

# 실습 개요

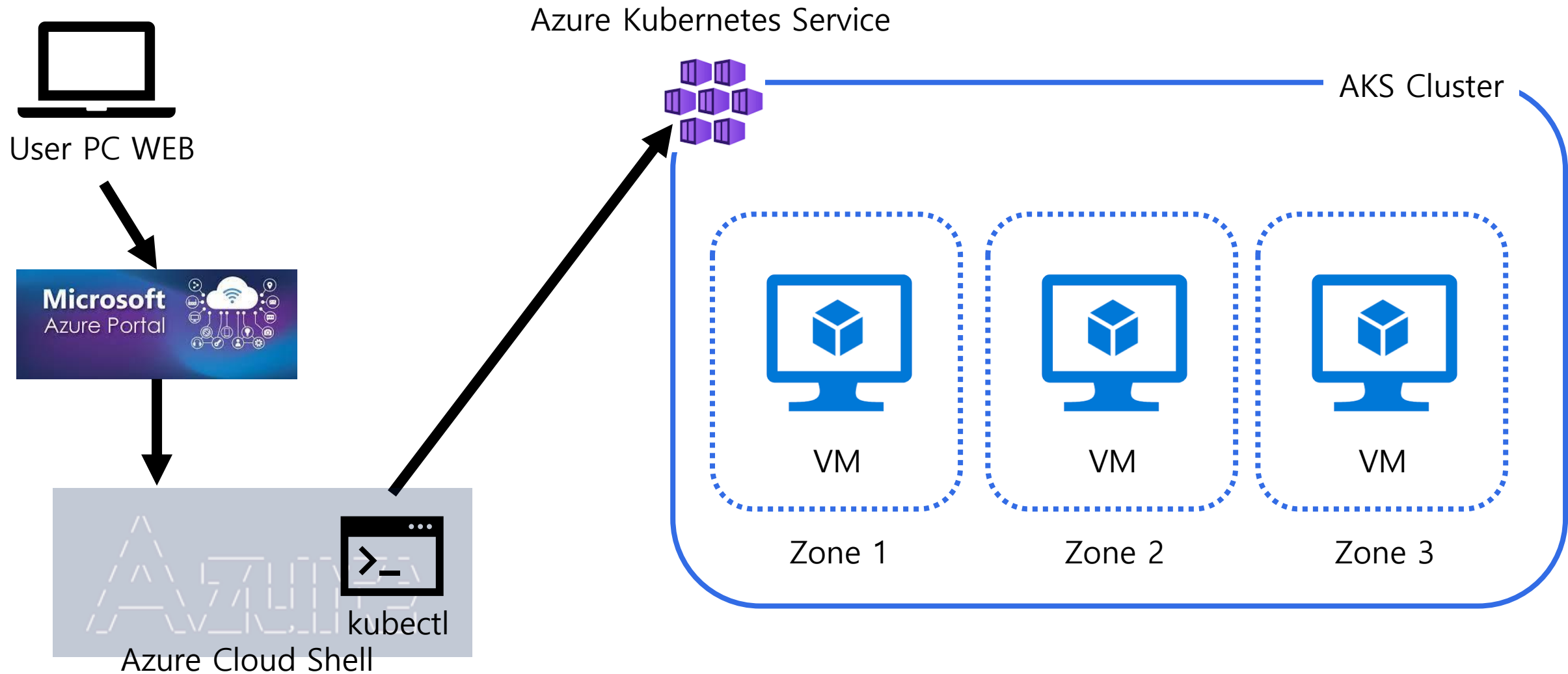
\*\*은 본인의 번호로 편집하여 입력

노란색 글씨 부분은 수강생 별 고유한 ID값, IP, NAME 등으로 편집하고, 명령어 수행

초록색 박스 명령어는 줄바꿈되는 명령어를 한 줄로 표현(복사 붙여넣기 할 때 사용)

한줄로 표현된 명령어

# 실습 환경 구성도 예시



# Lab 1.

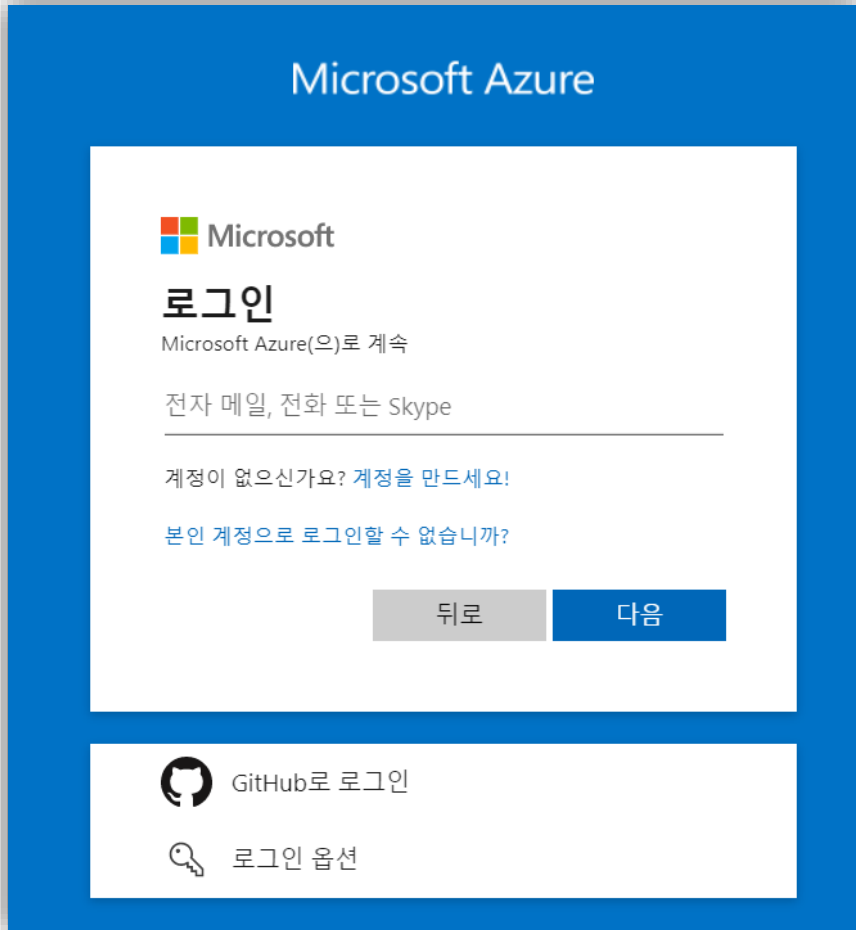
## 클러스터 배포

---

- Azure Portal 접속
- Cloud Shell 생성
- AKS 클러스터 배포
- AKS 클러스터 인증

# Azure Portal 로그인

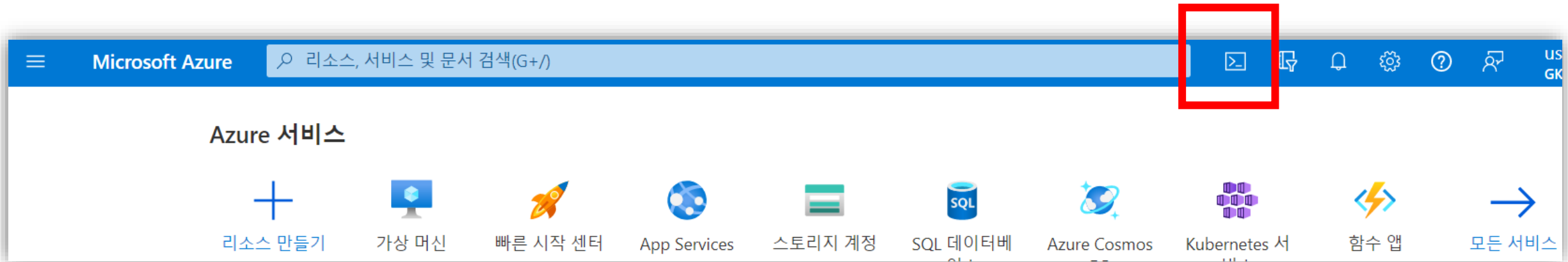
https://portal.azure.com/ 에 접속 후 제공받은 계정과 암호를 이용하여 azure 에 로그인



The image shows the Microsoft Azure login interface. It features a blue header with the 'Microsoft Azure' text. Below this, the Microsoft logo is displayed. The main heading is '로그인' (Login), followed by the text 'Microsoft Azure(으)로 계속' (Continue with Microsoft Azure). There is a text input field labeled '전자 메일, 전화 또는 Skype' (Email, phone, or Skype). Below the input field, there are two links: '계정이 없으신가요? 계정을 만드세요!' (Don't have an account? Create an account!) and '본인 계정으로 로그인할 수 없습니까?' (Can't sign in with your account?). At the bottom of the main section, there are two buttons: '뒤로' (Back) and '다음' (Next). Below the main section, there is a section for 'GitHub로 로그인' (Sign in with GitHub) and a link for '로그인 옵션' (Login options).

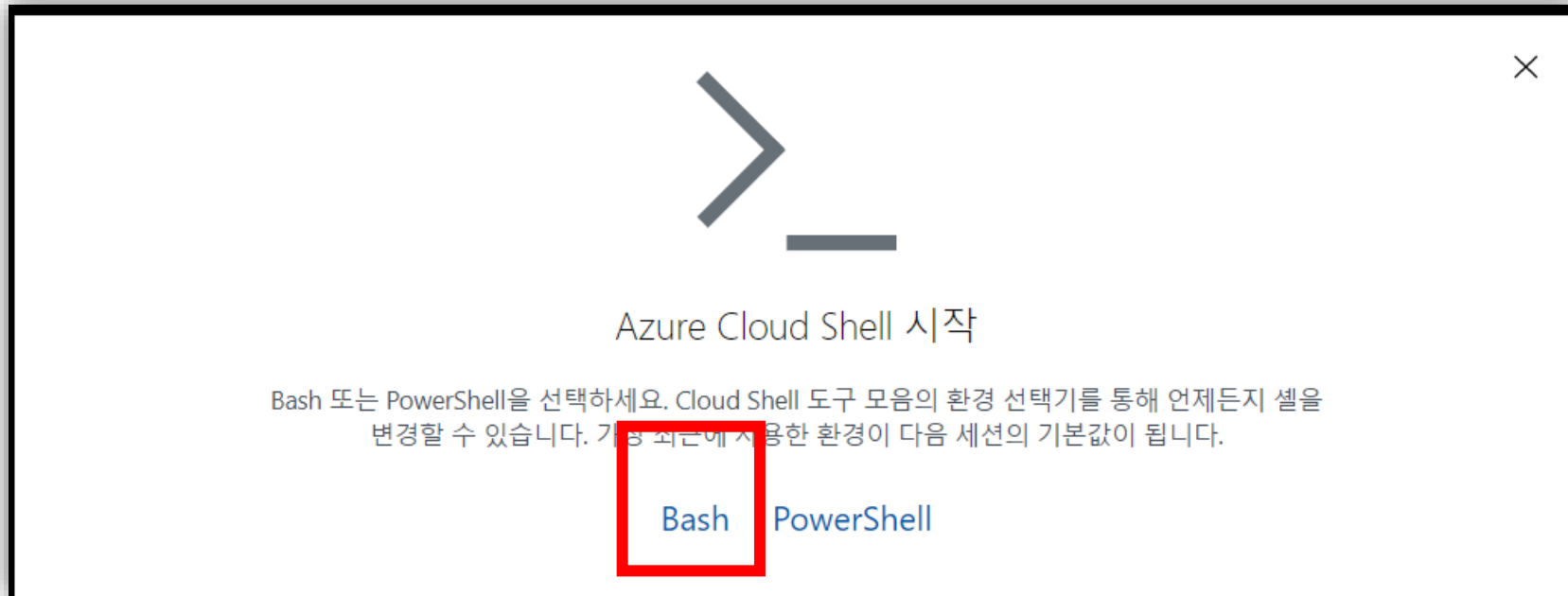
# CloudShell 생성

상단 검색창 우측 CloudShell 아이콘 클릭



# CloudShell 생성

Bash 클릭



# CloudShell 생성

구독 확인 후 스토리지 만들기 클릭

×

탐재된 스토리지가 없음

Azure Cloud Shell을 통해 파일을 유지하려면 Azure 파일 공유가 필요합니다. [자세한 정보](#)  
새 스토리지 계정이 만들어지고 이로 인해 소액의 월 비용이 발생합니다. [가격 보기](#)

\* 구독

Azure subscription 1

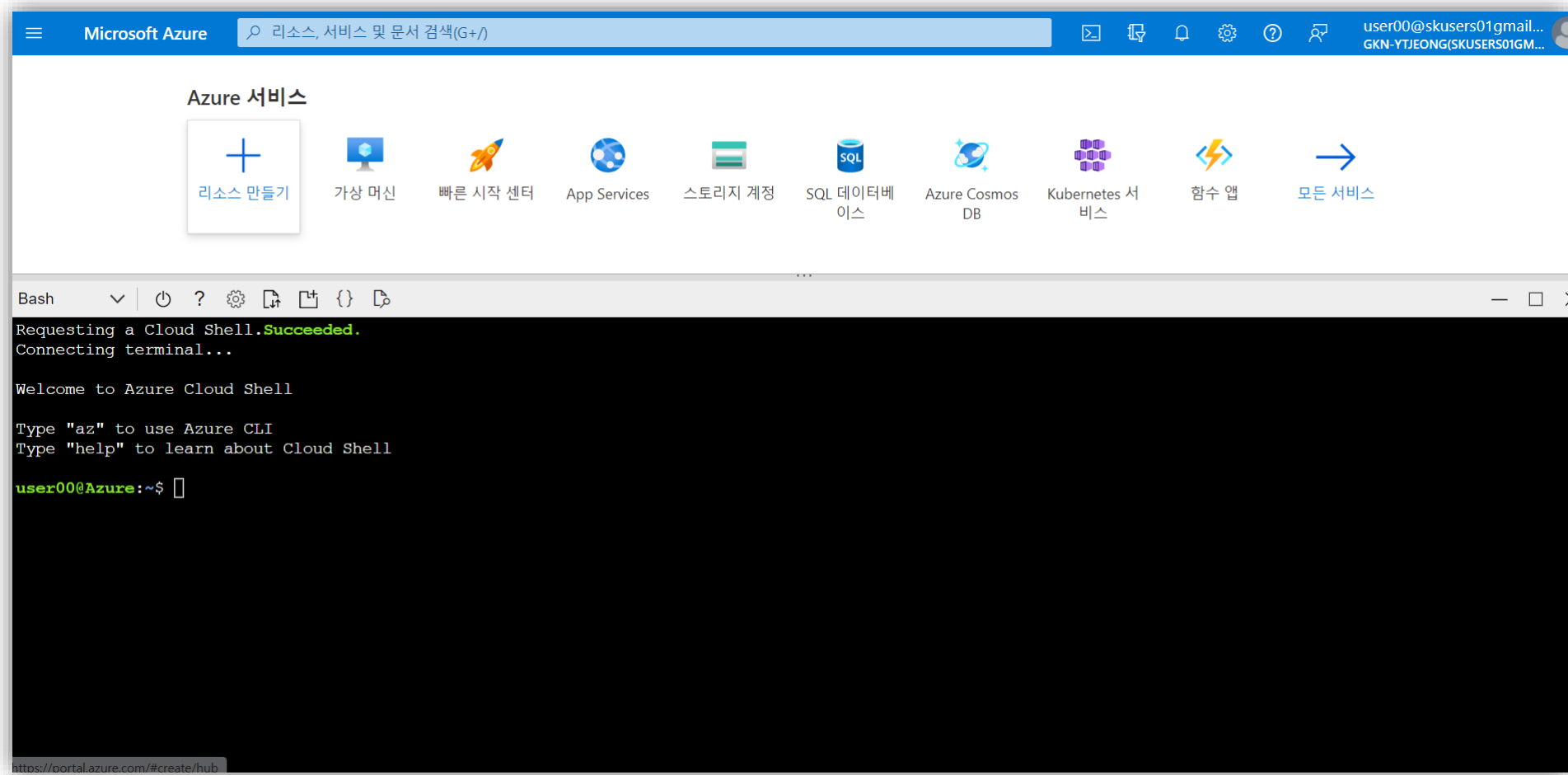
[고급 설정 표시](#)

스토리지 만들기

닫기

# CloudShell 생성

생성 완료





# 클러스터 배포

리소스그룹 생성

```
az group create --name user**-rg --location koreacentral
```

클러스터 배포

```
az aks create --resource-group user**-rg --name ₩  
user**-cluster --node-count 2 --generate-ssh-keys --enable-node-public-ip
```

```
az aks create --resource-group user**-rg --name user**-cluster --node-count 3 --generate-ssh-keys --enable-node-public-ip
```

# 클러스터 정보 인증 및 노드 확인

## 클러스터 정보 인증

```
az aks get-credentials --resource-group user**-rg --name user**-cluster
```

## 노드 확인

```
kubectl get nodes
```

```
user00@Azure:~$ kubectl get node
```

NAME	STATUS	ROLES	AGE	VERSION
aks-nodepool11-26564061-vmss000000	Ready	agent	8m56s	v1.21.9
aks-nodepool11-26564061-vmss000001	Ready	agent	8m55s	v1.21.9
aks-nodepool11-26564061-vmss000002	Ready	agent	8m55s	v1.21.9

# Kubectl 자동완성 기능 적용

Tab 키를 사용하여 자동완성 기능 적용,

kubectl 을 k 로 약어(short name) 설정

```
source <(kubectl completion bash)
echo "source <(kubectl completion bash)" >> ~/.bashrc
source /etc/bash_completion
alias k=kubectl
complete -F __start_kubectl k
```

# Lab 2

## 워크로드 배포

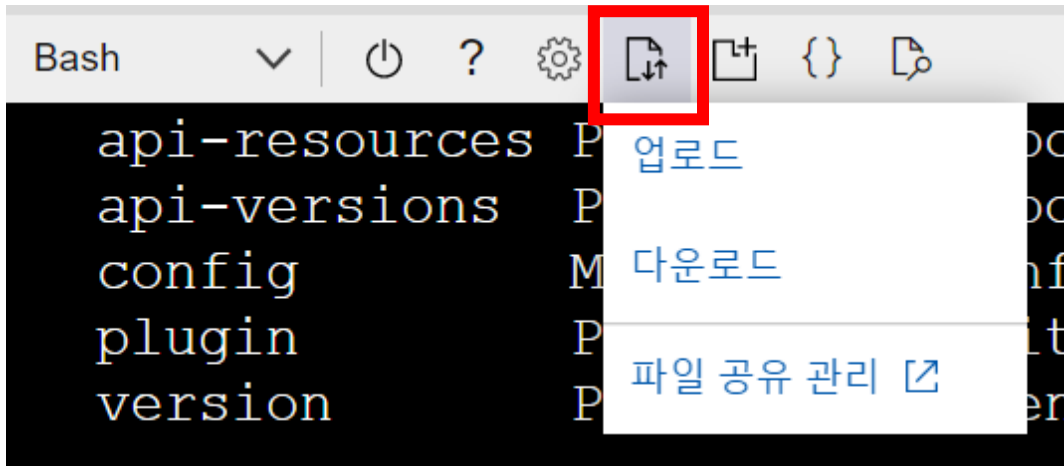
### part 1

---

- Pod
- Replicaset
- Deployment

# 실습 소스 파일 업로드

Cloud Shell 상단 파일 업로드/다운로드 버튼 클릭 후 업로드 클릭하여,  
제공받은 실습 소스파일 업로드



unzip 명령어로 압축 해제 및 디렉토리 이동

```
unzip AKS-YAML.zip
```

```
cd AKS-YAML
```

# Pod

Pod 조회

```
kubectl get pods
```

yaml 확인

```
cat pod.yaml
```

pod 생성

```
kubectl create -f pod.yaml
```

pod 세부 정보 조회

```
kubectl get pod -o wide  
kubectl describe pod
```

# Pod

yaml 없이 아래 조건으로 Pod 생성

```
pod name : no-yaml-pod  
container image : nginx  
container port : 80
```

pod 생성

```
kubectl run no-yaml-pod --image=nginx --port=80
```

pod 정보 확인 후 삭제

```
kubectl get pod  
kubectl describe pod  
kubectl delete pod --all
```

# Namespace

Namespace 조회

```
kubectl get namespace
```

yaml 확인

```
cat ns1.yaml
```

Namespace 생성

```
kubectl create -f ns1.yaml
```

Namespace 정보 조회

```
kubectl get ns
```



# Namespace안에 Pod 생성

Namespace 조회

```
kubectl create -f pod.yaml -n ns1
```

pod 조회

```
kubectl get pod  
kubectl get pod -n ns1
```

Namespace 삭제

```
kubectl delete ns ns1
```

# Replicaset

Replicaset 조회

```
kubectl get rs
```

yaml 확인

```
cat rs1.yaml
```

Replicaset 생성

```
kubectl create -f rs1.yaml
```

Replicaset 세부 조회

```
kubectl get rs  
kubectl describe rs
```

# Replicaset

Pod 조회

```
kubectl get pod
```

생성된 pod 중 1개 삭제

```
kubectl delete pod pod_name
```

Pod 자동 복구 확인

```
kubectl get pod
```

Replicaset 삭제

```
kubectl delete rs --all
```

# Deployment

deployment 조회

```
kubectl get deploy
```

yaml 확인

```
cat dp1.yaml
```

deployment 생성

```
kubectl create -f dp1.yaml
```

deployment 세부 조회

```
kubectl describe deploy
```

# Deployment

Pod 조회

```
kubectl get pod
```

생성된 pod 중 1개 삭제

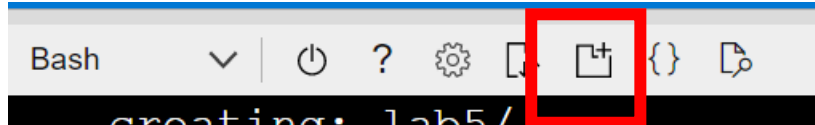
```
kubectl delete pod pod_name
```

Pod 자동 복구 확인

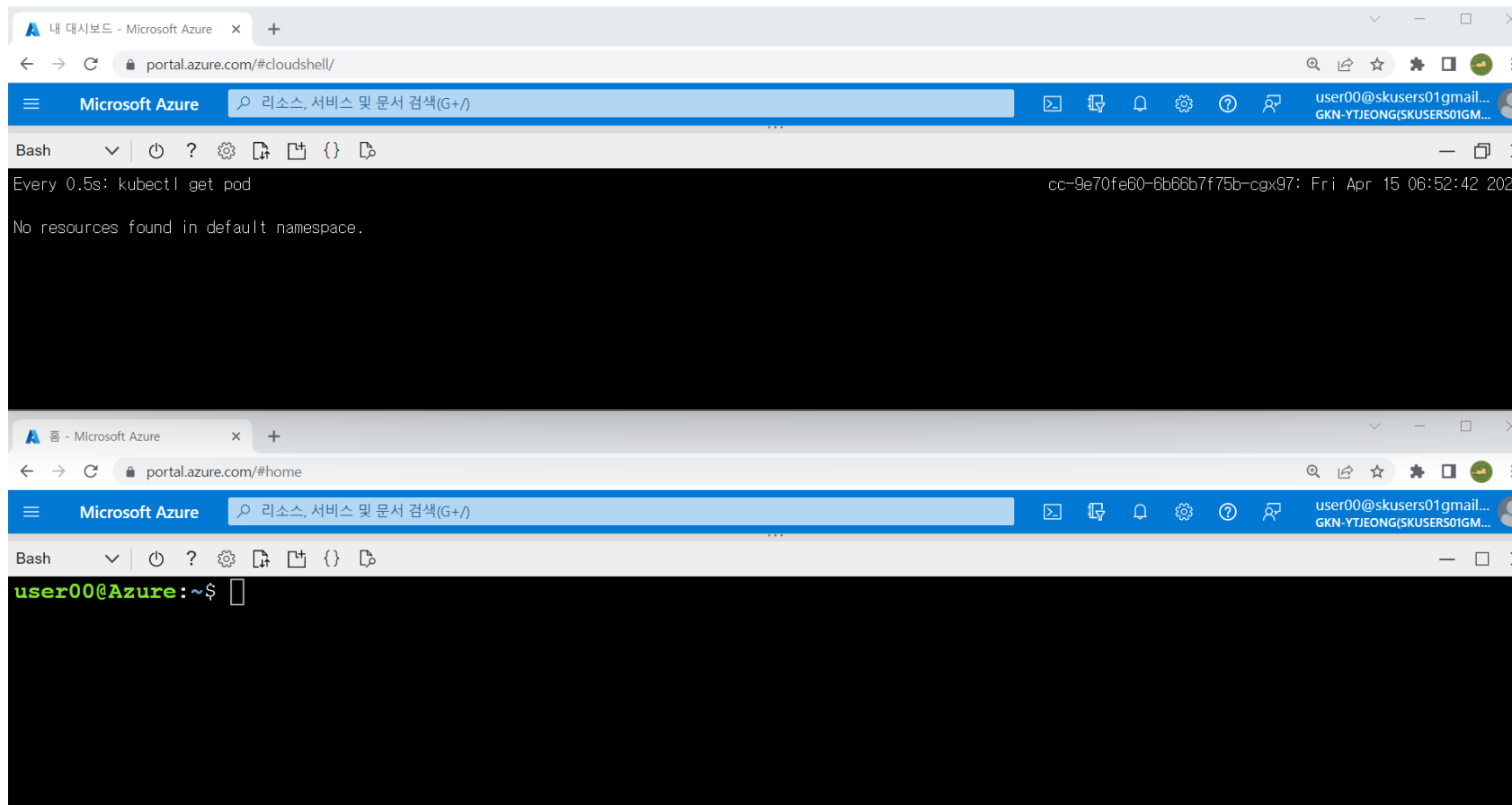
```
kubectl get pod
```

# Deployment

세션 열기 버튼 클릭하여 Cloud Shell 을 추가하여 모니터링용으로 사용



```
watch -n 0.5 kubectl get pod
```



# Deployment

아래 조건에 따라 create 명령으로 deployment 생성

deployment 조건

deployment name : dp2  
container image : nginx:1.14.0  
container port : 80  
replicas : 3

```
kubectl create deploy dp2 --image=nginx:1.14.0 --port=80 --replicas=3
```

생성 확인

```
kubectl get deploy  
kubectl get pod
```

# Deployment

컨테이너 이미지 업데이트 (1.14.0 -> 1.15.0)

```
kubectl set image deployment/dp2 nginx=nginx:1.15.0 --record=true
```

\* 이때 명령어 수행 직후 모니터링 터미널로 업데이트 방식 확인

\* record=true 값으로 해야 히스토리 확인 시 어떤 내용인지 확인 가능

업데이트 내역 확인

```
kubectl describe pod  
kubectl describe deploy
```



# Deployment

업데이트 방식 변경 (vi 편집기 사용 방식과 동일)

```
kubectl edit deploy dp2
```

좌측 박스의 내용을 우측 박스의 내용으로 수정

```
strategy:  
  rollingUpdate:  
    maxSurge: 25%  
    maxUnavailable: 25%  
  type: RollingUpdate
```

```
strategy:  
  type: Recreate
```

# Deployment

컨테이너 이미지 업데이트 (1.14.0 -> 1.15.0 -> 1.16.0)

```
kubectl set image deployment/dp2 nginx=nginx:1.16.0 --record=true
```

\* 이때 명령어 수행 직후 모니터링 터미널로 업데이트 방식 확인

업데이트 내역 확인

```
kubectl describe pod  
kubectl describe deploy
```

# Deployment

롤아웃 기록 확인

```
kubectl rollout history deploy/dp2
```

직전 버전으로 롤백

```
kubectl rollout undo deploy/dp2
```

롤백 이후 버전확인

```
kubectl describe deploy  
kubectl describe pod
```

# Deployment

리비전 히스토리를 지정하여 롤백

```
kubectl rollout undo deploy/dp2 --to-revision=1
```

롤백 이후 버전확인

```
kubectl describe deploy  
kubectl describe pod
```

# Deployment

## 스케일링

```
kubectl scale deploy/dp2 --replicas=5
```

## 스케일링 결과 확인

```
kubectl get pod  
kubectl describe deploy
```

## 리소스 삭제

```
kubectl delete deploy --all
```