

AZ-104. Challenge Lab 08

LAB 11. PowerShell DSC 확장으로 가상 머신 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
2023.08.30	1.0.0	우진환	LAB 11 내용 작성

목차

도전 과제	5
STEP 01. 스토리지 계정 만들기.....	5
STEP 02. WINDOWS 가상 머신 만들기.....	5
STEP 03. 구성 파일 만들기.....	5
STEP 04. POWERSHELL DSC 확장 추가.....	6
TASK 01. 스토리지 계정 만들기	7
TASK 02. AZURE 가상 머신 만들기	8
TASK 03. 구성 파일 만들기	10
TASK 04. POWERSHELL DSC 확장 추가	12

도전 과제

이 실습에서는 PowerShell DSC (Desired State Configuration) 확장을 사용하여 가상 머신을 구성합니다.

STEP 01. 스토리지 계정 만들기

1. 다음 속성을 사용하여 새 스토리지 계정을 만듭니다.

속성	값
리소스 그룹	rg1lod<xxxxxxxx>
스토리지 계정 이름	sa<xxxxxxxx>ais
성능	표준
중복	GRS(지역 중복 스토리지)

2. 새로 만든 스토리지 계정에 `scripts` 이름의 컨테이너를 만듭니다.

STEP 02. Windows 가상 머신 만들기

1. 다음 속성을 사용하여 새 가상 머신을 만듭니다.

속성	값
리소스 그룹	rg1lod<xxxxxxxx>
가상 머신 이름	VM1
가용성 옵션	인프라 중복이 필요하지 않습니다.
이미지	[smalldisk] Windows Server 2022 Datacenter: Azure Edition Core
크기	Standard_DS1_v2
사용자 이름	azureUser
암호	Pa55w.rd1234
인바운드 포트 선택	HTTP (80), RDP (3389)
부트 진단	사용 안 함

2. 새로 만든 가상 머신의 공용 IP 주소를 확인합니다.

STEP 03. 구성 파일 만들기

1. 다음 내용의 `WebApp.ps1` 파일을 만듭니다.

```
configuration WebApp
{
    Script FirewallRule
    {
        GetScript = { @ { Result = [string]$(netsh advfirewall firewall show rule name="http") } }
        SetScript = { netsh advfirewall firewall add rule name="http" dir=in action=allow protocol=TCP localport=80 }
        TestScript = {
            if ((netsh advfirewall firewall show rule name="http") -match 'No rules match the specified criteria') { $false }
            else { $true }
        }
    }
    Script IndexHtmlFile
    {
        GetScript = { @ { Result = (Test-Path -Path 'C:\inetpub\wwwroot\index.html') } }
        SetScript = {
```

```

$outFile = 'C:\inetpub\wwwroot\index.html'
Invoke-WebRequest
'https://raw.githubusercontent.com/LODSCContent/ChallengeLabs_ArmResources/master
/Labs/AIS/index.html' -OutFile $outFile
Unblock-File -Path $outFile
}
TestScript = { Test-Path -Path 'C:\inetpub\wwwroot\index.html' }
DependsOn = '[WindowsFeature]IIS'
}
WindowsFeature IIS
{
  Ensure = "Present"
  Name = "Web-Server"
}
}

```

2. WebApp.ps1 파일을 WebApp.zip 파일로 압축합니다.
3. 다음 내용으로 PowerShell 모듈 매니페스트 파일을 만들고 allnodes.psd1 이름으로 저장합니다.

```

@{
  AllNodes = @()
  NonNodeData = ""
}

```

4. 스토리지 계정의 scripts 컨테이너에 WebApp.zip 파일과 allnodes.psd1 파일을 업로드합니다.

STEP 04. PowerShell DSC 확장 추가

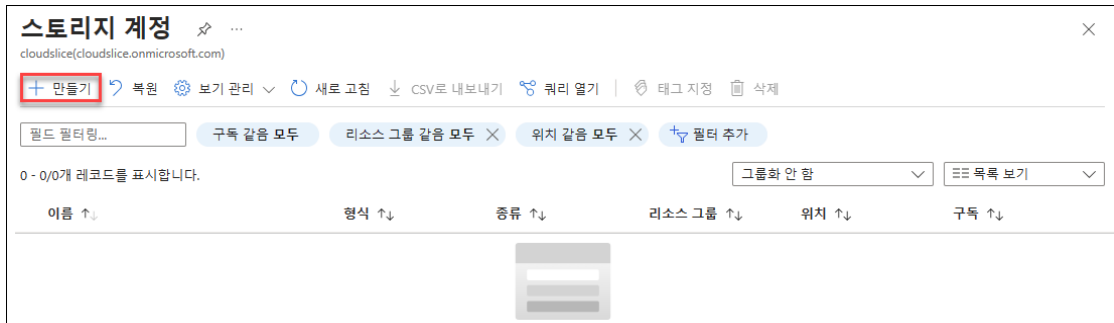
1. VM1 가상 머신에 PowerShell Desired State Configuration 확장을 다음 구성으로 추가합니다.

속성	값
Configuration Modules or Script	스토리지 계정에 업로드한 "WebApp.zip" 파일을 선택
Module-qualified Name of Configuration	WebApp.ps1\WebApp
Configuration Data PSD1 File	스토리지 계정에 업로드한 "allnodes.psd1" 파일을 선택
WMF Version	latest
Data Collection	Enable
Version	DSC 확장 버전 히스토리 페이지 에서 가장 최신 버전을 입력

2. PowerShell DSC 확장이 성공적으로 프로비저닝된 것을 확인합니다.
3. 브라우저에서 새 탭을 열고 VM1 가상 머신의 공용 IP 주소에 액세스하여 새 웹 앱이 표시되는지 확인합니다.

TASK 01. 스토리지 계정 만들기

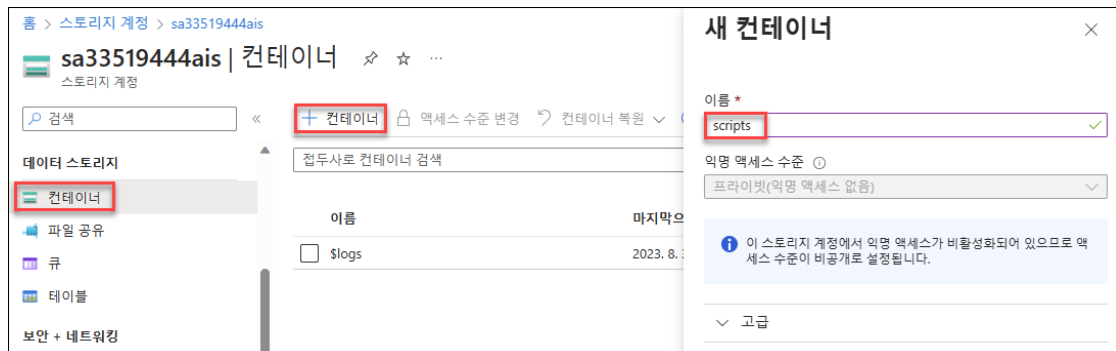
1. Azure 포털의 검색창에서 "스토리지 계정"을 검색한 후 클릭합니다. [스토리지 계정] 블레이드의 메뉴에서 [만들기]를 클릭합니다.



2. [저장소 계정 만들기] 블레이드의 [기본] 탭에서 아래와 같이 구성한 후 [검토]를 클릭합니다. [검토] 탭에서 [만들기]를 클릭합니다.

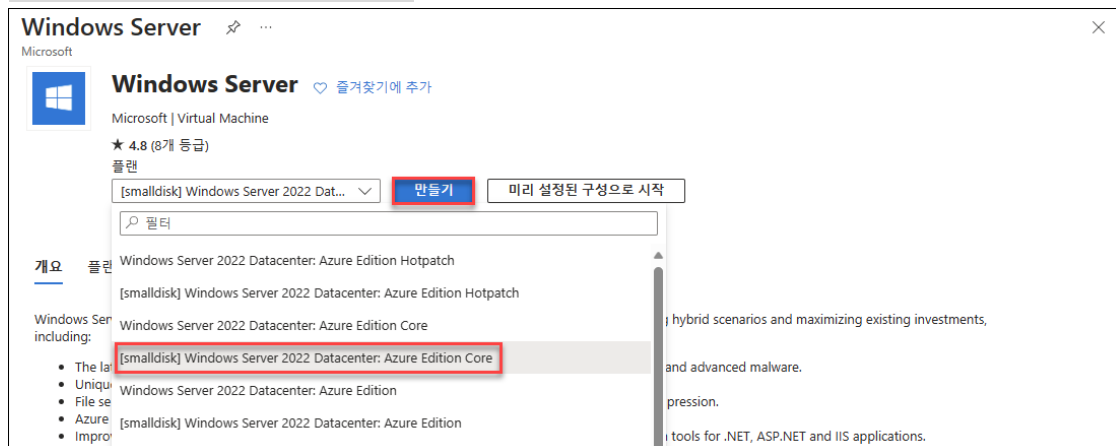
- [프로젝트 정보 - 리소스 그룹]: rg1lod<xxxxxxxx>
- [인스턴스 정보 - 스토리지 계정 이름]: sa<xxxxxxxx>ais
- [인스턴스 정보 - 지역]: (US) East US
- [인스턴스 정보 - 성능]: 표준
- [인스턴스 정보 - 중복]: GRS(지역 중복 스토리지)

3. 새로 만든 [sa<xxxxxxxx> ais 스토리지 계정] 블레이드의 [데이터 스토리지 - 컨테이너]로 이동한 후 메뉴에서 [컨테이너]를 클릭합니다. [새 컨테이너] 창에서 이름에 "scripts"를 입력한 후 [만들기]를 클릭합니다.



TASK 02. Azure 가상 머신 만들기

1. Azure 포털에서 [리소스 만들기]를 클릭한 후 "Windows Server"를 검색하고 [Windows Server] 타일을 클릭합니다. [Windows Server] 블레이드에서 "[smalldisk] Windows Server 2022 Datacenter: Azure Edition Core"를 선택하고 [만들기]를 클릭합니다.



2. [가상 머신 만들기] 블레이드의 [기본 사항] 탭에서 아래와 같이 구성한 후 [모니터링] 탭으로 이동합니다.

- [프로젝트 정보 - 리소스 그룹]: rg1lod<xxxxxxxx>
- [인스턴스 정보 - 가상 머신 이름]: VM1
- [인스턴스 정보 - 지역]: (US) East US
- [인스턴스 정보 - 가용성 옵션]: 인프라 중복이 필요하지 않습니다.
- [인스턴스 정보 - 보안 유형]: 신뢰할 수 있는 시작 가상 머신
- [인스턴스 정보 - 크기]: Standard_DS1_v2
- [관리자 계정 - 사용자 이름]: azureUser
- [관리자 계정 - 암호]: Pa55w.rd1234
- [인바운드 포트 규칙 - 공용 인바운드 포트]: 선택한 포트 허용
- [인바운드 포트 규칙 - 인바운드 포트 선택]: HTTP (80), RDP (3389)

가상 머신 만들기

기본 사항 디스크 네트워킹 관리 모니터링 고급 태그 검토 + 만들기

Linux 또는 Windows를 실행하는 가상 머신을 만듭니다. Azure Marketplace에서 이미지를 선택하거나 고유한 사용자 지정 이미지를 사용합니다. [기본] 탭을 완료하고 [검토 + 만들기]하여 기본 매개 변수로 가상 머신을 프로비전하거나, 전체 사용자 지정에 대해 각 탭을 검토합니다. [자세한 정보](#)

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

구독 *

리소스 그룹 * [새로 만들기](#)

인스턴스 정보

가상 머신 이름 *

지역 *

가용성 옵션

보안 유형

이미지 *

VM 아키텍처 ☐ Arm64 ☒ x64
Arm64는 선택한 이미지에서 지원되지 않습니다.

Azure Spot 할인으로 실행 ☐

크기 * [모든 크기 보기](#)
선택한 범위에 대한 정책 할당을 기준으로 한 항목 가용성입니다.
policyAssignment1142 ([정책 세부 정보](#))

관리자 계정

사용자 이름 *

암호 *

암호 확인 *

인바운드 포트 규칙
공중 인터넷에서 액세스할 수 있는 가상 머신 네트워크 포트를 선택하세요. [네트워킹] 탭에서 더 제한되거나 세분화된 네트워크 액세스를 지정할 수 있습니다.

공용 인바운드 포트 * ☐ 없음 ☒ 선택한 포트 허용

인바운드 포트 선택 *
인터넷의 모든 트래픽이 기본적으로 차단됩니다. [VM] > [네트워킹] 페이지에서 인바운드 포트 규칙을 변경할 수 있습니다.

라이선싱
Azure 하이브리드 혜택 사용하여 이미 소유한 라이선스로 최대 49%를 절약하세요. [자세한 정보](#)
기존 Windows Server 라이선스를 사용하시겠습니까? ☐

[Azure 하이브리드 혜택 준수 검토](#)

3. [모니터링] 탭에서 부트 진단을 "사용 안 함"으로 선택한 후 [검토 + 만들기]를 클릭합니다. [검토 + 만들기] 탭에서 [만들기]를 클릭합니다.

가상 머신 만들기

기본 사항 디스크 네트워킹 관리 **모니터링** 고급 태그 검토 + 만들기

VM에 대한 모니터링 옵션을 구성합니다.

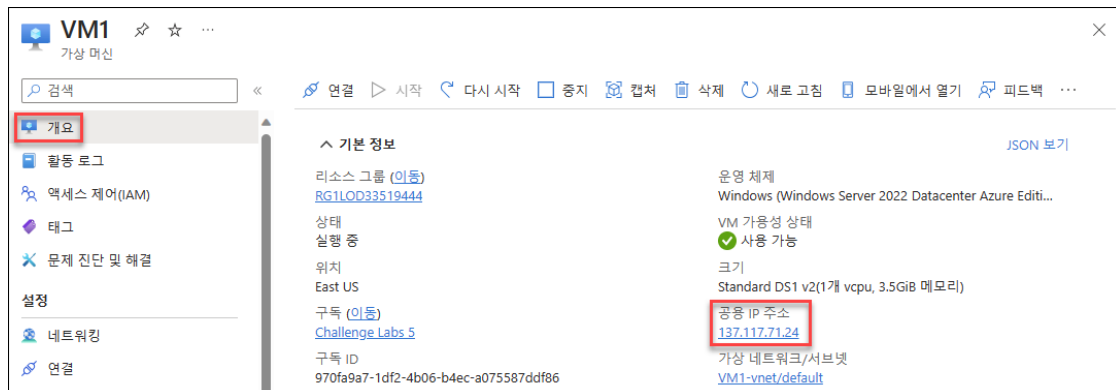
경고
권장 경고 규칙 사용 ☐

진단

부트 진단 ☐ 관리형 스토리지 계정으로 사용하도록 설정(권장) ☐ 사용자 지정 스토리지 계정으로 사용하도록 설정 ☒ 사용 안 함

OS 게스트 진단 사용 ☐

4. 새로 만든 [VM1 가상 머신] 블레이드의 [개요]에서 공용 IP 주소를 메모장에 복사합니다.



TASK 03. 구성 파일 만들기

1. 메모장을 열고 다음 내용을 작성한 다음 `WebApp.ps1` 이름으로 저장합니다.

```
configuration WebApp
{
    Script FirewallRule
    {
        GetScript = { @ { Result = [string]$(netsh advfirewall firewall show rule name="http") } }
        SetScript = { netsh advfirewall firewall add rule name="http" dir=in action=allow protocol=TCP localport=80 }
        TestScript = {
            if ((netsh advfirewall firewall show rule name="http") -match 'No rules match the specified criteria') { $false }
            else { $true }
        }
    }
    Script IndexHtmlFile
    {
        GetScript = { @ { Result = (Test-Path -Path 'C:\inetpub\wwwroot\index.html') } }
        SetScript = {
            $outFile = 'C:\inetpub\wwwroot\index.html'
            Invoke-WebRequest
            'https://raw.githubusercontent.com/LODSCContent/ChallengeLabs_ArmResources/master/Labs/AIS/index.html' -OutFile $outFile
            Unblock-File -Path $outFile
        }
        TestScript = { Test-Path -Path 'C:\inetpub\wwwroot\index.html' }
        DependsOn = '[WindowsFeature]IIS'
    }
    WindowsFeature IIS
    {
        Ensure = "Present"
        Name = "Web-Server"
    }
}
```

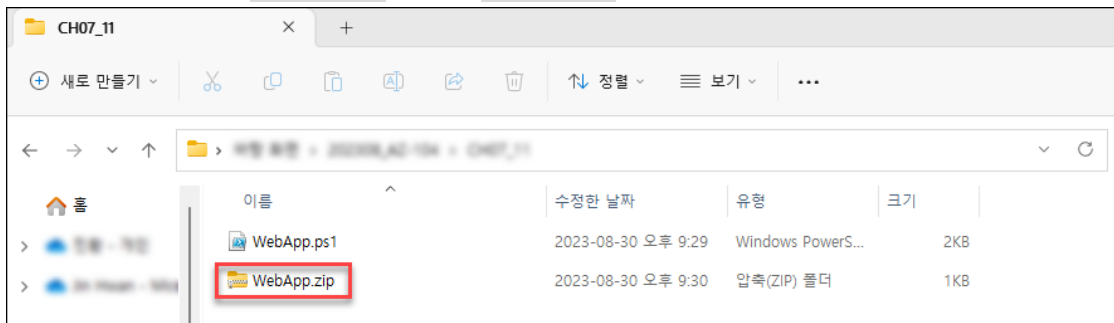
```

WebApp.ps1
파일 편집 보기

configuration WebApp
{
    Script FirewallRule
    {
        GetScript = { $Result = [string]$(netsh advfirewall firewall show rule name="http") } }
        SetScript = { netsh advfirewall firewall add rule name="http" dir=in action=allow protocol=TCP localport=80 }
        TestScript = { if ((netsh advfirewall firewall show rule name="http") -match 'No rules match the specified criteria') { $false } else { $true } }
    }
    Script IndexHtmlFile
    {
        GetScript = { $Result = (Test-Path -Path 'C:\inetpub\wwwroot\index.html') } }
        SetScript = { $outFile = 'C:\inetpub\wwwroot\index.html'
        Invoke-WebRequest 'https://raw.githubusercontent.com/L0D5Content/ChallengeLabs_ArmResources/master/Labs/AIS/index.html' -OutFile $outFile
        Unblock-File -Path $outFile }
        TestScript = { Test-Path -Path 'C:\inetpub\wwwroot\index.html' }
        DependsOn = '[WindowsFeature]IIS'
    }
    WindowsFeature IIS
    {
        Ensure = "Present"
        Name = "Web-Server"
    }
}

```

2. [파일 탐색기]를 열고 **WebApp.ps1** 파일을 **WebApp.zip** 파일로 압축합니다.



3. 메모장을 열고 다음 내용을 작성한 후 **allnodes.ps1** 이름으로 저장합니다. 빌트-인 DSC ConfigurationData 매개 변수를 사용하여 DSC 구성에서 사용할 데이터를 정의할 수 있습니다. 또한 여러 환경에 대해 단일 구성을 사용할 수 있습니다.

```

{
    AllNodes = @()
    NonNodeData = ""
}

```

allnodes.ps1

```

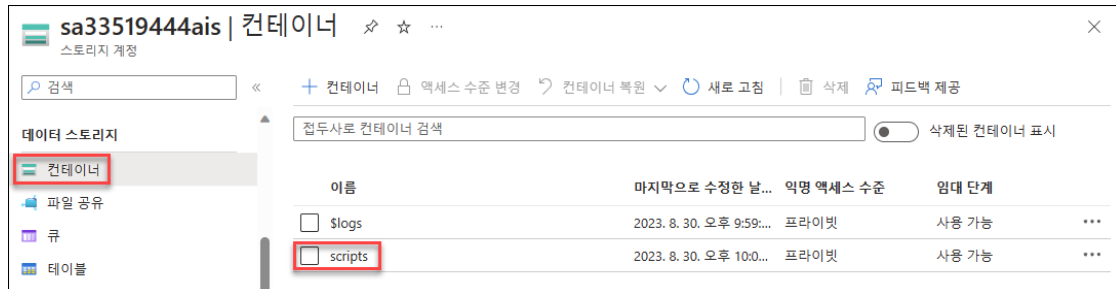
{
    AllNodes = @()
    NonNodeData = ""
}

```

4. Azure 포털의 검색창에서 "스토리지 계정"을 검색한 후 클릭합니다. [스토리지 계정] 블레이드에서 **sa<xxxxxxxx>ais** 스토리지 계정을 클릭합니다.



5. [sa<xxxxxxxx>ais] 스토리지 계정 블레이드의 [데이터 스토리지 - 컨테이너]로 이동한 후 **scripts** 컨테이너를 클릭합니다.

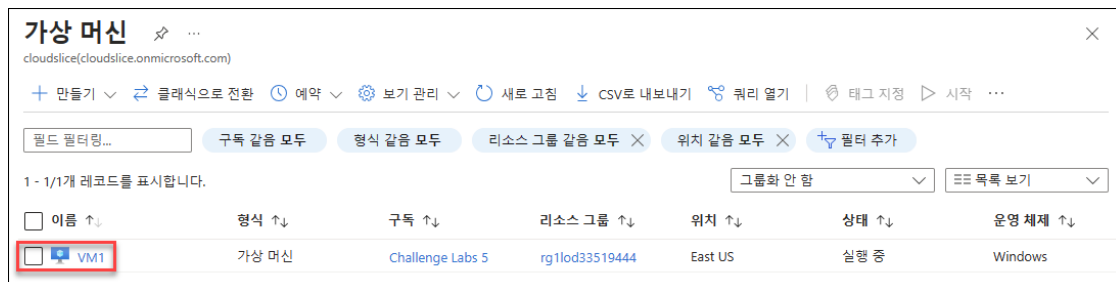


6. [scripts 컨테이너]에서 [업로드]를 클릭합니다. [Blob 업로드] 창에서 앞서 만들었던 WebApp.zip 파일과 allnodes.psd1 파일을 업로드합니다.



TASK 04. PowerShell DSC 확장 추가

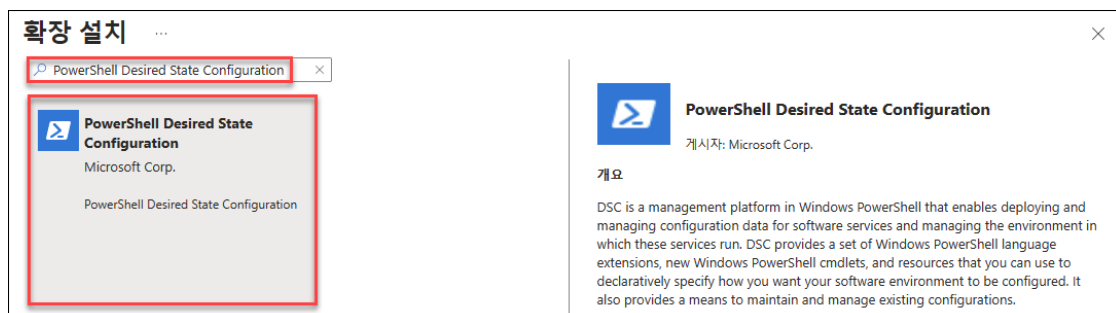
1. Azure 포털의 검색창에서 "가상 머신"을 검색한 후 클릭합니다. [가상 머신] 블레이드에서 VM1 가상 머신을 클릭합니다.



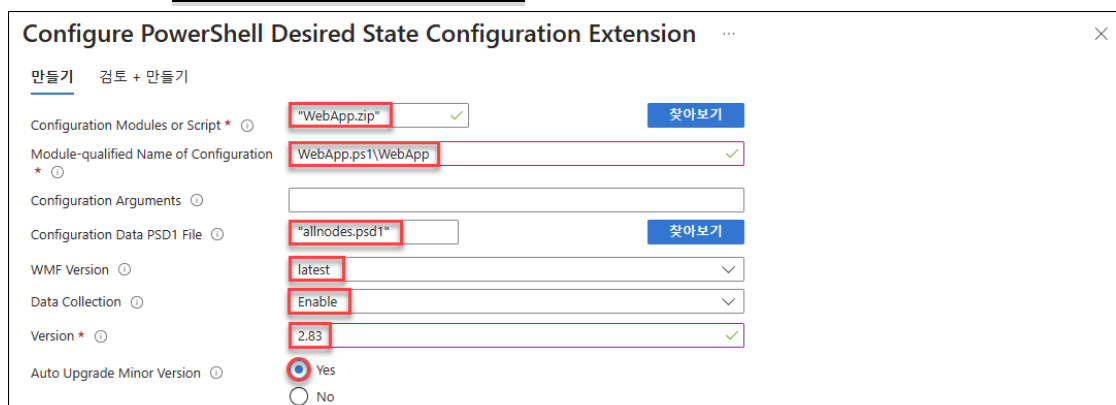
2. [VM1 가상 머신] 블레이드의 [설정 - 확장 프로그램 + 애플리케이션]으로 이동합니다. [확장] 탭에서 [추가]를 클릭합니다.



3. [확장 설치] 블레이드에서 "PowerShell Desired State Configuration"을 검색한 후 선택하고 [다음]을 클릭합니다.

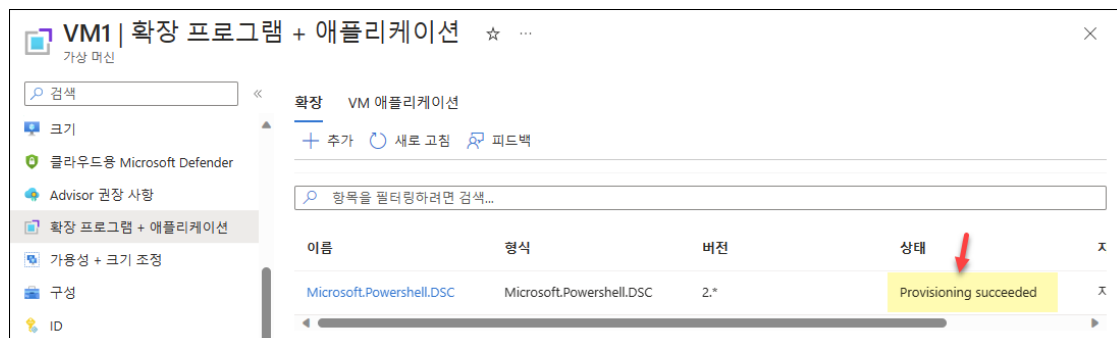


4. [Configure PowerShell Desired State Configuration Extension] 블레이드의 [만들기] 탭에서 아래와 같이 구성한 후 [다음]을 클릭합니다. [검토 + 만들기] 탭에서 [만들기]를 클릭합니다.
- Configuration Modules or Script: [찾아보기]를 클릭한 후 `cs<xxxxxxxx>ais` 스토리지 계정의 `scripts` 컨테이너에서 "WebApp.zip" 파일을 선택합니다.
 - Module-qualified Name of Configuration: WebApp.ps1\WebApp
 - Configuration Data PSD1 File: [찾아보기]를 클릭한 후 `cs<xxxxxxxx>ais` 스토리지 계정의 `scripts` 컨테이너에서 "allnodes.psd1" 파일을 선택합니다.
 - WMF Version: latest
 - Data Collection: Enable
 - Version: DSC 확장 버전 히스토리 페이지에서 최신 버전을 확인한 후 입력합니다.



5. DSC 확장이 배포될 때까지 3~5분 정도가 소요됩니다. 배포가 완료되면 [VM1 가상 머신]

블레이드의 [확장 프로그램 + 애플리케이션]으로 이동한 후 추가한 DSC 확장이 프로비저닝되었는지 확인합니다.



6. 브라우저에서 새 탭을 열고 VM1 가상 머신의 공용 IP에 액세스합니다. 아래와 같이 웹 앱이 표시되는 것을 확인합니다.

