

Kubernetes parancsok

1. **kubectl get nodes**

- Megjeleníti az összes elérhető node-ot a Kubernetes klaszterben.

2. **cd edu-cloud-native/k8s-intro**

- Átvált az adott könyvtárba, amely tartalmazza a szükséges konfigurációs fájlokat.

3. **ll**

- Kilistázza a könyvtár tartalmát, beleértve a fájlokat és azok tulajdonosait.

4. **ec2metadata --public-ipv4**

- Lekérdezi az EC2 instance nyilvános IPv4 címét.

5. **sudo docker build -t practice/06 .**

- Docker image-et készít az aktuális könyvtárban található Dockerfile alapján.

6. **sudo docker save practice/06 | sudo k3s ctr images import -**

- A létrehozott Docker image-et importálja a K3s klaszterbe.

7. **kubectl get pods**

- Kilistázza az aktuális namespace-ben található összes Podot.

8. **kubectl get pods -A**

- Kilistázza a klaszter összes namespace-ében lévő Podokat.

9. **kubectl apply -f deployment.yaml**

- Alkalmazza a deployment.yaml fájlban található konfigurációt a Kubernetes klaszterre.

10. **kubectl apply -f net-debug.yaml**

- Alkalmazza a net-debug.yaml fájlban található konfigurációt.

11. **kubectl get pods -o wide**

- Kilistázza a Podokat részletes információkkal, például IP címekkel.

12. **kubectl get pods -o yaml**

- Kilistázza a Podokat YAML formátumban, teljes konfigurációs adatokkal.

13. **kubectl describe pod <pod name>**

- Megjeleníti az adott Pod részletes leírását.

14. **kubectl describe deployment <deployment name>**

- Megjeleníti az adott Deployment részletes leírását.

15. **kubectl exec -it <name of your net-debug pod> -- bash**

- Belép az adott Podba, ahol interaktív Bash parancssort biztosít.

Kubernetes parancsok

16. **kubectl delete -f <config file to delete>**

- Törli a megadott konfigurációs fájl által létrehozott Kubernetes objektumokat.

17. **kubectl get pods --watch**

- Folyamatosan figyeli és frissíti a Podok állapotát.

18. **kubectl edit deployment my-server**

- Szerkeszti az adott Deploymentet (itt a replicas mezőt módosítjuk).

19. **kubectl apply -f service.yaml**

- Alkalmazza a service.yaml fájlt, létrehozva egy új Service-t a klaszterben.

20. **kubectl get services** vagy **kubectl get svc**

- Kilistázza a klaszter összes Service-jét.

21. **kubectl describe svc my-service**

- Részletes információkat jelenít meg az adott Service-ről.

22. **curl http://<IP of one of the my-service Pod>:8080/**

- HTTP kérést küld egy konkrét Pod IP címére és 8080 portjára.

23. **curl <http://my-service:80/>, curl <http://my-service.default:80/>, curl <http://my-service.default.svc.cluster.local>**

- HTTP kérést küld a Service domain nevei segítségével a klaszter belső hálózataán.

24. **kubectl apply -f service-external.yaml**

- Alkalmazza az service-external.yaml fájlt, létrehozva egy LoadBalancer típusú Service-t.

25. **kubectl get svc my-external-service -o json | jq .**

- JSON formátumban jeleníti meg az my-external-service részleteit a jq használatával.

26. **kubectl get svc my-external-service -o jsonpath={.status.loadBalancer}**

- Lekérdezi az my-external-service LoadBalancer státuszát és JSONPath formátumban jeleníti meg.

27. **kubectl apply -f ingress.yaml**

- Alkalmazza az ingress.yaml fájlt, létrehozva egy Ingress szabályt.

28. **kubectl get ingress**

- Kilistázza az összes Ingress szabályt a klaszterben.

29. **kubectl get pods -A --show-labels**

- Kilistázza az összes Podot a klaszterben a címkékkel együtt.

Kubernetes parancsok

30. **kubectl get pods -l app=my-server**

- Kijelöl azokat a Podokat, amelyek app=my-server címkével rendelkeznek.

31. **kubectl get pods -l 'app in (my-server, my-client)'**

- Kijelöl azokat a Podokat, amelyek app=my-server vagy app=my-client címkével rendelkeznek.

32. **kubectl get pods -L app**

- Kijelöl azokat a Podokat és megjeleníti az app címkét oszlopként.

33. **kubectl apply -f deployment-3.yaml**

- Alkalmazza a deployment-3.yaml fájlt, módosítva a Deploymentet.

34. **kubectl get svc my-external-service -o jsonpath={.status.loadBalancer}**

- Lekérdezi a my-external-service LoadBalancer IP címét.

35. **kubectl delete -f <config file to delete>**

- Törli az adott konfigurációs fájl alapján létrehozott objektumokat.

36. **kubectl get pods -o yaml**

- YAML formátumban listázza a Podok részletes konfigurációit.

Kubernetes parancsok

Példa kérdések:

1. Hány pod fut a kube-system namespace-ben?

- **Megoldás:** \$ kubectl get pods -n kube-system
- **Válasz:** A kubectl get pods -n kube-system parancs kiírja a kube-system namespace összes podját, és a listából megszámolható, hogy hány pod fut ebben a namespace-ben.

2. Melyik podok restartolódtak az utóbbi időben a teljes clusterben?

- **Megoldás:** \$ kubectl get pods -A --field-selector=status.phase=Running
- **Válasz:** A kubectl get pods -A parancs kimenetéből ellenőrizhető a "RESTARTS" oszlop, amely megmutatja, hogy egy pod hányszor indult újra.

3. Melyik szolgáltatás rendelkezik külső IP-vel (EXTERNAL-IP)?

- **Megoldás:** \$ kubectl get svc -A
- **Válasz:** A parancs kimenetében az "EXTERNAL-IP" oszlopban található értékek megmutatják, hogy van-e külső IP hozzárendelve a szolgáltatáshoz.

4. Melyik podok vannak kapcsolatban a "my-service" szolgáltatással?

- **Megoldás:** \$ kubectl describe svc my-service
- **Válasz:** A kubectl describe svc my-service parancs "Endpoints" részénél listázott IP címek mutatják, mely podok kapcsolódnak ehhez a szolgáltatáshoz.

5. Milyen IP című podok futnak a default namespace-ben?

- **Megoldás:** \$ kubectl get pods -o wide -n default
- **Válasz:** A parancs "IP" oszlopa megmutatja a default namespace-ben lévő podok IP címét.

6. Hány Kubernetes objektum tartalmazza az "app=my-server" labelt?

- **Megoldás:** \$ kubectl get all -l app=my-server
- **Válasz:** Az kubectl get all -l app=my-server parancs kilistázza az összes objektumot, amelyek rendelkeznek az "app=my-server" labellel, és a listából megszámolható ezek száma.

7. Van-e olyan pod, amely nem érte el a teljes "READY" állapotot?

- **Megoldás:** \$ kubectl get pods -A
- **Válasz:** A parancs kimenetében a "READY" oszlop ellenőrizhető, és minden podnál megfigyelhető, hogy elérte-e a teljesen kész állapotot.

8. Mi a "my-server" pod IP címe?

- **Megoldás:** \$ kubectl get pod my-server-<pod-id> -o wide
- **Válasz:** A "my-server" pod IP címe megtalálható a kimenet "IP" oszlopában.

Kubernetes parancsok

9. Hány szolgáltatás (Service) fut az összes namespace-ben?

- **Megoldás:** `$ kubectl get svc -A`
- **Válasz:** A `kubectl get svc -A` parancs kimenetéből megszámolható, hogy hány szolgáltatás fut az összes namespace-ben.

10. Mely podok tartoznak az "app=nginx" címkével rendelkező alkalmazáshoz?

- **Megoldás:** `$ kubectl get pods -l app=nginx -A`
- **Válasz:** A `kubectl get pods -l app=nginx -A` parancs megmutatja az összes olyan podot minden namespace-ben, amely az "app=nginx" címkével rendelkezik.

11. Mely szolgáltatások érhetőek el TCP protokollal?

- **Megoldás:** `$ kubectl get svc -o yaml | grep protocol: TCP`
- **Válasz:** Ez a parancs kilistázza azokat a szolgáltatásokat, amelyeknél a "protocol: TCP" van beállítva.

12. Mely podok használnak több mint egy konténert?

- **Megoldás:** `$ kubectl get pods -A -o custom-columns="NAMESPACE:.metadata.namespace,NAME:.metadata.name,CONTAINERS:.spec.containers[*].name" | grep ','`
- **Válasz:** Az ilyen podoknál több konténer neve is megjelenik, egymás mellett vesszővel elválasztva.

13. Van-e olyan pod, amely "Error" állapotban van?

- **Megoldás:** `$ kubectl get pods -A | grep Error`
- **Válasz:** A `kubectl get pods -A` parancs kimenetében lehet ellenőrizni az "Error" állapotú podokat.

14. Melyik namespace-ben fut a legtöbb pod?

- **Megoldás:** `$ kubectl get pods -A --no-headers | awk '{print $1}' | sort | uniq -c | sort -nr`
- **Válasz:** Az `awk` és `sort` parancsokat kombinálva a legtöbb podot tartalmazó namespace kiszámítható.

15. Melyik podot restartolták a legtöbbször?

- **Megoldás:** `$ kubectl get pods -A --sort-by='.status.containerStatuses[0].restartCount'`
- **Válasz:** A lista elején a legtöbbször újraindított pod található.

16. Mely node-on futnak a podok a "default" namespace-ben?

- **Megoldás:** `$ kubectl get pods -n default -o wide`
- **Válasz:** A parancs "NODE" oszlopa megmutatja, hogy melyik node-on futnak ezek a podok.

Kubernetes parancsok

17. Milyen portot használ a "my-service" szolgáltatás a cluster-ben?

- **Megoldás:** \$ kubectl get svc my-service -o yaml
- **Válasz:** Az "ports" szakasz alatt látható a port és targetPort, amelyek mutatják a szolgáltatás által használt portokat.

18. Hány szolgáltatás van "LoadBalancer" típusú a teljes clusterben?

- **Megoldás:** \$ kubectl get svc -A --field-selector spec.type=LoadBalancer
- **Válasz:** A kimenetből megszámolható, hogy hány "LoadBalancer" típusú szolgáltatás található.

19. Mely podok nem teljesítették a "Ready" állapotot?

- **Megoldás:** \$ kubectl get pods -A --field-selector=status.phase!=Running
- **Válasz:** Ez a parancs megmutatja az összes olyan podot, amely még nem érte el a "Running" vagy "Ready" állapotot.

20. Mely podok használják a legtöbb CPU erőforrást?

- **Megoldás:** \$ kubectl top pod -A --sort-by=CPU
- **Válasz:** A kubectl top pod kimenetéből megfigyelhető, hogy mely podok használják a legtöbb CPU-t.