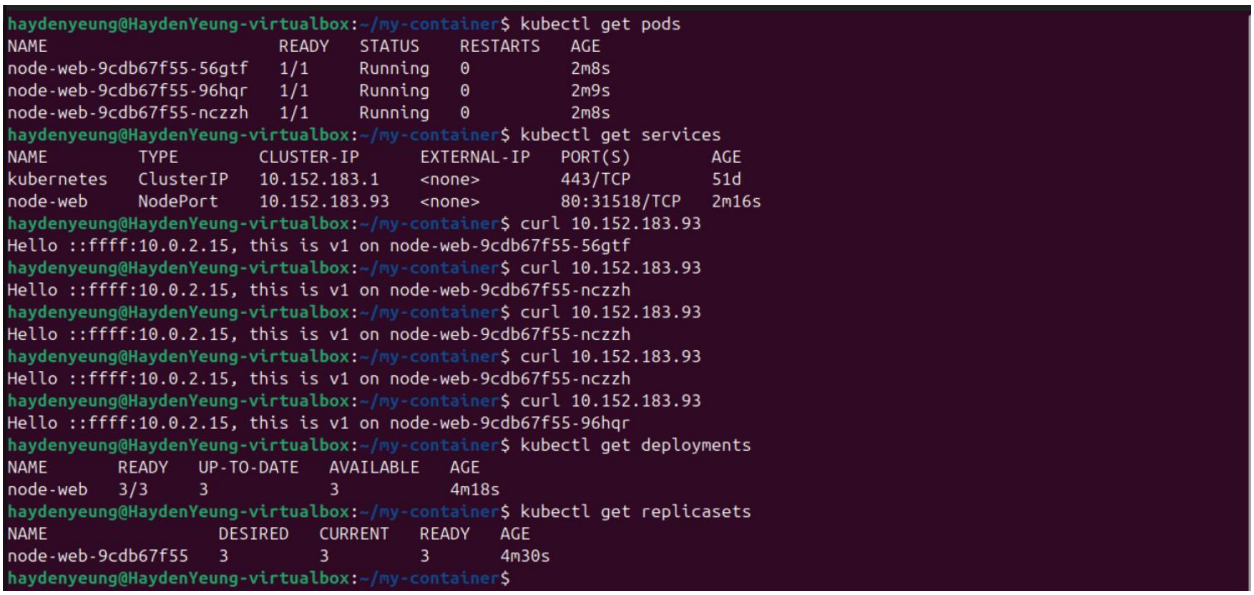
1. Lesson Recap

- This week, I learned concepts of StatefulSets, ReplicaSets and how normal Deployment & StatefulSets are different from one another (reflected through lab activities).
- In addition, I learned how to "cheat" in writing .yaml file for deploying a container through the including of dry-run in both "kubectl create deployment …" and "kubectl create service …".
- Revisit on how to "applying for" PVC & checking local files and display its contents.sssssssssssssssssss

2. Lab Activities



Followed instruction steps before attempted on Task 1:

- Created 3 container images (node-web:1, 2, and 3) & pushed them to localhost repository.
- Deployed 3 replicas of node-web:1 along with service that governing these latters.

Task 1 – Verify that these objects are related

"kubectl describe deployment node-web"

```
haydenyeung@HaydenYeung-virtualbox:~/my-container$ kubectl describe deployment node-web
Name:                   node-web
Namespace:              default
CreationTimestamp:      Sun, 20 Apr 2025 22:00:35 +1000
Labels:                 app=node-web
Annotations:            deployment.kubernetes.io/revision: 1
Selector:               app=node-web
Replicas:               3 desired | 3 updated | 3 total | 3 available | 0 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=node-web
  Containers:
   node-web:
    Image:         localhost:5000/node-web:1
    Port:          8080/TCP
    Host Port:     0/TCP
    Environment:   <none>
    Mounts:        <none>
  Volumes:         <none>
  Node-Selectors:  <none>
  Tolerations:     <none>
Conditions:
  Type           Status  Reason
  ----           ------  ------
  Available      True    MinimumReplicasAvailable
  Progressing    True    NewReplicaSetAvailable
OldReplicaSets:  <none>
NewReplicaSet:   node-web-9cdb67f55 (3/3 replicas created)
Events:          <none>
```

“kubectl describe replicasets node-web-9cdb67f55”

```
haydenyeung@HaydenYeung-virtualbox:~/my-container$ kubectl describe replicasets node-web-9cdb67f55
Name:           node-web-9cdb67f55
Namespace:      default
Selector:       app=node-web,pod-template-hash=9cdb67f55
Labels:         app=node-web
                pod-template-hash=9cdb67f55
Annotations:    deployment.kubernetes.io/desired-replicas: 3
                deployment.kubernetes.io/max-replicas: 4
                deployment.kubernetes.io/revision: 1
Controlled By:  Deployment/node-web
Replicas:       3 current / 3 desired
Pods Status:    3 Running / 0 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels:  app=node-web
           pod-template-hash=9cdb67f55
  Containers:
   node-web:
    Image:         localhost:5000/node-web:1
    Port:          8080/TCP
    Host Port:     0/TCP
    Environment:   <none>
    Mounts:        <none>
  Volumes:         <none>
  Node-Selectors:  <none>
  Tolerations:     <none>
Events:            <none>
```

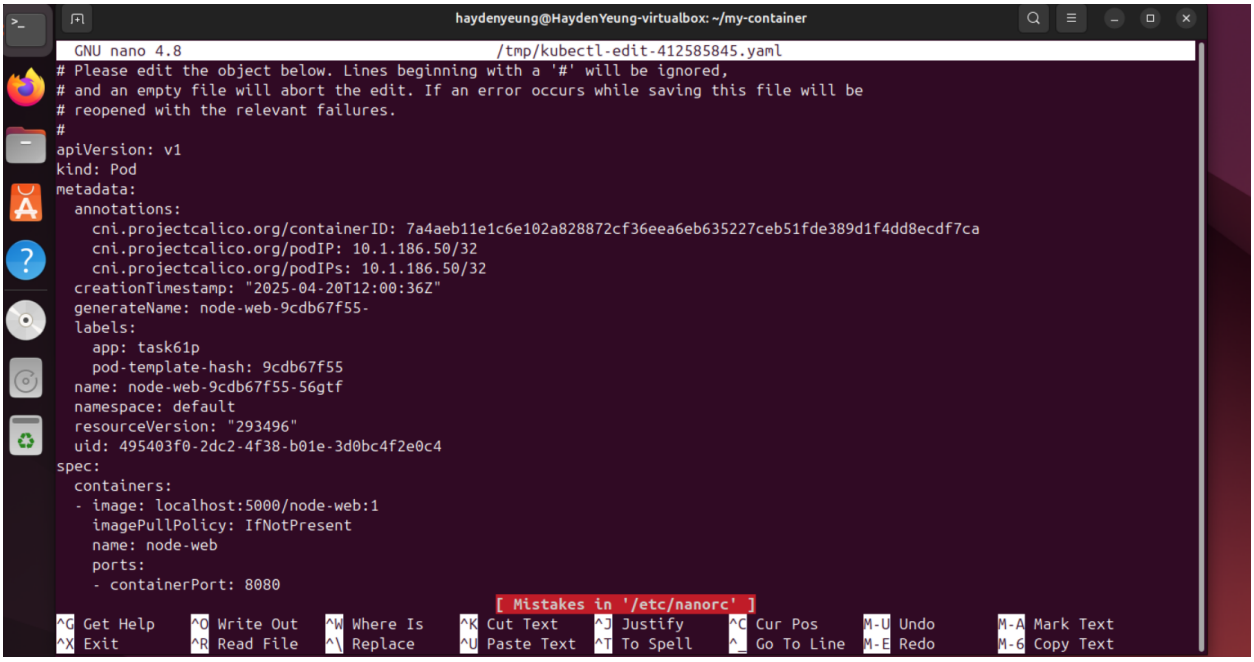
“kubectl describe pod node-web-9cdb67f55-56gtf”

```
haydenyeung@HaydenYeung-virtualbox:~/my-container$ kubectl describe pod node-web-9cdb67f55-56gtf
Name:             node-web-9cdb67f55-56gtf
Namespace:        default
Priority:         0
Service Account:  default
Node:             haydenyeung-virtualbox/10.0.2.15
Start Time:       Sun, 20 Apr 2025 22:00:36 +1000
Labels:           app=node-web
                  pod-template-hash=9cdb67f55
Annotations:      cni.projectcalico.org/containerID: 7a4aeb11e1c6e102a828872cf36eea6eb635227ceb51fde389d1f4dd8ecdf7ca
                  cni.projectcalico.org/podIP: 10.1.186.50/32
                  cni.projectcalico.org/podIPs: 10.1.186.50/32
Status:           Running
IP:               10.1.186.50
IPs:
  IP:           10.1.186.50
Controlled By:  ReplicaSet/node-web-9cdb67f55
Containers:
  node-web:
    Container ID:   containerd://8c15536581a42dd1a6f99d0cb09e42287ef7c93c07cc2afb04948e84557ab421
    Image:          localhost:5000/node-web:1
    Image ID:       localhost:5000/node-web@sha256:5513a828705234896c35e2a6af03388fbd3818f32548940cd4df7502496f40cf
    Port:           8080/TCP
    Host Port:      0/TCP
    State:          Running
      Started:      Sun, 20 Apr 2025 22:01:09 +1000
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-kjbfs (ro)
```

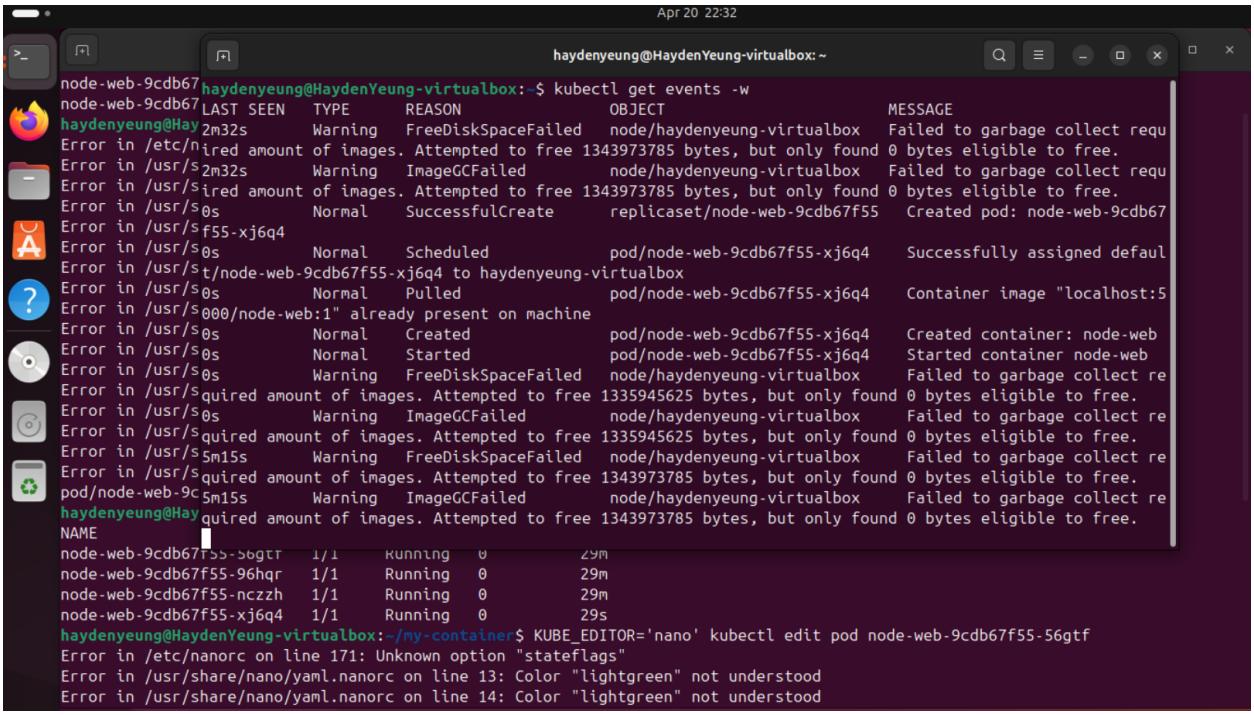
```
Conditions:
  Type                        Status
  PodReadyToStartContainers   True
  Initialized                 True
  Ready                       True
  ContainersReady             True
  PodScheduled                True
Volumes:
  kube-api-access-kjbfs:
    Type:                    Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    ConfigMapOptional:       <nil>
    DownwardAPI:             true
QoS Class:                   BestEffort
Node-Selectors:              <none>
Tolerations:                 node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                             node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:                      <none>
```

- All 3 of them were labeled as “app=node-web”
- In “deployment” output, I saw the following line: “NewReplicaSets: node-web-9cdb67f55 (3/3 replicas created)”.
- In “replicasets” output, I saw the line: “Controlled by: Deployment/node-web”.
- In “pod” output, I saw the line: “Controlled by: ReplicaSet/node-web-9cdb67f55”.

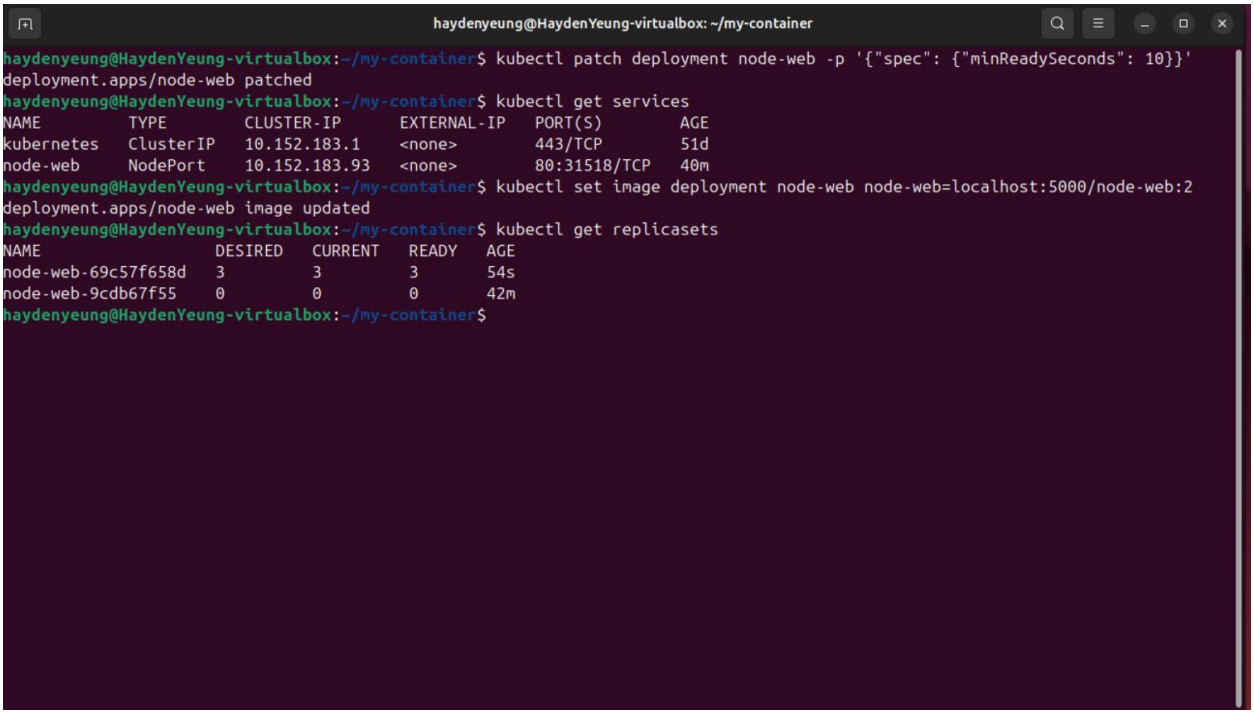
→ Thus, I could conclude that: Deployment “node-web” controls ReplicaSets “node-web-9cdb67f55”, this latter is controlling pod “node-web-9cdb67f55-56gtf”.



Changed the “labels: app” value from “node-web” to “task61p” and here were the results:



- A new pod called “node-web-9cdb67f55-xj6q4” was created.
- 4 pods were running instead of the original 3.



Updated the deployment's container image from "node-web:1" to "node-web:2"

Task 2 – Why is the old ReplicaSet left behind?

"kubectl describe deployment node-web"

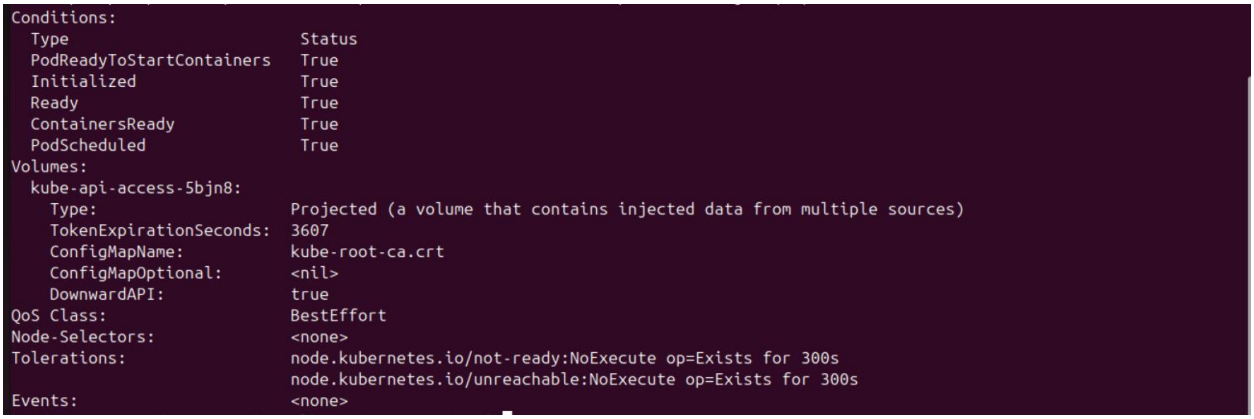
```
haydenyeung@HaydenYeung-virtualbox:~/my-container$ kubectl describe deployment node-web
Name:                   node-web
Namespace:              default
CreationTimestamp:      Sun, 20 Apr 2025 22:00:35 +1000
Labels:                 app=node-web
Annotations:            deployment.kubernetes.io/revision: 2
Selector:               app=node-web
Replicas:               3 desired | 3 updated | 3 total | 3 available | 0 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        10
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=node-web
  Containers:
   node-web:
    Image:         localhost:5000/node-web:2
    Port:          8080/TCP
    Host Port:     0/TCP
    Environment:   <none>
    Mounts:        <none>
  Volumes:         <none>
  Node-Selectors:  <none>
  Tolerations:     <none>
Conditions:
  Type           Status  Reason
  ----           ------  ------
  Available      True    MinimumReplicasAvailable
  Progressing    True    NewReplicaSetAvailable
OldReplicaSets:  node-web-9cdb67f55 (0/0 replicas created)
NewReplicaSet:   node-web-69c57f658d (3/3 replicas created)
Events:          <none>
```

“kubectl describe replicaset node-web-69c57f658d”

```
haydenyeung@HaydenYeung-virtualbox:~/my-container$ kubectl describe replicasets node-web-69c57f658d
Name:           node-web-69c57f658d
Namespace:      default
Selector:       app=node-web,pod-template-hash=69c57f658d
Labels:         app=node-web
                pod-template-hash=69c57f658d
Annotations:    deployment.kubernetes.io/desired-replicas: 3
                deployment.kubernetes.io/max-replicas: 4
                deployment.kubernetes.io/revision: 2
Controlled By:  Deployment/node-web
Replicas:       3 current / 3 desired
Pods Status:    3 Running / 0 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels:  app=node-web
           pod-template-hash=69c57f658d
  Containers:
   node-web:
    Image:         localhost:5000/node-web:2
    Port:          8080/TCP
    Host Port:     0/TCP
    Environment:   <none>
    Mounts:        <none>
  Volumes:         <none>
  Node-Selectors:  <none>
  Tolerations:     <none>
Events:            <none>
```

“kubectl describe pod node-web-69c57f658d-5bvhk”

```
haydenyeung@HaydenYeung-virtualbox:~/my-container$ kubectl describe pods node-web-69c57f658d-5bvhk
Name:             node-web-69c57f658d-5bvhk
Namespace:        default
Priority:         0
Service Account:  default
Node:             haydenyeung-virtualbox/10.0.2.15
Start Time:       Sun, 20 Apr 2025 22:42:47 +1000
Labels:           app=node-web
                  pod-template-hash=69c57f658d
Annotations:      cni.projectcalico.org/containerID: 53231e27d8bd9a256dfafe31ec2667b9aeeb3ed6bf9d03e9e1c27422d3a55aaf
                  cni.projectcalico.org/podIP: 10.1.186.6/32
                  cni.projectcalico.org/podIPs: 10.1.186.6/32
Status:           Running
IP:               10.1.186.6
IPs:
  IP:           10.1.186.6
Controlled By:  ReplicaSet/node-web-69c57f658d
Containers:
  node-web:
    Container ID:   containerd://efea87310dc1d18fa7e5f624312914ddebd59516b445511c13f17b400cd050c9
    Image:          localhost:5000/node-web:2
    Image ID:       localhost:5000/node-web@sha256:6aba11275003da62ae0306077dedb81ff7a2a38279b21f4a647809c4c763d003
    Port:           8080/TCP
    Host Port:      0/TCP
    State:          Running
      Started:      Sun, 20 Apr 2025 22:42:49 +1000
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-5bjn8 (ro)
```



```
Conditions:
  Type                        Status
  PodReadyToStartContainers   True
  Initialized                 True
  Ready                       True
  ContainersReady             True
  PodScheduled                True
Volumes:
  kube-api-access-5bjn8:
    Type:                     Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:   3607
    ConfigMapName:            kube-root-ca.crt
    ConfigMapOptional:        <nil>
    DownwardAPI:              true
QoS Class:                    BestEffort
Node-Selectors:               <none>
Tolerations:                  node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                              node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:                       <none>
```

Explanation:

This behavior is intentional and part of Kubernetes' design for **safe rollouts and rollbacks**. Here's why the old ReplicaSet persists:

1/ Rollback Capability

- Kubernetes retains the old ReplicaSet to allow instant rollback if the new version (node-web-69c57f658d) fails. You can revert to the previous version with:

"kubectl rollout undo deployment/node-web"

2/ Revision History

- Each ReplicaSet represents a **Deployment revision**.
- The old ReplicaSet (node-web-9cdb67f55) serves as historical record, viewable via:

"kubectl rollout history deployment/node-web"

3/ Rolling Update Mechanism

- During updates, Kubernetes:
  - Creates a new ReplicaSet (version 2).
  - Gradually scales up the new Pods while scaling down the old ones.
  - Ensures zero-downtime deployments.

Challenge Task – Try a StatefulSet

Copied the content of “node-web.yaml” to “stateful-web.yaml” and changed the value of “kind”: from “Deployment” to “StatefulSet”

```
41s         Normal    SuccessfulCreate      statefulset/node-web            create Pod node-web-0 in Sta
tefulSet node-web successful
14s         Normal    SuccessfulCreate      statefulset/node-web            create Pod node-web-1 in Sta
tefulSet node-web successful
12s         Normal    SuccessfulCreate      statefulset/node-web            create Pod node-web-2 in Sta
tefulSet node-web successful
5m15s       Normal    Killing               pod/node-web-9cdb67f55-56gtf    Stopping container node-web
5m15s       Warning   Unhealthy             pod/node-web-6fbc68446d-nsqjb   Readiness probe failed: Get
"http://10.1.186.49:8080/": dial tcp 10.1.186.49:8080: connect: connection refused
5m15s       Warning   Unhealthy             pod/node-web-6fbc68446d-ll6s2   Readiness probe failed: Get
"http://10.1.186.17:8080/": dial tcp 10.1.186.17:8080: connect: connection refused
5m15s       Warning   Unhealthy             pod/node-web-6fbc68446d-t2ql9   Readiness probe failed: Get
"http://10.1.186.5:8080/": dial tcp 10.1.186.5:8080: connect: connection refused
5m15s       Warning   Unhealthy             pod/node-web-6fbc68446d-nsqjb   Readiness probe failed: Get
"http://10.1.186.49:8080/": dial tcp 10.1.186.49:8080: connect: invalid argument
```

Got these errors upon attempted to apply “stateful-web.yaml”

Solutions:

- Include “serviceName: node-web” under “selector” field of “kind: StatefulSet” – StatefulSet requires this field to managing the network identity of all pods generated from “stateful-web.yaml”. In this case, Service named “node-web” is defined in Service section but the lack of this in StatefulSet caused the lost-connection between it and Service that it would governing over.
- Add line “initialDelaySeconds: 10” under “- readinessProbe:” – to allow the application has some time for starting process before checking.

```
5m15s       Normal    SuccessfulCreate              statefulset/node-web          create Pod node-web
-0 in StatefulSet node-web successful
5m15s       Normal    Scheduled                     pod/node-web-0                Successfully assign
ed default/node-web-0 to haydenyeung-virtualbox
5m15s       Normal    Started                       pod/node-web-0                Started container n
ode-web
5m16s       Normal    Pulled                        pod/node-web-0                Container image "lo
calhost:5000/node-web:1" already present on machine
5m15s       Normal    Created                       pod/node-web-0                Created container:
node-web
5m15s       Normal    SuccessfulCreate              statefulset/node-web          create Pod node-web
-1 in StatefulSet node-web successful
5m15s       Normal    Scheduled                     pod/node-web-1                Successfully assign
ed default/node-web-1 to haydenyeung-virtualbox
5m15s       Normal    Pulled                        pod/node-web-1                Container image "lo
calhost:5000/node-web:1" already present on machine
5m15s       Normal    Created                       pod/node-web-1                Created container:
node-web
5m15s       Normal    Started                       pod/node-web-1                Started container n
ode-web
5m15s       Normal    SuccessfulCreate              statefulset/node-web          create Pod node-web
-2 in StatefulSet node-web successful
5m15s       Normal    Scheduled                     pod/node-web-2                Successfully assign
ed default/node-web-2 to haydenyeung-virtualbox
5m15s       Normal    Pulled                        pod/node-web-2                Container image "lo
```

Results:

- All 3 generated pods from applied “stateful-web.yaml” were healthy.
- However, before this result: the set values for probe above caused K8s to considered these 3 pods unhealthy – because probe did the checking before the pods ran (applications in these pods were not yet opened port 8080).

Tested with the following commands:

```
haydenyeung@HaydenYeung-virtualbox:~/my-container$ kubectl get statefulsets
NAME       READY   AGE
node-web   3/3     18m
haydenyeung@HaydenYeung-virtualbox:~/my-container$ kubectl get services
NAME         TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
kubernetes   ClusterIP   10.152.183.1    <none>        443/TCP        51d
node-web     NodePort    10.152.183.66   <none>        80:31381/TCP   44m
haydenyeung@HaydenYeung-virtualbox:~/my-container$ kubectl get pods
NAME         READY   STATUS    RESTARTS   AGE
node-web-0   1/1     Running   0          18m
node-web-1   1/1     Running   0          18m
node-web-2   1/1     Running   0          18m
```

- Because in the content of the corrected version of “stateful-web.yaml”, the “kind” value was changed from “Deployment” to “StatefulSet” → There was no deployment to be found.

- A StatefulSet named “node-web” was found instead. Service and pods were found because the new version of “stateful-web.yaml” still using the same Service specs → the same with the case of pods.

“kubectl describe statefulsets node-web”

```
haydenyeung@HaydenYeung-virtualbox:~/my-container$ kubectl describe statefulsets node-web
Name:               node-web
Namespace:          default
CreationTimestamp:  Mon, 21 Apr 2025 01:21:47 +1000
Selector:           app=node-web
Labels:             app=node-web
Annotations:        <none>
Replicas:           3 desired | 3 total
Update Strategy:    RollingUpdate
  Partition:        0
Pods Status:        3 Running / 0 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels:  app=node-web
  Containers:
   node-web:
    Image:         localhost:5000/node-web:1
    Port:          8080/TCP
    Host Port:     0/TCP
    Readiness:     http-get http://:http/ delay=10s timeout=1s period=5s #success=1 #failure=3
    Environment:   <none>
    Mounts:        <none>
  Volumes:         <none>
  Node-Selectors:  <none>
  Tolerations:     <none>
Volume Claims:     <none>
Events:            <none>
```

“kubectl describe pod node-web-0”

```
haydenyeung@HaydenYeung-virtualbox:~/my-container$ kubectl describe pod node-web-0
Name:             node-web-0
Namespace:        default
Priority:         0
Service Account:  default
Node:             haydenyeung-virtualbox/10.0.2.15
Start Time:       Mon, 21 Apr 2025 01:21:47 +1000
Labels:           app=node-web
                  apps.kubernetes.io/pod-index=0
                  controller-revision-hash=node-web-69d75c8578
                  statefulset.kubernetes.io/pod-name=node-web-0
Annotations:      cni.projectcalico.org/containerID: 056c9555a5a2482b633d26645c021cd0eb233b9ddae6e8f617ccf52e15c28b38
                  cni.projectcalico.org/podIP: 10.1.186.32/32
                  cni.projectcalico.org/podIPs: 10.1.186.32/32
Status:           Running
IP:               10.1.186.32
IPs:
  IP:           10.1.186.32
Controlled By:  StatefulSet/node-web
Containers:
  node-web:
    Container ID:   containerd://7646f5547c0be9f61edac5e9f085f0cdbfa29958ad4ea0c0baccd09b263dcb00
    Image:          localhost:5000/node-web:1
    Image ID:       localhost:5000/node-web@sha256:5513a828705234896c35e2a6af03388fbd3818f32548940cd4df7502496f40cf
    Port:           8080/TCP
    Host Port:      0/TCP
    State:          Running
      Started:      Mon, 21 Apr 2025 01:21:48 +1000
    Ready:          True
    Restart Count:  0
    Readiness:      http-get http://:http/ delay=10s timeout=1s period=5s #success=1 #failure=3
```

- In “pod node-web-0”, I could see the line “Controlled By: StatefulSet/node-web”.

Try to Delete a pod through: “kubectl delete pod node-web-0”

```
LAST SEEN   TYPE      REASON                OBJECT                          MESSAGE
3m33s       Warning   FreeDiskSpaceFailed   node/haydenyeung-virtualbox   (combined from similar events)
: Failed to garbage collect required amount of images. Attempted to free 1933031833 bytes, but only foun
d 0 bytes eligible to free.
66s         Normal    Killing               pod/node-web-0                Stopping container node-web
35s         Normal    Scheduled             pod/node-web-0                Successfully assigned default/
node-web-0 to haydenyeung-virtualbox
34s         Normal    Pulled                pod/node-web-0                Container image "localhost:500
0/node-web:1" already present on machine
33s         Normal    Created               pod/node-web-0                Created container: node-web
33s         Normal    Started               pod/node-web-0                Started container node-web
2m36s       Warning   Unhealthy             pod/node-web-2                Readiness probe failed: Get "h
ttp://10.1.186.36:8080/": context deadline exceeded (Client.Timeout exceeded while awaiting headers)
```

- Pod “node-web-0” was recreated instead of a new pod name like in the case of Deployment.

Try to splitting a pod from the StatefulSet node-web through command: “KUBE_EDITOR='nano' kubectl edit pod node-web-0” and change the value of “label” from: “app: node-web” to “app: task61p”

```
5m15s       Normal    Killing              pod/node-web-0              Stopping container node-web
5m16s       Normal    SuccessfulCreate     statefulset/node-web        create Pod node-web-0 in State
fulSet node-web successful
5m15s       Normal    Scheduled            pod/node-web-0              Successfully assigned default/
node-web-0 to haydenyeung-virtualbox
5m15s       Normal    Pulled               pod/node-web-0              Container image "localhost:500
0/node-web:1" already present on machine
5m15s       Normal    Created              pod/node-web-0              Created container: node-web
5m15s       Normal    Started              pod/node-web-0              Started container node-web
```

```
haydenyeung@HaydenYeung-virtualbox:~/my-container$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
node-web-0    1/1     Running   0          7m18s
node-web-1    1/1     Running   0          40m
node-web-2    1/1     Running   0          40m
```

- This pod “node-web-0” was destroyed and recreated instead of being splitted to be an outsider like in Deployment scenarios.

Modified the original “myapp.js” file

```
const http = require('http');
const os = require('os');
const fs = require('fs');
const path = require('path');

const listenPort = 8080;
const stampFile = '/data/datestamp.txt'

console.log('v1 Server starting on host ' + os.hostname() + ' port ' + listenPort + '...');

const server = http.createServer((req, res) => {
        let clientIP = req.connection.remoteAddress;
        console.log('Processing request for ' + req.url + ' from ' + clientIP);

        try {
                res.writeHead(200);

                if (fs.existsSync(stampFile)) {
                        const data = fs.readFileSync(stampFile, 'utf8');
                        res.write(', I was created on' + data);
                } else {
                        res.write('No datestamp file is found\n');
                }

                res.end('Hello ' + clientIP + ', this is v1 on ' + os.hostname() + '\n');
        } catch (error) {
                res.writeHead(500);
```

Add the following code to “Container” spec in “stateful-web.yaml” and change the image name from localhost:5000/node-web:1 to localhost:5000/node-web:4 (despite that I forgot to change the v1 to v4 inside the context of “myapp.js”).

```
        volumeMounts:
        - name: data
          mountPath: /data
  volumeClaimTemplates:
  - metadata:
      name: data
    spec:
      accessModes: [ "ReadWriteOnce" ]
      resources:
        requests:
          storage: 1Gi
```

“kubectl apply -f stateful-web.yaml”

```
haydenyeung@HaydenYeung-virtualbox:~/my-container$ kubectl apply -f stateful-web.yaml
statefulset.apps/node-web created
service/node-web unchanged
haydenyeung@HaydenYeung-virtualbox:~/my-container$ kubectl get statefulset
NAME       READY   AGE
node-web   3/3     91s
haydenyeung@HaydenYeung-virtualbox:~/my-container$ kubectl get pods
NAME         READY   STATUS    RESTARTS   AGE
node-web-0   1/1     Running   0          97s
node-web-1   1/1     Running   0          83s
node-web-2   1/1     Running   0          69s
haydenyeung@HaydenYeung-virtualbox:~/my-container$ kubectl get pvc
NAME              STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS        VOLUMEATTRIB
UTESCLASS   AGE
data-node-web-0   Bound    pvc-86d2cf98-4ed0-40de-bb39-8c159a5dbf58   1Gi        RWO            microk8s-hostpath   <unset>
            11m
data-node-web-1   Bound    pvc-ab89aff7-c235-4b32-8d21-14db86b054f4   1Gi        RWO            microk8s-hostpath   <unset>
            10m
data-node-web-2   Bound    pvc-267aae81-394e-497b-8221-59c087494e7d   1Gi        RWO            microk8s-hostpath   <unset>
            10m
```

- Each pod has its own PVC.

“kubectl exec -it node-web-0 – sh -c “date > /data/datestamp.txt” & “kubectl port-forward node-web-0 8080:8080”

```
haydenyeung@HaydenYeung-virtualbox:~/my-container$ kubectl exec -it node-web-0 -- sh -c "date > /data/datestamp.txt"
haydenyeung@HaydenYeung-virtualbox:~/my-container$ kubectl port-forward node-web-0 8080:8080
Forwarding from 127.0.0.1:8080 -> 8080
Forwarding from [::1]:8080 -> 8080
Handling connection for 8080
```

```
haydenyeung@HaydenYeung-virtualbox: ~
haydenyeung@HaydenYeung-virtualbox:~$ curl http://localhost:8080/
, I was created onSun Apr 20 17:32:49 UTC 2025
Hello ::ffff:127.0.0.1, this is v1 on node-web-0
haydenyeung@HaydenYeung-virtualbox:~$
```

Testing with data storage of node-web-0

```
haydenyeung@HaydenYeung-virtualbox:~/my-container$ kubectl exec node-web-0 -- sh -c "echo 'Data from node-web-0' > /data/test.tx
t"
haydenyeung@HaydenYeung-virtualbox:~/my-container$ kubectl delete pod node-web-0
pod "node-web-0" deleted
haydenyeung@HaydenYeung-virtualbox:~/my-container$ kubectl exec node-web-0 -- cat /data/test.txt
Data from node-web-0
haydenyeung@HaydenYeung-virtualbox:~/my-container$ kubectl exec node-web-1 -- cat /data/text.txt
cat: /data/text.txt: No such file or directory
command terminated with exit code 1
haydenyeung@HaydenYeung-virtualbox:~/my-container$ kubectl exec node-web-2 -- cat /data/text.txt
cat: /data/text.txt: No such file or directory
command terminated with exit code 1
haydenyeung@HaydenYeung-virtualbox:~/my-container$
```

- /data/test.txt only presents in node-web-0 but not in other pods.
- Data still persisting after the original node-web-0 is deleted and “equipped” back on the new recreated node-web-0 (automated by StatefulSet).

3. Quiz Result