한국 마이크로소프트

Microsoft Technical Trainer

Enterprise Skills Initiative

AZ-104. LAB09C

Azure Kubernetes Service 구현

Microsoft

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.

문서 작성 연혁

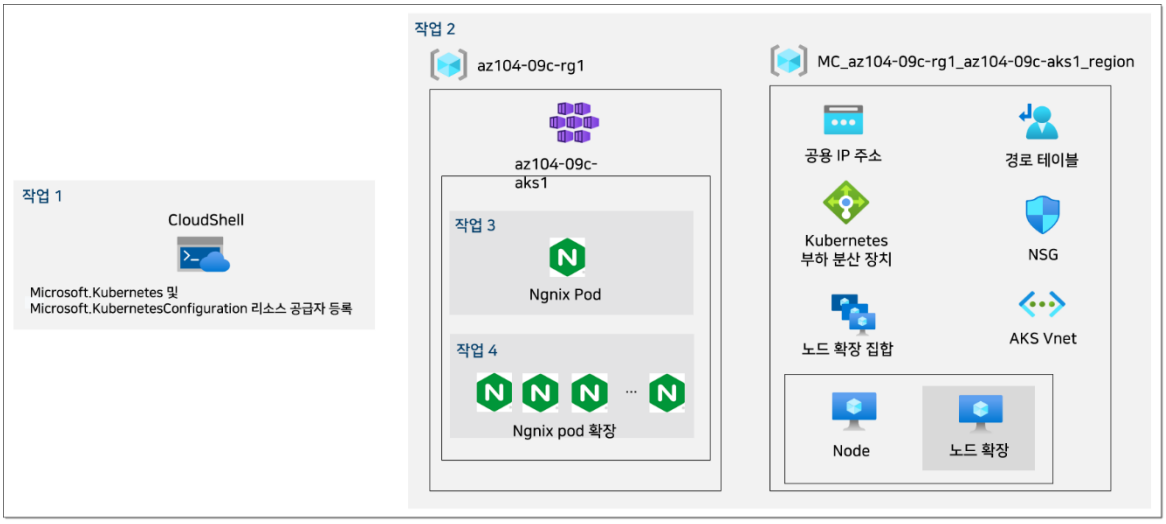
날짜	버전	작성자	변경 내용
2021.11.22	1.0.0	우진환	LAB09C 작성
2022.10.08	1.1.0	우진환	Azure 포털 변경 사항 적용
2023.02.09	1.2.0	우진환	Cloudslice 변경 사항 적용
2023.06.03	1.3.0	우진환	Cloudslice 변경 사항 적용

목차

실습 시나리오

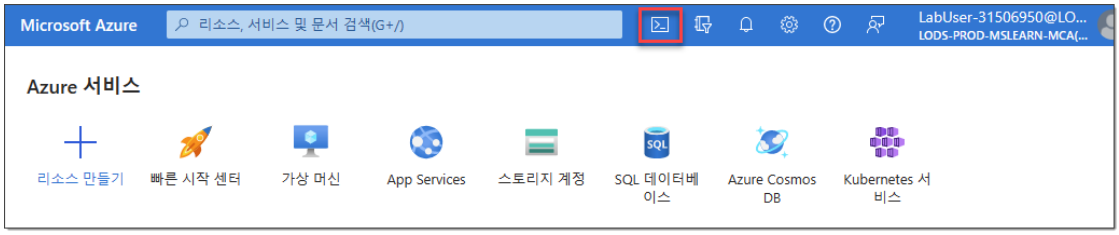
Contoso에는 Azure Container Instances를 사용하여 실행하기에 적합하지 않은 여러 다중 계층 애플리케이션이 있습니다. 컨테이너화된 워크로드로 실행할 수 있는지 여부를 확인하기 위해 Kubernetes를 컨테이너 오케스트레이터로 사용하여 평가하려고 합니다. 관리 오버헤드를 최소화하기 위해 간소화된 배포 환경 및 크기 조정 기능을 포함하여 Azure Kubernetes Service를 테스트하려고 합니다.

아키텍처 다이어그램

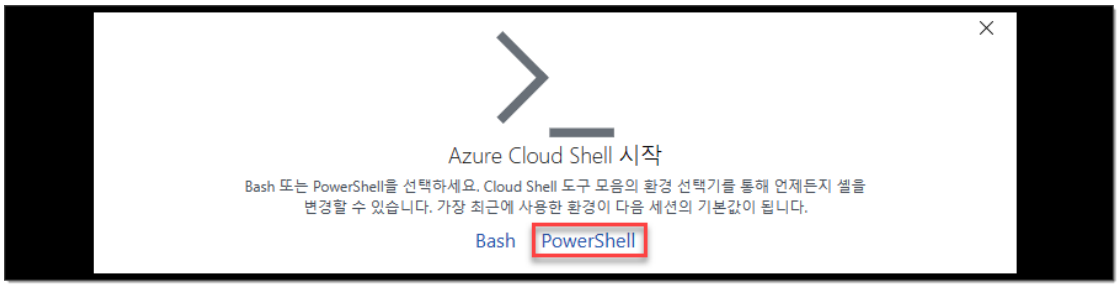


TASK 01. Cloud Shell 준비

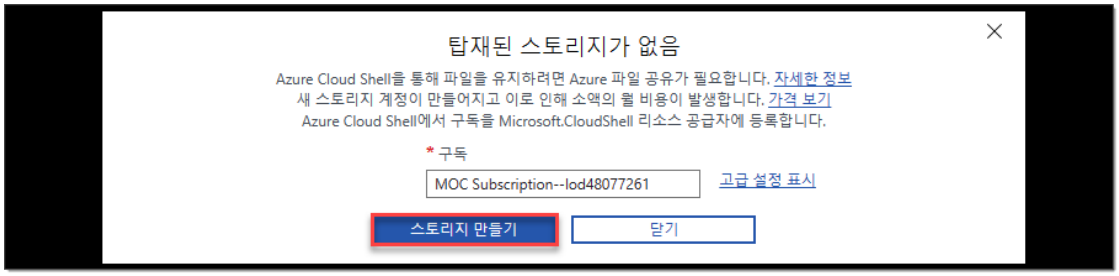
1. Azure 포털의 우측 상단에서 [Cloud Shell] 아이콘을 클릭합니다.



2. [Azure Cloud Shell 시작] 창에서 [PowerShell]을 클릭합니다.



3. [탑재된 스토리지가 없음] 페이지에서 [스토리지 만들기]를 클릭합니다.



TASK 02. `Microsoft.Kubernetes` 및 `Microsoft.KubernetesConfiguration` 리소스 공급자 등록

이 작업에서는 Azure Kubernetes Service 클러스터 배포에 필요한 리소스 공급자를 등록합니다.

1. Azure 포털에서 [Cloud Shell]을 실행합니다. [Cloud Shell]의 PowerShell 세션에서 다음 명령을 실행하여 Kubernetes와 관련된 리소스 공급자를 등록합니다.

```
# Kubernetes 리소스 공급자 등록
Register-AzResourceProvider -ProviderNamespace Microsoft.Kubernetes
Register-AzResourceProvider -ProviderNamespace Microsoft.KubernetesConfiguration
```

```
PowerShell
PS /home/labuser-31506950> # Kubernetes 리소스 공급자 등록
PS /home/labuser-31506950> Register-AzResourceProvider -ProviderNamespace Microsoft.Kubernetes

ProviderNamespace : Microsoft.Kubernetes
RegistrationState : Registering
ResourceTypes     : {connectedClusters, locations, locations/operationStatuses, registeredSubscriptions…}
Locations         : {West Europe, East US, West Central US, South Central US…}


PS /home/labuser-31506950> Register-AzResourceProvider -ProviderNamespace Microsoft.KubernetesConfiguration

ProviderNamespace : Microsoft.KubernetesConfiguration
RegistrationState : Registering
ResourceTypes     : {sourceControlConfigurations, extensions, fluxConfigurations, operations…}
Locations         : {East US, West Europe, West Central US, West US 2…}


PS /home/labuser-31506950>
```

2. [Cloud Shell]에서 다음 명령을 실행하여 리소스 공급자가 등록되었는지 확인합니다. `RegistrationState`에서 "`Registered`"로 표시되어야 합니다.

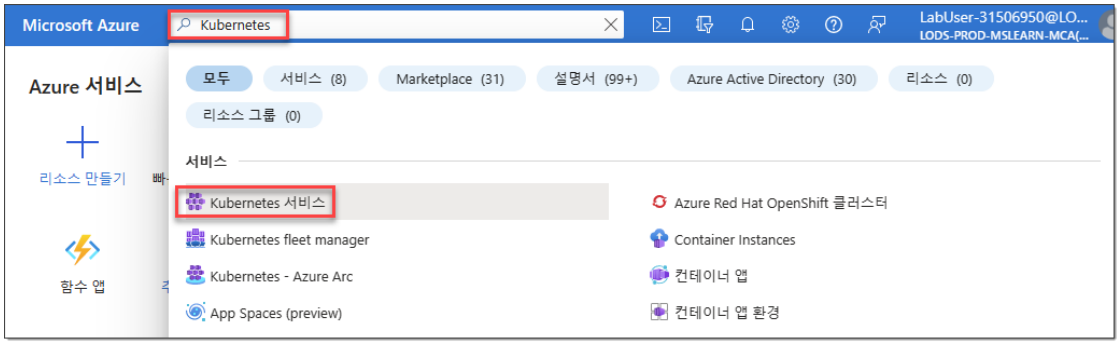
```
# Kubernetes 리소스 공급자 등록 확인
Get-AzResourceProvider -ProviderNamespace Microsoft.Kubernetes
```

```
PowerShell
PS /home/labuser-31506950> # Kubernetes 리소스 공급자 등록 확인
PS /home/labuser-31506950> Get-AzResourceProvider -ProviderNamespace Microsoft.Kubernetes

ProviderNamespace : Microsoft.Kubernetes
RegistrationState : Registered
ResourceTypes     : {connectedClusters}
Locations         : {West Europe, East US, West Central US, South Central US…}

ProviderNamespace : Microsoft.Kubernetes
RegistrationState : Registered
ResourceTypes     : {locations}
Locations         : {}

ProviderNamespace : Microsoft.Kubernetes
RegistrationState : Registered
ResourceTypes     : {locations/operationStatuses}
Locations         : {East US 2 EUAP, West Europe, East US, West Central US…}

ProviderNamespace : Microsoft.Kubernetes
RegistrationState : Registered
ResourceTypes     : {registeredSubscriptions}
Locations         : {}

ProviderNamespace : Microsoft.Kubernetes
RegistrationState : Registered
ResourceTypes     : {Operations}
Locations         : {}
```

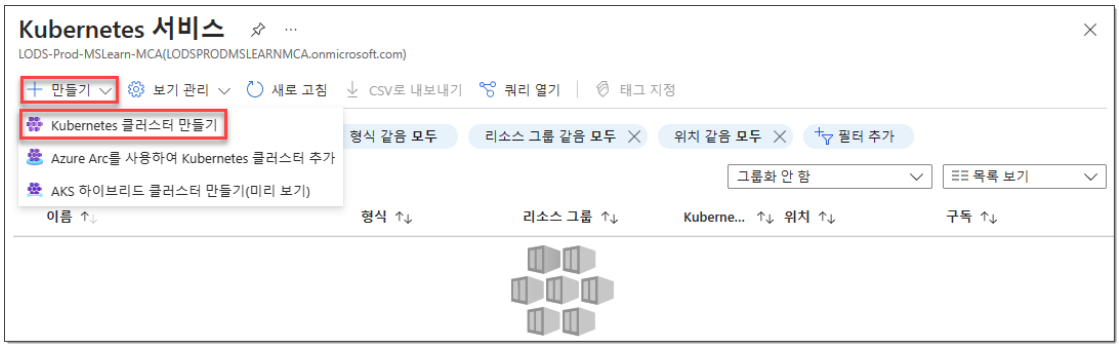
TASK 03. Azure Kubernetes Service 클러스터 배포

이 작업에서는 Azure 포털을 사용하여 Azure Kubernetes Service 클러스터를 배포합니다.

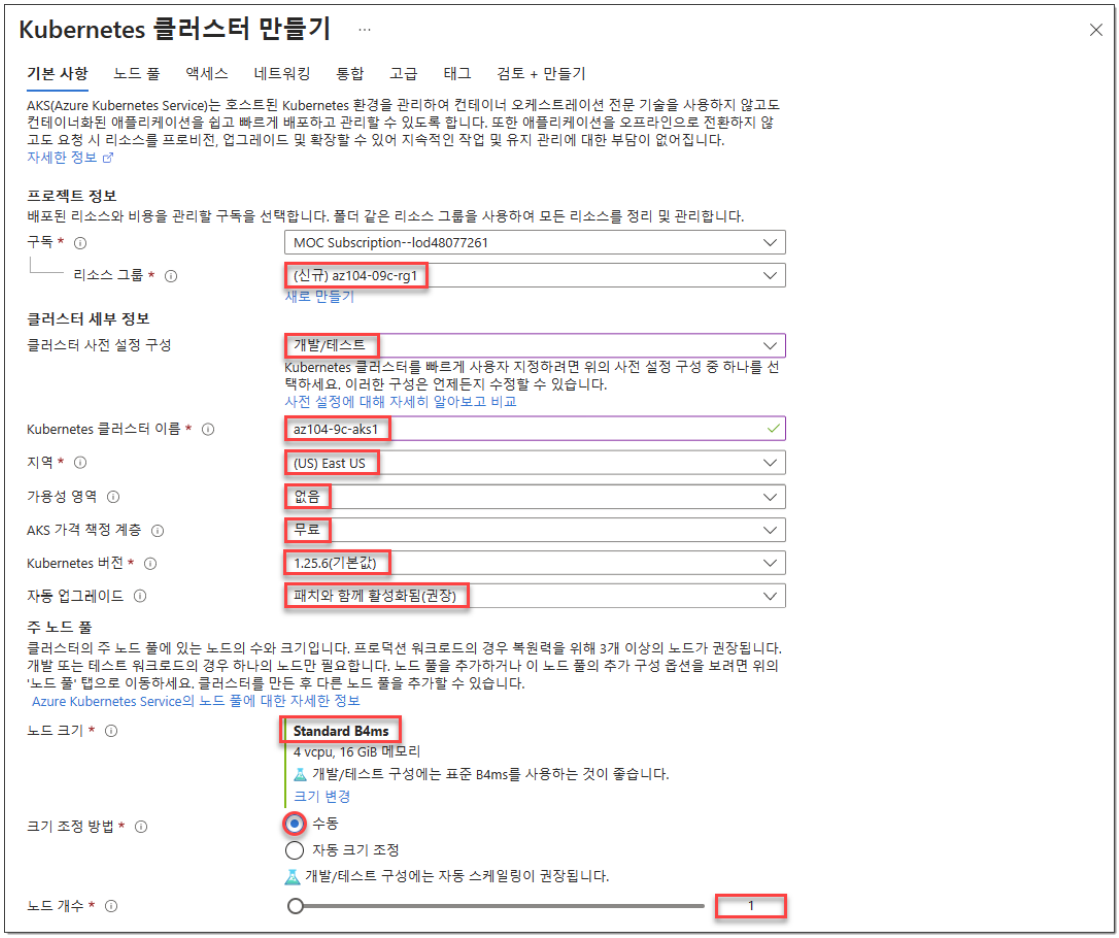
1. Azure 포털의 검색창에서 "`Kubernetes`"를 검색한 후 [Kubernetes 서비스]를 클릭합니다.



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



3. [Kubernetes 클러스터 만들기] 블레이드의 [기본 사항] 탭에서 아래와 같이 구성하고 [다음]을 클릭합니다. 클러스터 이름을 포함한 모든 설정은 CloudSlice의 Azure Policy로 지정되어 있기 때문에 반드시 아래 내용으로 설정을 구성해야 합니다.
   - [프로젝트 정보 - 리소스 그룹]: "새로 만들기"를 클릭한 후 "`az104-09c-rg1`" 이름을 입력합니다.
   - [클러스터 세부 정보 - 클러스터 사전 설정 구성]: 개발/테스트
   - [클러스터 세부 정보 - Kubernetes 클러스터 이름]: az104-9c-aks1
   - [클러스터 세부 정보 - 지역]: (US) East US
   - [클러스터 세부 정보 - 가용성 영역]: 없음
   - [클러스터 세부 정보 - AKS 가격 책정 계층]: 무료
   - [클러스터 세부 정보 - Kubernetes 버전]: 기본값 사용
   - [클러스터 세부 정보 - 자동 업그레이드]: 패치와 함께 활성화됨
   - [주 노드 풀 - 노드 크기]: 기본값을 사용합니다.
   - [주 노드 풀 - 크기 조정 방법]: 수동
   - [주 노드 풀 - 노드 개수]: 1



4. [노드 풀] 탭에서 "가상 노드 사용" 옵션을 선택하지 않고 [다음]을 클릭합니다.

5. [액세스] 탭에서 아래와 같이 구성한 후 [다음]을 클릭합니다.
   - 리소스 ID: 시스템에서 할당한 관리 ID
   - 인증 및 권한 부여: Kubernetes RBAC가 있는 로컬 계정

6. [네트워킹] 탭에서 아래와 같이 구성하고 [다음]을 클릭합니다.
   - 네트워크 구성: kubenet
   - DNS 이름 접두사: 기본값 사용

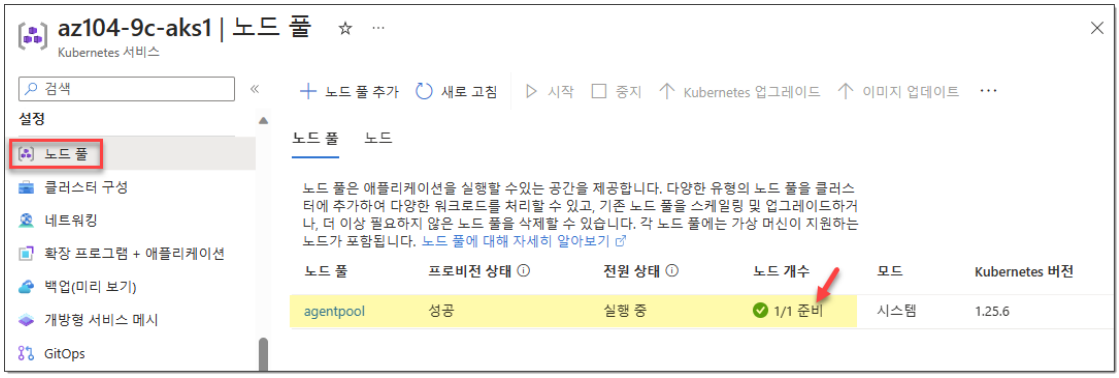
7. [통합] 탭에서 아래와 같이 구성한 후 [다음]을 클릭합니다. 프로덕션 환경에서는 모니터링을 활성화하는 것이 권장됩니다.
   - [클라우드용 Microsoft Defender - 기본 플랜을 무료로 사용]: 선택하지 않습니다.
   - [Azure Container Registry - 컨테이너 레지스트리]: 없음
   - [컨테이너 인사이트 - 컨테이너 로그 사용]: 선택하지 않습니다.
   - [관리되는 Prometheus - Prometheus 메트릭 사용]: 선택하지 않습니다.
   - [Grafana - Grafana 사용]: 선택하지 않습니다.
   - [경고 중 - 권장 경고 규칙 사용]: 선택하지 않습니다.
   - [Azure Policy - Azure Policy]: 사용 안 함

8. [고급] 탭에서 기본 설정을 유지하고 [검토 + 만들기]를 클릭합니다. [검토 + 만들기] 탭에서 [만들기]를 클릭합니다.

TASK 04. Azure Kubernetes Service 클러스터에 pod 배포

이 작업에서는 Azure Kubernetes Service 클러스터에 `pod`를 배포합니다.

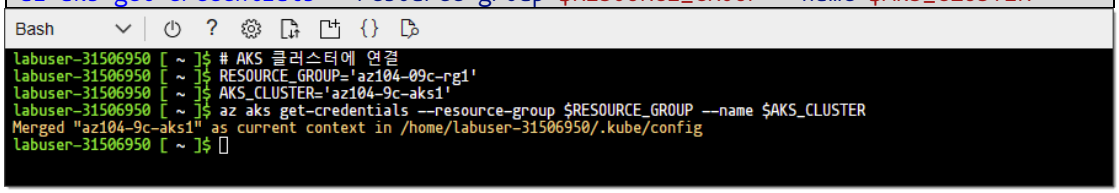
1. [`az104-9c-aks1` Kubernetes 서비스] 블레이드로 이동한 후 [설정 - 노드 풀]로 이동합니다. 하나의 노드로 구성된 단일 풀의 클러스터가 표시되는 것을 확인합니다.



2. Azure 포털에서 [Cloud Shell]을 열고 Bash 세션을 실행합니다.

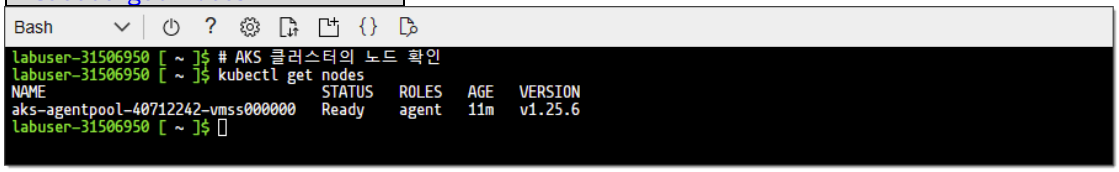
3. [Cloud Shell]의 Bash 세션에서 다음 명령을 실행하여 AKS 클러스터에 액세스하기 위한 자격 증명을 가져옵니다.

```
# AKS 클러스터에 연결
RESOURCE_GROUP='az104-09c-rg1'
AKS_CLUSTER='az104-9c-aks1'
az aks get-credentials --resource-group $RESOURCE_GROUP --name $AKS_CLUSTER
```



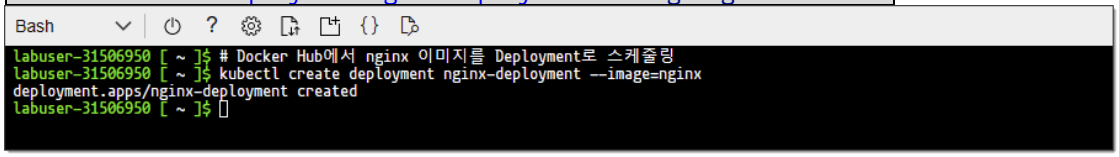
4. [Cloud Shell]에서 다음 명령을 실행하여 AKS 클러스터에 배포된 노드를 확인합니다. 출력 결과에서 노드의 상태가 `Ready`로 표시되는지 확인합니다.

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



5. [Cloud Shell]에서 다음 명령을 실행하여 Docker Hub에서 `nginx` 이미지를 배포합니다.

```
# Docker Hub 에서 nginx 이미지를 Deployment 로 스케줄링
kubectl create deployment nginx-deployment --image=nginx
```



6. [Cloud Shell]에서 다음 명령을 실행하여 Kubernetes `pod`가 생성되었는지 확인합니다.

```
# Pod 생성 확인
```

```
kubectl get pods
```

```
Bash
labuser-31506950 [ ~ ]$ # Pod 생성 확인
labuser-31506950 [ ~ ]$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-5fbdf85c67-8xj45   1/1     Running   0          41s
labuser-31506950 [ ~ ]$
```

7. [Cloud Shell]에서 다음 명령을 실행하여 `Deployment` 상태를 확인합니다.

```
# Deployment 상태 확인
kubectl get deployment
```

```
Bash
labuser-31506950 [ ~ ]$ # Deployment 상태 확인
labuser-31506950 [ ~ ]$ kubectl get deployment
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   1/1     1            1           64s
labuser-31506950 [ ~ ]$
```

8. [Cloud Shell]에서 다음 명령을 실행하여 인터넷에서 `pod`에 액세스할 수 있도록 구성합니다.

```
# Pod를 인터넷에서 액세스할 수 있도록 설정
kubectl expose deployment nginx-deployment --port=80 --type=LoadBalancer
```

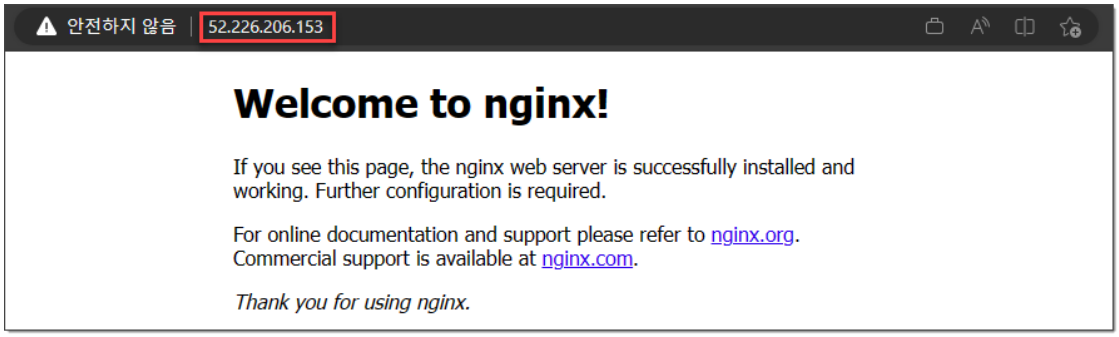
```
Bash
labuser-31506950 [ ~ ]$ # Pod를 인터넷에서 액세스할 수 있도록 설정
labuser-31506950 [ ~ ]$ kubectl expose deployment nginx-deployment --port=80 --type=LoadBalancer
service/nginx-deployment exposed
labuser-31506950 [ ~ ]$
```

9. [Cloud Shell]에서 다음 명령을 실행하여 공용 IP 주소가 프로비저닝되었는지 확인합니다. `ngnix-deployment` 항목의 `EXTERNALIP` 열에 공용 IP 주소가 표시되는지 확인합니다. 공용 IP 주소가 표시되지 않으면 조금 더 기다린 후 다시 명령을 실행하여 공용 IP 주소가 표시되는지 확인합니다. 표시되는 공용 IP 주소를 복사합니다.

```
# Service 개체 확인
kubectl get service
```

```
Bash
labuser-31506950 [ ~ ]$ # Service 개체 확인
labuser-31506950 [ ~ ]$ kubectl get service
NAME               TYPE           CLUSTER-IP    EXTERNAL-IP      PORT(S)        AGE
kubernetes         ClusterIP      10.0.0.1      <none>           443/TCP        15m
nginx-deployment   LoadBalancer   10.0.190.0    52.226.206.153   80:31838/TCP   27s
labuser-31506950 [ ~ ]$
```

10. 브라우저에서 새 탭을 열고 위에서 복사한 공용 IP 주소에 액세스합니다. 아래와 같이 "`Welcome to nginx!`" 페이지가 표시되는 것을 확인합니다.

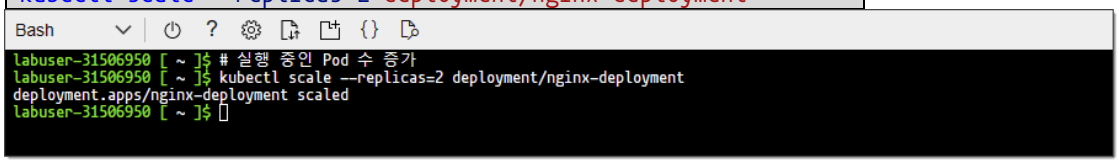


TASK 05. Azure Kubernetes Service 클러스터에서 컨테이너화된 워크로드 확장

이 작업에서는 `pod`의 수를 수평적으로 증가시킨(horizontally scale) 다음 클러스터 노드의 수를 증가시킵니다.

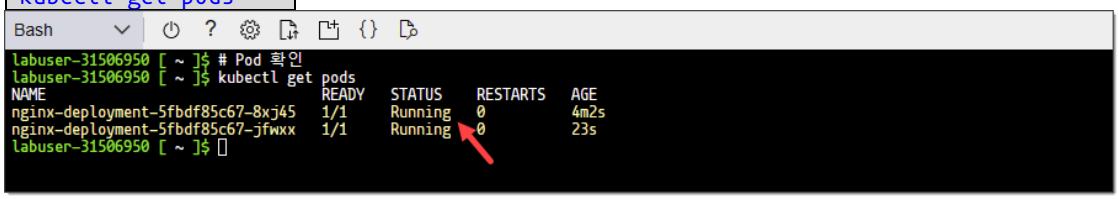
1. [Cloud Shell]을 열고 다음 명령을 실행하여 `pod`의 수를 2개로 증가시켜 `deployment`를 확장합니다.

```
# 실행 중인 Pod 수 증가
kubectl scale --replicas=2 deployment/nginx-deployment
```



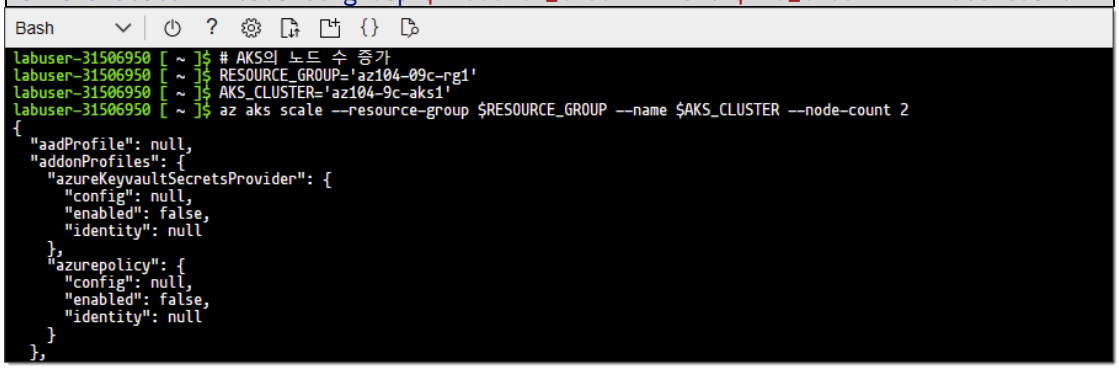
2. [Cloud Shell]에서 다음 명령을 실행하여 `deployment`가 확장의 결과를 확인합니다. `NGINX`를 실행하는 `pod`수가 2개로 증가한 것을 확인합니다.

```
# Pod 확인
kubectl get pods
```



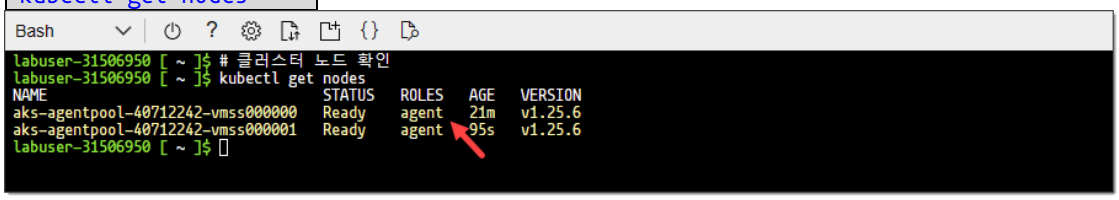
3. [Cloud Shell]에서 다음 명령을 실행하여 노드의 수를 2대로 증가시켜 클러스터를 scale out 합니다. 추가 노드가 프로비저닝될 때까지 기다립니다. 작업은 대략 3분 정도가 소요됩니다.

```
# AKS 의 노드 수 증가
RESOURCE_GROUP='az104-09c-rg1'
AKS_CLUSTER='az104-9c-aks1'
az aks scale --resource-group $RESOURCE_GROUP --name $AKS_CLUSTER --node-count 2
```



4. [Cloud Shell]에서 다음 명령을 실행하여 클러스터 확장의 결과를 확인합니다. 노드의 수가 2대로 증가된 것을 확인할 수 있습니다.

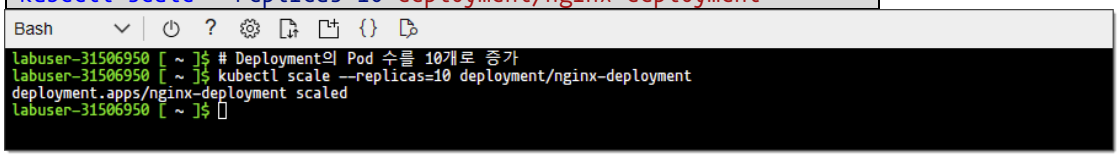
```
# 클러스터 노드 확인
kubectl get nodes
```



5. [Cloud Shell]에서 다음 명령을 실행하여 `deployment`를 다시 확장합니다.

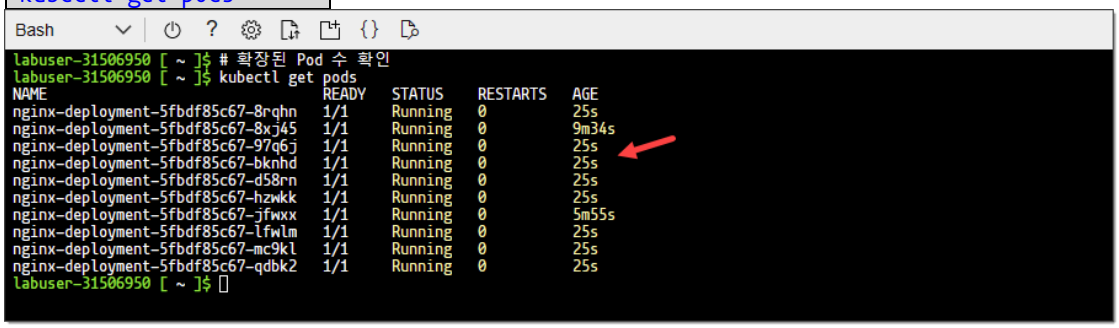
```
# Deployment 의 Pod 수를 10 개로 증가
```

```
kubectl scale --replicas=10 deployment/nginx-deployment
```



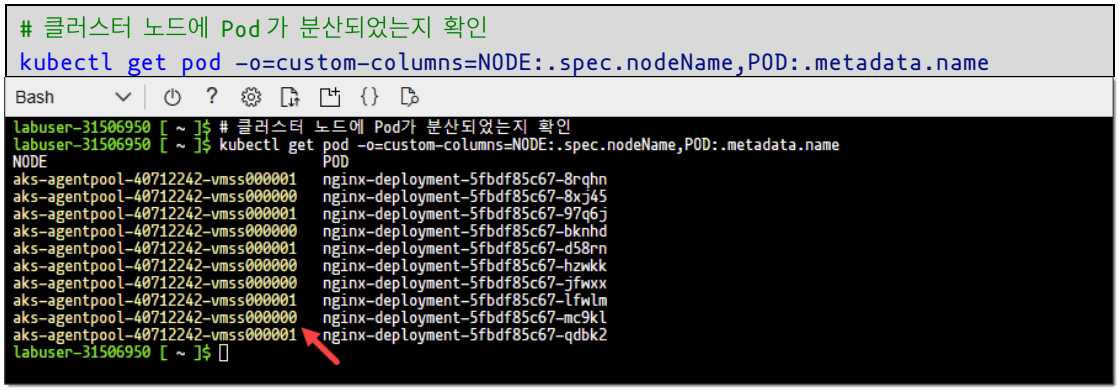
6. [Cloud Shell]에서 다음 명령을 실행하여 `deployment` 확장 결과를 확인합니다. `pod`의 수가 10개로 증가된 것을 확인할 수 있습니다.

```
# 확장된 Pod 수 확인
kubectl get pods
```



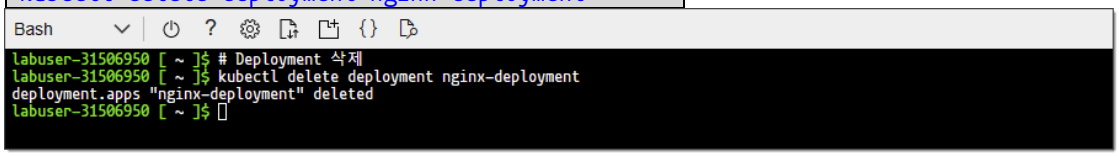
7. [Cloud Shell]에서 다음 명령을 실행하여 클러스터 노드간 `pod`가 어떻게 배포되어 있는지 확인합니다. `pod`가 노드간에 분산되어 있는 것을 확인할 수 있습니다.

```
# 클러스터 노드에 Pod가 분산되었는지 확인
kubectl get pod -o=custom-columns=NODE:.spec.nodeName,POD:.metadata.name
```



8. [Cloud Shell]에서 다음 명령을 실행하여 배포를 삭제합니다.

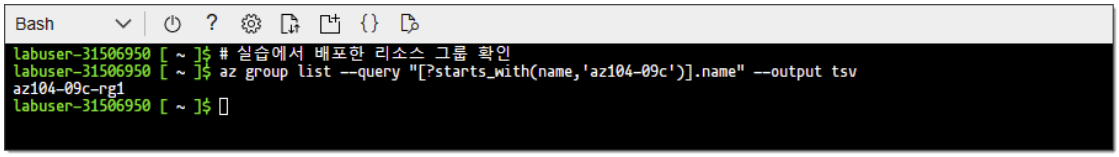
```
# Deployment 삭제
kubectl delete deployment nginx-deployment
```



TASK 06. 리소스 정리

1. [Cloud Shell]의 Bash 세션에서 다음 명령을 실행하여 이 실습에서 만든 리소스 그룹을 확인합니다.

```
# 실습에서 배포한 리소스 그룹 확인
az group list --query "[?starts_with(name,'az104-09c')].name" --output tsv
```



2. [Cloud Shell]에서 다음 명령을 실행하여 실습에서 만들었던 모든 리소스 그룹을 삭제합니다. 이 명령은 비동기식으로 실행되기 때문에 동일한 Bash 세션에서 다른 Azure CLI 명령을 실행할 수 있습니다. 하지만 리소스 그룹이 실제 삭제될 때까지는 몇 분 정도가 소요됩니다.

```
# 실습에서 사용한 리소스 그룹 삭제
az group list --query "[?starts_with(name,'az104-09c')].[name]" --output tsv | \
  xargs -L1 bash -c 'az group delete --name $0 --no-wait --yes'
```