

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 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 03 내용 작성

목차

도전 과제	5
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 클러스터에서 애플리케이션 실행	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

- 다음 속성을 사용하여 Azure Kubernetes 클러스터를 만듭니다.

속성	값
리소스 그룹	corp-data1od<xxxxxxxx>
클러스터 사전 설정 구성	프로덕션 표준
Kubernetes 클러스터 이름	aks-<xxxxxxxx>
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

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

속성	값
Cloud Shell 지역	미국 동부
리소스 그룹	corp-data1od<xxxxxxxx>
스토리지 계정	cs<xxxxxxxx> 계정을 생성
파일 공유	fs<xxxxxxxx> 파일 공유 생성

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

STEP 03. AKS 클러스터에서 애플리케이션 실행

- `kubectl create deployment` 명령을 실행하여 nginx 이미지를 사용하는 `Deployment`를 만듭니다.
- `kubectl get pods` 명령을 실행하여 실행 중인 `Pod` 목록을 확인합니다.
- `kubectl expose` 명령을 실행하여 `LoadBalancer Service` 유형을 사용하여 포트 80으로 `Pod`를 인터넷에 노출합니다.
- `kubectl get service` 명령을 실행하여 배포한 `Service`를 확인합니다. 표시되는 `EXTERNAL-IP`를 확인합니다.
- 브라우저에서 확인 IP에 액세스하여 nginx 기반 웹 앱이 표시되는지 확인합니다.

TASK 01. AKS 클러스터 만들기

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

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



- [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

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

Kubernetes 클러스터 만들기

기본 사항 **노드 풀** 네트워킹 통합 고급 태그 검토 + 만들기

노드 풀
기본 템플에 구성된 필수 주 노드 풀 외에도 선택적 노드 풀을 추가하여 다양한 워크로드를 처리할 수 있습니다.
[노드 풀에 대해 자세히 알아보기](#)

+ 노드 풀 추가 - 삭제

<input type="checkbox"/> 이름	모드	노드 크기	OS 유형	노드 개수	가용성 영역	최대 Pod/노드
<input checked="" type="checkbox"/> agentpool	시스템	Standard_D8ds_v5(변경)	Linux	2 - 5	1,2,3	110
<input type="checkbox"/> userpool	사용자	Standard_D8ds_v5(변경)	Linux	2 - 100	1,2,3	110

가상 노드 사용
가상 노드는 서버리스 Azure Container Instances에서 지원하는 버스트 가능 크기 조정을 허용합니다.
[가상 노드에 대한 자세한 정보](#)

가상 노드 사용 ☐

노드 풀 OS 디스크 암호화
기본적으로 AKS의 모든 디스크는 Microsoft 관리형 키를 사용하여 미사용 암호화됩니다. 암호화에 대한 추가 제어가 필요한 경우 Azure Key Vault에서 지원하는 디스크 암호화 집합을 사용하여 고유한 키를 제공할 수 있습니다. 디스크 암호화 집합은 클러스터의 모든 노드 풀에 대해 OS 디스크를 암호화하는 데 사용됩니다. [자세한 정보](#)

암호화 형식 (기본값) 플랫폼 관리형 키로 미사용 데이터 암호화

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.

노드 풀 업데이트

aks-33536407

노드 풀 이름 *

모드 * ☒ 사용자 ☒ 시스템
시스템 POD를 지원하려면 주 노드 풀이 시스템 노드 풀이어야 합니다.

OS 유형 ☒ Linux ☐ Windows
시스템 노드 풀에는 Linux가 필요합니다.

가용성 영역

Azure 스폿 인스턴스 사용 ☐
Azure 스폿 인스턴스는 시스템 노드 풀과 함께 사용할 수 없습니다.

노드 크기 *
2 vcpu, 7 GiB 메모리
[크기 선택](#)

크기 조정 방법 ☒ 수동 ☐ 자동 크기 조정 - 권장
이 옵션은 현재 실행 중인 워크로드에 맞게 클러스터의 크기를 자동으로 조정하는 것이 좋습니다.

노드 개수 *

선택적 설정

노드당 최대 Pod * 30 - 250

노드당 공용 IP 사용 ☐

레이블
레이블은 노드 등의 Kubernetes 리소스에 대한 식별 정보를 분류하거나 추가하는 데 사용할 수 있는 키/값 쌍입니다. 노드 풀의 레이블은 노드 풀의 각 노드에 적용됩니다. [자세한 정보](#)

키	값	효과
CriticalAddonsOnly	true	NoSchedule

테인트
테인트는 어떤 포드가 어떤 노드에서 예약될 수 있는지를 결정하기 위해 허용자와 함께 사용되는 튜플입니다. 포드가 노드에 예약되기 위해서는 해당 노드에 적용되는 모든 테인트를 용인해야 합니다. 노드 풀에 대한 테인트가 노드 풀의 각 노드에 적용됩니다. [자세한 정보](#)

키 값 효과



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

Kubernetes 클러스터 만들기 ...

기본 사항 **노드 풀** 네트워킹 통합 고급 태그 검토 + 만들기

노드 풀
기본 탭에 구성된 필수 주 노드 풀 외에도 선택적 노드 풀을 추가하여 다양한 워크로드를 처리할 수 있습니다.
[노드 풀에 대해 자세히 알아보기](#)

+ 노드 풀 추가 **삭제**

이름	모드	노드 크기	OS 유형	노드 개수	가용성 영역	최대 Pod/노드
agentpool	시스템	Standard_DS2_v2(변경)	Linux	1	없음	110
userpool	사용자	Standard_D8ds_v5(변경)	Linux	2 - 100	1,2,3	110

가상 노드 사용
가상 노드는 서버리스 Azure Container Instances에서 지원하는 베스트 가능 크기 조정을 허용합니다.
[가상 노드에 대한 자세한 정보](#)

가상 노드 사용 ☐

노드 풀 OS 디스크 암호화
기본적으로 AKS의 모든 디스크는 Microsoft 관리형 키를 사용하여 미사용 암호화됩니다. 암호화에 대한 추가 제어가 필요한 경우 Azure Key Vault에서 지원하는 디스크 암호화 집합을 사용하여 고유한 키를 제공할 수 있습니다. 디스크 암호화 집합은 클러스터의 모든 노드 풀에 대해 OS 디스크를 암호화하는 데 사용됩니다. [자세한 정보](#)

암호화 형식 (기본값) 플랫폼 관리형 키로 미사용 데이터 암호화

6. [네트워킹] 탭에서 기본값을 유지하고 [다음]을 클릭합니다.

Kubernetes 클러스터 만들기 ...

기본 사항 노드 풀 **네트워킹** 통합 고급 태그 검토 + 만들기

'kubenet' 또는 'Azure CNI'의 두 가지 네트워킹 옵션 중에서 선택할 수 있습니다.
 • **kubenet** 플러그인을 사용하면 Pod가 오버레이 IP 범위 방식으로 프라이빗 호스트 범위에서 IP를 받을 수 있습니다.
 • **Azure CNI** Vnet 플러그인을 사용하면 Pod가 Azure 서브넷에서 바로 IP를 받을 수 있습니다.

[Azure Kubernetes Service의 네트워킹에 대한 자세한 정보](#)

네트워크 구성 ☐ kubenet ☒ Azure CNI

자체 가상 네트워크 사용 ☐

DNS 이름 접두사 *

트래픽 라우팅
부하 분산 장치

보안
프라이빗 클러스터 사용 ☐
권한 있는 IP 범위 설정 ☐
네트워크 정책 ☐ 없음 ☒ Calico ☐ Azure

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: 고기
- [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. [탐재된 스토리지가 없음] 창에서 "고급 설정 표시" 링크를 클릭합니다. [탐재된 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<xxxxxxxx> 리소스 그룹을 선택합니다.
 - 스토리지 계정: "새로 만들기"를 선택하고 "cs<xxxxxxxx>" 이름을 입력합니다.
 - 파일 공유: "새로 만들기"를 선택하고 "fs<xxxxxxxx>" 이름을 입력합니다.

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

```
# AKS 클러스터에 연결
az aks get-credentials --resource-group corp-datalod<xxxxxxxx> \
  --name aks-<xxxxxxxx>

Bash
taylor-33536407 [ ~ ]$ # AKS 클러스터에 연결
taylor-33536407 [ ~ ]$ az aks get-credentials --resource-group corp-datalod33536407 --name aks-33536407
Merged "aks-33536407" as current context in /home/taylor-33536407/.kube/config
taylor-33536407 [ ~ ]$
```

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

```
# AKS 클러스터의 노드 확인
kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
aks-agentpool-38553913-vmss000000	Ready	agent	6m38s	v1.26.6

TASK 03. AKS 클러스터에서 애플리케이션 실행

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

```
# Deployment 배포
kubectl create deployment nginx-<xxxxxxxx> --image=nginx
```

Bash

```
taylor~33536407 [ ~ ] $ # Deployment 배포
taylor~33536407 [ ~ ] $ kubectl create deployment nginx-33536407 --image=nginx
deployment.apps/nginx-33536407 created
taylor~33536407 [ ~ ] $
```

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

```
# 실행 중인 Pod 확인
kubectl get pods
```

```
Bash
```

```
taylor~33536407 [ ~ ]$ # 실행 중인 Pod 확인
taylor~33536407 [ ~ ]$ kubectl get pods
```

NAME	STATUS	RESTARTS	AGE
nginx-33536407-7dccc57f8d5-42q89	1/1	Running	0
taylor~33536407 [~]\$			76s

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

```
# LoadBalancer 유형의 Service 만들기
kubectl expose deployment nginx-xxxxxxx --port=80 --type=LoadBalancer
```

Bash

```
taylor-33536407 [ ~ ]$ # LoadBalancer 유형의 Service 만들기
taylor-33536407 [ ~ ]$ kubectl expose deployment nginx-33536407 --port=80 --type=LoadBalancer
service/nginx-33536407 exposed
taylor-33536407 [ ~ ]$
```

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

```
# Service 목록 확인
kubectl get services
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.0.0.1	<none>	443/TCP	16m
nginx-33536407	LoadBalancer	10.0.196.89	4.236.211.66	80:31200/TCP	105s

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

