Q1: create a redis pod

```
controlplane $ ls
filesystem  snap
controlplane $ vim my_pod.yml
controlplane $ ls
filesystem  my_pod.yml  snap
controlplane $ kubectl apply -f my_pod.yml
error: error validating "my_pod.yml": error validating data: apiVersion
=false
controlplane $ vim my_pod.yml
controlplane $ vim my_pod.yml
controlplane $ kubectl apply -f my_pod.yml
error: error parsing my_pod.yml: error converting YAML to JSON: yaml: li
controlplane $ vim my_pod.yml
controlplane $ kubectl apply -f my_pod.yml
error: error validating "my_pod.yml": error validating data: apiVersion
=false
controlplane $ vim my_pod.yml
controlplane $ kubectl apply -f my_pod.yml
pod/redis created
controlplane $ kubectl get pods
NAME    READY   STATUS              RESTARTS   AGE
redis   0/1     ContainerCreating   0          5s
controlplane $ kubectl get pods
NAME    READY   STATUS              RESTARTS   AGE
redis   0/1     ContainerCreating   0          8s
controlplane $ kubectl get pods -w
NAME    READY   STATUS    RESTARTS   AGE
redis   1/1     Running   0          10s
```

```
Editor   Tab 1   +
apiVersion: v1
kind: Pod
metadata:
  name: redis
spec:
  containers:
    - name: redis-container
      image: redis:latest
~
```

Q2: create an nginx pod with the image nginx123

```
Editor   Tab 1   +
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
   - name: nginx-container
     image: nginx123
~
```

```
nginx   0/1     CrashLoopBackOff   0          5m36s
redis   1/1     Running            0          8m1s
^Ccontrolplane $ kubectl get pods -w
NAME    READY   STATUS         RESTARTS   AGE
nginx   0/1     ErrImagePull   0          6m1s
redis   1/1     Running        0          8m26s
```

Q4:

The status code is Imageloop back off

And the status is because the image doesnt exist

Q5:

To change the status code change the nginx image to nginx:latest

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
   - name: nginx-container
     image: nginx:latest
~
```

```
NAME    READY   STATUS             RESTARTS   AGE
nginx   0/1     ImagePullBackOff   0          7m5s
redis   1/1     Running            0          9m30s
controlplane $ vim my_pod.yml
controlplane $ kubectl apply -f my_pod.yml
pod/nginx configured
controlplane $
```

Q6: how many replica sets exist on the system

```
controlplane $ kubectl get rs
No resources found in default namespace.
controlplane $
```

Q7: create a replica set

Editor Tab 1 +

```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: replica-set-1
spec:
  replicas: 2
  selector:
    matchLabels:
      tier: busybox
  template:
    metadata:
      labels:
        tier: busybox
    spec:
      containers:
       - name: busybox-container
         image: busybox
~
```

```
controlplane $ kubectl describe rs/replica-set-1
Name:         replica-set-1
Namespace:    default
Selector:     tier=busybox
Labels:       <none>
Annotations:  <none>
Replicas:     2 current / 2 desired
Pods Status:  2 Running / 0 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels:  tier=busybox
  Containers:
   busybox-container:
    Image:         busybox
    Port:          <none>
    Host Port:     <none>
    Environment:   <none>
    Mounts:        <none>
  Volumes:         <none>
  Node-Selectors:  <none>
  Tolerations:     <none>
Events:
  Type    Reason            Age    From                   Message
  ----    ------            ----   ----                   -------
  Normal  SuccessfulCreate  2m25s  replicaset-controller  Created pod: replica-set-1-qfthv
  Normal  SuccessfulCreate  2m25s  replicaset-controller  Created pod: replica-set-1-c8hsv
controlplane $
```

Q8: Scaling to 3 replicas is done by editing the replicas in the my_pod file

```
controlplane $ vim my_pod.yml
controlplane $ kubectl apply -f my_pod.yml
replicaset.apps/replica-set-1 configured
controlplane $ kubectl describe rs/replica-set-1
Name:         replica-set-1
Namespace:    default
Selector:     tier=busybox
Labels:       <none>
Annotations:  <none>
Replicas:     3 current / 3 desired
Pods Status:  3 Running / 0 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels:  tier=busybox
  Containers:
   busybox-container:
    Image:         busybox
    Port:          <none>
    Host Port:     <none>
    Environment:   <none>
    Mounts:        <none>
  Volumes:         <none>
  Node-Selectors:  <none>
  Tolerations:     <none>
Events:
  Type    Reason            Age    From                   Message
  ----    ------            ----   ----                   -------
  Normal  SuccessfulCreate  3m14s  replicaset-controller  Created pod: replica-set-1-qfthv
  Normal  SuccessfulCreate  3m14s  replicaset-controller  Created pod: replica-set-1-c8hsv
  Normal  SuccessfulCreate  2s     replicaset-controller  Created pod: replica-set-1-rtbqx
controlplane $
```

Q9: How many ready pods are in the replicaset

```
replica-set-1-rtbqx   0/1     CrashLoopBackOff   7 (2m28s ago)   13m
controlplane $ kubectl get pods -w
NAME                  READY   STATUS             RESTARTS        AGE
nginx                 1/1     Running            1 (37m ago)     37m
redis                 1/1     Running            0               39m
replica-set-1-c8hsv   0/1     CrashLoopBackOff   8 (21s ago)     16m
replica-set-1-qfthv   0/1     CrashLoopBackOff   8 (43s ago)     16m
replica-set-1-rtbqx   0/1     CrashLoopBackOff   7 (2m33s ago)   13m
q^Ccontrolplane $ kubectl get rs
NAME            DESIRED   CURRENT   READY   AGE
replica-set-1   3         3         0       16m
controlplane $
```

None because busybox doesnt have a valid entrypoint nad exits as soon as it starts

Q10: deleting a pod in a replica set

```
controlplane $ kubectl describe rs/replica-set-1
Name:         replica-set-1
Namespace:    default
Selector:     tier=busybox
Labels:       <none>
Annotations:  <none>
Replicas:     3 current / 3 desired
Pods Status:  3 Running / 0 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels:  tier=busybox
  Containers:
   busybox-container:
    Image:         busybox
    Port:          <none>
    Host Port:     <none>
    Environment:   <none>
    Mounts:        <none>
  Volumes:         <none>
  Node-Selectors:  <none>
  Tolerations:     <none>
Events:
  Type    Reason            Age   From                   Message
  ----    ------            ----  ----                   -------
  Normal  SuccessfulCreate  6s    replicaset-controller  Created pod: replica-set-1-nsqrp
  Normal  SuccessfulCreate  6s    replicaset-controller  Created pod: replica-set-1-rjbk2
  Normal  SuccessfulCreate  6s    replicaset-controller  Created pod: replica-set-1-twvkf
controlplane $ kubectl delete pod/replica-set-1-nsqrp
pod "replica-set-1-nsqrp" deleted
controlplane $ kubectl get pods
NAME                READY   STATUS             RESTARTS      AGE
nginx               1/1     Running            1 (40m ago)   41m
redis               1/1     Running            0             43m
replica-set-1-9jt8w   0/1     CrashLoopBackOff   1 (2s ago)    5s
replica-set-1-rjbk2   0/1     Completed          2 (23s ago)   27s
replica-set-1-twvkf   0/1     Completed          2 (24s ago)   27s
controlplane $
```

K8s would replicate the pod and recreate another in its place

Q11: how many deployments and replicasets on the system

```
controlplane $ kubectl get deploy
No resources found in default namespace.
controlplane $ kubectl get rs
NAME            DESIRED   CURRENT   READY   AGE
replica-set-1   3         3         0       91s
controlplane $
```

Q12: creating a deployment

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: deployment-1
  labels:
    tier: busy
spec:
  replicas: 3
  selector:
    matchLabels:
      tier: busy
  template:
    metadata:
      labels:
        tier: busy
    spec:
      containers:
      - name: busybox
        image: nginx:1.14.2
        ports:

~
~
~
~
~
~
~
~
```

Q13:

```
controlplane $ kubectl get rs
NAME            DESIRED   CURRENT   READY   AGE
replica-set-1   3         3         0       91s
controlplane $ vim my_pod.yml
controlplane $ kubectl apply -f my_pod.yml
deployment.apps/deployment-1 created
controlplane $ kubectl get deploy
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
deployment-1   0/3     3            0           3s
controlplane $ kubectl get deploy -w
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
deployment-1   3/3     3            3           9s
^Ccontrolplane $ kubectl get rs
NAME                      DESIRED   CURRENT   READY   AGE
deployment-1-5f9c9bfc6f   3         3         3       21s
replica-set-1             3         3         0       4m35s
controlplane $
```

1 deployment and 2 replica sets

Q 14:

3 pods are ready with deployment 1

```
controlplane $ kubectl get rs
NAME           DESIRED   CURRENT   READY   AGE
replica-set-1   3         3         0       91s
controlplane $ vim my_pod.yml
controlplane $ kubectl apply -f my_pod.yml
deployment.apps/deployment-1 created
controlplane $ kubectl get deploy
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
deployment-1   0/3     3            0           3s
controlplane $ kubectl get deploy -w
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
deployment-1   3/3     3            3           9s
^Ccontrolplane $ kubectl get rs
NAME                      DESIRED   CURRENT   READY   AGE
deployment-1-5f9c9bfc6f   3         3         3       21s
replica-set-1             3         3         0       4m35s
controlplane $
```

Q15: using the recreate strategy

Editor Tab 1 +

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: deployment-1
  labels:
    tier: busy
spec:
  replicas: 3
  strategy:
    type: Recreate
  selector:
    matchLabels:
      tier: busy
  template:
    metadata:
      labels:
        tier: busy
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
```

Q16: describing the current deployment

```
controlplane $ vim my_pod.yml
controlplane $ kubectl apply -f my_pod.yml
deployment.apps/deployment-1 configured
controlplane $ kubectl describe deploy deployment-1
Name:               deployment-1
Namespace:          default
CreationTimestamp:  Sun, 09 Feb 2025 13:38:38 +0000
Labels:             tier=busy
Annotations:        deployment.kubernetes.io/revision: 2
Selector:           tier=busy
Replicas:           3 desired | 3 updated | 3 total | 3 available | 0 unavailable
StrategyType:       Recreate
MinReadySeconds:    0
Pod Template:
  Labels:  tier=busy
  Containers:
   nginx:
    Image:         nginx:1.14.2
    Port:          <none>
    Host Port:     <none>
    Environment:   <none>
    Mounts:        <none>
  Volumes:         <none>
  Node-Selectors:  <none>
  Tolerations:     <none>
Conditions:
  Type           Status  Reason
  ----           ------  ------
  Available      True    MinimumReplicasAvailable
  Progressing    True    NewReplicaSetAvailable
OldReplicaSets:  deployment-1-5f9c9bfc6f (0/0 replicas created)
NewReplicaSet:   deployment-1-66b49f6559 (3/3 replicas created)
Events:
  Type    Reason             Age   From                   Message
  ----    ------             ----  ----                   -------
  Normal  ScalingReplicaSet  6m7s  deployment-controller  Scaled up replica set deployment-1-5f9c9bfc6f to 3
  Normal  ScalingReplicaSet  11s   deployment-controller  Scaled down replica set deployment-1-5f9c9bfc6f to 0 from 3
  Normal  ScalingReplicaSet  8s    deployment-controller  Scaled up replica set deployment-1-66b49f6559 to 3
controlplane $
```

Q17: Rolling back would use the Rolling Update type with the same image which was busybox

Q18: create nginx deployement

Editor Tab 1 +

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
 name: nginx-deployment
 labels:
   app: nginx-app
   type: frontend
spec:
 replicas: 3
 selector:
   matchLabels:
     app: nginx-app
     type: frontend
 template:
   metadata:
     labels:
       app: nginx-app
       type: frontend
   spec:
     containers:
     - name: nginx-container
       image: nginx:latest
 strategy:
   type: RollingUpdate
   rollingUpdate:
     maxSurge: 1
     maxUnavailable: 1
```

```
controlplane $ kubectl apply -f my_deploy.yml
deployment.apps/nginx-deployment created
controlplane $ kubectl describe deploy nginx-deployment
Name:                   nginx-deployment
Namespace:              default
CreationTimestamp:      Sun, 09 Feb 2025 13:54:09 +0000
Labels:                 app=nginx-app
                        type=frontend
Annotations:            deployment.kubernetes.io/revision: 1
Selector:               app=nginx-app,type=frontend
Replicas:               3 desired | 3 updated | 3 total | 3 available | 0 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  1 max unavailable, 1 max surge
Pod Template:
  Labels:  app=nginx-app
           type=frontend
  Containers:
   nginx-container:
    Image:         nginx:latest
    Port:          <none>
    Host Port:     <none>
    Environment:   <none>
    Mounts:        <none>
  Volumes:         <none>
  Node-Selectors:  <none>
  Tolerations:     <none>
Conditions:
  Type           Status  Reason
  ----           ------  ------
  Available      True    MinimumReplicasAvailable
  Progressing    True    NewReplicaSetAvailable
OldReplicaSets:  <none>
NewReplicaSet:   nginx-deployment-75d94b4d46 (3/3 replicas created)
Events:
  Type    Reason             Age   From                   Message
  ----    ------             ----  ----                   -------
  Normal  ScalingReplicaSet  18s   deployment-controller  Scaled up replica set nginx-deployment-75d94b4d46 to 3
controlplane $
```

Q19: create static pod

First find the current nodes of the system

Kubectly get nodes

```
Connection to node01 closed.
controlplane $ kubectl get nodes
NAME           STATUS   ROLES           AGE   VERSION
controlplane   Ready    control-plane   12d   v1.31.0
node01         Ready    <none>          12d   v1.31.0
controlplane $ kubectl describe node01
error: the server doesn't have a resource type "node01"
```

Find the hostname by using kubectl describe node nodename

```
  Ready                   True    Sun, 09
Addresses:
  InternalIP:  172.30.2.2
  Hostname:    node01
Capacity:
  cpu:                1
```

Then ssh into the hostname:

```
controlplane $ ssh node01
Last login: Sun Feb  9 19:21:11 2025 from 10.244.8
node01 $
```

Creating a pod named static.yml

```
Editor   Tab 1   +
apiVersion: v1
kind: Pod
metadata:
  name: static-web
  labels:
    role: static
spec:
  containers:
    - name: web
      image: nginx
~
~
~
```

add the static pod path to the kubelet.conf on the node:

```
  user:
    client-certificate: /var/lib/kubelet/pki/kubelet-client-current.pem
    client-key: /var/lib/kubelet/pki/kubelet-client-current.pem
staticPodPath: /etc/kubernetes/manifessts/static.yml
~
~
~
~
~
~
~
~
```

Inspecting and checking the static pod using crictl to check the running containers on the worker node

```
node01 $ crictl ps
CONTAINER           IMAGE               CREATED             STATE               NAME                ATTEMPT             POD ID              POD
c4e8ab7cddcb2       97662d24417b3       27 minutes ago      Running             web                 1                   d935916fbc1db       static-web-node01
dbf5dffe23584       cbb01a7bd410d       About an hour ago   Running             coredns             1                   6458852f319ae       coredns-57888bfdc7-bbwzr
9a6ff5428dde7       cbb01a7bd410d       About an hour ago   Running             coredns             1                   e231ba73e3053       coredns-57888bfdc7-685jj
4a043490b4a86       e6ea68648f0cd       About an hour ago   Running             kube-flannel        1                   e8a37ea38d398       canal-phldr
ebe026cf02bf3       75392e3500e36       About an hour ago   Running             calico-node         1                   e8a37ea38d398       canal-phldr
ab794dca65845       ad83b2ca7b09e       About an hour ago   Running             kube-proxy          1                   f3af2fc89dd2b       kube-proxy-z2ps8
node01 $ man crictl
No manual entry for crictl
node01 $ crictl --help
NAME:
   crictl - client for CRI

USAGE:
   crictl [global options] command [command options]
```