe node controlplane

kubectl taint nodes controlplane node role.kubernetes.io/master:NoSchedule-

1. Node Affinity
2. How many Labels exist on node node01?

kubectl describe node node01

1. Apply a label color=blue to node node01

kubectl label node node01 color=blue

1. Create a new deployment named blue with the nginx image and 3 replicas.

kubectl create deployment blue --image=nginx --replicas=3

1. Which nodes can the pods for the blue deployment be placed on?

kubectl describe node node01

kubectl describe node controlplane (check taints)

or

kubectl describe node controlplane | grep -i taints

kubectl describe node node01 | grep -i taints

1. **Resource Limitations**
2. A pod called rabbit is deployed. Identify the CPU requirements set on the Pod

kubectl describe pod rabbit (Check In the requests)

1. Another pod called elephant has been deployed in the default namespace. It fails to get to a running state. Inspect this pod and identify the Reason why it is not running.

kubectl get pod --all-namespaces

(The status CrashLoopBackOff indicates that it is failing because the pod is out of memory. Identify the memory limit set on the POD.)

1. The elephant pod runs a process that consume 15Mi of memory. Increase the limit of the elephant pod to 20Mi. (Delete and recreate the pod if required. Do not modify anything other than the required fields.)

Create a pod naming elephant using polinix/stress

Edit the pod by changing the parameter

1. **Daemonsets**
2. How many DaemonSets are created in the cluster in all namespaces?

kubectl get daemonsets --all-namespaces

1. On how many nodes are the pods scheduled by the **DaemonSet** kube-proxy

kubectl describe daemonset kube-proxy --namespace=kube-system

1. Image used in pod for the daemonset kube-flannel-ds?

kubectl describe daemonsets kube-flannel-ds --namespace=kube-system

1. Deploy a **DaemonSet** for FluentD Logging.

 Name: elasticsearch

 Namespace: kube-system

 Image: k8s.gcr.io/fluentd-elasticsearch:1.20

kubectl create deployment elasticsearch --image=k8s.gcr.io/fluentd-elasticsearch:1.20 -n kube-system --dry-run=client -o yaml > fluentd.yaml

kubectl create -f fluentd.yaml

1. **Static Pods**
2. How many static pods exist in this cluster in all namespaces?

Kubectl get pods –all-namespaces

look for those with -controlplane appended in the name

kubectl describe pods kube-scheduler-controlplane --namespace=kube-system

1. Which of the below components is NOT deployed as a static pod?

kubectl get pods –all-namespaces

look for the pod from the list that does not end with -controlplane

1. On which nodes are the static pods created currently?

kubectl get pods - -all-namespaces -o wide

1. What is the path of the directory holding the static pod definition files?

Etc/kubernetes/manifests

1. What is the docker image used to deploy the kube-api server as a static pod?

/etc/kubernetes/manifests/kube-apiserver.yaml

1. Create a static pod named static-busybox that uses the busybox image and the command sleep 1000

Create a pod definition file called static-busybox.yaml with the provided specs

and place it under /etc/kubernetes/manifests directory.

1. **Test Multiple Schedulers**
2. What is the name of the POD that deploys the default kubernetes scheduler in this environment?

kubectl get pods --namespace=kube-sytem

-Kube-scheduler-controlplane

1. What is the image used to deploy the kubernetes scheduler?

(Inspect the kubernetes scheduler pod and identify the image)

kubectl describe pod kube-scheduler-controlplane --namespace=kube-system

1. Deploy an additional scheduler to the cluster following the given specification.

(Use the manifest file used by kubeadm tool. Use a different port than the one used by the current one.)

 Namespace: kube-system

 Name: my-scheduler

 Status: Running

 Custom Scheduler Name

Create a kube-scheduler.yaml in ------🡪 /etc/Kubernetes/manifests

- --leader-elect=false

- --port=10282

- --scheduler-name=my-scheduler

- --secure-port=0

Kubectl create -f my-scheduler.yaml

kubectl get pods kube-scheduler --namespace=kube-system

1. A POD definition file is given. Use it to create a POD with the new custom scheduler.

File is located at /root/nginx-pod.yaml

* Name: nginx
* Uses custom scheduler
* Status: Running

Vim nginx-pod.yaml

Add “SchedulerName: my-scheduler”

1. **Logging and Monitoring**

Practice Test Monitor Cluster Components

1. Deploy the metrics-server by creating all the components downloaded.

Run the kubectl create -f . command from within the downloaded repository.

Kubectl create -f kubectl Kubernetes-metrices-sever/

kubectl top node

1. Identify the node that consumes the most CPU.
2. Identify the POD that consumes the most Memory

Kubectl top node

1. Identify the POD that consumes the most Memory.

kubectl top pod

1. **Practice Test managing Application Logs**
2. A user - USER5 - has expressed concerns accessing the application. Identify the cause of the issue.

(Inspect the logs of the POD)

Kubectl logs <podname>

1. A user is reporting issues while trying to purchase an item. Identify the user and the cause of the issue.

Inspect the logs of the webapp in the POD

kubectl logs <podname> -c simple-webapp

1. **Application Life Cycle Management**

**Pratice Test Rolling Updates**

1. Inspect the deployment and identify the number of PODs deployed by it

Check the replicas

Kubectl get deployments

Kubectl describe deployment <deployment name>

1. What container image is used to deploy the applications?
2. Inspect the deployment and identify the current strategy

Kubectl describe deployment <deployment name>

(Rolling Update)

1. If you were to upgrade the application now what would happen?

Pods will be updated few at a time

1. Let us try that. Upgrade the application by setting the image on the deployment to kodekloud/webapp-color:v2

(Do not delete and re-create the deployment. Only set the new image name for the existing deployment.)

kubectl edit deployment frontend

(/root/curl-test.sh)

1. Up to how many PODs can be down for upgrade at a time

(Consider the current strategy settings and number of PODs – 4)

Look in Max Unavailable under RoolingUpdate 🡪25%

1. Change the deployment strategy to Recreate

(Do not delete and re-create the deployment. Only update the strategy type for the existing deployment.)

kubectl edit deployment frontend

(Also delete rolling Update as strategy.rollingupdate)

1. Upgrade the application by setting the image on the deployment to kodekloud/webapp-color:v3

kubectl edit deployment frontend

**Practice Commands and Arguments**

1. What is the command used to run the pod ubuntu-sleeper?

Kubectl describe pod

Check🡪command option

1. Create a pod with the ubuntu image to run a container to sleep for 5000 seconds. Modify the file ubuntu-sleeper-2.yaml.

 Pod Name: ubuntu-sleeper-2

 Command: sleep 5000

Edit the file and add the command

1. Create a pod using the file named ubuntu-sleeper-3.yaml. There is something wrong with it. Try to fix it!

Pod Name: ubuntu-sleeper-3

Command: sleep 1200

1. Inspect the file Dockerfile given at /root/webapp-color. What command is run at container startup?
2. Inspect the file Dockerfile2 given at /root/webapp-color. What command is run at container startup?
3. Create a pod with the given specifications. By default it displays a blue background. Set the given command line arguments to change it to green

kubectl run webapp-green --image=kodekloud/webapp-color --restart=Never --dry-run -o yaml > pod.yaml

or

kubectl run webapp-green --image=kodekloud/webapp-color --dry-run=client -o yaml > pod.yaml

vim pod.yaml

kubectl apply -f pod.yaml

**Practice Test ENV Variable**

1. What is the environment variable name set on the container in the pod?

Kubectl describe pod <podname>

1. Update the environment variable on the POD to display a green background

(Note: Delete and recreate the POD. Only make the necessary changes. Do not modify the name of the Pod.)

kubectl get pod webapp-color -o yaml > pod.yaml

kubectl delete pod webapp-color

vim pod.yaml

kubectl apply -f pod.yaml

1. How many ConfigMaps exist in the environment?

kubectl get configmaps

1. Identify the database host from the config map db-config

kubectl describe configmap db-config

1. Create a new ConfigMap for the webapp-color POD. Use the spec given below.

Kubectl create configmap webapp-configmap - -from-literal=APP\_COLOR=darkblue

1. Update the environment variable on the POD to use the newly created ConfigMap

kubectl delete pod webapp-color

kubectl explain pods --recursive | grep envFrom -A3

vim pod.yaml