K8s_LAB01

1- Create a pod with the name “imperative-nginx” and with the image nginx and latest tag. using Imperative command (not yaml).

```
Editor   Tab 1   +
controlplane $ kubectl run imperative-nginx --image=nginx
Error from server (AlreadyExists): pods "imperative-nginx" already exists
controlplane $ kubectl get pod
NAME               READY   STATUS    RESTARTS   AGE
imperative-ngin    1/1     Running   0          101s
imperative-nginx   1/1     Running   0          76s
controlplane $
```

2- Create a pod with the name webserver and with the image “nginx123” Use a pod-definition YAML file.

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

```
controlplane $ vim pod-definition.yaml
controlplane $ kubectl apply -f pod-definition.yaml
pod/webserver created
controlplane $ kubectl get pod
NAME               READY   STATUS         RESTARTS   AGE
imperative-ngin    1/1     Running        0          17m
imperative-nginx   1/1     Running        0          17m
webserver          0/1     ErrImagePull   0          18s
controlplane $
```

3- What is the nginx pod status?

```
Editor   Tab 1   +
controlplane $ kubectl get pod
NAME              READY   STATUS             RESTARTS   AGE
imperative-ngin   1/1     Running            0          20m
imperative-nginx  1/1     Running            0          20m
webserver         0/1     ImagePullBackOff   0          3m2s
controlplane $
```

4- Change the nginx pod image to “nginx” check the status again

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

```
controlplane $ vim pod-definition.yaml
controlplane $ kubectl apply -f pod-definition.yaml
pod/webserver configured
controlplane $ kubectl get pod
NAME              READY   STATUS    RESTARTS   AGE
imperative-ngin   1/1     Running   0          25m
imperative-nginx  1/1     Running   0          25m
webserver         1/1     Running   0          8m
controlplane $
```

5- How many pods are running in the system? Type the command to show this

```
controlplane $ kubectl get pod
NAME              READY   STATUS    RESTARTS   AGE
imperative-ngin   1/1     Running   0          26m
imperative-nginx  1/1     Running   0          26m
webserver         1/1     Running   0          9m2s
controlplane $
```

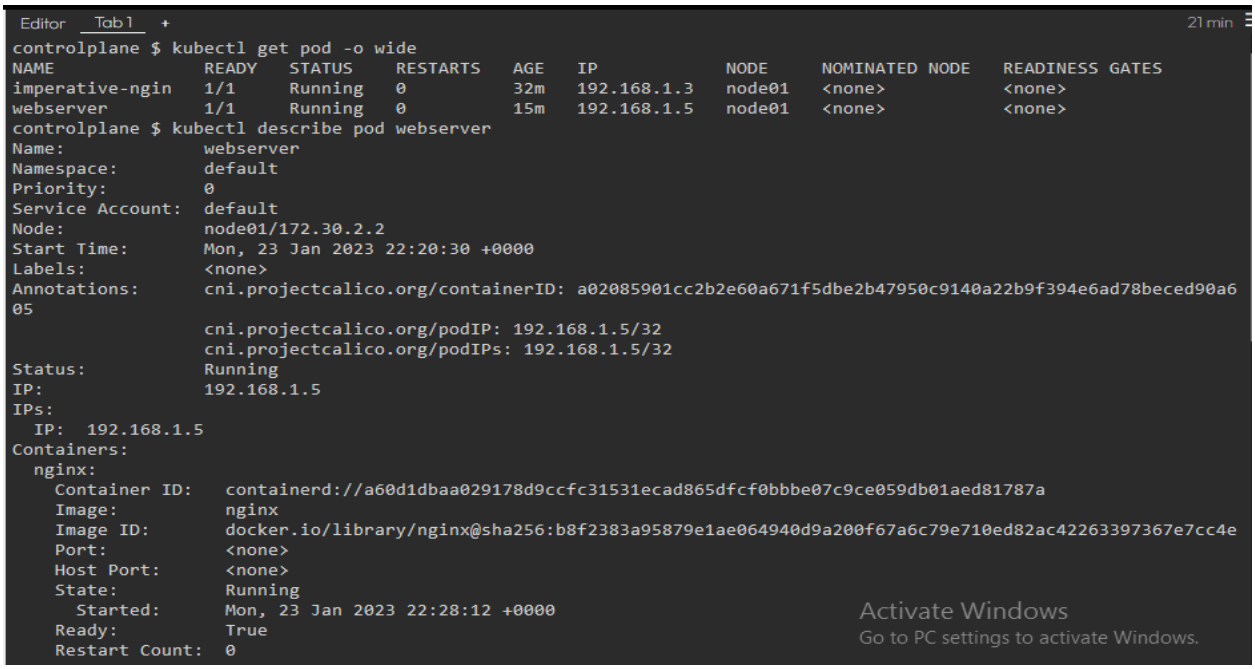
6- What does READY column in the output of get pods command indicate?

Answer : It indicates the number of containers which are ready.

7- Delete first pod named imperative-nginx you just created. Type the command to do this

```
controlplane $ kubectl delete pod/imperative-nginx
pod "imperative-nginx" deleted
controlplane $ kubectl get pod
NAME              READY   STATUS    RESTARTS   AGE
imperative-ngin   1/1     Running   0          30m
webserver         1/1     Running   0          13m
controlplane $
```

8- Which node is pod named webserver running on (list two commands to do this)



9- Get a shell to the running container i.e ssh into it (figure out the command).

10- Run cat /etc/os-release inside the container.

11- Exit from the shell (/bin/bash) session.

12- Get logs of pod, what are logs and what they are used for?

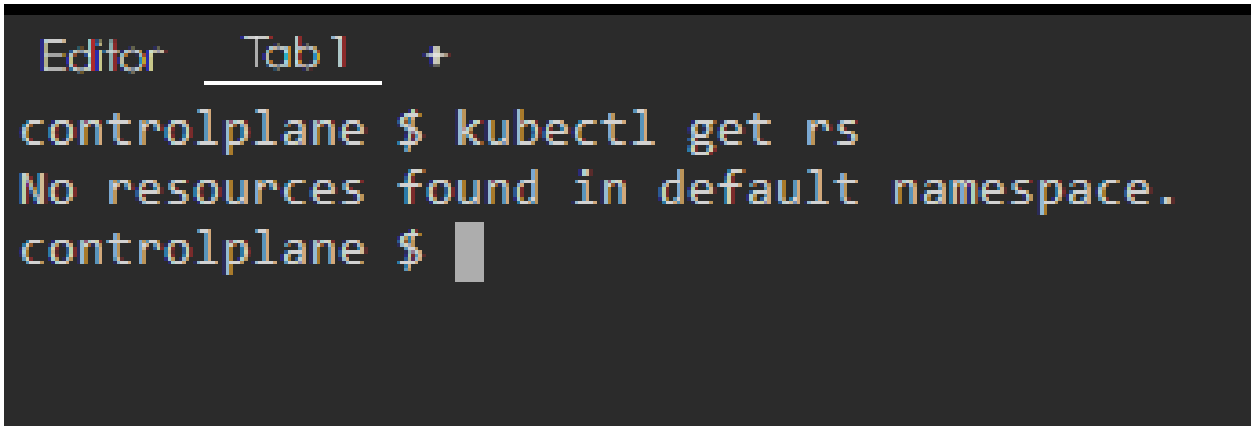
Answer :

Log files are a historical record of everything and anything that happens within a system, including events such as transactions, errors and intrusions.

Logs are using in monitoring across systems to detect particular log events and patterns in log data. Monitoring in real-time for anomalies or inactivity to gauge system health.

```
controlplane $ kubectl logs webserver
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2023/01/23 22:28:12 [notice] 1#1: using the "epoll" event method
2023/01/23 22:28:12 [notice] 1#1: nginx/1.23.3
2023/01/23 22:28:12 [notice] 1#1: built by gcc 10.2.1 20210110 (Debian 10.2.1-6)
2023/01/23 22:28:12 [notice] 1#1: OS: Linux 5.4.0-131-generic
2023/01/23 22:28:12 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2023/01/23 22:28:12 [notice] 1#1: start worker processes
2023/01/23 22:28:12 [notice] 1#1: start worker process 29
controlplane $
```


13- How many ReplicaSets exist on the system?



14- create a ReplicaSet with name= replica-set-1 image= busybox replicas= 3

Editor Tab 1 +

```yaml
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: replica-set-1
  labels:
    app: guestbook
    tier: frontend
spec:
  replicas: 3
  selector:
    matchLabels:
      tier: s:
      tier: frontend
  template:
    metadata:
      labels:
        tier: frontend
    spec:
      containers:
      - name: busybox-1
        image: busybox
        tty  : true
~
```

```
controlplane $ touch replica-set-1.yaml
controlplane $ vim replica-set-1.yaml
controlplane $ kubectl apply -f replica-set-1.yaml
replicaset.apps/replica-set-1 unchanged
controlplane $ kubectl get pod
NAME                  READY   STATUS    RESTARTS   AGE
replica-set-1-7sx2k   1/1     Running   0          5m7s
replica-set-1-pt794   1/1     Running   0          5m7s
replica-set-1-r72fl   1/1     Running   0          5m7s
controlplane $
```

15- Scale the ReplicaSet replica-set-1 to 5 PODs.

```
controlplane $ kubectl scale --replicas=5 -f replica-set-1.yaml
replicaset.apps/replica-set-1 scaled
controlplane $
```

16- How many PODs are READY in the replica-set-1?

```
controlplane $ kubectl get pod
NAME                READY   STATUS    RESTARTS   AGE
replica-set-1-4fp9w   1/1     Running   0          75s
replica-set-1-7sx2k   1/1     Running   0          8m48s
replica-set-1-pt794   1/1     Running   0          8m48s
replica-set-1-r72fl   1/1     Running   0          8m48s
replica-set-1-r9vpq   1/1     Running   0          75s
controlplane $
```

17- Delete any one of the 5 PODs then check How many PODs exist now? Why are there still 5 PODs, even after you deleted one? Answer : Because the replica set is working to make the number of PODs is constant if any pod was deleted replica set will create another one.

```
controlplane $ kubectl get pod
NAME                READY   STATUS    RESTARTS   AGE
replica-set-1-4fp9w   1/1     Running   0          75s
replica-set-1-7sx2k   1/1     Running   0          8m48s
replica-set-1-pt794   1/1     Running   0          8m48s
replica-set-1-r72fl   1/1     Running   0          8m48s
replica-set-1-r9vpq   1/1     Running   0          75s
controlplane $ kubectl delete pod/replica-set-1-4fp9w
pod "replica-set-1-4fp9w" deleted
controlplane $ kubectl get pod
NAME                READY   STATUS    RESTARTS   AGE
replica-set-1-569lc   1/1     Running   0          6m56s
replica-set-1-7sx2k   1/1     Running   0          18m
replica-set-1-pt794   1/1     Running   0          18m
replica-set-1-r72fl   1/1     Running   0          18m
replica-set-1-r9vpq   1/1     Running   0          10m
controlplane $
```