

한국 마이크로소프트

Microsoft Technical Trainer

Enterprise Skills Initiative

AZ-104. Challenge Lab 09

LAB 05. AKS를 사용하여 보안 구성

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 않습니다.

© 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 05 내용 작성

목차

도전 과제	5
STEP 01. AKS 클러스터 배포.....	5
STEP 02. 관리자로 AKS 클러스터 연결.....	5
STEP 03. NON-ADMINISTRATIVE 사용자로 AKS 클러스터에 컨테이너 배포	5
TASK 01. RBAC을 사용하는 AKS 클러스터 배포	7
TASK 02. 관리자로 AKS 클러스터에 연결	10
TASK 03. NON-ADMINISTRATIVE 사용자로 AKS 클러스터에 컨테이너 배포	11

도전 과제

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

- AKS 클러스터를 만들고 클러스터에 RBAC 역할을 할당합니다.
- Azure Cloud Shell을 사용하여 관리자로 AKS 클러스터에 연결합니다.
- AKS 클러스터에 개발자로 애플리케이션을 배포한 다음 클러스터 보안을 확인합니다.

STEP 01. AKS 클러스터 배포

- 다음 속성을 사용하여 AKS 클러스터를 배포합니다.

속성	값
리소스 그룹	corp-datalod<XXXXXXXX>
클러스터 사전 설정 구성	표준
Kubernetes 클러스터 이름	aks
가용성 영역	없음
노드 크기	Standard DS2_v2 - 2 vcpu, 8 GiB 메모리
노드 수	최대 2 개
인증 및 권한 부여	Kubernetes RBAC 가 있는 로컬 계정
컨테이너 모니터링	모두 사용하지 않습니다.

- 새로 만든 Kubernetes 서비스에서 액세스 제어를 구성합니다. User1-<XXXXXXXX> 계정이 "Azure Kubernetes Service 클러스터 사용자 역할"을 가지도록 설정합니다.

STEP 02. 관리자로 AKS 클러스터 연결

- Cloud Shell의 Bash 세션을 다음 정보를 사용하여 시작합니다.

속성	값
리소스 그룹	corp-datalod<XXXXXXXX>
Cloud Shell 지역	미국 동부
스토리지 계정	cs<XXXXXXXX>
파일 공유	cloudshell

- Cloud Shell에서 `az aks get-credentials` 명령을 실행하여 AKS 클러스터에 연결합니다.
- Cloud Shell에서 `kubectl config current-context` 명령을 실행하여 컨텍스트를 확인합니다.
- Cloud Shell에서 `kubectl get` 명령을 실행하여 노드를 확인합니다.

STEP 03. Non-administrative 사용자로 AKS 클러스터에 컨테이너 배포

- User1-<XXXXXXXX> 계정으로 포털에 로그인한 후 다음 정보를 사용하여 Cloud Shell의 Bash 세션을 시작합니다.

속성	값
리소스 그룹	corp-datalod<XXXXXXXX>
Cloud Shell 지역	미국 동부
스토리지 계정	cs<XXXXXXXX>

파일 공유	cloudshell
-------	------------

2. Cloud Shell에서 `az aks get-credentials` 명령을 사용하여 AKS 클러스터에 연결합니다.
3. Cloud Shell에서 `kubectl config` 명령을 실행하여 현재 컨텍스트를 확인합니다.
4. Cloud Shell에서 `az aks stop` 명령을 실행하여 AKS 클러스터를 중지합니다. 권한 부족으로 인해 명령이 실패하는 것을 확인합니다.
5. Cloud Shell에서 `kubectl create deployment` 명령을 실행하여 nginx 이미지를 사용하는 컨테이너를 만듭니다.
6. Cloud Shell에서 `kubectl get` 명령을 실행하여 배포한 Pod를 확인합니다.
7. Cloud Shell에서 `kubectl expose deployment` 명령을 실행하여 새 Service 개체를 만듭니다.
8. Cloud Shell에서 `kubectl get services` 명령을 실행하여 공용 IP를 확인한 후 연결을 확인합니다.

TASK 01. RBAC을 사용하는 AKS 클러스터 배포

1. Azure 포털의 검색창에서 "Kubernetes 클러스터"를 검색한 후 클릭합니다. [Kubernetes 서비스] 블레이드에서 [만들기 - Kubernetes 클러스터 만들기]를 클릭합니다.



2. [Kubernetes 클러스터 만들기] 블레이드의 [기본 사항] 탭에서 아래와 같이 구성한 후 [액세스] 탭으로 이동합니다.

- [프로젝트 정보 - 리소스 그룹]: corp-datalod<XXXXXXXX>
- [클러스터 세부 정보 - 클러스터 사전 설정 구성]: 표준(\$\$)
- [클러스터 세부 정보 - Kubernetes 클러스터 이름]: aks
- [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]: 기본값 버전을 사용합니다.
- [클러스터 세부 정보 - 자동 업그레이드]: 패치와 함께 활성화됨
- [주 노드 풀 - 노드 크기]: Standard DS2 v2
- [주 노드 풀 - 크기 조정 방법]: 자동 크기 조정
- [주 노드 풀 - 노드 수 범위]: 최대 2대로 설정합니다.

Kubernetes 클러스터 만들기

기본 사항 노드 풀 액세스 네트워킹 통합 고급 태그 검토 + 만들기

AKS(Azure Kubernetes Service)는 호스트된 Kubernetes 환경을 관리하여 컨테이너 오케스트레이션 전문 기술을 사용하지 않고도 컨테이너화된 애플리케이션을 쉽고 빠르게 배포하고 관리할 수 있도록 합니다. 또한 애플리케이션을 오프라인으로 전환하지 않고도 요청 시 리소스를 프로비전, 업그레이드 및 확장할 수 있어 지속적인 작업 및 유지 관리에 대한 부담이 없어집니다.

[자세한 정보](#)

프로젝트 정보
배포된 리소스와 비용을 관리할 구독을 선택합니다. 폴더 같은 리소스 그룹을 사용하여 모든 리소스를 정리 및 관리합니다.

구독 * ① Challenge Labs 3
리소스 그룹 * ① corp-datalod33123656
[새로 만들기](#)

클러스터 세부 정보
클러스터 사전 설정 구성

표준(\$\$)
Kubernetes 클러스터를 빠르게 사용자 지정하려면 위의 사전 설정 구성 중 하나를 선택하세요. 이러한 구성은 언제든지 수정할 수 있습니다.
[사전 설정에 대해 자세히 알아보고 비교](#)

Kubernetes 클러스터 이름 * ① aks
지역 * ① (US) East US
가용성 영역 ① 없음
표준 구성에는 고가용성이 권장됩니다.

AKS 가격 책정 계층 ① 표준
Kubernetes 버전 * ① 1.26.6(기본값)
자동 업그레이드 ① 패치와 함께 활성화됨(권장)

주 노드 풀
클러스터의 주 노드 풀에 있는 노드의 수와 크기입니다. 프로덕션 워크로드의 경우 복원력을 위해 3개 이상의 노드가 권장됩니다. 개발 또는 테스트 워크로드의 경우 하나의 노드만 필요합니다. 노드 풀을 추가하거나 이 노드 풀의 추가 구성 옵션을 보려면 위의 '노드 풀' 탭으로 이동하세요. 클러스터를 만든 후 다른 노드 풀을 추가할 수 있습니다.
[Azure Kubernetes Service의 노드 풀에 대한 자세한 정보](#)

노드 크기 * ① Standard DS2 v2
표준 DS2_v2는 표준 구성에 권장됩니다.
[크기 변경](#)

크기 조정 방법 * ① ☒ 자동 크기 조정
표준 구성에는 자동 스케일링을 사용하는 것이 좋습니다.

노드 수 범위 * ① 1 2

3. [액세스] 탭에서 인증 및 권한 부여를 "Kubernetes RBAC가 있는 로컬 계정"으로 선택한 후 [통합] 탭으로 이동합니다.

Kubernetes 클러스터 만들기

기본 사항 노드 풀 액세스 네트워킹 통합 고급 태그 검토 + 만들기

리소스 ID ① 시스템에서 할당한 관리 ID
기본적으로 Azure는 관리 ID를 사용합니다. 서비스 주체를 사용하려면 CLI를 사용하세요. [자세한 정보](#)

인증 및 권한 부여 ① Kubernetes RBAC가 있는 로컬 계정

클러스터가 배포되면 Kubernetes CLI를 사용하여 RBAC 구성을 관리합니다. [자세한 정보](#)

4. [통합] 탭에서 아래와 같이 구성한 후 [검토 + 만들기]를 클릭합니다. [검토 + 만들기] 탭에서 [만들기]를 클릭합니다.

- 컨테이너 인사이트: 선택하지 않습니다.
- 관리되는 Prometheus: 선택하지 않습니다.
- 관리 Grafana: 선택하지 않습니다.
- 경고 중: 선택하지 않습니다.
- Azure Policy: 사용 안 함

Kubernetes 클러스터 만들기 ...

기본 사항 노드 풀 액세스 네트워킹 **통합** 고급 태그 검토 + 만들기

AKS 클러스터를 추가 서비스와 연결합니다.

클라우드용 Microsoft Defender
클라우드용 Microsoft Defender는 하이브리드 클라우드 워크로드 전반에 걸쳐 통합 보안 관리 및 고급 위협 보호를 제공합니다.
[자세한 정보](#)

✓ 구독은 클라우드용 Microsoft Defender 기본 플랜으로 보호됩니다.

Azure Container Registry
클러스터를 Azure Container Registry에 연결하여 개인 이미지 레지스트리에서 원활한 배포를 지원합니다.
[Azure Container Registry에 대한 자세한 정보](#)

컨테이너 레지스트리: 없음

① Azure Kubernetes Service로 Azure Container Registry에 연결하려면 사용자 액세스 관리자 역할이 있는 구독 기여자이거나 구독 소유자여야 합니다.

Azure Monitor
AKS에 기본적으로 포함된 CPU 및 메모리 메트릭 외에도 컨테이너 인사이트를 사용하도록 설정하여 클러스터의 전체 성능 및 상태에 대한 보다 포괄적인 데이터를 살펴볼 수 있습니다. 청구는 데이터 수집 및 보존 설정을 기준으로 이루어집니다.
[컨테이너 성능 및 상태 모니터링에 대한 자세한 정보](#)
[가격에 대한 자세한 정보](#)

컨테이너 인사이트
컨테이너 로그 사용 ☐
표준 구성에는 Azure Monitor가 권장됩니다.

관리되는 Prometheus
관리 Prometheus는 컨테이너화된 워크로드를 모니터링하는 데 확장성 있는 고가용성 보안 메트릭 플랫폼을 제공합니다.
[자세한 정보](#)

Prometheus 메트릭 사용 ☐

관리 Grafana
Azure Monitor 작업 영역에 저장된 관리되는 Prometheus 데이터를 시각화하기 위해 Grafana의 완전 관리형 인스턴스를 선택합니다. [가격에 대한 자세한 정보](#)

Grafana 사용 ☐

경고 중
권장 경고 규칙 사용 ☐

Azure Policy
Azure Policy를 통해 일관된 중앙 집중식 방식으로 AKS 클러스터의 대규모 적용 및 보호 기능을 적용하세요.
[AKS용 Azure Policy에 대한 자세한 정보](#)

Azure Policy: ☐ 사용 ☒ 사용 안 함

5. [aks Kubernetes 서비스] 블레이드의 [액세스 제어]로 이동한 후 메뉴에서 [추가 - 역할 할당 추가]를 클릭합니다.

aks | 액세스 제어(IAM) ☆ ...

Kubernetes 서비스

검색

개요 활동 로그 **액세스 제어(IAM)** 태그 문제 진단 및 해결 클라우드용 Microsoft Defender

Kubernetes 리소스 네임스페이스

« **추가** ↓ 역할 할당 다운로드 열 편집 새로 고침 제거 사용자 의견

역할 할당 추가

역할 거부 할당 클래식 관리자

공통 관리자 추가

내 액세스
이 리소스에 대한 내 액세스 수준 보기

내 액세스 보기

액세스 권한 확인
사용자, 그룹, 서비스 보안 주체 또는 관리 ID가 이 리소스에 대해 보유한 액세스 권한 수준을 검토하세요. [자세한 정보](#)

액세스 권한 확인

6. [역할 할당 추가] 블레이드의 [역할]에서 "Kubernetes"로 검색한 후 "Azure Kubernetes Service 클러스터 사용자 역할"을 선택하고 [다음]을 클릭합니다.

역할 할당 추가 ...

역할 구성원 검토 + 할당

역할 정의는 권한 컬렉션입니다. 기본 제공 역할을 사용하거나 사용자 지정 역할을 만들 수 있습니다. [자세한 정보](#)

할당 유형

작업 가능 역할 권한 있는 관리자 역할

가상 머신을 만드는 기능과 같은 작업 기능을 기반으로 Azure 리소스에 대한 액세스 권한을 부여합니다.

× 형식: 모두 범주: 모두

이름 ↑	설명 ↑	형식 ↑	범주 ↑	세부 정보
Azure Kubernetes Service 기여자 역할	Azure Kubernetes Service 클러스터를 읽고 ...	BuiltInRole	컨테이너	보기
Azure Kubernetes Service 클러스터 관리 역할	클러스터 관리 자격 증명 작업을 나열합니다.	BuiltInRole	컨테이너	보기
Azure Kubernetes Service 클러스터 사용자 역할	클러스터 사용자 자격 증명 작업을 나열합...	BuiltInRole	컨테이너	보기
Azure Kubernetes Service Cluster Monitoring User	List cluster monitoring user credential action.	BuiltInRole	없음	보기
Azure Kubernetes Service RBAC 관리자	리소스 할당량 및 네임스페이스 업데이트 ...	BuiltInRole	컨테이너	보기

7. [구성원] 탭에서 "구성원 선택"을 클릭합니다. [구성원 선택] 창에서 "User1-`<XXXXXXXXXX>`"를 검색한 후 선택하고 [선택]을 클릭합니다. [구성원] 탭에서 [검토 + 할당]을 클릭합니다. [검토 + 할당] 탭에서 [검토 + 할당]을 클릭합니다.

홈 > Kubernetes 서비스 > aks | 액세스 제어(IAM) >

역할 할당 추가 ...

역할 구성원 검토 + 할당

선택한 역할: Azure Kubernetes Service 클러스터 사용자 역할

다음에 대한 액세스 할당: ☒ 사용자, 그룹 또는 서비스 주체 ☐ 관리 ID

구성원 **+ 구성원 선택**

이름	개체 ID	유형
선택한 구성원 없음		

Description: 선택 사항

구성원 선택

선택

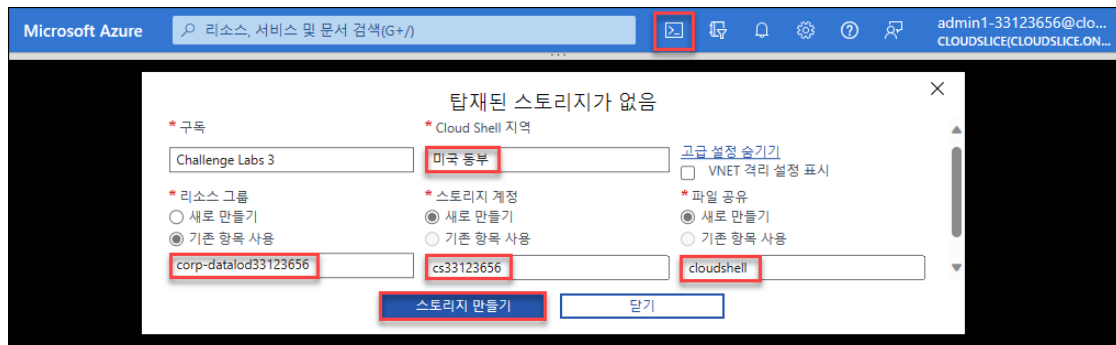
사용자, 그룹 또는 서비스 보안 주체가 없습니다.

선택한 구성원:

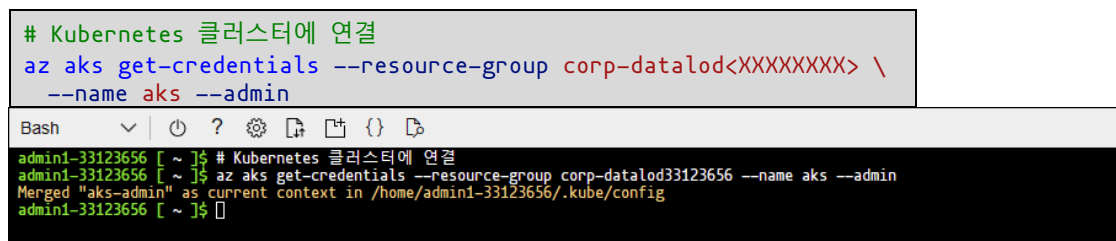
US	User1-33123656	제거
	User1-33123656@cloudslice.onmicros...	

TASK 02. 관리자로 AKS 클러스터에 연결

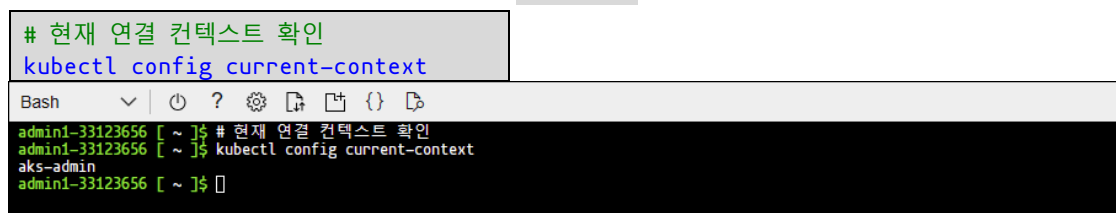
- Azure 포털의 우측 상단에서 [Cloud Shell] 아이콘을 클릭한 후 "Bash"를 선택합니다. [탐재된 스토리지가 없음] 창에서 "고급 설정 표시"를 클릭합니다. [탐재된 스토리지가 없음] 페이지에서 아래와 같이 구성한 후 [스토리지 만들기]를 클릭합니다.
 - Cloud Shell 지역: 미국 동부
 - 리소스 그룹: corp-datalod`<XXXXXXXXXX>`
 - 스토리지 계정: cs`<XXXXXXXXXX>`
 - 파일 공유: cloudshell



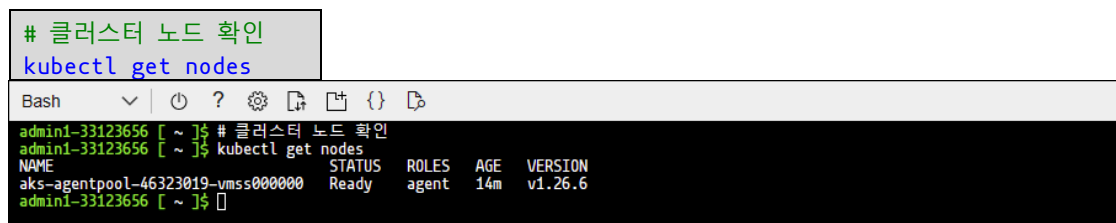
2. [Cloud Shell]의 Bash 세션에서 다음 명령을 실행하여 AKS 클러스터에 연결합니다.



3. [Cloud Shell]의 Bash 세션에서 다음 명령을 실행하여 클러스터 연결에 대한 현재 컨텍스트를 확인합니다. 명령 출력에서 현재 컨텍스트가 "aks-admin"으로 표시되는지 확인합니다.



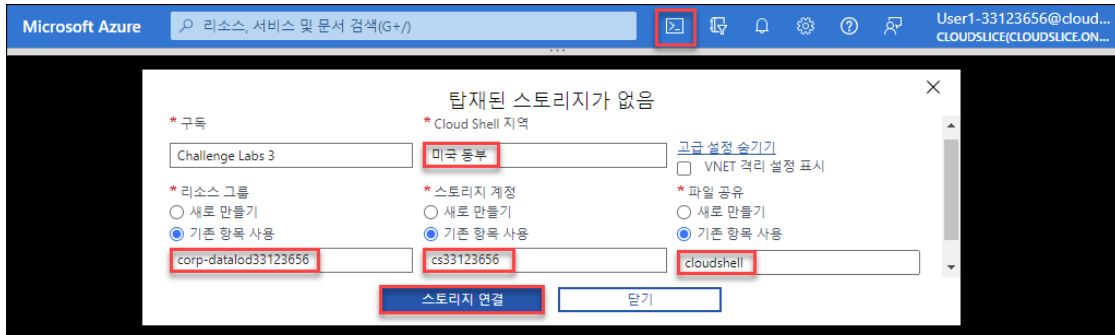
4. [Cloud Shell]의 Bash 세션에서 다음 명령을 실행하여 클러스터 노드 목록을 확인합니다.



TASK 03. non-administrative 사용자로 AKS 클러스터에 컨테이너 배포

1. Microsoft Edge 브라우저에서 InPrivate 창을 열고 Azure 포털에 연결합니다. 개발자 계정(**User1-XXXXXXXXXX**)을 사용하여 로그인합니다.
2. Azure 포털의 우측 상단에서 [Cloud Shell] 아이콘을 클릭한 후 "**Bash**"를 선택합니다. [탐재된 스토리지가 없음] 창에서 "**고급 설정 표시**"를 클릭합니다. [탐재된 스토리지가 없음] 페이지에서 아래와 같이 구성한 후 [스토리지 연결]을 클릭합니다. 로그인 사용자 계정은 Azure 포털에 읽기 액세스 권한만 있기 때문에 파일 공유 탐색과 관련된 오류가 발생하며 이는 무시할 수 있습니다.
 - Cloud Shell 지역: 미국 동부
 - 리소스 그룹: corp-datalodXXXXXXXXXX

- 스토리지 계정: "기존 항목 사용"을 선택한 후 "cs<XXXXXXXXXX>" 계정을 선택합니다.
- 파일 공유: "기존 항목 사용"을 선택한 후 "cloudshell"을 입력합니다.



- [Cloud Shell]의 Bash 세션에서 다음 명령을 실행하여 사용자 컨텍스트로 Kubernetes 클러스터에 연결합니다.

```
# Kubernetes 에 사용자 컨텍스트로 연결
az aks get-credentials --resource-group corp-datalod<XXXXXXXXXX> --name aks
```

```
Bash user1-33123656 [ ~ ]$ # Kubernetes에 사용자 컨텍스트로 연결
user1-33123656 [ ~ ]$ az aks get-credentials --resource-group corp-datalod33123656 --name aks
Merged "aks" as current context in /home/user1-33123656/.kube/config
user1-33123656 [ ~ ]$
```

- [Cloud Shell]의 Bash 세션에서 다음 명령을 실행하여 클러스터에 연결된 현재 컨텍스트를 확인합니다. 컨텍스트가 "aks"로 표시되는 것을 확인합니다.

```
# 현재 연결 컨텍스트 확인
kubectl config current-context
```

```
Bash user1-33123656 [ ~ ]$ # 현재 연결 컨텍스트 확인
user1-33123656 [ ~ ]$ kubectl config current-context
aks
user1-33123656 [ ~ ]$
```

- [Cloud Shell]의 Bash 세션에서 다음 명령을 실행하여 Kubernetes 클러스터를 중지합니다. 현재 컨텍스트에서 클러스터에 대한 권한이 없기 때문에 아래와 같이 오류가 발생하는 것을 확인할 수 있습니다.

```
# AKS 클러스터 중지
az aks stop --resource-group corp-datalod<XXXXXXXXXX> --name aks
```

```
Bash user1-33123656 [ ~ ]$ # AKS 클러스터 중지
user1-33123656 [ ~ ]$ az aks stop --resource-group corp-datalod33123656 --name aks
(Message: The client 'User1-33123656@cloudslice.onmicrosoft.com' with object id 'c280331e-0272-4bc3-8c15-da9343e72c5d' does not have permission to perform action 'Microsoft.ContainerService/managedClusters/stop/action' over scope '/subscriptions/4a5a6077-91a6-4b7a-ae62-a2ed402b1a1d33123656/providers/Microsoft.ContainerService/managedClusters/aks' or the scope is invalid. If access was recently granted, please refresh your credentials.
Code: AuthorizationFailed
Message: The client 'User1-33123656@cloudslice.onmicrosoft.com' with object id 'c280331e-0272-4bc3-8c15-da9343e72c5d' does not have permission to perform action 'Microsoft.ContainerService/managedClusters/stop/action' over scope '/subscriptions/4a5a6077-91a6-4b7a-ae62-a2ed402b1a1d33123656/providers/Microsoft.ContainerService/managedClusters/aks' or the scope is invalid. If access was recently granted, please refresh your credentials.
ls.
user1-33123656 [ ~ ]$
```

- [Cloud Shell]의 Bash 세션에서 nginx 이미지를 사용하는 Deployment를 배포합니다.

```
# nginx 이미지로 새 Deployment 배포
kubectl create deployment nginx-<XXXXXXXXXX> --image=nginx
```

```
Bash
user1-33123656 [ ~ ]$ # nginx 이미지로 새 Deployment 배포
user1-33123656 [ ~ ]$ kubectl create deployment nginx-33123656 --image=nginx
deployment.apps/nginx-33123656 created
user1-33123656 [ ~ ]$
```

7. [Cloud Shell]의 Bash 세션에서 다음 명령을 실행하여 애플리케이션에 대한 Pod가 생성되었는지 확인합니다.

```
# 배포한 Pod 확인
kubectl get pods
```

```
Bash
user1-33123656 [ ~ ]$ # 배포한 Pod 확인
user1-33123656 [ ~ ]$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-33123656-b76b9897f-x64z9      1/1     Running   0           69s
user1-33123656 [ ~ ]$
```

8. [Cloud Shell]의 Bash 세션에서 다음 명령을 실행하여 Service 개체를 만들어 Pod를 인터넷에 노출시킵니다.

```
# Service 개체 배포
kubectl expose deployment nginx-XXXXXXX --port=80 --type=LoadBalancer
```

```
Bash
user1-33123656 [ ~ ]$ # Service 개체 배포
user1-33123656 [ ~ ]$ kubectl expose deployment nginx-33123656 --port=80 --type=LoadBalancer
service/nginx-33123656 exposed
user1-33123656 [ ~ ]$
```

9. [Cloud Shell]의 Bash 세션에서 다음 명령을 실행하여 Service 개체에 할당된 공용 IP 주소를 확인합니다. EXTERNAL-IP 열의 값이 공용 IP이며 이 IP 주소를 메모장에 복사합니다.

```
# 배포한 Service 확인
kubectl get services
```

```
Bash
user1-33123656 [ ~ ]$ # 배포한 Service 확인
user1-33123656 [ ~ ]$ kubectl get services
NAME                TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
kubernetes          ClusterIP   10.0.0.1      <none>         443/TCP          29m
nginx-33123656      LoadBalancer 10.0.222.92   20.81.35.207  80:31324/TCP     41s
user1-33123656 [ ~ ]$
```

10. 브라우저에서 새 탭을 열고 위에서 복사한 공용 IP 주소에 액세스합니다. 아래와 같이 NGINX 웹 페이지가 표시되는 것을 확인합니다.

