### Commands

The following commands can be used for practice of code shared.

Note: errors as some are expected and are shown.

#### Pods

1. kubectl get pods
2. cd src
3. kubectl create -f 1.1-basic\_pod.yaml
4. kubectl get pods
5. kubectl describe pod mypod | more
6. kubectl delete pod mypod
7. kubectl create -f 1.2-port\_pod.yaml
8. kubectl describe pod mypod | more
9. curl 192.168.###.###:80 (Replace ###.### with the IP address octets from the describe output)  
   # This command will time out (see the next lesson to understand why)
10. kubectl describe pod mypod | more
11. kubectl delete pod mypod
12. kubectl create -f 1.4-resources\_pod.yaml
13. kubectl describe pod mypod | more

Note: kubectl will accept the singular or plural form of resource kinds. For example kubectl get pods and kubectl get pod are equivalent.

#### Services

1. kubectl create -f 2.1-web\_service.yaml
2. kubectl get services
3. kubectl describe service webserver
4. kubectl describe nodes | grep -i addresses -A 1
5. curl 10.0.0.100:3#### (replace #### with the actual port digits)

#### Multi-Container Pods

1. kubectl create -f 3.1-namespace.yaml
2. kubectl create -f 3.2-multi\_container.yaml -n microservice
3. kubectl get -n microservice pod app
4. kubectl describe -n microservice pod app
5. kubectl logs -n microservice app counter --tail 10
6. kubectl logs -n microservice app poller -f

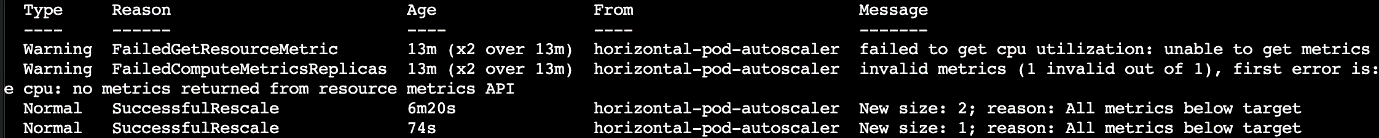
#### Service Discovery

1. kubectl create -f 4.1-namespace.yaml
2. kubectl create -f 4.2-data\_tier.yaml -n service-discovery
3. kubectl get pod -n service-discovery
4. kubectl describe service -n service-discovery data-tier
5. kubectl create -f 4.3-app\_tier.yaml -n service-discovery
6. kubectl create -f 4.4-support\_tier.yaml -n service-discovery
7. kubectl get pods -n service-discovery
8. kubectl logs -n service-discovery support-tier poller -f

#### Deployments

1. kubectl create -f 5.1-namespace.yaml
2. kubectl create -n deployments -f 5.2-data\_tier.yaml -f 5.3-app\_tier.yaml -f 5.4-support\_tier.yaml
3. kubectl get -n deployments deployments
4. kubectl -n deployments get pods
5. kubectl scale -n deployments deployment support-tier --replicas=5
6. kubectl -n deployments get pods
7. kubectl delete -n deployments pods support-tier-... support-tier-... --wait=false (You can use tab completion to display the possible values to replace ... with)
8. watch -n 1 kubectl -n deployments get pods
9. kubectl scale -n deployments deployment app-tier --replicas=5
10. kubectl -n deployments get pods
11. kubectl describe -n deployments service app-tier

#### Autoscaling

1. # kubectl create -f metrics-server/ # metrics-server is pre-installed on the playground so you don't need to run this command
2. watch kubectl top pods -n deployments
3. kubectl create -f 6.1-app\_tier\_cpu\_request.yaml -n deployments
4. kubectl apply -f 6.1-app\_tier\_cpu\_request.yaml -n deployments
5. kubectl get -n deployments deployments app-tier
6. kubectl create -f 6.2-autoscale.yaml -n deployments
7. watch -n 1 kubectl get -n deployments deployments app-tier # This can take up to 13 minutes to take effect. You may want to skip waiting. The image below shows the output of kubectl describe -n deployments hpa after waiting long enough  
   
8. kubectl api-resources
9. kubectl describe -n deployments hpa
10. kubectl get -n deployments hpa
11. kubectl edit -n deployments hpa
12. watch -n 1 kubectl get -n deployments deployments app-tier

#### Rolling Updates and Rollbacks

1. kubectl delete -n deployments hpa app-tier
2. kubectl edit -n deployments deployment app-tier
3. watch -n 1 kubectl get -n deployments deployments app-tier
4. kubectl edit -n deployments deployment app-tier
5. kubectl rollout -n deployments status deployment app-tier
6. tmux
7. kubectl edit -n deployments deployments app-tier (left terminal)
8. kubectl rollout -n deployments status deployment app-tier (right terminal)
9. kubectl rollout -n deployments pause deployment app-tier (left terminal)
10. kubectl get deployments -n deployments app-tier (left terminal)
11. kubectl rollout -n deployments resume deployment app-tier (left terminal)
12. kubectl rollout -n deployments undo deployment app-tier
13. kubectl scale -n deployments deployment app-tier --replicas=1

#### Probes

1. kubectl create -f 7.1-namespace.yaml
2. kubectl create -f 7.2-data\_tier.yaml -n probes
3. kubectl get deployments -n probes -w
4. kubectl create -f 7.3-app\_tier.yaml -n probes
5. kubectl get -n probes deployments app-tier -w
6. kubectl get -n probes pods
7. kubectl logs -n probes app-tier-... | cut -d' ' -f5,8-11 (You can use tab completion to display the possible values to replace ... with)

#### Init Containers

1. kubectl apply -f 8.1-app\_tier.yaml -n probes
2. kubectl describe pod -n probes app-tier... (You can use tab completion to display the possible values to replace ... with)
3. kubectl logs -n probes app-tier-... await-redis (You can use tab completion to display the possible values to replace ... with)

#### Volumes

1. kubectl -n deployments logs support-tier-... poller --tail 1 (You can use tab completion to display the possible values to replace ... with)
2. kubectl exec -n deployments data-tier-... -it -- /bin/bash (You can use tab completion to display the possible values to replace ... with)
3. kill 1
4. kubectl -n deployments get pods
5. kubectl -n deployments logs support-tier-... poller --tail 1 (You can use tab completion to display the possible values to replace ... with)  
   Note: It takes around a couple of minutes for the effects of the restart to settle. The poller will stop updating and report the last value before restarting until it can reach the new data tier value. Try again after a minute if you don't see a relatively small value)
6. kubectl create -f 9.1-namespace.yaml
7. aws ec2 describe-volumes --region=us-west-2 --filters="Name=tag:Type,Values=PV" --query="Volumes[0].VolumeId" --output=text
8. vol\_id=$(aws ec2 describe-volumes --region=us-west-2 --filters="Name=tag:Type,Values=PV" --query="Volumes[0].VolumeId" --output=text)
9. sed -i "s/INSERT\_VOLUME\_ID/$vol\_id/" 9.2-pv\_data\_tier.yaml
10. kubectl create -n volumes -f 9.2-pv\_data\_tier.yaml -f 9.3-app\_tier.yaml -f 9.4-support\_tier.yaml
11. kubectl describe -n volumes pvc
12. kubectl describe -n volumes pod data-tier-... (You can use tab completion to display the possible values to replace ... with)
13. kubectl logs -n volumes support-tier-... poller --tail 1 (You can use tab completion to display the possible values to replace ... with)  
    Note: It takes a few minutes for all of the readiness checks to pass and for the counter to start incrementing. If you don't see a counter value output then try again after a minute or two.
14. kubectl delete -n volumes deployments data-tier
15. kubectl get -n volumes pods
16. kubectl create -n volumes -f 9.2-pv\_data\_tier.yaml
17. kubectl logs -n volumes support-tier-... poller --tail 1 (You can use tab completion to display the possible values to replace ... with)

#### ConfigMaps and Secrets

1. kubectl create -f 10.1-namespace.yaml
2. kubectl create -n config -f 10.2-data\_tier\_config.yaml -f 10.3-data\_tier.yaml
3. kubectl exec -n config data-tier-... -it -- /bin/bash (You can use tab completion to display the possible values to replace ... with)
4. cat /etc/redis/redis.conf
5. redis-cli CONFIG GET tcp-keepalive
6. exit
7. kubectl edit -n config configmaps redis-config
8. kubectl exec -n config data-tier-... -- redis-cli CONFIG GET tcp-keepalive (You can use tab completion to display the possible values to replace ... with)
9. kubectl rollout -n config restart deployment data-tier
10. kubectl exec -n config data-tier-... -- redis-cli CONFIG GET tcp-keepalive (You can use tab completion to display the possible values to replace ... with)
11. kubectl create -f 10.4-app\_tier\_secret.yaml -n config
12. kubectl describe -n config secret app-tier-secret
13. kubectl edit -n config secrets app-tier-secret
14. kubectl create -f 10.5-app\_tier.yaml -n config
15. kubectl exec -n config app-tier-... -- env (You can use tab completion to display the possible values to replace ... with)