

# Azure DevOps 400 과정

강사 : 주민규

## Jenkins with Azure VM

### // Azure Resource Group 생성

```
$ az group create -n TestRG02 -l eastus
```

### // Azure VM 생성

```
$ az vm create -g TestRG02 -n vm01 --admin-username mingyu --  
image UbuntuLTS --ssh-key-value ~/.ssh/mod10_rsa.pub
```

```
$ sudo apt update
```

```
$ sudo apt upgrade
```

### // Node.js 설치

```
$ sudo curl -sL https://deb.nodesource.com/setup_12.x | sudo -E bash -  
$ sudo apt install -y nodejs
```

**// NginX 가 설치되어 있지 않아서 3000 포트로 연결되어 있지 않음, VM 에서 아래 부분을 수정하여 Nginx 를 Reverse Proxy로 설정한다.**

```
$ sudo apt install nginx
```

```
$ sudo apt update
```

```
$ sudo nano /etc/nginx/sites-available/default
```

```
try_files $uri $uri/ =404; 주식 처리 후
```

```
proxy_pass http://127.0.0.1:3000/;
```

```
$ sudo service nginx restart
```

### // 실행 후 아래와 같이 node\_modules 폴더가 있는지 확인!

```
mingyu@vm01:/var/lib/jenkins/workspace/JenkinsOnVM$ ls  
index.js node_modules package-lock.json package.json
```

### // Jenkins 설치 준비 - Java 도 설치

```
$ sudo apt install default-jre  
$ java -version  
$ wget -q -O - https://pkg.jenkins.io/debian/jenkins-ci.org.key | sudo  
apt-key add -
```

### // Repository 추가

```
$ echo deb https://pkg.jenkins.io/debian-stable binary/ | sudo tee /etc/  
apt/sources.list.d/jenkins.list
```

### // Jenkins 설치

```
$ sudo apt install jenkins  
$ sudo apt update
```

### // Jenkins 포트 확인 : 기본 8080

```
$ sudo nano /etc/default/jenkins
```

### // Jenkins 시작

```
$ sudo systemctl start jenkins
```

### // Jenkins 상태 확인

```
$ sudo systemctl status jenkins
```

### // VM 포트 열기

```
$ az vm open-port -g testrg02 -n vm01 --port 8080 --priority 1001  
$ az vm open-port -g testrg02 -n vm01 --port 80 --priority 1002
```

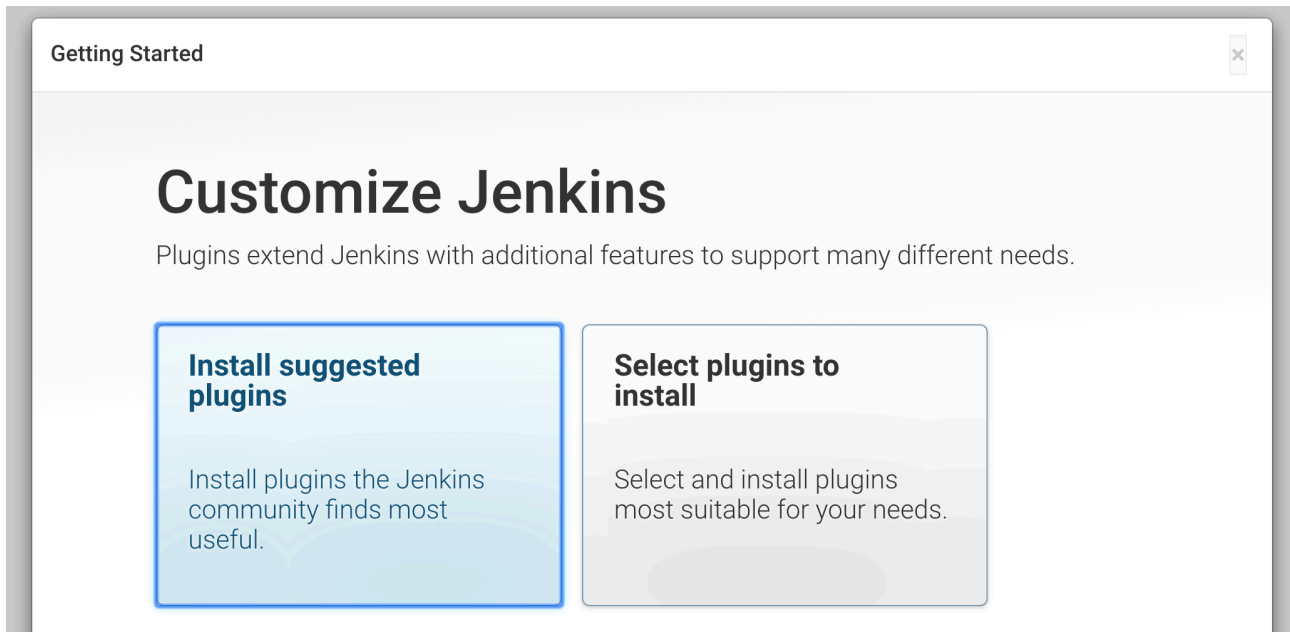
### // Jenkins Password 확인

```
$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

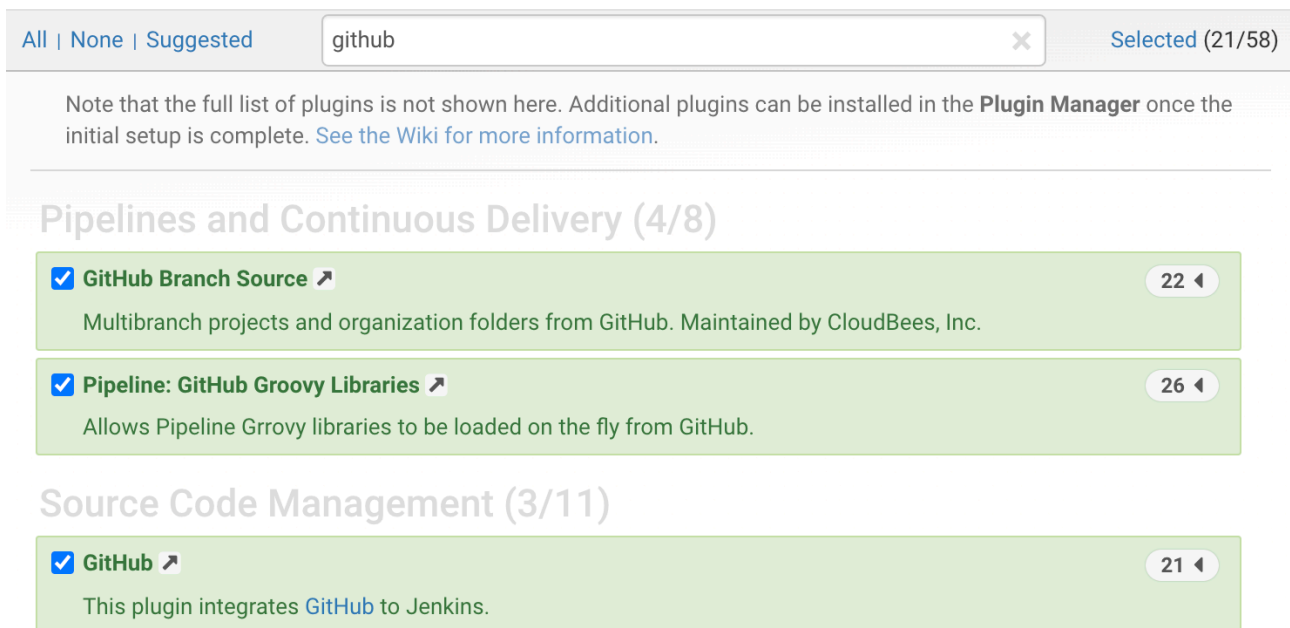
### // 웹브라우저로 Jenkins 페이지 접속

```
http://52.152.145.186:8080/
```

## // 우측 Select Plugins to install 선택



## // 깃허브 검색 후 선택 -> 설치



# Getting Started

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	✓ Credentials Binding	<pre> ** Pipeline: Nodes and Processes ** JUnit ** Matrix Project ** Resource Disposer Workspace Cleanup Ant ** JavaScript GUI Lib: ACE Editor bundle ** JavaScript GUI Lib: jQuery bundles (jQuery and jQuery UI) ** Pipeline: SCM Step ** Pipeline: Groovy ** Pipeline: Job ** Apache HttpComponents Client 4.x API ** Display URL API Mailer ** Pipeline: Basic Steps Gradle </pre>
✓ Timestampers	✓ Workspace Cleanup	✓ Ant	✓ Gradle	
✓ Pipeline	↻ GitHub Branch Source	↻ Pipeline: GitHub Groovy Libraries	✓ Pipeline: Stage View	
↻ Git	↻ Subversion	↻ SSH Build Agents	↻ Matrix Authorization Strategy	
↻ PAM Authentication	↻ LDAP	↻ Email Extension	✓ Mailer	
↻ GitHub				

// 헬로우 월드 앱을 Fork 한다.

<https://github.com/Azure-Samples/nodejs-docs-hello-world>

<https://github.com/CodingSta/nodejs-docs-hello-world>

// 아래 설정을 하면, 깃허브에 소스 갱신 등의 이벤트가 있을 때, **Payload URL**로 푸시를 전달 한다.

// 깃허브의 해당 레포지토리에서 우상단 **Settings** 누르고 좌측 탭 **Webhooks**를 선택.

Add webhook(웹훅 추가)를 선택한 다음, 필터 상자에서 Jenkins를 입력.

1. Payload URL(페이로드 URL)에 대해 `http://<publicip>:8080/github-webhook/`를 입력합니다. 후행 슬래시(/)를 포함.
2. Content type(콘텐츠 형식)에 대해 `application/x-www-form-urlencoded`를 선택
3. 이벤트는 Just the push event 선택.
4. Active 선택.

## Webhooks / Add webhook

We'll send a POST request to the URL below with details of any subscribed events. You can choose the data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information in the [developer documentation](#).

### Payload URL \*

### Content type

### Secret

### Which events would you like to trigger this webhook?

- ☒ Just the push event.
- ☐ Send me **everything**.
- ☐ Let me select individual events.

### ☒ Active

We will deliver event details when this hook is triggered.

// 위 설정을 한 후에, Jenkins 가 코드 커밋 등의 GitHub 이벤트에 대해서 응답하게 하려면 Jenkins 에서 아래와 같이 설정한다.



1. Jenkins 웹 사이트에서 홈 페이지에서 새 작업 만들기를 선택
2. HelloWorld를 작업 이름으로 입력합니다. 프리스타일 프로젝트를 선택
3. General 탭 에서 GitHub project 선택 후 포크된 리포지토리 URL 입력
4. <https://github.com/CodingSta/nodejs-docs-hello-world>

General   소스 코드 관리   빌드 유발   빌드 환경   Build   빌드 후 조치

설명

[Plain text] [미리보기](#)

☒ GitHub project

Project url   `https://github.com/CodingSta/nodejs-docs-hello-world`

1. 소스 코드 관리 섹션에서 Git 선택하고 아래 입력.
2. <https://github.com/CodingSta/nodejs-docs-hello-world.git>

소스 코드 관리

☐ None

☒ Git

Repositories

Repository URL   `https://github.com/CodingSta/nodejs-docs-hello-world.git`

Credentials   `- none -`   [Add](#)

1. 빌드 유발(트리거 빌드)에서 GitHub hook trigger for GITScm polling 선택

### 빌드 유발

- ☐ 빌드를 원격으로 유발 (예: 스크립트 사용)
- ☐ Build after other projects are built
- ☐ Build periodically
- ☒ GitHub hook trigger for GITScm polling
- ☐ Poll SCM

8. 빌드 섹션 아래에서 빌드 단계 추가를 선택하고 echo “Test” 입력.

### Build

Add build step ▼

- Execute Windows batch command
- Execute shell
- Invoke Ant
- Invoke Gradle script
- Invoke top-level Maven targets
- Run with timeout
- Set build status to "pending" on GitHub commit

9. 위에서 설정한 것을 바꾸려면 좌측 메뉴에서 “구성” 을 선택.





대시보드로 돌아가기

상태

변경사항

작업공간

작업공간 초기화

Build Now

Project 삭제

구성

GitHub Hook Log

GitHub

Rename

## 10. GitHub 통합 테스트

- 메뉴에서 Github 를 선택하고, 포크한 소스를 수정하여 커밋 한다.
- index.js 파일을 선택하고 Hello World 를 다른 string 으로 수정한다.
- Commit 버튼을 누르고 나면, 아래와 같이 젠킨스 페이지에서 빌드가 된다.


All					
S	W	Name ↓	최근 성공	최근 실패	최근 소요 시간
		<a href="#">HelloMinGyu</a>	52 sec - <a href="#">#1</a>	—	4.1 sec

- 빌드 번호 링크를 선택하고 왼쪽에 있는 콘솔 출력을 선택.

 프로젝트로 돌아가기

 상태

 바뀐점

 Console Output

 빌드 정보 수정

 Delete build ‘#1’

 Polling Log

 Git Build Data

 No Tags

 빌드 #1 (2020



No changes.



[Started by GitHub pus](#)



Revision: 15431e1f87

• refs/remotes/or

- 코드가 GitHub에서 로드되면 Jenkins가 수행하는 단계와 콘솔에 메시지 Test을 출력하는 빌드 작업을 확인 가능.
- GitHub에서 커밋이 수행될 때마다 웹후크는 Jenkins에 도달하며, 이 방법으로 새 빌드를 트리거

**// Shell Command 는 아래와 같이 BUILD\_ID=dontkillME && 를 붙여야 한다.**

```
echo "Test by MinGyu"
npm install
BUILD_ID=dontkillME && nohup node index.js >> nohup.out 2>&1 &
```

## AzureDevOps with Azure VM

### Azure Ubuntu VM 에서 CI/CD 구성

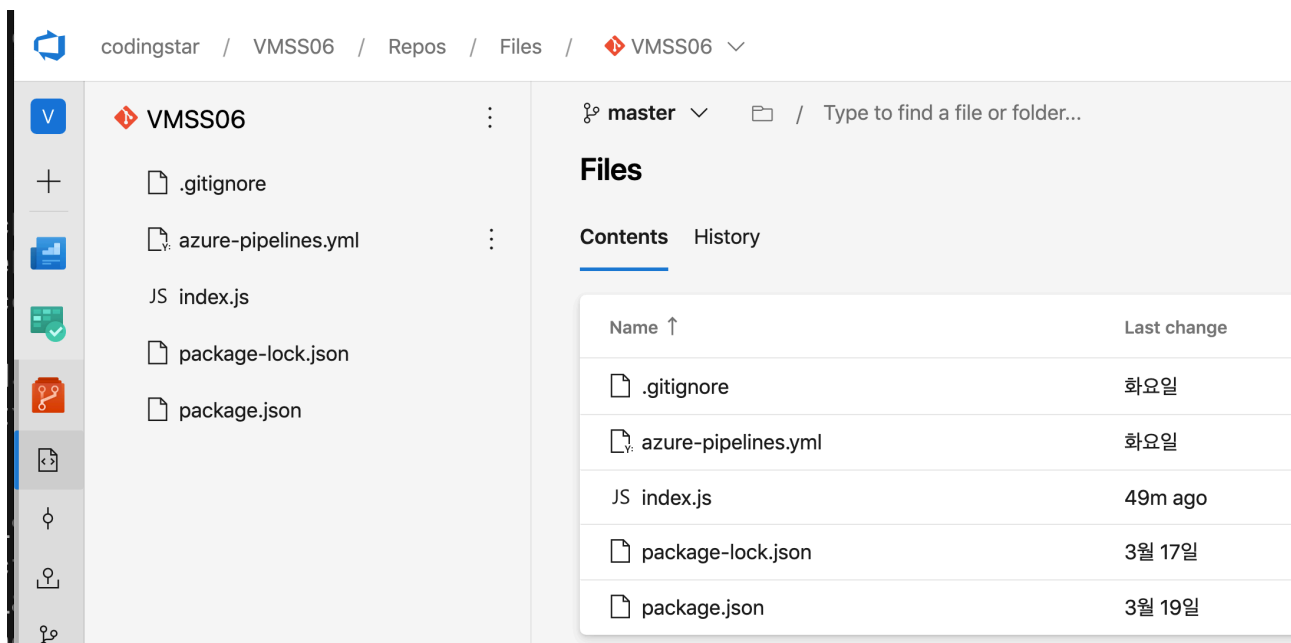
// Azure VM 생성 및 포트 열기

// Node.js 설치

// Nginx 가 설치 및 Nginx 를 Reverse Proxy로 설정하여 Node.js 와 연결.

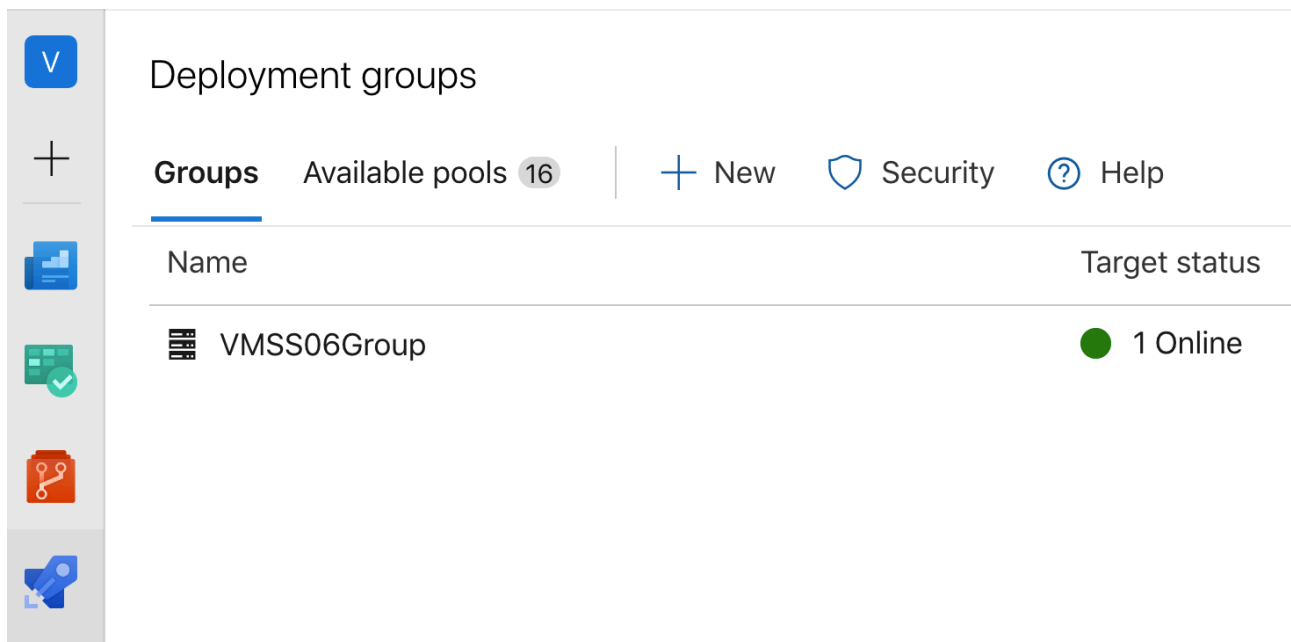
// NginX restart

Azure DevOps 조직에 로그인하고, 프로젝트 생성 및 Deployment Group 생성.  
Repos 에서 git 명령어로 소스 코드 파일을 업로드 한다.



The screenshot shows the Azure DevOps Repos interface for a repository named VMSS06. The left sidebar contains a navigation menu with icons for Home, Add, Repos, Files, and VMSS06. The main area displays the file structure of the repository, including .gitignore, azure-pipelines.yml, JS index.js, package-lock.json, and package.json. The right pane shows the 'Files' section with a table of file details.

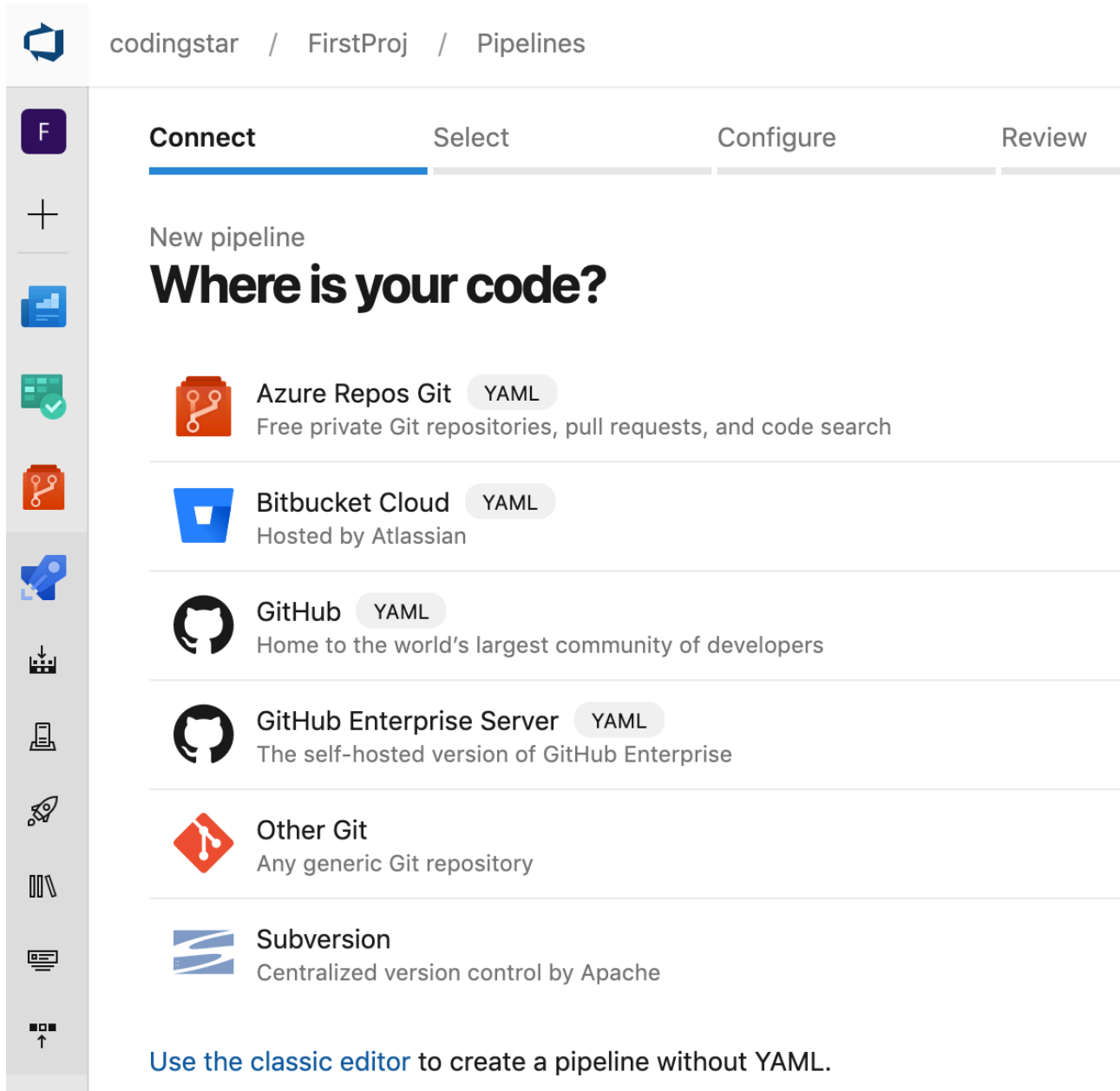
Name ↑	Last change
.gitignore	화요일
azure-pipelines.yml	화요일
JS index.js	49m ago
package-lock.json	3월 17일
package.json	3월 19일



The screenshot shows the Azure DevOps Deployment groups interface. The left sidebar contains a navigation menu with icons for Home, Add, Groups, Available pools, and VMSS06Group. The main area displays the 'Deployment groups' section with a table of group details.

Name	Target status
VMSS06Group	1 Online

// Build Pipeline 을 설정한다. Azure Repos Git 을 선택하고, 선택한 소스에 azure-pipelines.yml 파일이 있다면, 추가 구성은 필요 없음.



The screenshot shows the 'Connect' step of the 'New pipeline' wizard in Azure DevOps. The breadcrumb navigation at the top reads 'codingstar / FirstProj / Pipelines'. The left sidebar contains various icons, including a purple 'F' icon. The main content area has four tabs: 'Connect' (active), 'Select', 'Configure', and 'Review'. Below the tabs, the text 'New pipeline' is followed by the heading 'Where is your code?'. A list of code hosting providers is displayed, each with an icon, name, and a 'YAML' button. The providers are: Azure Repos Git (Free private Git repositories, pull requests, and code search), Bitbucket Cloud (Hosted by Atlassian), GitHub (Home to the world's largest community of developers), GitHub Enterprise Server (The self-hosted version of GitHub Enterprise), Other Git (Any generic Git repository), and Subversion (Centralized version control by Apache). At the bottom, there is a link: 'Use the classic editor to create a pipeline without YAML.'

**GitHub** 를 선택하면 로그인을 위해 리디렉션될 수 있습니다. **GitHub** 자격 증명을 입력합니다.

리포지토리 목록이 표시되면 원하는 샘플 앱 리포지토리를 선택합니다.

## azure-pipelines.yml 의 Sample

```
1  trigger:
2  - master
3
4  variables:
5  | vmImageName: 'ubuntu-18.04'
6
7  stages:
8  | - stage: Build
9  |   jobs:
10  |   | - job: Build
11  |   |   pool:
12  |   |   | vmImage: $(vmImageName)
13  |   |   steps:
14  |   |   | - script: |
15  |   |   |   npm install
16  |   |   |   npm run build --if-present
17  |   |   |   npm run test --if-present
18  |   |   | displayName: 'npm install, build and test'
19
20  |   Settings
21  |   | - task: CopyFiles@2
22  |   |   inputs:
23  |   |   | Contents: '**' ## update to match what you want to copy
24  |   |   | TargetFolder: '$(Build.ArtifactStagingDirectory)'
25
26  |   Settings
27  |   | - task: PublishBuildArtifacts@1
28  |   |   inputs:
29  |   |   | pathToPublish: '$(Build.ArtifactStagingDirectory)' # dist or build files
```

## Release pipeline 을 작성한다.

All pipelines > New release pipeline

Pipeline Tasks Variables Retention Options History

Artifacts | + Add

Stages | + Add

\_VMSS06

Stage 1  
1 job, 2 tasks

Schedule not set

Artifact

Build - \_VMSS06

Project \* VMSS06

Source (build pipeline) \* VMSS06

Default version \* Latest

Source alias \* \_VMSS06

The artifacts published by each v pipelines. The latest successful b drop.

좌측 탭을 눌러서 아티팩트를 선택한다.

그리고 좌측 앱의 번개표시를 눌러서 아래 부분을 활성화 시켜준다.

## Continuous deployment trigger

Build: \_VMSS06

☒ Enabled

Creates a release every time a new build is available.

그리고 Task 메뉴를 눌러서 아래와 같이 셋팅한다.

Pipeline **Tasks** ▾ Variables Retention Options History

Stage 1  
Deployment process

Deployment group job  
Run on deployment group

Stop  
Command line

Start  
Command line

Command line ⓘ

Task version 2.\* ▾

Display name \*  
Stop

Script \*  
pm2 stop index.js

Advanced ^

Working Directory  
\$(System.DefaultWorkingDirectory)/\_VMSS06/drop

☐ Fail on Standard Error ⓘ

Control Options ^

☒ Enabled

☒ Continue on error

## VMSS DevOps

미리 설정해 둔 OS Disk 이미지를 활용하려면...

// 갤러리 생성

```
$ az sig create -g TestRG03 --gallery-name gallery01
```

// 현재 만들어진 VM 의 ID 가져오기

```
$ az vm get-instance-view -g TestRG03 -n vm01 --query id
```

```
출력 : "/subscriptions/8c3bd745-5ec6-46f3-9b52-a455b309bdd6/  
resourceGroups/TestRG03/providers/Microsoft.Compute/  
virtualMachines/vm01"
```

// OS Disk 이미지 정의 생성

```
$ az sig image-definition create -g TestRG03 --gallery-name  
gallery01 --gallery-image-definition vm01ImgDef --offer mingyu --  
sku minSKU --os-type Linux --os-state specialized --publisher  
jumingyu
```

// 15분 이상 걸림 : 이미지 버전 생성 : 위에서 쿼리한 VM ID 이용.

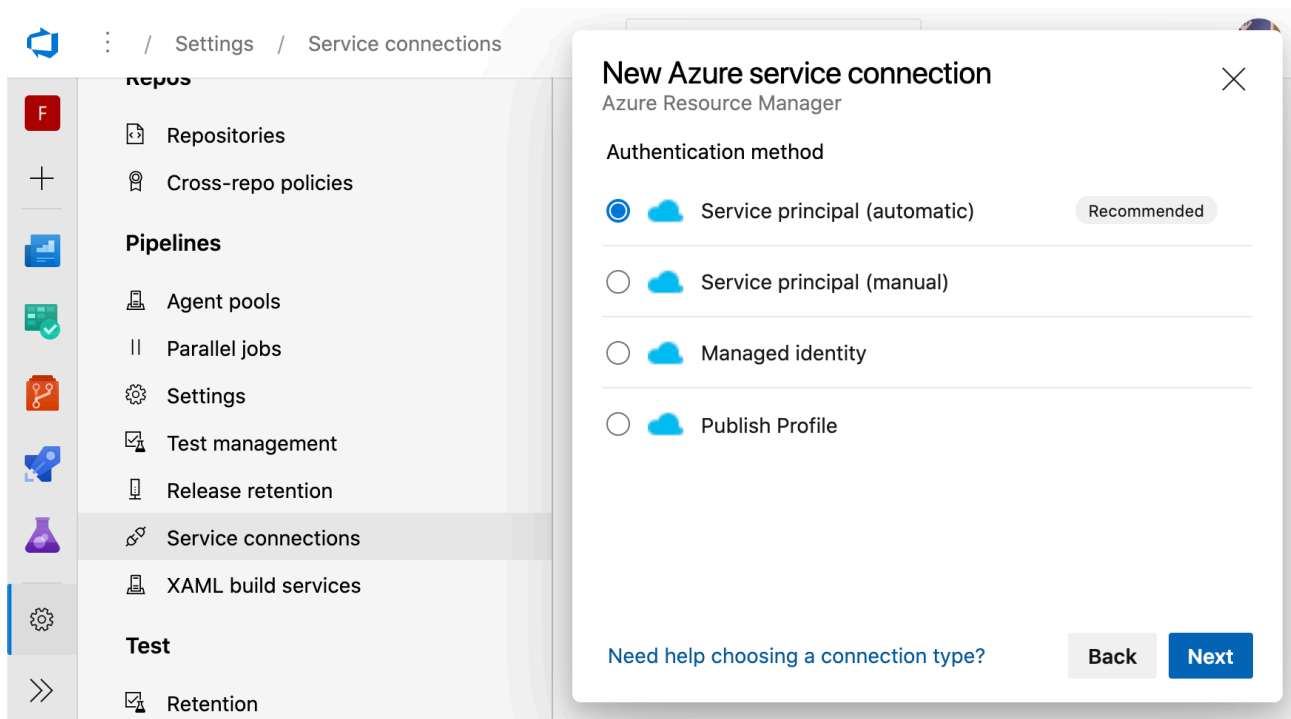
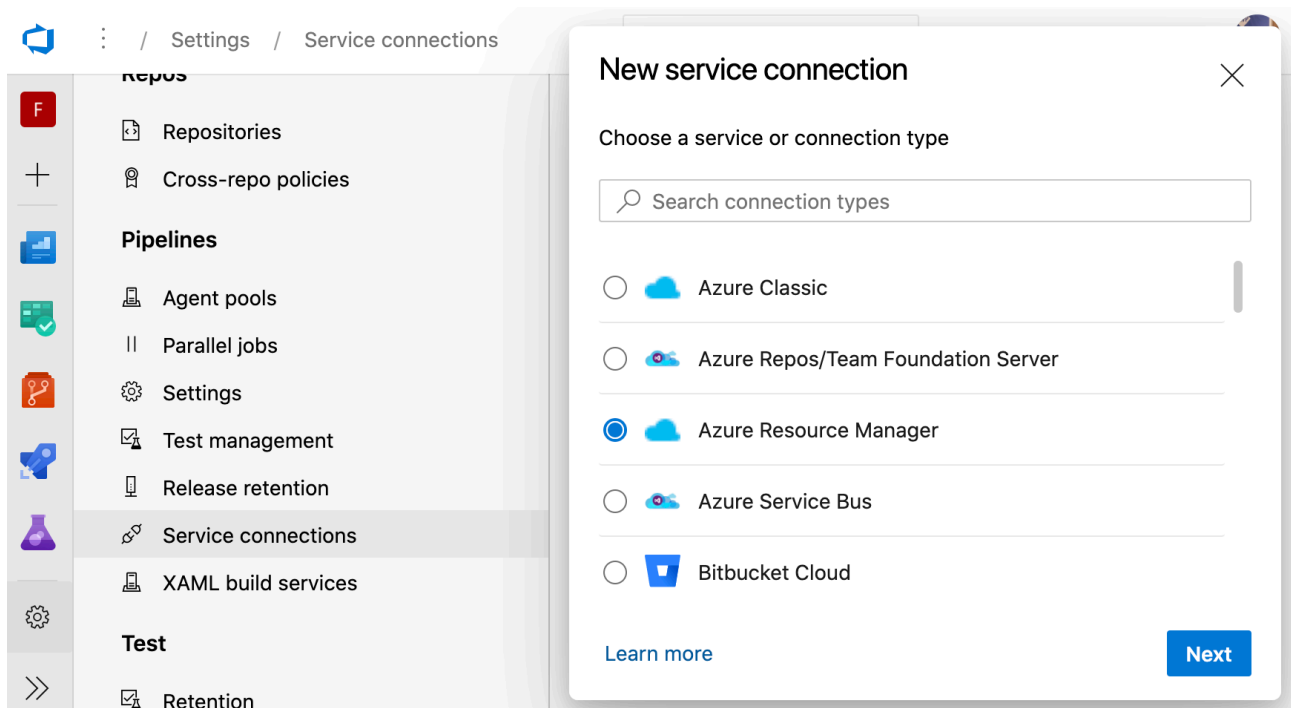
```
$ az sig image-version create -g TestRG03 --gallery-name gallery01  
--gallery-image-definition vm01ImgDef --gallery-image-version  
1.0.0 --target-regions "eastus" --replica-count 1 --managed-image  
"/subscriptions/8c3bd745-5ec6-46f3-9b52-a455b309bdd6/  
resourceGroups/TestRG03/providers/Microsoft.Compute/  
virtualMachines/vm01"
```

// VMSS 생성

```
$ az vmss create -g TestRG02 -n vmss01 --image UbuntuLTS --  
upgrade-policy-mode automatic --admin-username mingyu --vm-  
sku Standard_ds1_v2 --lb-sku Standard --ssh-key-value ~/.ssh/  
id_vm01.pub
```



## DevOps 에서 프로젝트 생성 후, 아래와 같이 설정.



## New Azure service connection

Azure Resource Manager using service principal (automatic)

Scope level

☒ Subscription  
☐ Management Group  
☐ Machine Learning Workspace

Subscription

Visual Studio Enterprise (8c3bd745-5ec6-46f3-9b52-a45...)

Resource group

Details

Service connection name

Visual Studio Enterprise 2

Description (optional)

Security

☒ Grant access permission to all pipelines

[Learn more](#)  
[Troubleshoot](#)

[Back](#)
[Save](#)

배포를 위한 VMSS 준비 : VMSS 의 메뉴에서 Extensions 메뉴에 들어가서 아래를 선택한다.

Microsoft Azure

Search resources, services, and docs (G+)

[Home](#) > [Resource groups](#) > [TestRG02](#) > [vmss01](#) | [Extensions](#) > [New resource](#) > Azure Pipelines Agent For Linux

### New resource

Agent for Cloud Workload Protection (Linux)  
Symantec Corp.

Stackify Retrace Linux Agent  
Stackify

Agent for Linux Server Monitoring  
Site24x7

SentinelOne Linux Extension  
SentinelOne

Rapid7 Insight Agent  
Rapid7 Inc.

Azure Pipelines Agent For Linux  
Microsoft Corp.

## Azure Pipelines Agent For Linux

Microsoft Corp.

### Configure deployment group agents with the Azure DevOps account

Design and automate release pipelines seamlessly across your environment platform using the **Azure Pipelines service** within **Azure DevOps Services**

**Azure Pipelines Agent** extension can be installed to deploy software using Services to the virtual machines hosted in Microsoft Azure. When installed, it automatically downloads the Azure Pipelines Agent MSI, installs it, and registers it with the specified Deployment Group within your Azure DevOps Organization.

**Deployment groups** help to organize and define permission boundaries across multiple machine agent pools. When you create an Azure Pipelines build or release, you could specify the Deployment Group and choose "tags" to deploy to multiple agent pools in parallel.

Legal TermsBy clicking the Create button, I acknowledge that I am getting terms of Microsoft Corp. and that the [legal terms](#) of Microsoft Corp. apply to it.

Publisher

Microsoft Corp.

Useful Links

[Azure Pipelines](#)  
[Deployment Groups](#)

## // 아래와 같이 설정

Home > Resource groups > TestRG02 > vmss01 | Extensions > New resource > Azure Pipelines Agent For Linux > Install extension

### Install extension

Azure DevOps Organization Url \* ⓘ  
 ✓

Team Project \* ⓘ  
 ✓


Deployment Group \* ⓘ  
 ✓







Agent Name ⓘ

Personal Access Token \* ⓘ  
 ✓

Tags ⓘ




## // Deployment Groups 에서의 Available pools 설정

 codingstar / FirstVMSS / Pipelines / Deployment groups

### Deployment groups

Groups   **Available pools** 11

Deployment pool		Target status
 environment-14-482b7d65-d161-4440-a...	+	1 Offline
 environment-15-819d7fbe-f440-4adb-9c...	+	1 Offline
 <a href="#">FirstVMSS-FirstVMSSGroup</a>	+	

## Organization Settings

codingstar

Search Settings

## General

- Overview
- Projects
- Users
- Billing
- Auditing
- Global notifications
- Usage
- Extensions
- Azure Active Directory

## Security

Policies

## Deployment pools &gt; environment-14-482b7d65-d161-444...

Details Targets 1 Save Security

Name

environment-14-482b7d65-d161-4440-a2d6-876baf455855

Type of target to register:

Linux

System

Registration script

```
mkdir azagent;cd azagent;curl -fsSL https://vstsagentpackage.azureedge.net/2.170.1/tar.gz;tar -zxvf vstsagent.2.170.1.tar.gz;./config.sh --deploymentpool --deploymentpoolid "a2d6-876baf455855" --acceptteeeula https://dev.azure.com/codingstar/ --url https://dev.azure.com/codingstar/ "environment-14-482b7d65-d161-4440-a2d6-876baf455855" --url https://dev.azure.com/codingstar/
```

☐ Use a personal access token in the script for authentication

Copy script to the clipboard

Save

Copy to clipboard