Labs commands and files.

1. Create pod with command

```
kubectl run NAME –image=name –restart=Never
```

2. Create namespace with command

```
kubectl create namespace NAME
```

3. Create deployment with command

```
kubectl create deployment NAME –image=NAME –replicas=COUNT(NUM) --port=80 –restart=Always
```

4. Expose deployment with command

```
kubectl expose deployment NAME --type=TYPE --name=SERVICES-NAME
```

5. Verify the status of pod,deployment,replicas

```
kubectl get pod,deploy,replicas
kubectl describe pod,deploy,replicas
kubectl logs POD-NAME
```

6. Understanding the node

```
kubectl get node
kubectl describe node
kubectl get event
```

7. Create value for configmap

```
{
mkdir primary
echo c > primary/cyan
echo m > primary/magenta
echo y > primary/yellow
echo k > primary/black
echo "known as key" >> primary/black
echo blue > favorite
}
```

8. Create Configmap

```
kubectl create configmap colors \
        --from-literal=text=black  \
        --from-file=./favorite  \
        --from-file=./primary/
```

9. Create pod to use configmap

```
apiVersion: v1
   kind: Pod
   metadata:
     name: shell-demo
   spec:
     containers:
     - name: nginx
       image: nginx
       env:
       - name: ilike
         valueFrom:
          configMapKeyRef:
           name: colors
           key: favorite
```

10. Create pod from file

```
kubectl create -f simpleshell.yaml
```

11. Configmap again

```
apiVersion: v1
   kind: ConfigMap
   metadata:
     name: fast-car
     namespace: default
   data:
     car.make: Ford
     car.model: Mustang
     car.trim: Shelby
```

12. Create configmap from file

```
kubectl create -f car-map.yaml
```

13. Create a pod using configmap as volume

```
apiVersion: v1
   kind: Pod
   metadata:
     name: shell-demo
   spec:
     containers:
     containers:
        - name: nginx
          image: nginx
          volumeMounts:
          - name: car-vol
            mountPath: /etc/cars
      volumes:
        - name: car-vol
          configMap:
            name: fast-car
```

14. Using hostpath

```
apiVersion: v1
```

```
kind: Pod
metadata:
  name: nginx-hostpath
spec:
  containers:
  - image: nginx
    name: nginx-hostpath
    volumeMounts:
    - mountPath: /test-host
      name: test-hostvolume
  volumes:
  - name: test-hostvolume
    hostPath:
      # directory location on host
      path: /data
      # this field is optional
      type: Directory
```

15. Exec into the POD

```
kubectl exec -it POD -- /bin/bash
```