# 한국 마이크로소프트

# Microsoft Technical Trainer

Enterprise Skills Initiative

AZ-104. Challenge Lab 09

LAB 03. AKS를 사용하여 애플리케이션 배포



이 문서는 Microsoft Technical Trainer팀에서 ESI 교육 참석자분들에게 제공해 드리는 문서입니다.



#### 요약

이 내용들은 표시된 날짜에 Microsoft에서 검토된 내용을 바탕으로 하고 있습니다. 따라서, 표기된 날짜 이후에 시장의 요구사항에 따라 달라질 수 있습니다. 이 문서는 고객에 대한 표기된 날짜 이후에 변화가 없다는 것을 보증하지 않습니다.

이 문서는 정보 제공을 목적으로 하며 어떠한 보증을 하지는 않습니다.

저작권에 관련된 법률을 준수하는 것은 고객의 역할이며, 이 문서를 마이크로소프트의 사전 동의 없이 어떤 형태(전자 문서, 물리적인 형태 막론하고) 어떠한 목적으로 재 생산, 저장 및 다시 전달하는 것은 허용되지 않습니다.

마이크로소프트는 이 문서에 들어있는 특허권, 상표, 저작권, 지적 재산권을 가집니다. 문서를 통해 명시적으로 허가된 경우가 아니면, 어떠한 경우에도 특허권, 상표, 저작권 및 지적 재산권은 다른 사용자에게 허용되지 아니합니다.

© 2023 Microsoft Corporation All right reserved.

Microsoft®는 미합중국 및 여러 나라에 등록된 상표입니다. 이 문서에 기재된 실제 회사 이름 및 제품 이름은 각 소유자의 상표일 수 있습니다.

## 문서 작성 연혁

날짜	버전	작성자	변경 내용
2023.08.31	1.0.0	우진환	LAB 03 내용 작성



# 목차

도전 과제	3
STEP 01. AKS 클러스터 만들기	5
STEP 02. CLOUD SHELL을 사용하여 AKS 클러스터에 연결	5
STEP 03. AKS 클러스터에서 애플리케이션 실행	5
TASK 01. AKS 클러스터 만들기	5
TASK 02. CLOUD SHELL을 사용하여 AKS 클러스터에 연결	9
TASK 03 AKS 클러스턴에서 애플리케이션 실해	10



#### 도전 과제

이 실습에서는 AKS (Azure Kubernetes Service)를 사용하여 컨테이너화된 애플리케이션을 배포합니다.

- AKS 클러스터를 만듭니다.
- Azure Cloud Shell을 사용하여 AKS 클러스터에 연결합니다.
- AKS 클러스터에 애플리케이션을 컨테이너로 실행합니다.

#### STEP 01. AKS 클러스터 만들기

1. 다음 속성을 사용하여 Azure Kubernetes 클러스터를 만듭니다.

속성	값
리소스 그룹	corp-datalod <xxxxxxxxx></xxxxxxxxx>
클러스터 사전 설정 구성	프로덕션 표준
Kubernetes 클러스터 이름	aks- <xxxxxxxxx< td=""></xxxxxxxxx<>
가용성 영역	없음
시스템 노드 풀	Standard_DS2_v2
노드 개수	1
통합	모든 통합 기능을 사용하지 않음

### STEP 02. Cloud Shell을 사용하여 AKS 클러스터에 연결

1. 다음 속성을 사용하여 [Cloud Shell]의 Bash 세션을 만듭니다.

속성	값
Cloud Shell 지역	미국 동부
리소스 그룹	corp-datalod <xxxxxxxxx></xxxxxxxxx>
스토리지 계정	cs <xxxxxxxx> 계정을 생성</xxxxxxxx>
파일 공유	fs <xxxxxxxxx> 파일 공유 생성</xxxxxxxxx>

- 2. [Cloud Shell]의 Bash 세션에서 az aks get-credentials 명령을 사용하여 AKS 클러스터에 연결합니다.
- 3. kubectl get nodes 명령을 실행하여 클러스터 노드를 확인합니다.

#### STEP 03. AKS 클러스터에서 애플리케이션 실행

- 1. kubectl create deployment 명령을 실행하여 nginx 이미지를 사용하는 Deployment를 만듭니다.
- 2. kubectl get pods 명령을 실행하여 실행 중인 Pod 목록을 확인합니다.
- 3. kubectl expose 명령을 실행하여 LoadBalancer Service 유형을 사용하여 포트 80으로 Pod를 인터넷에 노출합니다.
- 4. kubectl get service 명령을 실행하여 배포한 Service를 확인합니다. 표시되는 EXTERNAL-IP를 확인합니다.
- 5. 브라우저에서 확인 IP에 액세스하여 nginx 기반 웹 앱이 표시되는지 확인합니다.

#### TASK 01. AKS 클러스터 만들기

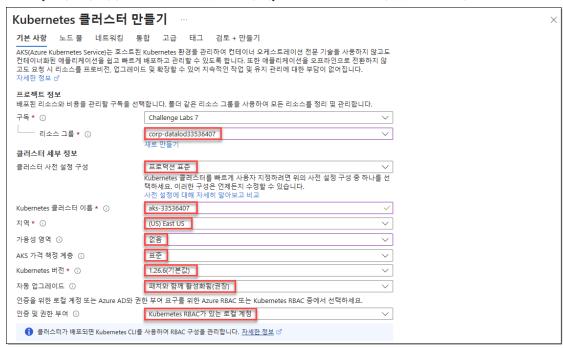
1. Azure 포털의 검색창에서 "Kubernetes 서비스"를 검색한 후 클릭합니다. [Kubernetes 서비스]



블레이드의 메뉴에서 [만들기 - Kubernetes 클러스터 만들기]를 클릭합니다.



- 2. [Kubernetes 클러스터 만들기] 블레이드의 [기본 사항] 탭에서 아래와 같이 구성한 후 [다음]을 클릭합니다.
  - [프로젝트 정보 리소스 그룹]: corp-datalod<xxxxxxxx>
  - [클러스터 세부 정보 클러스터 사전 설정 구성]: 프로덕션 표준
  - [클러스터 세부 정보 Kubernetes 클러스터 이름]: aks-<xxxxxxxx>
  - [클러스터 세부 정보 지역]: (US) East US
  - [클러스터 세부 정보 가용성 영역]: 없음
  - [클러스터 세부 정보 AKS 가격 책정 계층]: 표준
  - [클러스터 세부 정보 Kubernetes 버전]: 기본값을 선택합니다.
  - [클러스터 세부 정보 자동 업그레이드]: 패치와 함께 활성화됨(권장)
  - [클러스터 세부 정보 인증 및 권한 부여]: Kubernetes RBAC가 있는 로컬 계정

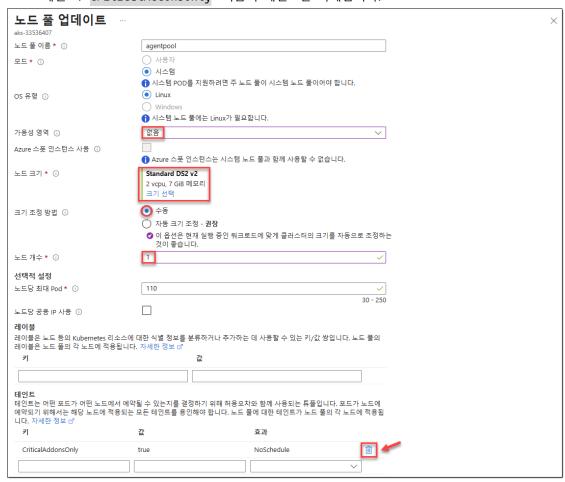


3. [노드 풀] 탭에서 agentpool 이름의 노드 풀을 클릭합니다.





- 4. [노드 풀 업데이트] 블레이드에서 아래와 같이 구성한 후 [업데이트]를 클릭합니다. 언급되지 않는 설정은 기본값을 유지합니다.
  - 가용성 영역: 없음
  - 노드 크기: Standard DS2 v2
  - 크기 조정 방법: 수동
  - 노드 개수: 1
  - 테인트: "CriticalAddonsOnly" 이름의 테인트를 삭제합니다.





5. [노드 풀] 탭에서 "userpool" 이름의 노드 풀을 선택하고 [삭제]를 클릭합니다. 그런 다음 [다음]을 클릭합니다.



6. [네트워킹] 탭에서 기본값을 유지하고 [다음]을 클릭합니다.

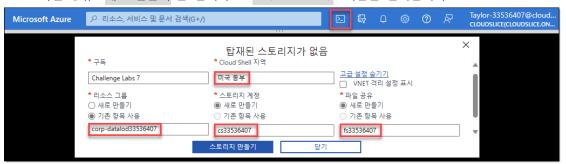


- 7. [통합] 탭에서 아래와 같이 구성한 후 [검토 + 만들기]를 클릭합니다. [검토 + 만들기] 탭에서 [만들기]를 클릭합니다.
  - [Azure Container Registry 컨테이너 레지스트리]: 없음
  - Azure Monitor: 끄기
  - [경고 중 권장 경고 규칙 사용]: 선택하지 않습니다.
  - [Azure Policy Azure Policy]: 사용 안 함

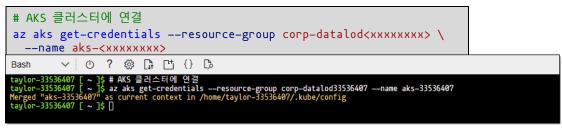


#### TASK 02. Cloud Shell을 사용하여 AKS 클러스터에 연결

- 1. Azure 포털에서 [Cloud Shell] 아이콘을 클릭한 후 "Bash"를 클릭합니다. [탑재된 스토리지가 없음] 창에서 "고급 설정 표시" 링크를 클릭합니다. [탑재된 스토리지가 없음] 페이지에서 아래와 같이 구성한 후 [스토리지 만들기]를 클릭합니다.
  - Cloud Shell 지역: 미국 동부
  - 리소스 그룹: "기존 항목 사용"을 선택하고 corp-datalod<xxxxxxxxx> 리소스 그룹을 선택합니다.
  - 스토리지 계정: "새로 만들기"를 선택하고 "cs<xxxxxxxx>" 이름을 입력합니다.
  - 파일 공유: "새로 만들기"를 선택하고 "fs<xxxxxxxxxx" 이름을 입력합니다.



2. [Cloud Shell]의 Bash 세션에서 다음 명령을 실행하여 AKS 클러스터에 연결합니다.





3. [Cloud Shell]의 Bash 세션에서 다음 명령을 실행하여 실행 중인 클러스터 노드를 확인합니다. 한 대의 클러스터 노드가 실행 중인 것을 확인합니다.

### TASK 03. AKS 클러스터에서 애플리케이션 실행

1. [Cloud Shell]의 Bash 세션에서 다음 명령을 실행하여 nginx 이미지를 사용하는 애플리케이션의 Deployment를 만듭니다.

2. [Cloud Shell]의 Bash 세션에서 다음 명령을 실행하여 AKS 클러스터에서 실행 중인 Pod 목록을 확인합니다. 하나의 Pod가 실행 중인 것을 확인합니다.

3. [Cloud Shell]의 Bash 세션에서 다음 명령을 실행합니다. 이 명령은 LoadBalancer 유형의 Service를 만들고 80 포트를 노출하여 Pod를 연결합니다.

4. [Cloud Shell]의 Bash 세션에서 다음 명령을 실행하여 실행 중인 Service의 상태를 확인합니다. 출력 결과에서 EXTERNAL-IP에 표시되는 IP 주소를 복사합니다.

5. 브라우저에서 새 탭을 열고 위에서 복사한 IP 주소에 액세스합니다. 아래와 같이 nginx 기반으로 실행되는 웹 애플리케이션이 표시되는지 확인합니다.



