

한국 마이크로소프트

# Microsoft Technical Trainer

Enterprise Skills Initiative

**AZ-104. LAB09A**

**웹 앱 구현**

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 허여되지 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
2021.11.22	1.0.0	우진환	LAB09A 작성
2022.10.08	1.1.0	우진환	Azure 포털 변경 사항 적용
2023.02.08	1.2.0	우진환	Cloudslice 변경 사항 적용
2023.06.03	1.3.0	우진환	Cloudslice 변경 사항 적용

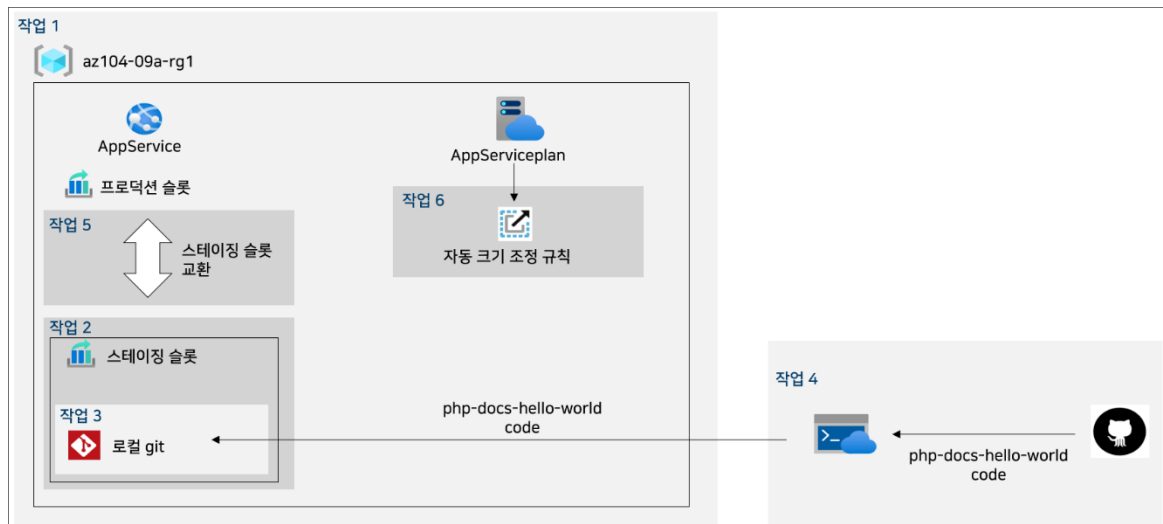
## 목차

실습 시나리오 .....	5
아키텍처 다이어그램 .....	5
TASK 1. CLOUD SHELL 준비 .....	5
TASK 02. AZURE 웹 앱 만들기 .....	6
TASK 03. 스테이징 배포 슬롯 만들기 .....	8
TASK 04. 웹 앱 배포 설정 구성 .....	9
TASK 05. 스테이징 배포 슬롯에 코드 배포 .....	10
TASK 06. 스테이징 슬롯 교환(SWAP) .....	12
TASK 07. AZURE 웹 앱의 자동 크기 조정(AUTOSCALING) 구성 및 테스트 .....	13
TASK 08. 리소스 정리 .....	17

## 실습 시나리오

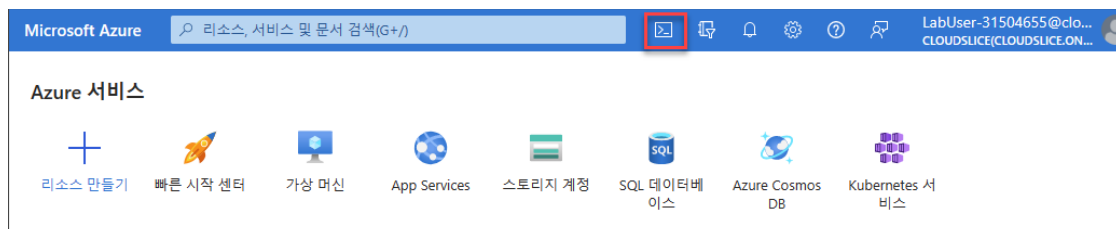
현재 회사의 온-프레미스 데이터 센터에서 호스팅되는 Contoso의 웹 사이트를 호스팅하기 위해 Azure Web Apps의 사용을 평가해야 합니다. 웹 사이트는 PHP 런타임 스택을 사용하여 Windows 서버에서 실행되고 있습니다. 또한, Azure 웹 앱 배포 슬롯을 활용하여 DevOps 사례를 구현할 수 있는 방법을 결정해야 합니다.

## 아키텍처 다이어그램

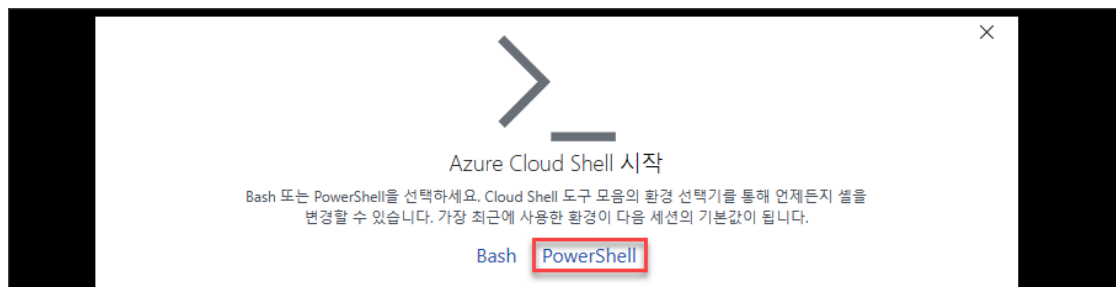


## TASK 1. Cloud Shell 준비

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



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



3. [탐재된 스토리지가 없음] 페이지에서 "고급 설정 표시" 링크를 클릭합니다.



4. [탐재된 스토리지가 없음] 페이지에서 아래와 같이 구성하고 [스토리지 만들기]를 클릭합니다.

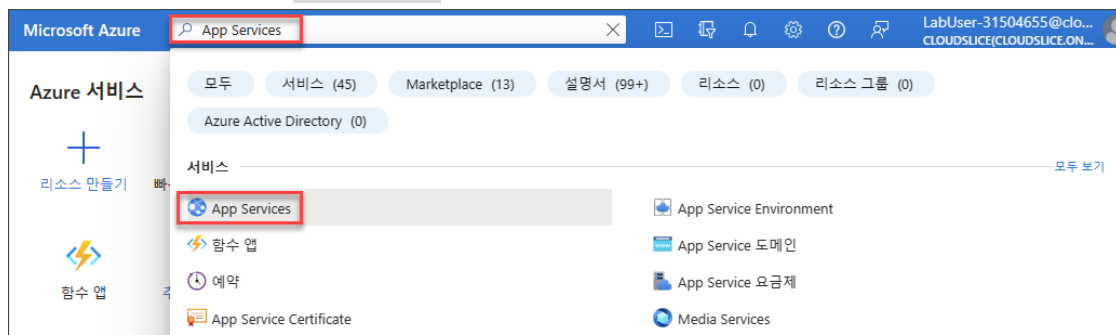
- Cloud Shell 지역: 미국 동부
- 리소스 그룹: "기존 항목 사용"을 선택한 후 "az104-09a-rg1-lod<xxxxxxxx>" 리소스 그룹을 선택합니다.
- 스토리지 계정: "새로 만들기"를 선택한 후 "cloudshell<xxxxxxxx>"을 입력합니다. <xxxxxxxx>은 사용중인 계정에 포함되어 있는 숫자입니다.
- 파일 공유: "새로 만들기"를 선택한 후 "shellstorage"를 입력합니다.



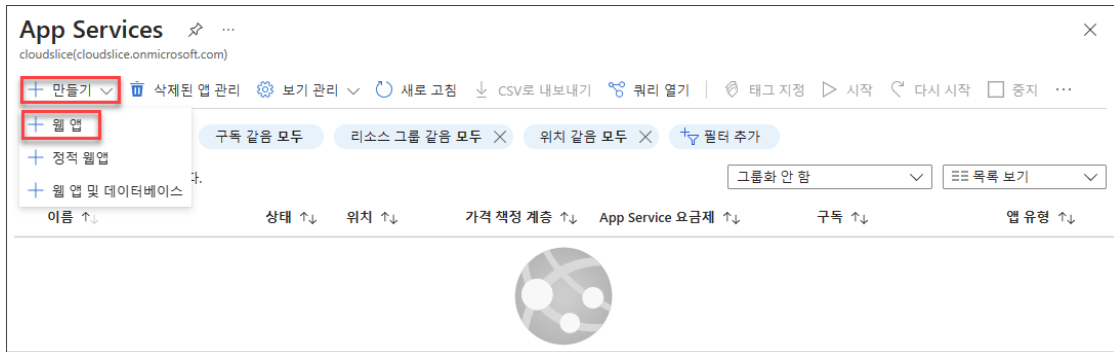
## TASK 02. Azure 웹 앱 만들기

이 작업에서는 Azure 웹 앱을 만듭니다.

1. Azure 포털의 검색창에서 "App Service"를 검색한 후 [App Services]를 클릭합니다.



2. [App Services] 블레이드의 메뉴에서 [만들기 - 웹 앱]을 클릭합니다.



3. [웹앱 만들기] 블레이드의 [기본] 탭에서 아래와 같이 구성한후 [검토 + 만들기]를 클릭합니다.

- [프로젝트 세부 정보 - 리소스 그룹]: **az104-09a-rg1-lod<xxxxxxxx>**
- [인스턴스 정보 - 이름]: **labuser<xxxxxxxx>**
- [인스턴스 정보 - 게시]: 코드
- [인스턴스 정보 - 런타임 스택]: PHP 8.2
- [인스턴스 정보 - 운영 체제]: Linux
- [인스턴스 정보 - 지역]: East US
- [가격 책정 플랜]: 기본값을 사용합니다.
- [영역 중복 - 영역 중복]: 사용 안 함

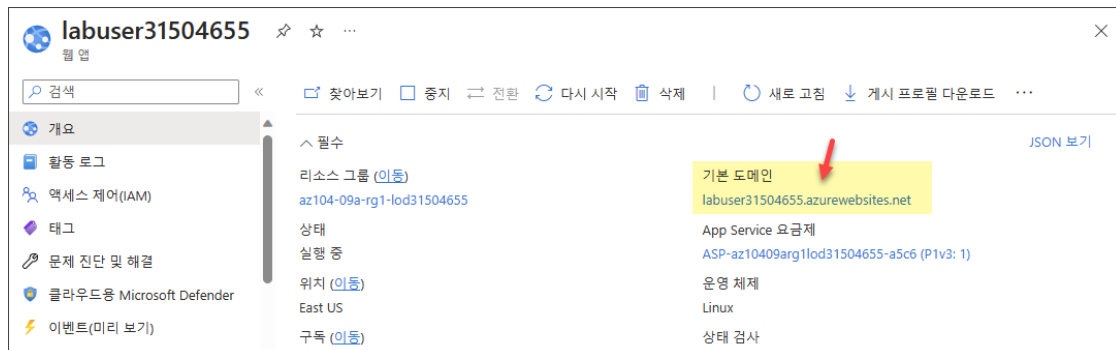
4. [검토 + 만들기] 탭에서 [만들기]를 클릭합니다. 리소스 배포가 완료되면 [리소스로 이동]을 클릭합니다.



### TASK 03. 스테이징 배포 슬롯 만들기

이 작업에서는 스테이징 배포 슬롯을 만듭니다.

1. [App service] 블레이드의 [개요]에서 표시되는 URL을 클릭하여 새 브라우저 탭에서 기본 웹 페이지가 표시되는 것을 확인합니다.



2. 기본 웹 페이지를 닫고 [App Service] 블레이드의 [배포 - 배포 슬롯]으로 이동한 후 메뉴에서 [슬롯 추가]를 클릭합니다. 현재 시점에서 이 웹 앱은 "프로덕션" 배포 슬롯만 있습니다.

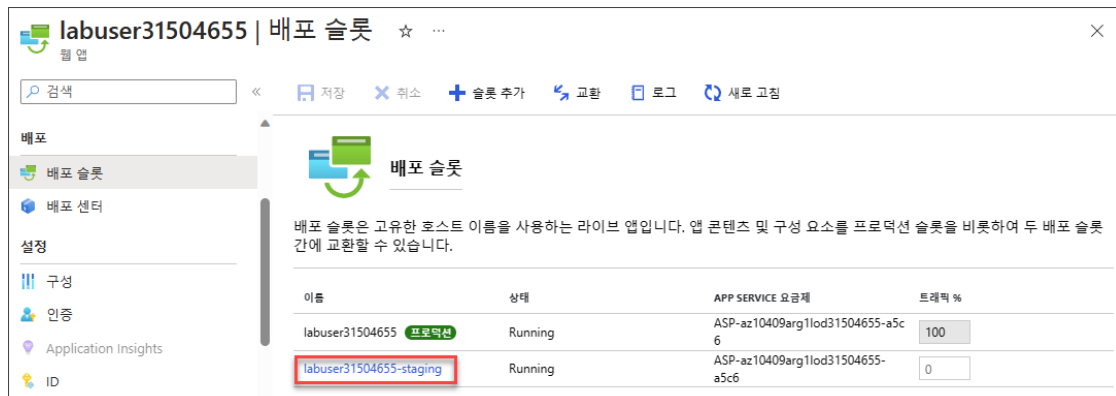


3. [슬롯 추가] 창에서 다음과 같이 구성하고 [추가]를 클릭합니다.

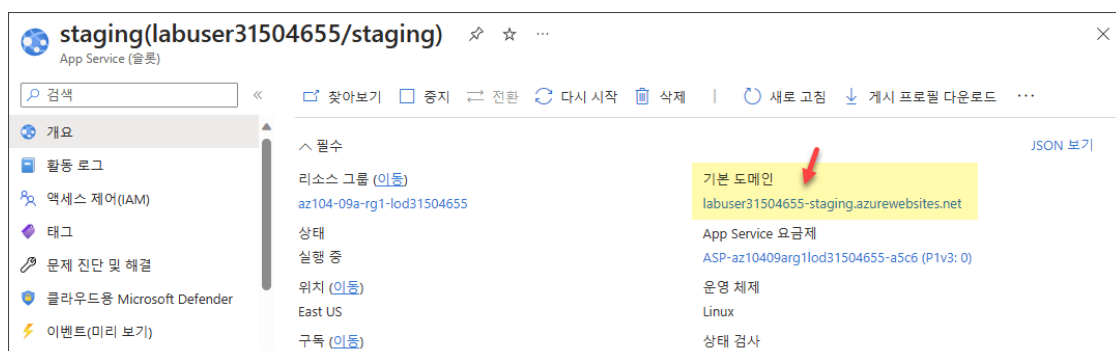
- 이름: staging
- 다음의 설정 복제: 설정을 복제하지 않음



4. [App Service] 블레이드의 [배포 - 배포 슬롯]에서 새로 만들어진 스테이징 슬롯이 표시되는 것을 확인하고 이를 클릭합니다.



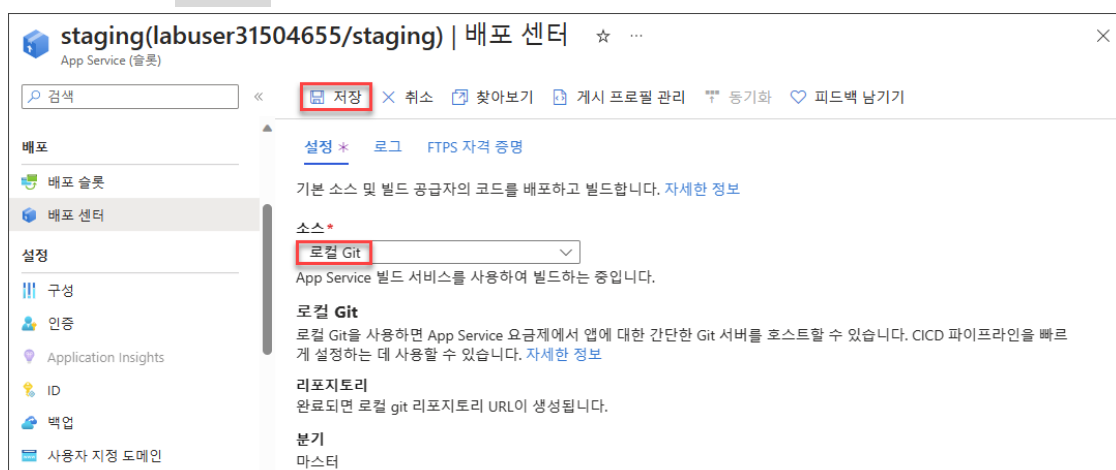
5. [App Service (슬롯)] 블레이드가 표시됩니다. 이 스테이징 슬롯을 검토하고 프로덕션 슬롯과 다른 URL을 가지고 있는 것을 확인합니다.



## TASK 04. 웹 앱 배포 설정 구성

이 작업에서는 웹 앱 배포 설정을 구성합니다.

1. [staging App Service (슬롯)] 스테이징 배포 슬롯 블레이드에서 [배포 - 배포 센터]로 이동합니다. [설정] 탭에서 소스를 "로컬 Git"으로 선택한 후 메뉴에서 [저장]을 클릭합니다.

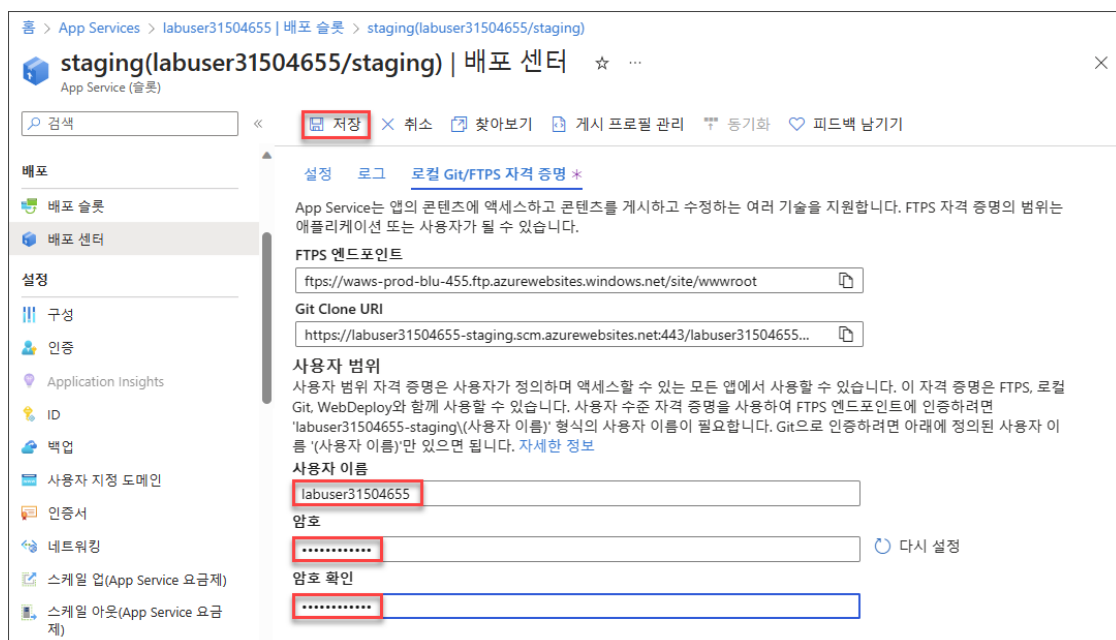


2. [배포 센터]에서 표시되는 Git Clone URI 값을 메모장에 복사합니다. 이 Git Clone URL은 다음 작업에서 필요합니다.



3. [배포 센터]의 [로컬 Git/FTP/FTP 자격 증명] 탭으로 이동한 후 "사용자 범위" 섹션에서 아래와 같이 구성하고 [저장]을 클릭합니다.

- 사용자 이름: Labuser<xxxxxxxx> 이름을 입력합니다. 반드시 **q** 문자를 포함하지 않아야 합니다.
- 암호: 복잡성을 만족하는 암호(Pa55w.rd1234)를 입력합니다. 암호는 8글자 이상이어야 하며 대문자, 소문자, 숫자, 기호를 포함해야 합니다.



## TASK 05. 스테이징 배포 슬롯에 코드 배포

이 작업에서는 스테이징 배포 슬롯에 코드를 배포합니다.

1. Azure 포털에서 [Cloud Shell]을 실행합니다. PowerShell 세션에서 다음 명령을 실행하여 웹 앱 코드를 포함하고 있는 원격 리포지토리를 클론합니다.

```
# 원격 리포지토리 클론
git clone https://github.com/Azure-Samples/php-docs-hello-world
```

```
PowerShell
PS /home/labuser-31504655> # 원격 리포지토리 클론
PS /home/labuser-31504655> git clone https://github.com/Azure-Samples/php-docs-hello-world
Cloning into 'php-docs-hello-world'...
remote: Enumerating objects: 26, done.
remote: Total 26 (delta 0), reused 0 (delta 0), pack-reused 26
Receiving objects: 100% (26/26), 5.64 KiB | 5.64 MiB/s, done.
Resolving deltas: 100% (6/6), done.
PS /home/labuser-31504655>
```

2. [Cloud Shell]에서 다음 명령을 실행하여 샘플 웹 앱 코드를 포함하고 있는 로컬 리포지토리의 새로 만들어진 복제본으로 이동합니다.

```
# 로컬 리포지토리 복제본으로 이동
Set-Location -Path $HOME/php-docs-hello-world/

PowerShell
PS /home/labuser-31504655> # 로컬 리포지토리 복제본으로 이동
PS /home/labuser-31504655> Set-Location -Path $HOME/php-docs-hello-world/
PS /home/labuser-31504655/php-docs-hello-world>
```

3. [Cloud Shell] 창에서 다음 명령을 실행하여 원격 git을 추가합니다.
  - [deployment\_user\_name]에 배포 자격 증명 사용자 이름을 입력하고 [git\_clone\_url]은 앞서 메모장에 복사했던 Git Clone URL을 붙여 넣습니다.
  - git remote add 뒤의 이름이 배포 자격 증명 사용자 이름과 일치할 필요는 없지만 고유해야 합니다.

```
# 원격 Git 추가
git remote add [deployment_user_name] [git_clone_url]

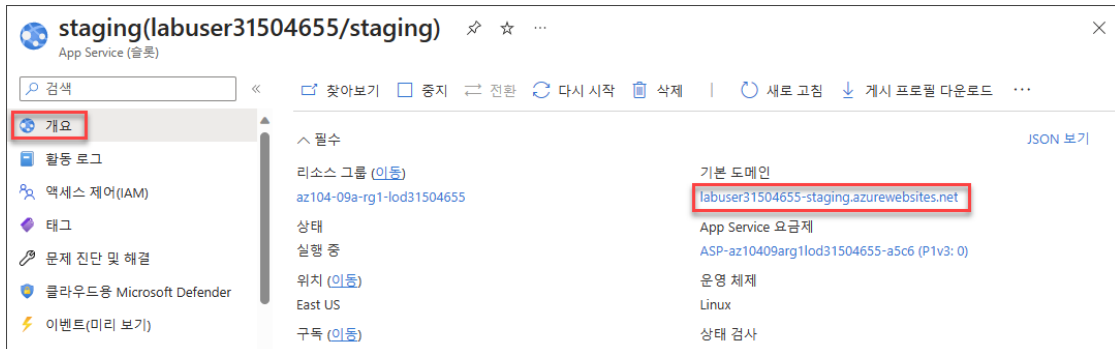
PowerShell
PS /home/labuser-31504655/php-docs-hello-world> # 원격 Git 추가
PS /home/labuser-31504655/php-docs-hello-world> git remote add labuser31504655 https://labuser31504655-staging.scm.azurewebsites.net:443/labuser31504655.g
it
PS /home/labuser-31504655/php-docs-hello-world>
```

4. [Cloud Shell] 창에서 다음 명령을 실행하여 로컬 리포지토리의 샘플 웹 앱 코드를 Azure 웹 앱 스테이징 배포 슬롯에 푸시합니다. [deployment\_user\_name]은 배포 자격 증명의 사용자 이름을 사용합니다.
  - 자격 증명 입력이 표시되면 스테이징 배포 슬롯에서 설정했던 사용자 이름과 암호를 입력합니다.

```
# 로컬 리포지토리의 샘플 코드를 Azure 스테이징 슬롯으로 푸시
git push [deployment_user_name] master

PowerShell
PS /home/labuser-31504655/php-docs-hello-world> # 로컬 리포지토리의 샘플 코드를 Azure 스테이징 슬롯으로 푸시
PS /home/labuser-31504655/php-docs-hello-world> git push labuser31504655 master
Username for 'https://labuser31504655-staging.scm.azurewebsites.net:443': labuser31504655
Password for 'https://labuser31504655@labuser31504655-staging.scm.azurewebsites.net:443':
Enumerating objects: 26, done.
Counting objects: 100% (26/26), done.
Delta compression using up to 2 threads
Compressing objects: 100% (17/17), done.
Writing objects: 100% (26/26), 5.64 KiB | 5.64 MiB/s, done.
Total 26 (delta 6), reused 26 (delta 6), pack-reused 0
remote: Deploy Async
remote: Updating branch 'master'.
```

5. [Cloud Shell]을 닫고 스테이징 슬롯 블레이드로 이동합니다. [staging App Service (슬롯)] 블레이드의 [개요]에서 표시되는 URL을 클릭합니다.



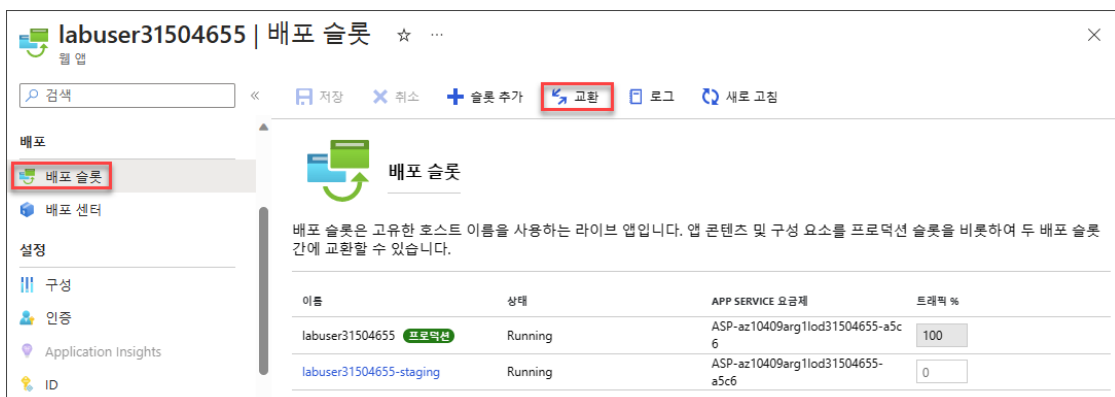
6. 다음과 같이 "Hello World!" 페이지가 표시되는 것을 확인합니다.



## TASK 06. 스테이징 슬롯 교환(Swap)

이 작업에서는 스테이징 슬롯을 프로덕션 슬롯과 교환합니다.

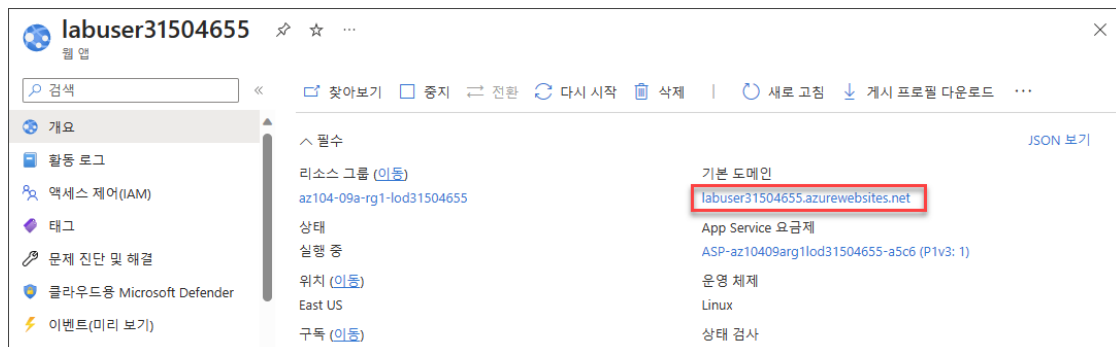
1. Azure 포털에서 웹 앱의 프로덕션 슬롯으로 이동합니다. [App Service] 블레이드의 [배포 - 배포 슬롯]으로 이동한 후 메뉴에서 [교환]을 클릭합니다.



2. [교환] 창에서 기본 설정을 검토하고 [교환]을 클릭합니다. 교환 작업이 완료되면 [닫기]를 클릭합니다.



3. [App Service] 블레이드의 [개요]에서 표시되는 URL을 클릭합니다.



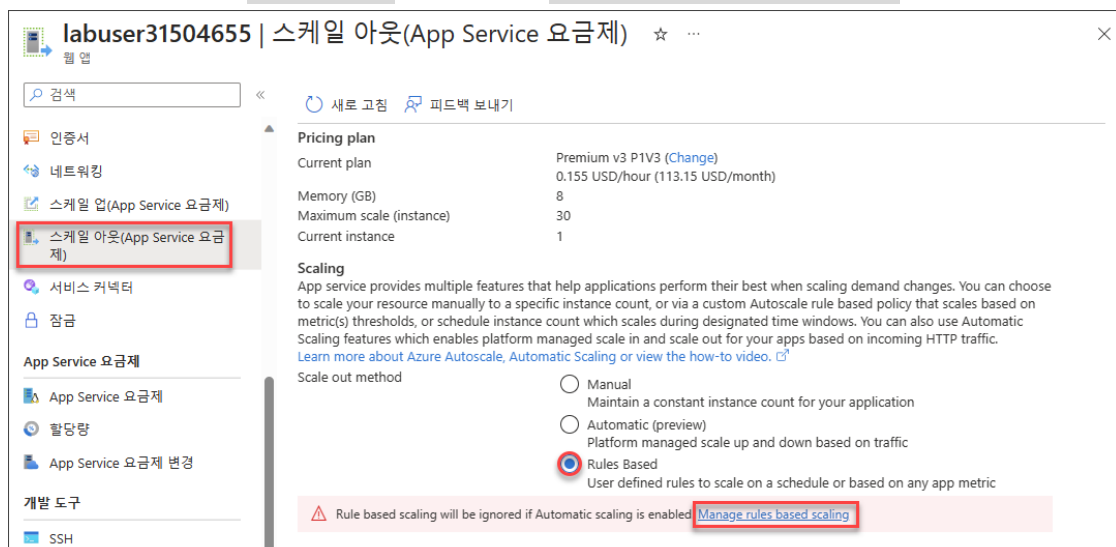
4. 새로 열린 탭에서 스테이징 슬롯에 배포했던 웹 사이트의 "Hello World!" 페이지가 표시되는 것을 확인합니다.



## TASK 07. Azure 웹 앱의 자동 크기 조정(autoscaling) 구성 및 테스트

이 작업에서는 Azure 웹 앱의 자동 크기 조정을 구성하고 테스트합니다.

1. 웹 앱의 프로덕션 슬롯 블레이드로 이동합니다. [App Service] 블레이드의 [설정 - 스케일 아웃(App Service 요금제)]로 이동한 후 "Rules Based"를 선택하고 "Manage rules based scaling" 링크를 클릭합니다.



2. [자동 크기 조정 설정] 블레이드의 [구성] 탭에서 "사용자 지정 자동 크기 조정"을 선택합니다. "기본값" 타일에서 크기 조정 모드를 "메트릭 기준 크기 조정"으로 선택하고 "규칙 추가" 링크를 클릭합니다.

**자동 크기 조정 설정** ...

ASP-az10409arg1lod31504655-a5c6 (App Service 요금제)

저장 취소 새로 고침 로그 피드백

**구성** 실행 기록 JSON 알리기 진단 설정

자동 스케일링은 수요가 변경될 때 애플리케이션이 최상의 성능을 유지할 수 있도록 하기 위해 기본 제공되는 기능입니다. 리소스 크기를 특정 인스턴스 수(으)로 수동으로 조정하거나, 메트릭 임계값에 따라 크기 조정되는 사용자 지정 자동 스케일링 정책을 통해 조정하거나, 지정된 기간 동안 스케일링되는 인스턴스 수를(를) 예약도록 선택할 수 있습니다. 자동 스케일링을 사용하면 요청에 따라 물(를) 추가하고 인스턴스 수를 제거하여 리소스의 성능을 향상하고 비용 효율적으로 유지할 수 있습니다. [Azure 자동 스케일링에 대해 자세히 알아보거나 방법 동영상](#)을 확인하세요.

리소스 크기를 조정하는 방법 선택

수동 크기 조정  
고정 인스턴스 수 유지

사용자 지정 자동 크기 조정  
모든 메트릭을 기반으로 일정에 따라 크기 조정

사용자 지정 자동 크기 조정

자동 크기 조정 설정 이름 \* ASP-az10409arg1lod31504655-a5c6-자동 크기 조정-142

리소스 그룹 az104-09a-rg1-lod31504655

인스턴스 수 1

**기본값 \*** 자동 생성된 기본 스케일링 조건

삭제 경고

크기 조정 모드 ☒ 메트릭 기준 크기 조정 ☐ 특정 인스턴스 수(으)로 확장

규칙

인스턴스 제한

일정

이 크기 조정 조건은 일치하는 다른 크기 조정 조건이 없을 때 실행됩니다.

+ 크기 조건 추가

3. [크기 조정 규칙] 창에서 다음과 같이 구성한 후 [추가]를 클릭합니다.

- 메트릭 원본: 현재 리소스
- 메트릭 네임스페이스: 표준 메트릭
- 메트릭 이름: CPU Percentage
- 연산자: 보다 큼
- 크기 조정 작업을 트리거하는 메트릭 임계값: 10
- 기간(분): 1
- 시간 조직 통계: 최대값
- 시간 집계: 최대값
- 작업: 다음을 기준으로 개수 늘이기
- 휴지 기간(분): 5
- 인스턴스 수: 1

**크기 조정 규칙**

메트릭 원본  
현재 리소스(ASP-az10409arg1lod31504655-a5c6)

리소스 종류  
App Service 요금제

리소스  
ASP-az10409arg1lod31504655-a5c6

Criteria  
메트릭 네임스페이스 \*  
표준 메트릭

메트릭 이름  
CPU Percentage

1분 시간 조직

자원 이름  
Instance

연산자  
=

크기 값  
모든 값

가운데에 여러 값을 선택하는 경우 자동 크기 조정이 각 값의 메트릭을 개별적으로 평가하지 않고, 선택한 값의 메트릭을 집계합니다.

시간(분) \* ①  
1

시간 단위(분) ①  
1

시간 조직 룰 \* ①  
최대값

시간 집계 \* ①  
최대값

Action  
작업 \*  
다음과 기준으로 개수 늘이기

휴지 시간(분) \* ①  
5

인스턴스 수 \*  
1

4. [App Service | 스케일 아웃] 페이지의 "인스턴스 제한"을 다음과 같이 구성하고 [저장]을 클릭합니다.

최소값	최대값	기본값
1	2	1

**자동 크기 조정 설정**

ASP-az10409arg1lod31504655-a5c6 (App Service 요금제)

구성 실행 기록 JSON 알리기 진단 설정

자동 스케일링은 수요가 변경될 때 애플리케이션이 최상의 성능을 유지할 수 있도록 하기 위해 기본 제공되는 기능입니다. 리소스 크기를 특정 인스턴스 수(으)로 수동으로 조정하거나, 메트릭 임계값에 따라 크기 조정을 하는 사용자 지정 자동 스케일링 정책을 통해 조정하거나, 지정된 기간 동안 스케일링되는 인스턴스 수를(들) 예약도 선택할 수 있습니다. 자동 스케일링을 사용하면 요청에 따라(들) 추가하고 인스턴스 수를 제거하여 리소스의 성능을 향상하고 비용 효율적으로 유지할 수 있습니다. Azure 자동 스케일링에 대해 자세히 알아보거나 방법 동영상을 확인하세요.

리소스 크기를 조정하는 방법 선택

수동 크기 조정  
고정 인스턴스 수 유지

사용자 지정 자동 크기 조정  
모든 메트릭을 기반으로 일정에 따라 크기 조정

사용자 지정 자동 크기 조정

자동 크기 조정 설정 이름 \*  
ASP-az10409arg1lod31504655-a5c6-자동 크기 조정-142

리소스 그룹  
az104-09a-rg1-lod31504655

인스턴스 수  
1

기본값 \* 자동 생성된 기본 스케일링 조건

삭제 경고

크기 조정 모드  
규칙

스케일 아웃  
시기 ASP-az10409arg1lod3... (최소값) CpuPercentage > 10 수를 1만큼 증가

인스턴스 제한

최소값 \* ①  
1

최대값 \* ①  
2

기본값 \* ①  
1

이 크기 조정 조건은 일치하는 다른 크기 조정 조건이 없을 때 실행됩니다.

+ 크기 조건 추가

5. Azure 포털에서 [Cloud Shell]을 열고 PowerShell 세션에서 다음 명령을 실행하여 Azure 웹 앱의 URL을 확인합니다.

```
# Azure 웹 앱 URL 확인
$rgName = 'az104-09a-rg1-lod<xxxxxxxx>'
$webapp = Get-AzWebApp -ResourceGroupName $rgName

PowerShell
PS /home/labuser-31504655> # Azure 웹 앱 URL 확인
PS /home/labuser-31504655> $rgName = 'az104-09a-rg1-lod31504655'
PS /home/labuser-31504655> $webapp = Get-AzWebApp -ResourceGroupName $rgName
PS /home/labuser-31504655> 
```



6. [Cloud Shell]에서 다음 명령을 실행하여 웹 앱에 대한 HTTP 요청을 전송하는 무한 반복을 시작합니다.

```
# 웹 앱에 요청 전송  
while ($true) { Invoke-WebRequest -Uri $webapp.DefaultHostName }  
  
PowerShell v | ? ⚙️ 📄 {} 🔍
```

```
PS /home/labuser-31504655> # 웹 앱에 요청 전송  
PS /home/labuser-31504655> while ($true) { Invoke-WebRequest -Uri $webapp.DefaultHostName }
```

```
StatusCode      : 200  
StatusDescription : OK  
Content         : Hello World!  
RawContent     : HTTP/1.1 200 OK  
                Date: Sat, 03 Jun 2023 06:26:24 GMT  
                Server: nginx/1.22.1  
                Transfer-Encoding: chunked  
                X-Powered-By: PHP/8.2.5  
                Content-Type: text/html; charset=utf-8  
  
                Hello World!  
Headers       : {[Date, System.String[]], [Server, System.String[]], [Transfer-Encoding, System.String[]], [X-Powered-By, System.String[]]-}  
Images        : []  
InputFields   : []  
Links         : []  
RawContentLength : 12  
RelationLink  : []
```

7. [Cloud Shell]을 최소화하고 웹 앱 블레이드로 다시 이동합니다. [자동 크기 조정 설정] 블레이드의 [실행 기록] 탭으로 이동합니다.

- 현재 실행 중인 **Observed Capacity**의 수가 "2"로 표시되는 것을 확인합니다.
- 작업에 "**Autoscale scale up completed**" 이벤트가 표시되는 것을 확인합니다.



8. App Service의 런타임 스택과 OS 버전에서 지원되는 경우 cordless 모니터링이 지원됩니다. 이 경우 [App Service] 블레이드의 [모니터링 - 프로세스 탐색기]로 이동하면 아래와 같은 내용을 확인할 수 있습니다.

- 프로세스 탐색기는 인스턴스 수와 리소스 사용률을 쉽게 모니터링 할 수 있도록 도와줍니다.
- [새로 고침]을 클릭하여 최신 내용을 확인합니다. 어느 정도 시간 후 인스턴스 수가 하나 더 증가하는 것을 확인할 수 있습니다.

