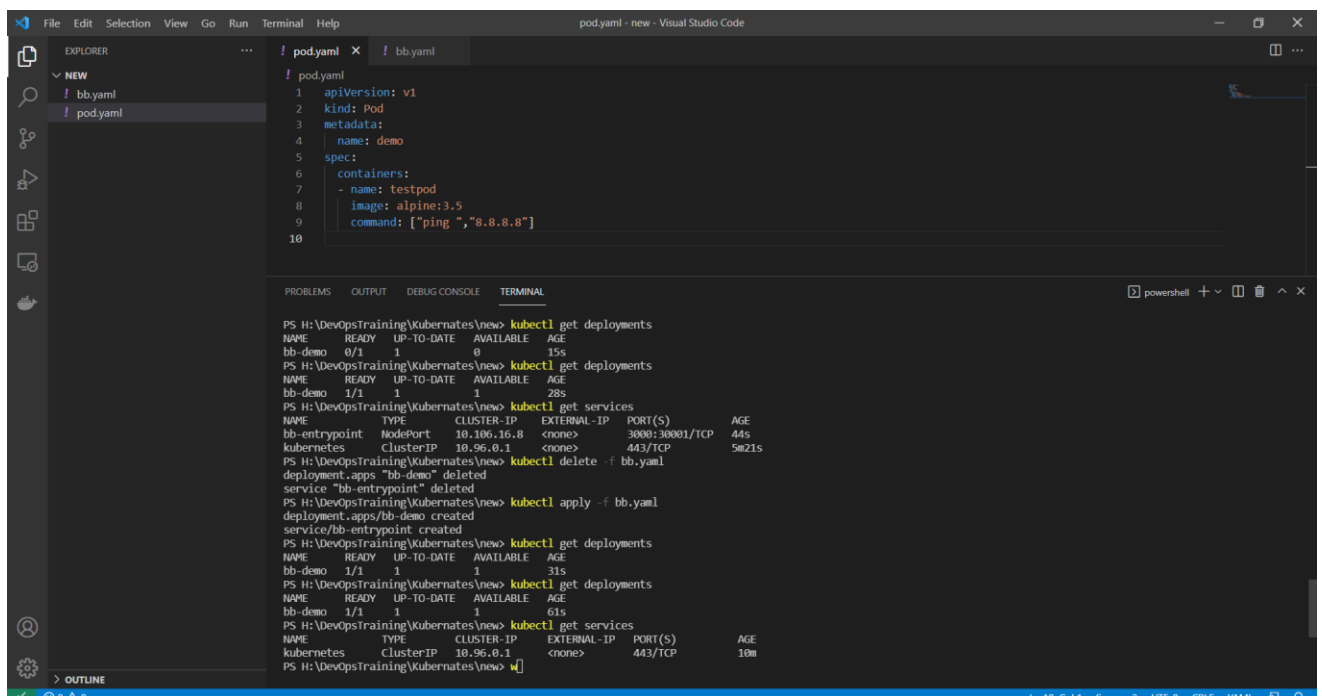


KUBERNETES

Sample Kubernetes Commands

1. `kubectl apply -f pod.yaml`
2. `kubectl get pods`
3. `kubectl logs demo`
4. `kubectl delete -f pod.yaml`
5. `docker swarm init`
6. `docker swarm join-token`
7. `docker swarm leave`
8. `docker service create --name demo alpine:3.5 ping 8.8.8.8`
9. `docker service ps demo`
10. `docker service logs demo`
11. `kubectl apply -f bb.yaml`
12. `kubectl get deployments`
13. `kubectl get services`
14. `kubectl delete -f bb.yaml`



The screenshot shows the Visual Studio Code interface with a file explorer on the left, a code editor in the center, and a terminal window at the bottom.

File Explorer: Shows a project named "NEW" with two files: "bb.yaml" and "pod.yaml".

Code Editor: Displays the content of "pod.yaml":

```
1 apiVersion: v1
2 kind: Pod
3 metadata:
4   name: demo
5 spec:
6   containers:
7   - name: testpod
8     image: alpine:3.5
9     command: ["ping", "8.8.8.8"]
10
```

Terminal: Shows the output of several kubectl commands executed in a PowerShell session:

```
PS H:\DevOpsTraining\Kubernetes\new> kubectl get deployments
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
bb-demo   0/1     1            0           15s
PS H:\DevOpsTraining\Kubernetes\new> kubectl get deployments
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
bb-demo   1/1     1            1           28s
PS H:\DevOpsTraining\Kubernetes\new> kubectl get services
NAME                TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
bb-entrypoint       NodePort    10.106.16.8   <none>        3000:30001/TCP   44s
kubernetes          ClusterIP   10.96.0.1     <none>        443/TCP          5021s
PS H:\DevOpsTraining\Kubernetes\new> kubectl delete -f bb.yaml
deployment.apps "bb-demo" deleted
service "bb-entrypoint" deleted
PS H:\DevOpsTraining\Kubernetes\new> kubectl apply -f bb.yaml
deployment.apps/bb-demo created
service/bb-entrypoint created
PS H:\DevOpsTraining\Kubernetes\new> kubectl get deployments
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
bb-demo   1/1     1            1           31s
PS H:\DevOpsTraining\Kubernetes\new> kubectl get deployments
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
bb-demo   1/1     1            1           61s
PS H:\DevOpsTraining\Kubernetes\new> kubectl get services
NAME                TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
kubernetes          ClusterIP   10.96.0.1     <none>        443/TCP          10m
```

```
File Edit Selection View Go Run Terminal Help
bb.yaml - new - Visual Studio Code

EXPLORER
NEW
! bb.yaml
! pod.yaml

! pod.yaml
1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4   name: bb-demo
5   namespace: default
6 spec:
7   replicas: 1
8   selector:
9     matchLabels:
10      bb: web
11   template:
12     metadata:
13       labels:
14         bb: web
15     spec:
16       containers:
17         - name: bb-site
18           image: abhinjith/getting-started
19 ---
20 apiVersion: v1
21 kind: Service
22 metadata:
23   name: bb-entrypoint
24   namespace: default
25 spec:
26   type: NodePort
27   selector:
28     bb: web
29   ports:
30     - port: 3000
31       targetPort: 3000
32       nodePort: 30001

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 10m
PS H:\DevOpsTraining\Kubernetes\new>
```

```
File Edit Selection View Go Run Terminal Help
bb.yaml - new - Visual Studio Code

EXPLORER
NEW
! bb.yaml
! pod.yaml

! bb.yaml
6 selector:

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS H:\DevOpsTraining\Kubernetes\new>
PS H:\DevOpsTraining\Kubernetes\new>
PS H:\DevOpsTraining\Kubernetes\new> docker swarm init
Swarm initialized: current node (hmdspg0dyf8aqcxw0pr9tb) is now a manager.

To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-3xdnu0h77gnqgyw4h2blbsyzollpmfhuhkk29688180xqjaksd-13gwh7w6mi3eluzrhjdgngjog 192.168.65.3:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.

PS H:\DevOpsTraining\Kubernetes\new> docker swarm join-token
"docker swarm join-token" requires exactly 1 argument.
See 'docker swarm join-token --help'.

Usage: docker swarm join-token [OPTIONS] (worker|manager)

Manage join tokens
PS H:\DevOpsTraining\Kubernetes\new> docker service create --name demo alpine:3.5 ping 8.8.8.8
nvtxi4pa906c96zpw5nfe61vb
overall progress: 1 out of 1 tasks
1/1: running [=====>]
verify: Service converged
PS H:\DevOpsTraining\Kubernetes\new> docker service ps demo
ID NAME IMAGE NODE DESIRED STATE CURRENT STATE ERROR PORTS
teq1warrlpv demo.1 alpine:3.5 docker-desktop Running Running 57 seconds ago
PS H:\DevOpsTraining\Kubernetes\new> docker service logs demo
demo.1.teq1warrlpv@docker-desktop | PING 8.8.8.8 (8.8.8.8): 56 data bytes
demo.1.teq1warrlpv@docker-desktop | 64 bytes from 8.8.8.8: seq=0 ttl=37 time=21.686 ms
demo.1.teq1warrlpv@docker-desktop | 64 bytes from 8.8.8.8: seq=1 ttl=37 time=22.002 ms
demo.1.teq1warrlpv@docker-desktop | 64 bytes from 8.8.8.8: seq=2 ttl=37 time=26.052 ms
demo.1.teq1warrlpv@docker-desktop | 64 bytes from 8.8.8.8: seq=3 ttl=37 time=22.648 ms
demo.1.teq1warrlpv@docker-desktop | 64 bytes from 8.8.8.8: seq=4 ttl=37 time=21.967 ms
demo.1.teq1warrlpv@docker-desktop | 64 bytes from 8.8.8.8: seq=5 ttl=37 time=75.730 ms
demo.1.teq1warrlpv@docker-desktop | 64 bytes from 8.8.8.8: seq=6 ttl=37 time=37.976 ms
demo.1.teq1warrlpv@docker-desktop | 64 bytes from 8.8.8.8: seq=7 ttl=37 time=21.798 ms
demo.1.teq1warrlpv@docker-desktop | 64 bytes from 8.8.8.8: seq=8 ttl=37 time=22.216 ms
demo.1.teq1warrlpv@docker-desktop | 64 bytes from 8.8.8.8: seq=9 ttl=37 time=23.104 ms
demo.1.teq1warrlpv@docker-desktop | 64 bytes from 8.8.8.8: seq=10 ttl=37 time=22.515 ms
demo.1.teq1warrlpv@docker-desktop | 64 bytes from 8.8.8.8: seq=11 ttl=37 time=23.365 ms
```