Deployment and Configuring Autoscaling For Stateless Application

Contents

1 Introduction 2

2 Documentation 3
- 2.1 Kubernetes Documentation 3
- 2.2 Linux Commands and VIM Commands 3

3 Pre-Requisite 4

4 Deploying Scalable Application & Running Nginx server 5
- 4.1 Scaling Nginx Deployment Replicas Using Scale Command 7
- 4.2 Auto-Healing With Deployment Controller 8
- 4.3 Update Deployment & Rolling Out New Versions Pod 8
- 4.4 Update Deployment & Fixing the Failed Pods 10
- 4.5 Rollback to previous Deployment version 15
- 4.6 Playing with Deployment Parameters – Canary Deployment 15
- 4.7 Deleting the resources created in this lab exercise 17

5 Configuring Autoscaling for Stateless Application 18
- 5.1 Installing metrics-server in the kubernetes cluster 18
- 5.2 Creating Deployment With Resource Limit Defined 19
- 5.3 Verify Cluster & Pod Level Metrics By Metrics-Server 21
- 5.4 Creating Horizontal Pod Autoscaler (HPA) 22
- 5.5 Demonstrating autoscaling of pod on load increase 23
- 5.6 Deleting the deployment resource created in this lab Exercise 25

6 Troubleshooting 26
- 6.1 Getting Error While Creating nginx-hpa.yaml File 26
- 6.2 Getting Error while Creating New Pod for Auto-Scaling 27
- 6.3 Getting Error While Creating nginx-hpa.yaml File 28

7 Question 30

8 Summry 31

1 INTRODUCTION

A **Kubernetes cluster** is a set of node machines for running containerized applications. If you're running **Kubernetes**, you're running a **cluster**. At a minimum, a **cluster** contains a control plane and one or more compute machines, or nodes.

This guide Covers:

- Deploying Scalable Stateless Application
- Configuring Autoscaling For Stateless Application

2 DOCUMENTATION

2.1 Kubernetes Documentation

1. Autoscaling in Kubernetes

   https://kubernetes.io/blog/2016/07/autoscaling-in-kubernetes/
2. Bootstrapping clusters with kubeadm

   https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/
3. Troubleshooting kubeadm

   https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/troubleshooting-kubeadm/
4. Using kubectl to Create a Deployment

   https://kubernetes.io/docs/tutorials/kubernetes-basics/deploy-app/deploy-intro/
5. Cluster Networking

   https://kubernetes.io/docs/concepts/cluster-administration/networking/

2.2 Linux Commands and VIM Commands

1. Basic Linux Commands

   https://maker.pro/linux/tutorial/basic-linux-commands-for-beginners

   https://www.hostinger.in/tutorials/linux-commands
2. Basic VIM Commands

   https://coderwall.com/p/adv71w/basic-vim-commands-for-getting-started
3. Popular VIM Commands

   https://www.keycdn.com/blog/vim-commands

3 PRE-REQUISITE

Ensure that you have completed following three activity guides (or you have an Ubuntu Server)

- Create account (Trial or Paid) on Azure Cloud.

  Note*: Follow Activity Guide* ***AG_ Deploy_App_On_Pod_&_Basic_Networking_ed*** *** *from portal*

4 DEPLOYING SCALABLE APPLICATION & RUNNING NGINX SERVER

Note: No need to copy paste or type this file manually you can get this file from

```
$ git clone https://github.com/mamtajha-ts/Kubernetes.git
$ cd Kubernetes
```

```
Cloning into 'Kubernetes'...
remote: Enumerating objects: 179, done.
remote: Counting objects: 100% (179/179), done.
remote: Compressing objects: 100% (125/125), done.
remote: Total 179 (delta 61), reused 167 (delta 50), pack-reused 0
Receiving objects: 100% (179/179), 17.08 MiB | 39.22 MiB/s, done.
Resolving deltas: 100% (61/61), done.
root@master:/home/ubuntu# cd Kubernetes/
root@master:/home/ubuntu/Kubernetes# ls
Dockerfile                          nfs-pvc.yaml
README.md                           nfspv-pod.yaml
__pycache__                         nginx-deployment.yaml
app.py                              nginx-hpa.yaml
apple.yaml                          nginx-svc.yaml
banana.yaml                         nodeaffinity-deployment.yaml
counter-pod.yaml                    nodeaffinity1-deployment.yaml
daemonset.yaml                      nodeanti-affinity-deployment.yaml
demo-pod.yaml                       nodeanti-affinity1-deployment.yaml
docker-compose.yaml                 oke-admin-service-account.yaml
elasticsearch-rbac.yaml             pod-dynamicpv-oci.yaml
elasticsearch-stfullset-oci.yaml    pod-dynamicpv.yaml
elasticsearch-stfullset.yaml        podaffinity-deployment.yaml
elasticsearch-svc.yaml              podaffinity1-deployment.yaml
elasticsearch.yaml                  podanti-affinity-deployment.yaml
example-ingress.yaml                podanti-affinity1-deployment.yaml
filebeat-agent.yaml                 pvc-oci.yaml
fluentd.yaml                        pvc.yaml
foo-allow-to-hello.yaml             quota-pod.yaml
guestbook-frontend-svc.yaml         quota-pod1.yaml
guestbook-frontend.yaml             quota.yaml
hello-allow-from-foo.yaml           readiness-pod.yaml
ingress-app1.yaml                   readiness-svc.yaml
ingress-app2.yaml                   redis-master-svc.yaml
ingress-route.yaml                  redis-master.yaml
kibana-elk.yaml                     redis-slave-svc.yaml
kibana.yaml                         redis-slave.yaml
label-deployment.yaml               requirements.txt
liveness-pod.yaml                   role-dev.yaml
logstash-configmap.yaml             rolebind.yaml
logstash-deployment.yaml            script.sh
logstash-svc.yaml                   security-cxt-nonroot.yaml
metrics-server.yaml                 security-cxt-priv.yaml
multi-container.yaml                security-cxt-readonly.yaml
multi-pod-configmap.yaml            security-cxt-rmcap.yaml
multi-pod-nginx.yaml                security-cxt-time.yaml
multi-prod-consumer.yaml            security-cxt.yaml
namespace.yaml                      tt-pod.yaml
network-policy.yaml                 tt-pod1.yaml
nfs-pv.yaml
root@master:/home/ubuntu/Kubernetes#
```

1. Create the nginx-deployment.yaml file to define the highly available nginx server

```
$ vim nginx-deployment.yaml
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.12
        ports:
        - containerPort: 80
~
~
~
~
~
~
~
~
```

2. Creating resources using the nginx-deployment.yaml file

```
$ kubectl create -f nginx-deployment.yaml
```

```
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl create -f nginx-deployment.yaml
deployment.apps/nginx-deployment created
```

3. Listing the deployments in the cluster

```
$ kubectl get deployment
```

```
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl get deployment
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   2/2     2            2           71s
root@kubeadm-master:/home/ubuntu/Kubernetes#
```

4. Listing all resources running in default namespace in the cluster

```
$ kubectl get all
```

```
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl get all
NAME                                     READY   STATUS    RESTARTS   AGE
pod/nginx-deployment-7c68bd84c6-5xjk9    1/1     Running   0          10s
pod/nginx-deployment-7c68bd84c6-hvsjv    1/1     Running   0          10s

NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   37d

NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nginx-deployment    2/2     2            2           10s

NAME                                           DESIRED   CURRENT   READY   AGE
replicaset.apps/nginx-deployment-7c68bd84c6    2         2         2       10s
root@kubeadm-master:/home/ubuntu/Kubernetes#
```

5. Listing all the pod and using -o wide option to list the nodes hosting the nginx replicas

```
$ kubectl get pods -o wide
```

```
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl get pods -o wide
NAME                                READY   STATUS    RESTARTS   AGE    IP          NODE      NOMINATED NODE   READINESS GATES
nginx-deployment-7c68bd84c6-5xjk9   1/1     Running   0          107s   10.32.0.2   worker2   <none>           <none>
nginx-deployment-7c68bd84c6-hvsjv   1/1     Running   0          107s   10.32.0.3   worker2   <none>           <none>
root@kubeadm-master:/home/ubuntu/Kubernetes#
```

4.1 Scaling Nginx Deployment Replicas Using Scale Command

1. Using kubectl scale command to specify the number of replicas for the nginx deployment

```
$ kubectl scale --replicas=5 deployment nginx-deployment
```

```
root@kubeadm-master:/home/ubuntu/Kubernetes#
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl scale --replicas=5 deployment nginx-deployment
deployment.apps/nginx-deployment scaled
root@kubeadm-master:/home/ubuntu/Kubernetes#
```

2. Listing all the pod replicas and using -o wide option to list the nodes hosting the nginx replicas

```
$ kubectl get pods -o wide
```

```
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl get pods -o wide
NAME                                READY   STATUS    RESTARTS   AGE    IP          NODE      NOMINATED NODE   READINESS GATES
nginx-deployment-7c68bd84c6-5xjk9   1/1     Running   0          3m3s   10.32.0.2   worker2   <none>           <none>
nginx-deployment-7c68bd84c6-86lmk   1/1     Running   0          56s    10.36.0.2   worker1   <none>           <none>
nginx-deployment-7c68bd84c6-hvsjv   1/1     Running   0          3m3s   10.32.0.3   worker2   <none>           <none>
nginx-deployment-7c68bd84c6-r98s6   1/1     Running   0          56s    10.32.0.4   worker2   <none>           <none>
nginx-deployment-7c68bd84c6-rs72l   1/1     Running   0          56s    10.36.0.3   worker1   <none>           <none>
root@kubeadm-master:/home/ubuntu/Kubernetes#
```

3. Describing the deployment resource so see detailed information about the nginx deployment

```
$ kubectl describe deployment nginx
```

```
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl describe deployment nginx
Name:                   nginx-deployment
Namespace:              default
CreationTimestamp:      Tue, 25 Aug 2020 03:44:18 +0000
Labels:                 app=nginx
Annotations:            deployment.kubernetes.io/revision: 1
Selector:               app=nginx
Replicas:               5 desired | 5 updated | 5 total | 5 available | 0 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=nginx
  Containers:
   nginx:
    Image:        nginx:1.12
    Port:         80/TCP
    Host Port:    0/TCP
    Environment:  <none>
    Mounts:       <none>
  Volumes:        <none>
Conditions:
  Type           Status  Reason
  ----           ------  ------
  Progressing    True    NewReplicaSetAvailable
  Available      True    MinimumReplicasAvailable
OldReplicaSets:  <none>
NewReplicaSet:   nginx-deployment-7c68bd84c6 (5/5 replicas created)
Events:
  Type    Reason             Age    From                   Message
  ----    ------             ----   ----                   -------
  Normal  ScalingReplicaSet  3m25s  deployment-controller  Scaled up replica set nginx-deployment-7c68bd84c6 to 2
  Normal  ScalingReplicaSet  78s    deployment-controller  Scaled up replica set nginx-deployment-7c68bd84c6 to 5

root@kubeadm-master:/home/ubuntu/Kubernetes#
```

4.2 Auto-Healing With Deployment Controller

1. Deleting one the pods manually and observing the auto-healing behaviour of deployment

Note: This nginx-deployment-7c68bd84c6-5xjk9 is for example please don't use this in command delete your OWN pod.

$ kubectl delete pod nginx-deployment-7c68bd84c6-5xjk9

$ kubectl get pods

```
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl delete pod nginx-deployment-7c68bd84c6-5xjk9
pod "nginx-deployment-7c68bd84c6-5xjk9" deleted
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-7c68bd84c6-86lmk   1/1     Running   0          2m47s
nginx-deployment-7c68bd84c6-dcmpz   1/1     Running   0          10s
nginx-deployment-7c68bd84c6-hvsjv   1/1     Running   0          4m54s
nginx-deployment-7c68bd84c6-r98s6   1/1     Running   0          2m47s
nginx-deployment-7c68bd84c6-rs72l   1/1     Running   0          2m47s
root@kubeadm-master:/home/ubuntu/Kubernetes#
```

4.3 Update Deployment & Rolling Out New Versions Pod

1. Using kubectl set command to update the nginx image version

```
$ kubectl --record deployment/nginx-deployment set image deployment/nginx-deployment nginx=nginx:1.17.0
```

```
root@kubeadm-master:/home/ubuntu/Kubernetes#
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl --record deployment/nginx-deployment set image deployment/nginx-deployment nginx=nginx:1.17.0
deployment.apps/nginx-deployment image updated
deployment.apps/nginx-deployment image updated
root@kubeadm-master:/home/ubuntu/Kubernetes#
```

2. Describe the deployment to confirm the image version is updated to nginx:1.17.0

```
$ kubectl describe deployment nginx
```

```
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl describe deployment nginx
Name:                   nginx-deployment
Namespace:              default
CreationTimestamp:      Tue, 25 Aug 2020 03:44:18 +0000
Labels:                 app=nginx
Annotations:            deployment.kubernetes.io/revision: 2
                        kubernetes.io/change-cause: kubectl deployment/nginx-deployment set image deployment/nginx-deployment nginx=nginx:1.17.0 --record=true
Selector:               app=nginx
Replicas:               5 desired | 5 updated | 5 total | 5 available | 0 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=nginx
  Containers:
   nginx:
    Image:        nginx:1.17.0
    Port:         80/TCP
    Host Port:    0/TCP
    Environment:  <none>
    Mounts:       <none>
  Volumes:        <none>
Conditions:
  Type           Status  Reason
  ----           ------  ------
  Available      True    MinimumReplicasAvailable
  Progressing    True    NewReplicaSetAvailable
OldReplicaSets:  <none>
NewReplicaSet:   nginx-deployment-599969c8d9 (5/5 replicas created)
Events:
  Type    Reason             Age   From                   Message
  ----    ------             ----  ----                   -------
  Normal  ScalingReplicaSet  20m   deployment-controller  Scaled up replica set nginx-deployment-7c68bd84c6 to 2
  Normal  ScalingReplicaSet  17m   deployment-controller  Scaled up replica set nginx-deployment-7c68bd84c6 to 5
  Normal  ScalingReplicaSet  57s   deployment-controller  Scaled up replica set nginx-deployment-599969c8d9 to 2
  Normal  ScalingReplicaSet  57s   deployment-controller  Scaled down replica set nginx-deployment-7c68bd84c6 to 4
  Normal  ScalingReplicaSet  57s   deployment-controller  Scaled up replica set nginx-deployment-599969c8d9 to 3
  Normal  ScalingReplicaSet  55s   deployment-controller  Scaled down replica set nginx-deployment-7c68bd84c6 to 3
  Normal  ScalingReplicaSet  55s   deployment-controller  Scaled up replica set nginx-deployment-599969c8d9 to 4
  Normal  ScalingReplicaSet  55s   deployment-controller  Scaled down replica set nginx-deployment-7c68bd84c6 to 2
  Normal  ScalingReplicaSet  55s   deployment-controller  Scaled up replica set nginx-deployment-599969c8d9 to 5
  Normal  ScalingReplicaSet  55s   deployment-controller  Scaled down replica set nginx-deployment-7c68bd84c6 to 1
  Normal  ScalingReplicaSet  53s   deployment-controller  Scaled down replica set nginx-deployment-7c68bd84c6 to 0
```

3. Verify the status of deployment, replicaset and pods. Notice that new replicaset is created and new set of pods are also running to server the newer version

```
$ kubectl get pods

$ kubectl get all
```

```
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-599969c8d9-d8gsg   1/1     Running   0          97s
nginx-deployment-599969c8d9-d9rlh   1/1     Running   0          95s
nginx-deployment-599969c8d9-dqjhk   1/1     Running   0          97s
nginx-deployment-599969c8d9-qdvn8   1/1     Running   0          97s
nginx-deployment-599969c8d9-zvrt9   1/1     Running   0          95s
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl get all
NAME                                    READY   STATUS    RESTARTS   AGE
pod/nginx-deployment-599969c8d9-d8gsg   1/1     Running   0          103s
pod/nginx-deployment-599969c8d9-d9rlh   1/1     Running   0          101s
pod/nginx-deployment-599969c8d9-dqjhk   1/1     Running   0          103s
pod/nginx-deployment-599969c8d9-qdvn8   1/1     Running   0          103s
pod/nginx-deployment-599969c8d9-zvrt9   1/1     Running   0          101s

NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   37d

NAME                               READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nginx-deployment   5/5     5            5           20m

NAME                                          DESIRED   CURRENT   READY   AGE
replicaset.apps/nginx-deployment-599969c8d9   5         5         5       103s
replicaset.apps/nginx-deployment-7c68bd84c6   0         0         0       20m
root@kubeadm-master:/home/ubuntu/Kubernetes#
```

4. Display the rollout history of the deployment to verify the number of revisions

```
$ kubectl rollout history deployment/nginx-deployment
```

```
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl rollout history deployment/nginx-deployment
deployment.apps/nginx-deployment
REVISION  CHANGE-CAUSE
1         <none>
2         kubectl deployment/nginx-deployment set image deployment/nginx-deployment nginx=nginx:1.17.0 --record=true

root@kubeadm-master:/home/ubuntu/Kubernetes#
```

4.4 Update Deployment & Fixing the Failed Pods

1. Using kubectl set command to update the nginx image version

```
$ kubectl set image deployment/nginx-deployment nginx=nginx:1.191 --record=true
```

```
root@kubeadm-master:/home/ubuntu/Kubernetes#
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl set image deployment/nginx-deployment nginx=nginx:1.191 --record=true
deployment.apps/nginx-deployment image updated
```

2. Display the rollout history of the deployment to verify the number of revisions

```
$ kubectl rollout history deployment/nginx-deployment
```

```
root@kubeadm-master:/home/ubuntu/Kubernetes#
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl rollout history deployment/nginx-deployment
deployment.apps/nginx-deployment
REVISION  CHANGE-CAUSE
1         <none>
2         kubectl deployment/nginx-deployment set image deployment/nginx-deployment nginx=nginx:1.17.0 --record=true
3         kubectl set image deployment/nginx-deployment nginx=nginx:1.191 --record=true
```

3. Describe the deployment to verify if all pods are rollout. But you would find that the rollout is not successful and deployment is stuck.

$ kubectl get deployment nginx-deployment

```
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl get deployment
NAME                 READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment     4/5     3            4           26m
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl get pods
NAME                                 READY   STATUS         RESTARTS   AGE
nginx-deployment-54c65fbcf8-bghvl    0/1     ErrImagePull   0          44s
nginx-deployment-54c65fbcf8-fgg6z    0/1     ErrImagePull   0          44s
nginx-deployment-54c65fbcf8-mnvh7    0/1     ErrImagePull   0          44s
nginx-deployment-599969c8d9-d8gsg    1/1     Running        0          7m7s
nginx-deployment-599969c8d9-dqjhk    1/1     Running        0          7m7s
nginx-deployment-599969c8d9-qdvn8    1/1     Running        0          7m7s
nginx-deployment-599969c8d9-zvrt9    1/1     Running        0          7m5s
root@kubeadm-master:/home/ubuntu/Kubernetes#
```

4. Describe the deployment to check the reason for the stucked deployment

$ kubectl describe deployment nginx

```
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl describe deployment nginx
Name:                   nginx-deployment
Namespace:              default
CreationTimestamp:      Tue, 25 Aug 2020 03:44:18 +0000
Labels:                 app=nginx
Annotations:            deployment.kubernetes.io/revision: 3
                        kubernetes.io/change-cause: kubectl set image deployment/nginx-deployment nginx=nginx:1.191 --record=true
Selector:               app=nginx
Replicas:               5 desired | 3 updated | 7 total | 4 available | 3 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=nginx
  Containers:
   nginx:
    Image:        nginx:1.191
    Port:         80/TCP
    Host Port:    0/TCP
    Environment:  <none>
    Mounts:       <none>
  Volumes:        <none>
Conditions:
  Type           Status  Reason
  ----           ------  ------
  Available      True    MinimumReplicasAvailable
  Progressing    True    ReplicaSetUpdated
OldReplicaSets:  nginx-deployment-599969c8d9 (4/4 replicas created)
NewReplicaSet:   nginx-deployment-54c65fbcf8 (3/3 replicas created)
Events:
  Type    Reason             Age                    From                   Message
  ----    ------             ----                   ----                   -------
  Normal  ScalingReplicaSet  27m                    deployment-controller  Scaled up replica set nginx-deployment-7c68bd84c6 to 2
  Normal  ScalingReplicaSet  25m                    deployment-controller  Scaled up replica set nginx-deployment-7c68bd84c6 to 5
  Normal  ScalingReplicaSet  8m41s                  deployment-controller  Scaled up replica set nginx-deployment-599969c8d9 to 2
  Normal  ScalingReplicaSet  8m41s                  deployment-controller  Scaled down replica set nginx-deployment-7c68bd84c6 to 4
  Normal  ScalingReplicaSet  8m41s                  deployment-controller  Scaled up replica set nginx-deployment-599969c8d9 to 3
  Normal  ScalingReplicaSet  8m39s                  deployment-controller  Scaled down replica set nginx-deployment-7c68bd84c6 to 3
  Normal  ScalingReplicaSet  8m39s                  deployment-controller  Scaled up replica set nginx-deployment-599969c8d9 to 4
  Normal  ScalingReplicaSet  8m39s                  deployment-controller  Scaled down replica set nginx-deployment-7c68bd84c6 to 2
  Normal  ScalingReplicaSet  8m39s                  deployment-controller  Scaled up replica set nginx-deployment-599969c8d9 to 5
  Normal  ScalingReplicaSet  8m39s                  deployment-controller  Scaled down replica set nginx-deployment-7c68bd84c6 to 1
  Normal  ScalingReplicaSet  8m37s                  deployment-controller  Scaled down replica set nginx-deployment-7c68bd84c6 to 0
  Normal  ScalingReplicaSet  2m18s (x3 over 2m18s)  deployment-controller  (combined from similar events): Scaled up replica set nginx-deployment-54c65fbcf8 to 3
```

5. Check the status of the rollout using status command

$ kubectl rollout status deployment/nginx-deployment

```
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl rollout status deployment/nginx-deployment
Waiting for deployment "nginx-deployment" rollout to finish: 3 out of 5 new replicas have been updated...
^Croot@kubeadm-master:/home/ubuntu/Kubernetes#
```

6. To investigate the failed/stuck deployment we will pause the deployment and make the changes to fix it

```
$ kubectl rollout pause deployment.v1.apps/nginx-deployment
```

```
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl rollout pause deployment.v1.apps/nginx-deployment
deployment.apps/nginx-deployment paused
```

7. Describe the deployment and see that the progress deadline has exceeded

```
$ kubectl describe deployment nginx
```

```
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl describe deployment nginx
Name:                   nginx-deployment
Namespace:              default
CreationTimestamp:      Tue, 25 Aug 2020 03:44:18 +0000
Labels:                 app=nginx
Annotations:            deployment.kubernetes.io/revision: 3
                        kubernetes.io/change-cause: kubectl set image deployment/nginx-deployment nginx=nginx:1.191 --record=true
Selector:               app=nginx
Replicas:               5 desired | 3 updated | 7 total | 4 available | 3 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=nginx
  Containers:
   nginx:
    Image:        nginx:1.191
    Port:         80/TCP
    Host Port:    0/TCP
    Environment:  <none>
    Mounts:       <none>
  Volumes:        <none>
Conditions:
  Type           Status  Reason
  ----           ------  ------
  Available      True    MinimumReplicasAvailable
  Progressing    False   ProgressDeadlineExceeded
OldReplicaSets:  nginx-deployment-599969c8d9 (4/4 replicas created)
NewReplicaSet:   nginx-deployment-54c65fbcf8 (3/3 replicas created)
Events:
  Type    Reason             Age                From                   Message
  ----    ------             ----               ----                   -------
  Normal  ScalingReplicaSet  39m                deployment-controller  Scaled up replica set nginx-deployment-7c68bd84c6 to 2
  Normal  ScalingReplicaSet  37m                deployment-controller  Scaled up replica set nginx-deployment-7c68bd84c6 to 5
  Normal  ScalingReplicaSet  20m                deployment-controller  Scaled up replica set nginx-deployment-599969c8d9 to 2
  Normal  ScalingReplicaSet  20m                deployment-controller  Scaled down replica set nginx-deployment-7c68bd84c6 to 4
  Normal  ScalingReplicaSet  20m                deployment-controller  Scaled up replica set nginx-deployment-599969c8d9 to 3
  Normal  ScalingReplicaSet  20m                deployment-controller  Scaled down replica set nginx-deployment-7c68bd84c6 to 3
  Normal  ScalingReplicaSet  20m                deployment-controller  Scaled up replica set nginx-deployment-599969c8d9 to 4
  Normal  ScalingReplicaSet  20m                deployment-controller  Scaled down replica set nginx-deployment-7c68bd84c6 to 2
  Normal  ScalingReplicaSet  20m                deployment-controller  Scaled up replica set nginx-deployment-599969c8d9 to 5
  Normal  ScalingReplicaSet  20m                deployment-controller  Scaled down replica set nginx-deployment-7c68bd84c6 to 1
  Normal  ScalingReplicaSet  20m                deployment-controller  Scaled down replica set nginx-deployment-7c68bd84c6 to 0
  Normal  ScalingReplicaSet  14m (x3 over 14m)  deployment-controller  (combined from similar events): Scaled up replica set nginx-deployment-54c65fbcf8 to 3
```

8. Fix the deployment by setting the correct image.Using kubectl set command to update the nginx image version

```
$ kubectl set image deployment/nginx-deployment nginx=nginx:1.19.1 --record=true
```

```
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl set image deployment.v1.apps/nginx-deployment nginx=nginx:1.19.1 --record=true
deployment.apps/nginx-deployment image updated
```

9. Changes will not apply till we resume the deployment. Verify the deployment is still in paused state.

```
$ kubectl get pods
$ kubectl rollout history deployment/nginx-deployment
```

```
$ kubectl get rs
```

```
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl get pods
NAME                                READY   STATUS             RESTARTS   AGE
nginx-deployment-54c65fbcf8-bghvl   0/1     ErrImagePull       0          16m
nginx-deployment-54c65fbcf8-fgg6z   0/1     ImagePullBackOff   0          16m
nginx-deployment-54c65fbcf8-mnvh7   0/1     ImagePullBackOff   0          16m
nginx-deployment-599969c8d9-d8gsg   1/1     Running            0          22m
nginx-deployment-599969c8d9-dqjhk   1/1     Running            0          22m
nginx-deployment-599969c8d9-qdvn8   1/1     Running            0          22m
nginx-deployment-599969c8d9-zvrt9   1/1     Running            0          22m
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl rollout history deployment.v1.apps/nginx-deployment
deployment.apps/nginx-deployment
REVISION  CHANGE-CAUSE
1         <none>
2         kubectl deployment/nginx-deployment set image deployment/nginx-deployment nginx=nginx:1.17.0 --record=true
3         kubectl set image deployment/nginx-deployment nginx=nginx:1.191 --record=true

root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl get rs
NAME                          DESIRED   CURRENT   READY   AGE
nginx-deployment-54c65fbcf8   3         3         0       16m
nginx-deployment-599969c8d9   4         4         4       23m
nginx-deployment-7c68bd84c6   0         0         0       42m
```

10. So if we wish to make multiple fixes in our deployment we can go ahead with it. Lets make some more changes to the Pod definition.

```
$ kubectl set resources deployment.v1.apps/nginx-deployment -c=nginx --limits=cpu=200m,memory=512Mi
```

```
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl set resources deployment.v1.apps/nginx-deployment -c=nginx --limits=cpu=200m,memory=512Mi
deployment.apps/nginx-deployment resource requirements updated
root@kubeadm-master:/home/ubuntu/Kubernetes#
```

11. We can resume the deployment using resume command. So that all changes we have made to the deployment are applied in one go.

```
$ kubectl rollout resume deployment.v1.apps/nginx-deployment
```

```
root@kubeadm-master:/home/ubuntu/Kubernetes#
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl rollout resume deployment.v1.apps/nginx-deployment
deployment.apps/nginx-deployment resumed
```

12. Verify the rollout status and Pod status is up and running on fixing the deployment.

```
$ kubectl get rs
$ kubectl rollout history deployment.v1.apps/nginx-deployment
$ kubectl get pods
$ kubectl get deployment nginx-deployment
$ kubectl rollout status deployment.v1.apps/nginx-deployment
```

```
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl get rs
NAME                          DESIRED   CURRENT   READY   AGE
nginx-deployment-54c65fbcf8   0         0         0       17m
nginx-deployment-599969c8d9   0         0         0       24m
nginx-deployment-7c68bd84c6   0         0         0       43m
nginx-deployment-86986479c6   5         5         5       15s
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl rollout history deployment.v1.apps/nginx-deployment
deployment.apps/nginx-deployment
REVISION  CHANGE-CAUSE
1         <none>
2         kubectl deployment/nginx-deployment set image deployment/nginx-deployment nginx=nginx:1.17.0 --record=true
3         kubectl set image deployment/nginx-deployment nginx=nginx:1.191 --record=true
4         kubectl set image deployment.v1.apps/nginx-deployment nginx=nginx:1.19.1 --record=true

root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-86986479c6-2t9cj   1/1     Running   0          31s
nginx-deployment-86986479c6-7qb7t   1/1     Running   0          31s
nginx-deployment-86986479c6-9485c   1/1     Running   0          30s
nginx-deployment-86986479c6-ck8pz   1/1     Running   0          31s
nginx-deployment-86986479c6-l8vdb   1/1     Running   0          27s
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl get deployment nginx-deployment
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   5/5     5            5           43m
root@kubeadm-master:/home/ubuntu/Kubernetes#
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl rollout status deployment.v1.apps/nginx-deployment
deployment "nginx-deployment" successfully rolled out
```

$ kubectl describe deployment nginx-deployment

```
deployment "nginx-deployment" successfully rolled out
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl describe deployment nginx-deployment
Name:                   nginx-deployment
Namespace:              default
CreationTimestamp:      Tue, 25 Aug 2020 03:44:18 +0000
Labels:                 app=nginx
Annotations:            deployment.kubernetes.io/revision: 4
                        kubernetes.io/change-cause: kubectl set image deployment.v1.apps/nginx-deployment nginx=nginx:1.19.1 --record=true
Selector:               app=nginx
Replicas:               5 desired | 5 updated | 5 total | 5 available | 0 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=nginx
  Containers:
   nginx:
    Image:      nginx:1.19.1
    Port:       80/TCP
    Host Port:  0/TCP
    Limits:
      cpu:        200m
      memory:     512Mi
    Environment:  <none>
    Mounts:       <none>
  Volumes:        <none>
Conditions:
  Type           Status  Reason
  ----           ------  ------
  Available      True    MinimumReplicasAvailable
  Progressing    True    NewReplicaSetAvailable
OldReplicaSets:  <none>
NewReplicaSet:   nginx-deployment-86986479c6 (5/5 replicas created)
Events:
  Type    Reason             Age                From                   Message
  ----    ------             ----               ----                   -------
  Normal  ScalingReplicaSet  44m                deployment-controller  Scaled up replica set nginx-deployment-7c68bd84c6 to 2
  Normal  ScalingReplicaSet  42m                deployment-controller  Scaled up replica set nginx-deployment-7c68bd84c6 to 5
  Normal  ScalingReplicaSet  25m                deployment-controller  Scaled up replica set nginx-deployment-599969c8d9 to 2
  Normal  ScalingReplicaSet  25m                deployment-controller  Scaled down replica set nginx-deployment-7c68bd84c6 to 4
  Normal  ScalingReplicaSet  25m                deployment-controller  Scaled up replica set nginx-deployment-599969c8d9 to 3
  Normal  ScalingReplicaSet  25m                deployment-controller  Scaled down replica set nginx-deployment-7c68bd84c6 to 3
  Normal  ScalingReplicaSet  25m                deployment-controller  Scaled up replica set nginx-deployment-599969c8d9 to 4
  Normal  ScalingReplicaSet  25m                deployment-controller  Scaled down replica set nginx-deployment-7c68bd84c6 to 2
  Normal  ScalingReplicaSet  25m                deployment-controller  Scaled up replica set nginx-deployment-599969c8d9 to 5
  Normal  ScalingReplicaSet  25m                deployment-controller  Scaled down replica set nginx-deployment-7c68bd84c6 to 1
  Normal  ScalingReplicaSet  25m                deployment-controller  Scaled down replica set nginx-deployment-7c68bd84c6 to 0
  Normal  ScalingReplicaSet  19m (x3 over 19m)  deployment-controller  (combined from similar events): Scaled up replica set nginx-deployment-54c65fbcf8 to 3
  Normal  ScalingReplicaSet  85s                deployment-controller  Scaled down replica set nginx-deployment-54c65fbcf8 to 0
  Normal  ScalingReplicaSet  84s                deployment-controller  Scaled up replica set nginx-deployment-86986479c6 to 3
  Normal  ScalingReplicaSet  83s                deployment-controller  Scaled down replica set nginx-deployment-599969c8d9 to 3
  Normal  ScalingReplicaSet  83s                deployment-controller  Scaled up replica set nginx-deployment-86986479c6 to 4
  Normal  ScalingReplicaSet  80s                deployment-controller  Scaled down replica set nginx-deployment-599969c8d9 to 2
  Normal  ScalingReplicaSet  80s                deployment-controller  Scaled up replica set nginx-deployment-86986479c6 to 5
  Normal  ScalingReplicaSet  80s                deployment-controller  Scaled down replica set nginx-deployment-599969c8d9 to 1
  Normal  ScalingReplicaSet  80s                deployment-controller  Scaled down replica set nginx-deployment-599969c8d9 to 0
```

4.5 Rollback to previous Deployment version

1. Rolling back the deployment to the previous version (revision = 2). Also verify pods are recreated with revision=2 version of image. And we are back with 5 pods with older image

```
$ kubectl rollout undo deployment/nginx-deployment --to-revision=2
$ kubectl get pods
$ kubectl rollout history deployment/nginx-deployment
```

```
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl rollout undo deployment/nginx-deployment --to-revision=2
deployment.apps/nginx-deployment rolled back
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl get pods
NAME                                READY   STATUS        RESTARTS   AGE
nginx-deployment-599969c8d9-5g5jq   1/1     Running       0          5s
nginx-deployment-599969c8d9-bv77g   1/1     Running       0          5s
nginx-deployment-599969c8d9-kjqbh   1/1     Running       0          5s
nginx-deployment-599969c8d9-nfm46   1/1     Running       0          3s
nginx-deployment-599969c8d9-t2bzq   1/1     Running       0          3s
nginx-deployment-86986479c6-7qb7t   1/1     Terminating   0          64m
nginx-deployment-86986479c6-9485c   0/1     Terminating   0          64m
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl rollout history deployment.v1.apps/nginx-deployment
deployment.apps/nginx-deployment
REVISION  CHANGE-CAUSE
1         <none>
3         kubectl set image deployment/nginx-deployment nginx=nginx:1.191 --record=true
4         kubectl set image deployment.v1.apps/nginx-deployment nginx=nginx:1.19.1 --record=true
5         kubectl deployment/nginx-deployment set image deployment/nginx-deployment nginx=nginx:1.17.0 --record=true

root@kubeadm-master:/home/ubuntu/Kubernetes#
```

4.6 Playing with Deployment Parameters – Canary Deployment

1. Modifying maxSurge parameter value from 25% to 90%. With this when can controller the percentage of pods we want to upgrade in one shot.
2. We can also modify the image version to nginx:1.18.

```
$ kubectl edit deployment nginx-deployment
```

```
spec:
  progressDeadlineSeconds: 600
  replicas: 5
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app: nginx
  strategy:
    rollingUpdate:
      maxSurge: 90%
      maxUnavailable: 25%
    type: RollingUpdate
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: nginx
    spec:
      containers:
      - image: nginx:1.18
        imagePullPolicy: IfNotPresent
        name: nginx
        ports:
        - containerPort: 80
          protocol: TCP
        resources: {}
```

3. Lets apply the modified changes and update the deployment

```
$ kubectl get all
$ kubectl describe deployment nginx-deployment
```

Note: That 90% of 5 .i.e. 4 of the replicas will get updated in one go and another one will update in next set.

```
root@kubeadm-master:/home/ubuntu/Kubernetes#
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl edit deployment nginx-deployment
^[[Adeployment.apps/nginx-deployment edited
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl get all
NAME                                    READY   STATUS        RESTARTS   AGE
pod/nginx-deployment-756ffb5cc8-6pn7m   1/1     Running       0          5s
pod/nginx-deployment-756ffb5cc8-gqbfr   1/1     Running       0          5s
pod/nginx-deployment-756ffb5cc8-gs5vp   1/1     Running       0          5s
pod/nginx-deployment-756ffb5cc8-gwsxd   1/1     Running       0          5s
pod/nginx-deployment-756ffb5cc8-gxtst   1/1     Running       0          5s
pod/nginx-deployment-7c68bd84c6-5qnk7   0/1     Terminating   0          5m11s
pod/nginx-deployment-7c68bd84c6-8l4pn   0/1     Terminating   0          5m11s
pod/nginx-deployment-7c68bd84c6-d6prj   0/1     Terminating   0          5m49s
pod/nginx-deployment-7c68bd84c6-fsn6b   0/1     Terminating   0          5m49s
pod/nginx-deployment-7c68bd84c6-tl8h4   0/1     Terminating   0          5m11s

NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   37d

NAME                               READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nginx-deployment   5/5     5            5           142m

NAME                                          DESIRED   CURRENT   READY   AGE
replicaset.apps/nginx-deployment-54c65fbcf8   0         0         0       117m
replicaset.apps/nginx-deployment-599969c8d9   0         0         0       123m
replicaset.apps/nginx-deployment-756ffb5cc8   5         5         5       8m53s
replicaset.apps/nginx-deployment-7c68bd84c6   0         0         0       142m
replicaset.apps/nginx-deployment-86986479c6   0         0         0       99m
```

```
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl describe deployment nginx-deployment
Name:                   nginx-deployment
Namespace:              default
CreationTimestamp:      Tue, 25 Aug 2020 03:44:18 +0000
Labels:                 app=nginx
Annotations:            deployment.kubernetes.io/revision: 10
                        kubernetes.io/change-cause: kubectl deployment/nginx-deployment set image deployment/nginx-deployment
Selector:               app=nginx
Replicas:               5 desired | 5 updated | 5 total | 5 available | 0 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 90% max surge
Pod Template:
  Labels:  app=nginx
  Containers:
   nginx:
    Image:        nginx:1.18
    Port:         80/TCP
    Host Port:    0/TCP
    Environment:  <none>
    Mounts:       <none>
  Volumes:        <none>
Conditions:
  Type           Status  Reason
  ----           ------  ------
  Available      True    MinimumReplicasAvailable
  Progressing    True    NewReplicaSetAvailable
OldReplicaSets:  <none>
NewReplicaSet:   nginx-deployment-756ffb5cc8 (5/5 replicas created)
```

4.7 Deleting the resources created in this lab exercise

```
$ kubectl delete -f nginx-deployment.yaml
```

5 CONFIGURING AUTOSCALING FOR STATELESS APPLICATION

5.1 Installing metrics-server in the kubernetes cluster

1. Viewing the contents of metrics-server.yaml file which has all the resource definition needed for metrics-server to run

```
$ vim metrics-server.yaml
```

2. Deploying the resources from metrics-server.yaml file

```
$ kubectl create -f metrics-server.yaml
```

```
root@kubeadm-master:/home/ubuntu/Kubernetes#
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl create -f metrics-server.yaml
clusterrole.rbac.authorization.k8s.io/system:aggregated-metrics-reader created
clusterrolebinding.rbac.authorization.k8s.io/metrics-server:system:auth-delegator created
rolebinding.rbac.authorization.k8s.io/metrics-server-auth-reader created
apiservice.apiregistration.k8s.io/v1beta1.metrics.k8s.io created
serviceaccount/metrics-server created
deployment.apps/metrics-server created
service/metrics-server created
clusterrole.rbac.authorization.k8s.io/system:metrics-server created
clusterrolebinding.rbac.authorization.k8s.io/system:metrics-server created
root@kubeadm-master:/home/ubuntu/Kubernetes#
root@kubeadm-master:/home/ubuntu/Kubernetes#
```

3. Verifying the status of the metrics-server deployment and pod created in kube-system namespace

```
$ kubectl get deployment metrics-server -n kube-system
```

```
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl get deployment metrics-server -n kube-system
NAME             READY   UP-TO-DATE   AVAILABLE   AGE
metrics-server   1/1     1            1           10m
root@kubeadm-master:/home/ubuntu/Kubernetes#
```

```
$ kubectl get pods -n kube-system
```

```
root@kubeadm-master:/home/ubuntu/Kubernetes#
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl get pods -n kube-system
NAME                                     READY   STATUS    RESTARTS   AGE
coredns-66bff467f8-2rm87                 1/1     Running   1          26h
coredns-66bff467f8-p79n6                 1/1     Running   1          26h
etcd-kubeadm-master                      1/1     Running   1          26h
kube-apiserver-kubeadm-master            1/1     Running   1          26h
kube-controller-manager-kubeadm-master   1/1     Running   2          26h
kube-proxy-2l4dc                         1/1     Running   1          25h
kube-proxy-kjnrs                         1/1     Running   1          26h
kube-proxy-psr9m                         1/1     Running   1          25h
kube-scheduler-kubeadm-master            1/1     Running   2          26h
metrics-server-5577df4865-bhbcq          1/1     Running   0          10s
weave-net-6mqkq                          2/2     Running   3          25h
weave-net-78tzp                          2/2     Running   3          25h
weave-net-nhlbc                          2/2     Running   3          25h
root@kubeadm-master:/home/ubuntu/Kubernetes#
```

5.2 Creating Deployment With Resource Limit Defined

1. Create file called nginx-hpa.yaml file and define the resource limit and resource for the nginx server

```
$ vim nginx-hpa.yaml
```

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-hpa
  labels:
    app: nginx-hpa
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx-hpa
  template:
    metadata:
      labels:
        app: nginx-hpa
    spec:
      containers:
      - image: nginx
        name: nginx-cont
        resources:
          limits:
            cpu: 300m
            memory: 200Mi
          requests:
            cpu: 150m
            memory: 100Mi

~
```

2. Create resource using nginx-hpa.yaml file

```
$ kubectl create -f nginx-hpa.yaml
```

```
root@kubeadm-master:/home/ubuntu/Kubernetes#
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl create -f nginx-hpa.yaml
deployment.apps/nginx-hpa created
root@kubeadm-master:/home/ubuntu/Kubernetes#
```

3. Verify the status of newly created deployment and pod

```
$ kubectl get all
```

```
root@kubeadm-master:/home/ubuntu/Kubernetes#
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl get all
NAME                                  READY   STATUS    RESTARTS   AGE
pod/nginx-deployment-7c68bd84c6-txq9r   1/1     Running   1          12h
pod/nginx-hpa-568d658b68-gjlql          1/1     Running   0          30s

NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   29h

NAME                              READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nginx-deployment   1/1     1            1           12h
deployment.apps/nginx-hpa          1/1     1            1           30s

NAME                                         DESIRED   CURRENT   READY   AGE
replicaset.apps/nginx-deployment-7c68bd84c6   1         1         1       12h
replicaset.apps/nginx-hpa-568d658b68          1         1         1       30s
root@kubeadm-master:/home/ubuntu/Kubernetes#
```

5.3 Verify Cluster & Pod Level Metrics By Metrics-Server

1. Use kubectl top command to display cluster and pod level metrics

```
$ kubectl top nodes
```

```
root@kubeadm-master:/home/ubuntu/Kubernetes#
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl top nodes
NAME             CPU(cores)   CPU%   MEMORY(bytes)   MEMORY%
kubeadm-master   98m          4%     1149Mi          30%
worker1          33m          3%     446Mi           24%
worker2          31m          3%     648Mi           36%
root@kubeadm-master:/home/ubuntu/Kubernetes#
```

```
$ kubectl top pods
```

```
root@kubeadm-master:/home/ubuntu/Kubernetes#
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl top pods
NAME                                CPU(cores)   MEMORY(bytes)
nginx-deployment-7c68bd84c6-txq9r   0m           3Mi
nginx-hpa-568d658b68-gjlql          0m           3Mi
root@kubeadm-master:/home/ubuntu/Kubernetes#
```

5.4 Creating Horizontal Pod Autoscaler (HPA)

1. Creating HPA for the nginx-deployment deployment resourcespecifying the auto-scaling condition and limits

```
$ kubectl autoscale deployment nginx-hpa --cpu-percent=2 --min=1 --max=3
```

```
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl autoscale deployment nginx-hpa --cpu-percent=2 --min=1 --max=3
horizontalpodautoscaler.autoscaling/nginx-hpa autoscaled
root@kubeadm-master:/home/ubuntu/Kubernetes#
```

2. Listing the HPAs in the cluster and seeing the details provided in the command output

```
$ kubectl get hpa
```

```
root@kubeadm-master:/home/ubuntu/Kubernetes#
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl get hpa
NAME        REFERENCE             TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
nginx-hpa   Deployment/nginx-hpa  0%/2%     1         3         1          22s
root@kubeadm-master:/home/ubuntu/Kubernetes#
```

3. Listing all the resources in default namespace in the cluster

```
$ kubectl get all
```

```
root@kubeadm-master:/home/ubuntu/Kubernetes#
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl get all
NAME                                  READY   STATUS    RESTARTS   AGE
pod/nginx-deployment-7c68bd84c6-txq9r 1/1     Running   1          13h
pod/nginx-hpa-568d658b68-gjlql        1/1     Running   0          4m12s

NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   29h

NAME                              READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nginx-deployment  1/1     1            1           13h
deployment.apps/nginx-hpa         1/1     1            1           4m12s

NAME                                        DESIRED   CURRENT   READY   AGE
replicaset.apps/nginx-deployment-7c68bd84c6 1         1         1       13h
replicaset.apps/nginx-hpa-568d658b68        1         1         1       4m12s

NAME                                        REFERENCE             TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
horizontalpodautoscaler.autoscaling/nginx-hpa Deployment/nginx-hpa 0%/2%     1         3         1          49s
root@kubeadm-master:/home/ubuntu/Kubernetes#
```

5.5 Demonstrating autoscaling of pod on load increase

1. Get the pods ip address so that we can curl on that nginx container ip address

```
$ kubectl get pods -o wide
```

```
root@kubeadm-master:/home/AzureUser/Kubernetes# kubectl get pods -o wide
NAME                        READY   STATUS    RESTARTS   AGE    IP          NODE      NOMINATED NODE   READINESS GATES
nginx-hpa-568d658b68-zvt2v  1/1     Running   0          2m7s   10.38.0.2   worker1   <none>           <none>
root@kubeadm-master:/home/AzureUser/Kubernetes#
```

2. Creating a load-generator pod which shall create continuous request on the nginx server

3. Open a separate ssh terminal and set the cluster config on that terminal

```
$ sudo su
$ export KUBECONFIG=$HOME/admin.conf
```

4. execute the below command to create a new pod

```
$ kubectl run -it --rm load-generator --image=busybox /bin/sh
```

5. Execute an infinite loop within the pod to load the nginx-hpa pod (ip address of the pod was fetched in step 1)

```
# while true; do wget -q -O- http://10.38.0.2; done
```

```
root@kubeadm-master:/home/ubuntu#
root@kubeadm-master:/home/ubuntu#
root@kubeadm-master:/home/ubuntu# kubectl run -it --rm load-generator --image=busybox /bin/sh
If you don't see a command prompt, try pressing enter.
/ #
/ #
/ # while true; do wget -q -O- http://10.46.0.2; done
```

6. Get back to the previous terminal. Check the status of HPA and check that the resource utilization increases in nginx-hpa pod and it scales to max number of pods

```
$ kubectl get hpa
$ kubectl top pods
$ kubectl get pods
```

```
root@kubeadm-master:/home/ubuntu/Kubernetes#
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl get hpa
NAME        REFERENCE              TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
nginx-hpa   Deployment/nginx-hpa   0%/2%     1         3         1          3m24s
root@kubeadm-master:/home/ubuntu/Kubernetes#
root@kubeadm-master:/home/ubuntu/Kubernetes#
root@kubeadm-master:/home/ubuntu/Kubernetes#
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl get hpa
NAME        REFERENCE              TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
nginx-hpa   Deployment/nginx-hpa   15%/2%    1         3         3          5m11s
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl top pods
NAME                               CPU(cores)   MEMORY(bytes)
load-generator                     228m         1Mi
nginx-deployment-7c68bd84c6-txq9r  0m           4Mi
nginx-hpa-568d658b68-gjlql         47m          3Mi
nginx-hpa-568d658b68-sbvmn         0m           2Mi
nginx-hpa-568d658b68-vfwcb         0m           2Mi
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl get pods
NAME                               READY   STATUS    RESTARTS   AGE
load-generator                     1/1     Running   0          3m30s
nginx-deployment-7c68bd84c6-txq9r  1/1     Running   1          13h
nginx-hpa-568d658b68-gjlql         1/1     Running   0          8m46s
nginx-hpa-568d658b68-sbvmn         1/1     Running   0          95s
nginx-hpa-568d658b68-vfwcb         1/1     Running   0          95s
root@kubeadm-master:/home/ubuntu/Kubernetes#
```

7. On the load generator terminal, press ctrl+c. Exit from the pod. Pod gets auto deleted.

8. Get back to the main terminal. Wait for 4-5 minutes and verify that on reduction of load nginx-hpa pod scales down to min replica count

```
$ kubectl top pods
```

```
root@kubeadm-master:/home/AzureUser/Kubernetes# kubectl top pods
NAME                          CPU(cores)   MEMORY(bytes)
nginx-hpa-568d658b68-49gtg    0m           3Mi
nginx-hpa-568d658b68-ldfqq    0m           3Mi
nginx-hpa-568d658b68-zvt2v    0m           3Mi
```

```
$ kubectl get pods -w
```

```
root@kubeadm-master:/home/ubuntu/Kubernetes#
root@kubeadm-master:/home/ubuntu/Kubernetes# kubectl get pods -w
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-7c68bd84c6-txq9r   1/1     Running   1          13h
nginx-hpa-568d658b68-gjlql          1/1     Running   0          12m
nginx-hpa-568d658b68-sbvmn          1/1     Running   0          5m29s
nginx-hpa-568d658b68-vfwcb          1/1     Running   0          5m29s




nginx-hpa-568d658b68-sbvmn          1/1     Terminating   0          6m49s
nginx-hpa-568d658b68-vfwcb          1/1     Terminating   0          6m49s
nginx-hpa-568d658b68-sbvmn          0/1     Terminating   0          6m50s
nginx-hpa-568d658b68-vfwcb          0/1     Terminating   0          6m50s
nginx-hpa-568d658b68-sbvmn          0/1     Terminating   0          6m51s
nginx-hpa-568d658b68-sbvmn          0/1     Terminating   0          6m51s
^Croot@kubeadm-master:/home/ubuntu/Kubernetes# kubectl get pods
NAME                                READY   STATUS        RESTARTS   AGE
nginx-deployment-7c68bd84c6-txq9r   1/1     Running       1          13h
nginx-hpa-568d658b68-gjlql          1/1     Running       0          14m
nginx-hpa-568d658b68-vfwcb          0/1     Terminating   0          7m2s
```

5.6 Deleting the deployment resource created in this lab Exercise

```
$ kubectl delete -f nginx-hpa.yaml
$ kubectl delete -f metrics-server.yaml
$ kubectl delete hpa nginx-hpa
```

6 TROUBLESHOOTING

6.1 Getting Error While Creating nginx-hpa.yaml File

Issue: Getting error while creating the nginx-hpa.yaml file

```
root@Master-ma:/home/ubuntu/Kubernetes# vim nginx-hpa.yaml
root@Master-ma:/home/ubuntu/Kubernetes# kubectl create -f nginx-hpa.yaml
error: error parsing nginx-hpa.yaml: error converting YAML to JSON: yaml: line 3: mapping values are not allowed in this conte
xt
root@Master-ma:/home/ubuntu/Kubernetes#
```

Reason: YAML file is wrong for nginx-hpa.yaml.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-hpa
  labels:
    app: nginx-hpa
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx-hpa
  template:
    metadata:
      labels:
        app: nginx-hpa
    spec:
      containers:
        name: nginx-cont
      - image: nginx
        resources:
          limits:
            cpu: 300m
            memory: 200Mi
          requests:
            cpu: 150m
            memory: 100Mi
~
```

Fix: Too Much gap between the Request and CPU. Coorect file give below

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-hpa
  labels:
    app: nginx-hpa
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx-hpa
  template:
    metadata:
      labels:
        app: nginx-hpa
    spec:
      containers:
      - image: nginx
        name: nginx-cont
        resources:
          limits:
            cpu: 300m
            memory: 200Mi
          requests:
            cpu: 150m
            memory: 100Mi

~
```

6.2 Getting Error while Creating New Pod for Auto-Scaling

Issue: While executing the below command getting error like below image

```
$ kubectl run -it --rm load-generator --image=busybox /bin/sh
```

```
root@worker2-ma:/home/ubuntu# sudo su
root@worker2-ma:/home/ubuntu# export KUBECONFIG=$HOME/admin.conf
root@worker2-ma:/home/ubuntu# kubectl run -it --rm load-generator --image=busybo
x /bin/sh
W0802 19:18:51.464133   78671 loader.go:223] Config not found: /root/admin.conf
The connection to the server localhost:8080 was refused - did you specify the right host or port?
root@worker2-ma:/home/ubuntu# kubectl run -it --rm load-generator --image=busybo                                    x /bin/sh
W0802 19:19:23.008778   78826 loader.go:223] Config not found: /root/admin.conf
The connection to the server localhost:8080 was refused - did you specify the right host or port?
root@worker2-ma:/home/ubuntu# kubectl run -it --rm load-generator --image=busybo                                    x /bin/sh
W0802 19:19:25.980687   78850 loader.go:223] Config not found: /root/admin.conf
The connection to the server localhost:8080 was refused - did you specify the right host or port?
root@worker2-ma:/home/ubuntu#
```

Reason: Getting above error because running above on Worker Node not on Master node

Fix: Run this command on Master node to check autoscaling create a duplicate session of master node execure this command

```
nginx-hpa-568d658b68-trjnr   0m           6Mi
root@Master-ma:/home/ubuntu/Kubernetes#
root@Master-ma:/home/ubuntu/Kubernetes#
root@Master-ma:/home/ubuntu/Kubernetes#
root@Master-ma:/home/ubuntu/Kubernetes# kubectl get pods
NAME                          READY   STATUS    RESTARTS   AGE
nginx-hpa-568d658b68-trjnr   1/1     Running   0          43m
root@Master-ma:/home/ubuntu/Kubernetes# kubectl get hpa
NAME        REFERENCE              TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
nginx-hpa   Deployment/nginx-hpa   0%/2%     1         3         1          40m
root@Master-ma:/home/ubuntu/Kubernetes# kubectl top pods
NAME                          CPU(cores)   MEMORY(bytes)
nginx-hpa-568d658b68-trjnr   0m           6Mi
root@Master-ma:/home/ubuntu/Kubernetes#
root@Master-ma:/home/ubuntu/Kubernetes#
root@Master-ma:/home/ubuntu/Kubernetes#
root@Master-ma:/home/ubuntu/Kubernetes# kubectl get pods
NAME                          READY   STATUS    RESTARTS   AGE
nginx-hpa-568d658b68-trjnr   1/1     Running   0          45m
root@Master-ma:/home/ubuntu/Kubernetes#
root@Master-ma:/home/ubuntu/Kubernetes#
root@Master-ma:/home/ubuntu/Kubernetes#
root@Master-ma:/home/ubuntu/Kubernetes#
```

6.3 Getting Error While Creating nginx-hpa.yaml File

Issue: Getting error while creating the nginx-hpa.yaml file

```
getting this error: root@kubernete-master:/home/AzureUser/Kubernetes#  vim
nginx-hpa.yaml
root@kubernete-master:/home/AzureUser/Kubernetes#  kubectl create -f
nginx-hpa.yaml
Error from server (BadRequest): error when creating "nginx-hpa.yaml":
Deployment in version "v1" cannot be handled as a Deployment:
v1.Deployment.Spec: v1.DeploymentSpec.Template: v1.PodTemplateSpec.Spec:
v1.PodSpec.Containers: []v1.Container: v1.Container.Resources:
v1.ResourceRequirements.Requests: Limits: unmarshalerDecoder: unable to
parse quantity's suffix, error found in #10 byte of ...|y":"200mi"},"request|...,
bigger context ...|esources":{"limits":
{"cpu":"300m","memory":"200mi"},"requests":
{"cpu":"150m","memory":"100mi"}}}}}}|...
root@kubernete-master:/home/AzureUser/Kubernetes#
```

2:41 AM

Reason: YAML file is wrong for nginx-hpa.yaml.

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-hpa
  labels:
    app: nginx-hpa
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx-hpa
  template:
    metadata:
      labels:
        app: nginx-hpa
    spec:
      containers:
      - image: nginx
        name: nginx-cont
        resources:
          limits:
            cpu: 300m
            memory: 200mi
          requests:
                                                    3,1           Top
```

Fix: Too Much gap between the Request and CPU. Coorect file give below

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-hpa
  labels:
    app: nginx-hpa
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx-hpa
  template:
    metadata:
      labels:
        app: nginx-hpa
    spec:
      containers:
      - image: nginx
        name: nginx-cont
        resources:
          limits:
            cpu: 300m
            memory: 200Mi
          requests:
            cpu: 150m
            memory: 100Mi

~
```

7 QUESTION

Q) kubectl run redis --image=redis --restart=never why we need to use --restart=never in the above cmd

Ans:

Always means that the container will be restarted even if it exited with a zero exit code (i.e. successfully). This is useful when you don't care why the container exited, you just want to make sure that it is always running (e.g. a web server). This is the default.

OnFailure means that the container will only be restarted if it exited with a non-zero exit code (i.e. something went wrong). This is useful when you want accomplish a certain task with the pod, and ensure that it completes successfully - if it doesn't it will be restarted until it does.

Never means that the container will not be restarted regardless of why it exited.

8 SUMMRY

In this guide we Covered:

- Deploying Scalable Stateless Application
- Configuring Autoscaling For Stateless Application