

CKA 실전문제풀이 (30문제)

김영철강사

2025년 4월11일버전

[문제1 출제] ETCD Backup

작업 클러스터 : k8s

https://127.0.0.1:2379에서 실행 중인 etcd의 snapshot을 생성하고 snapshot을 /data/etcd-snapshot.db에 저장합니다.

그런 다음 /data/etcd-snapshot-previous.db에 있는 기존의 이전 스냅샷을 복원합니다.

etcdctl을 사용하여 서버에 연결하기 위해 다음 TLS 인증서/키가 제공됩니다.

CA certificate: /etc/kubernetes/pki/etcd/ca.crt

Client certificate: /etc/kubernetes/pki/etcd/server.crt

Client key: /etc/kubernetes/pki/etcd/server.key

[정답]

=====

[문제2 출제] Cluster upgrade

작업 클러스터 : k8s

마스터 노드의 모든 Kubernetes control plane및 node 구성 요소를 버전 1.31.7 버전으로 업그레이드합니다.

"주의사항" 반드시 Master Node에서 root권한을 가지고 작업을 실행해야 한다.

[정답]

=====

[문제3] ServiceAccount, Role, RoleBinding

작업 클러스터 : k8s

애플리케이션 운영중 특정 namespace의 Pod들을 모니터할수 있는 서비스가 요청되었습니다. api-access 네임스페이스의 모든 pod를 view할 수 있도록 다음의 작업을 진행하시오.

1. api-access라는 새로운 namespace에 pod-viewer라는 이름의 Service Account를 만듭니다.
2. podreader-role이라는 이름의 Role과 podreader-rolebinding이라는 이름의 RoleBinding을 만듭니다.
3. 앞서 생성한 ServiceAccount를 API resource Pod에 대하여 watch, list, get을 허용하도록 매핑하시오.

[정답]

[문제4 출제] ServiceAccount, ClusterRole, ClusterRoleBinding

작업 클러스터 : k8s

작업 Context에서 애플리케이션 배포를 위해 새로운 ClusterRole을 생성하고 특정 namespace의 ServiceAccount를 바인드하시오.

다음의 resource type에서만 Create가 허용된 ClusterRole deployment-clusterrole을 생성합니다.

Resource Type: Deployment StatefulSet DaemonSet

미리 생성된 namespace api-access 에 ckd-token이라는 새로운 ServiceAccount를 만듭니다.

ClusterRole deployment-clusterrole을 namespace api-access 로 제한된 새 ServiceAccount ckd-token에 바인딩 하세요

[정답]

[문제5 출제] Authentication and Authorization (실제출제문제)

작업클러스터 : k8s

Context You have been asked to create a new ClusterRole for a deployment pipeline and bind it to a specific ServiceAccount
scoped to a specific namespace.

Task:

- Create namespace : app-team1
- Create a new ClusterRole named deployment-clusterrole , which only allows to create the following resource types:
Deployment StatefulSet DaemonSet
- Create a new ServiceAccount named cicd-token in the existing namespace app-team1
- Bind the new ClusterRole deployment-clusterrole to the new ServiceAccount cicd-token , limited to the namespace app-team1

[정답]

=====

[문제6 출제] Pod 생성하기

작업 클러스터 : k8s

'cka-exam'이라는 namespace를 만들고, 'cka-exam' namespace에 아래와 같은 Pod를 생성하시오.

pod Name: pod-01

image: busybox

환경변수 : CERT = "CKA-cert"

command: /bin/sh

args: "-c", "while true; do echo \$(CERT); sleep 10;done"

[정답]

=====

[문제7 출제] pod의 로그 확인해서 결과 추출하기

작업 클러스터 : hk8s

Pod "custom-app"의 log를 모니터링하고 'file not found'메세지를 포함하는 로그라인을 추출하세요.

추출된 결과는 /opt/REPORT/2022/custom-app-log에 기록하세요.

[정답]

=====

[문제8 출제] static pod 생성하기

작업 클러스터 : hk8s

hk8s-w1 노드에 nginx-static-pod.yaml 라는 이름의 Static Pod를 생성하세요.

pod name: nginx-static-pod

image: nginx

port : 80

[정답]

=====

[문제9 출제] multi container Pod 생성하기

작업 클러스터 : k8s

4개의 컨테이너를 동작시키는 eshop-frontend Pod를 생성하시오.

pod image: nginx, redis, memcached, consul

[정답]

[문제10 출제] sidecar container Pod 생성하기

작업 클러스터 : k8s

현재 운영중인 eshop-cart-app Pod의 로그를 Kubernetes built-in logging 아키텍처(예: kubectl logs)에 통합하는

로그 스트리밍 사이드카 컨테이너를 운영하시오.

busybox 이미지를 사용하여 price라는 이름의 sidecar container를 기존 eshop-cart-app에 추가합니다.

새 price 컨테이너는 다음과 같은 command를 실행해야 합니다.

Command: /bin/sh, -c, "tail -n+1 -f /var/log/cart-app.log"

/var/log에 마운트 된 볼륨을 사용하여 사이드카 컨테이너에서 로그 파일 cart-app.log를 사용해야 합니다.

eshop-cart-app Pod와 cart-app 컨테이너를 수정하지 마시오

[정답]

=====

[문제11 출제] Deployment & Scaling

작업 클러스터 : k8s

a. webserver 라는 이름으로 deployment를 생성하시오

Name: webserver

replicas 개수 2개

label: app_env_stage=dev

container name: webserver

container image: nginx:1.14

b. 그런다음 webserver Deployment의 pod 수를 3개로 확장하시오.

c. 그런다음 Rolling Update를 통한 nginx version=1.15으로 Rolling Update 하시오 기록하시오
(추가문제)

[정답]

=====

[문제12 출제] Rolling update & Roll back

작업 클러스터 : k8s

Deployment를 이용해 nginx 파드를 3개 배포한 다음 컨테이너 이미지 버전을 rolling update하
고 update record를 기록합니다.

마지막으로 컨테이너 이미지를 previous version으로 roll back 합니다.

name: eshop-payment

Image : nginx

Image version: 1.16

update image version: 1.17

label: app=payment, environment=production

[정답]

=====

[문제13 출제] 노드 비우기

작업 클러스터 : k8s

k8s-worker2 노드를 스케줄링 불가능하게 설정하고, 해당 노드에서 실행 중인 모든 Pod을 다른
node로 reschedule 하세요.

[정답]

[문제14 출제] Ready 노드 확인하기

[패턴1번문제]

작업 클러스터 : hk8s

Ready 상태(NoSchedule로 taint된 node는 제외)인 node를 찾아 그 수를

/var/CKA2023/notaint_ready_node에 기록하세요..

[정답]

실제문제) Check to see how many nodes are ready (not including nodes tainted NoSchedule) and write the number to

/var/CKA2023/notaint_ready_node.txt

[패턴2번문제]

작업클러스터 : hk8s

실제문제) Count the Number of Nodes That Are Ready to Run Normal WorkloadsCluster

Determine how many nodes in the cluster are ready to run normal workloads (i.e., workloads that do not have any special tolerations).

Output this number to the file /var/CKA2023/notaint_ready_node

[정답]

=====

[문제15 출제] Pod Scheduling

작업 클러스터 : k8s

다음의 조건으로 pod를 생성하세요.

Name: eshop-store

Image: nginx

Nodeselector: disktype=ssd

[정답]

=====

[문제16] ClusterIP type의 서비스 운영

작업 클러스터 : k8s

'devops' namespace에서 운영되고 있는 eshop-order deploymen의 Service를 만드세요.

Service Name: eshop-order-svc

Type: ClusterIP

Port: 80

[정답]

=====

[문제17 출제] Pod를 이용한 Named Service 구성

작업 클러스터 : k8s

1.미리 배포한 'front-end'에 기존의 nginx 컨테이너의 포트 '80/tcp'를 expose하는 'http'라는 이름을 추가합니다.

2.컨테이너 포트 http를 expose하는 'front-end-svc'라는 새 service를 만듭니다.

3.또한 준비된 node의 'NodePort'를 통해 개별 Pods를 expose되도록 Service를 구성합니다.

[정답]

=====

[문제18 출제] Access the Service from outside the Cluster via NodePort

작업 클러스터 : k8s

'front-end' deployment의 nginx 컨테이너를 expose하는 'front-end-nodesvc'라는 새 service를 만듭니다.

Front-end로 동작중인 Pod에는 node의 30200 포트로 접속되어야 합니다.

구성 테스트 curl k8s-worker1:30200 연결 시 nginx 홈페이지가 표시되어야 합니다.

[정답]

=====

[문제19 출제] NetworkPolicy

[문제패턴1]

작업 클러스터 : k8s

Create a new NetworkPolicy named allow-port-from-namespace in the existing namespace devops.

Ensure that the new NetworkPolicy allows Pods in namespace migops to connect to port 9000 of Pods in namespace devops.

Further ensure that the new NetworkPolicy:

- * does not allow access to Pods, which don't listen on port 9000
- * does not allow access from Pods, which are not in namespace migops

[정답]

[문제패턴2]

작업 클러스터 : k8s

Create a new NetworkPolicy named allow-port-from-namespace in the existing namespace devops.

Ensure that the new NetworkPolicy allows Pods in namespace migops(using label team=migops) to connect to port 80 of Pods in namespace devops.

Further ensure that the new NetworkPolicy: does not allow access to Pods, which don't listen on port 80 does not allow access from Pods, which are not in namespace migops

[정답]

[문제패턴3]

작업 클러스터 : k8s

default namespace에 다음과 같은 pod를 생성하세요.

name: poc

image: nginx

port: 80

label: app=poc

"partition=customera"를 사용하는 namespace에서만 poc의

80포트로 연결할 수 있도록 default namespace에 'allow-web-

from-customera'라는 network Policy를 설정하세요.

보안 정책상 다른 namespace의 접근은 제한합니다.

[정답]

=====

[문제20 출제] Ingress 구성

[문제패턴1]

작업 클러스터 : k8s

1.ingress-nginx namespace에서 nginx이미지를 app=nginx 레이블을 가지고

실행하는 nginx pod를 구성하세요.

2. 현재 appjs-servic와 nginx 서비스는 이미 동작 중입니다.

단, appjs-service type : NodePort 변경하세요

3. app-ingress.yaml 파일을 생성하고, 다음 조건의 ingress 를 구성하세요.

- name: app-ingress
- / 접속했을 때 nginx 서비스로 연결
- /app 접속했을 때 appjs-service 서비스로 연결

[정답]

=====

[문제패턴2]

작업 클러스터 : k8s

Create a new nginx Ingress resource as follows:

- * Name: ping
- * Namespace: ing-internal
- * Exposing service hi on path /hi using service port 5678

[정답]

=====

[문제21] Service and DNS Lookup 구성

작업클러스터 : k8s

image nginx를 사용하는 resolver pod를 생성하고 resolver-service라는 service를 구성합니다.

클러스터 내에서 service와 pod 이름을 조회할 수 있는지 테스트합니다.

dns 조회에 사용하는 pod 이미지는 busybox:1.28이고, service와 pod 이름 조회는 nslookup을 사용합니다.

service 조회 결과는 /var/CKA2022/nginx.svc에 pod name 조회 결과는 /var/CKA2022/nginx.pod 파일에 기록합니다.

[정답]

=====

[문제22] emptyDir Volume을 공유하는 multi-pod 운영 하세요

작업 클러스터 : k8s

다음 조건에 맞춰서 nginx 웹서버 pod가 생성한 로그파일을 받아서 STDOUT으로 출력하는 busybox 컨테이너를 운영하시오.

Pod Name: weblog

Web container:

- Image: nginx:1.17
- Volume mount : /var/log/nginx
- Readwrite

Log container:

- Image: busybox
- args: /bin/sh, -c, "tail -n+1 -f /data/access.log"
- Volume mount : /data
- readonly

emptyDir 볼륨을 통한 데이터 공유

[정답]

=====

[문제23 출제] HostPath Volume 구성

1./data/cka/ fluentd.yaml 파일을 만들어 새로운 Pod 생성하세요

신규생성 Pod Name : fluentd, image : fluentd, namespace : default)

2. 위 조건을 참고하여 다음 조건에 맞게 볼륨마운트를 설정하시오.

1) Worker node의 도커 컨테이너 디렉토리 /var/lib/docker/containers 동일 디렉토리로 pod에 마운트 하시오.

2) Worker node의 /var/log 디렉토리를 fluentd Pod에 동일이름의 디렉토리 마운트하시오.

[정답]

=====

[문제24 출제] PersistentVolume 만들기

작업 클러스터 : k8s

pv001라는 이름으로 size 1Gi, access mode ReadWriteMany를 사용하여 persistent volume을 생성합니다.

volume type은 hostPath이고 위치는 /tmp/app-config입니다.

[정답]

=====

[문제25 출제] PVC를 사용하는 애플리케이션 Pod 운영하기 문제

작업클러스터 : k8s

다음의 조건에 맞는 새로운 PersistentVolumeClaim 생성하시오.

Name: pv-volume

Class: app-hostpath-sc

Capacity: 10Mi

앞서 생성한 pv-volume PersistentVolumeClaim을 mount하는 Pod를 생성하시오.

Name: web-server-pod

Image: nginx

Mount path: /usr/share/nginx/html

Volume에서 ReadWriteMany 액세스 권한을 가지도록 구성합니다.

[정답]

=====

[문제26 출제] Pod Scale-out

작업 클러스터 : k8s

Expand the number of running Pods in "eshop-order" to 5.

- namespace: devops
- deployment: eshop-order
- replicas: 5

[정답]

=====

[문제27] Persistent Volume 정보 보기

작업클러스터 : hk8s

클러스터에 구성된 모든 PV를 capacity별로 sort하여 /var/CKA2022/my-pv-list 파일에 저장하십시오.

PV 출력 결과를 sort하기 위해 kubectl 명령만 사용하고, 그 외 리눅스 명령은 적용하지 마시오.

[정답]

=====

[문제28 출제] 클러스터 리소스 정보 보기

작업클러스터 : k8s

'name=overloaded-cpu' 레이블을 사용하는 Pod들 중 CPU 소비율이 가장 높은 Pod의 이름을 찾아서 /var/CKA2023/custom-app-log에 기록하십시오.

[정답]

=====

29. Rolling Update (출제)

Cluster: kubectl config use-context k8s

Create a deployment as follows: TASK:

name: nginx-app

Using container nginx with version 1.11.10-alpine The deployment should contain 3 replicas

Next, deploy the application with new version 1.11.13-alpine, by performing a rolling update

Finally, rollback that update to the previous version 1.11.10-alpine

[정답]

=====

[문제30 출제] Worker Node 동작 문제 해결

작업클러스터 : k8s

Worker Node 동작 문제 해결

k8s-worker2 라는 이름의 worker node가 현재 NotReady 상태에 있습니다. 이 상태의 원인을 조사하고k8s-worker2 노드를

Ready 상태로 전환하여 영구적으로 유지되도록 운영하시오.

[정답]