Masaa Learning

Go Zero to One

MLOps with Devops

Index

Pre-requisites

You need an **Agent** to run Azure Devops pipeline. An **Agent** is a service to run jobs defined in the pipeline. Essentially you have 2 types of agents

- **Microsoft Hosted Agent:** Agents that are managed by Microsoft. It requires at least 2-3 business day to get provisioned
- **Self Hosted Agent :** Making our own machine as agent

Steps To Configure Self Hosted Agent

- Install Visual Studio Code (VS code) in your Local Machine (Laptop/Desktop)
- Install Python in your Local Machine
- Install Git in your Local Machine
- Configure Git with Username and Email to commit changes
  - Open Git Bash
  - Type command
    - git config --global user.name ”your name”
    - git config –global user.email “your email id”

- Add Path to Environment Variables if not already added (see here)
  - Bash (path: *C:\Program Files\Git*)
  - Pip (path: *C:\Users\moham\AppData\Local\Programs\Python\Python310\Scripts*)
  - Python (path: *C:\Users\moham\AppData\Local\Programs\Python\Python310*)

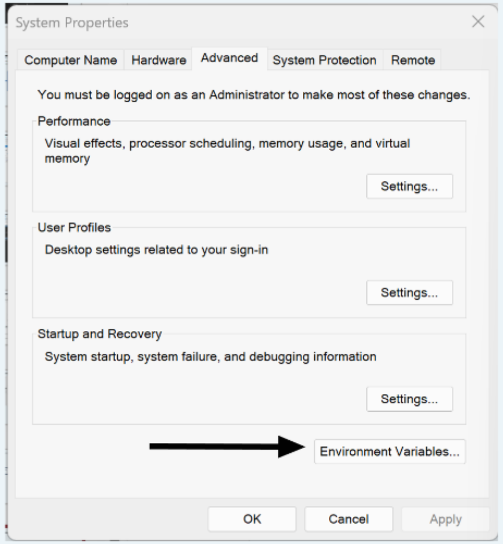
- Set your Execution Policy to **Unrestricted** if it is **Restricted** to unzip extracted folders while installing Self-Hosted Agent (or you can manually extract it)
  - Open your PowerShell as Administrator
  - To check Execution policy type command: **Get-ExecutionPolicy**
  - If it is Restricted type command:
    Set-ExecutionPolicy -ExecutionPolicy RemoteSigned or
    set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy Unrestricted –Force

- Install Self-Hosted-Agent if Azure Host Agents are not provisioned (see here)

Adding Path to Environment Variables

- Search for environment variables on your system
- Click on **Environment variables**
- Under **system variables** section, select **path** and click **edit**
- Click on **New** – add your path and click **ok**

1. Go to Environment Variables


System Properties

Computer Name | Hardware | Advanced | System Protection | Remote

You must be logged on as an Administrator to make most of these changes.

Performance

Visual effects, processor scheduling, memory usage, and virtual memory

Settings...

User Profiles

Desktop settings related to your sign-in

Settings...

Startup and Recovery

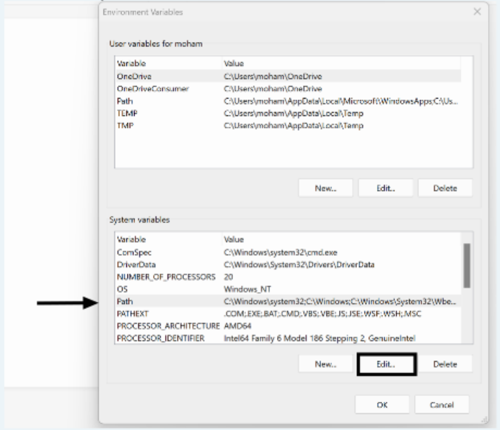
System startup, system failure, and debugging information

Settings...

Environment Variables...

OK | Cancel | Apply

2. Under System variables - select Path - click edit


Environment Variables

User variables for moham

Variable	Value
OneDrive	C:\Users\moham\OneDrive
OneDriveConsumer	C:\Users\moham\OneDrive
Path	C:\Users\moham\AppData\Local\Microsoft\WindowsApps;C:\Us...
TEMP	C:\Users\moham\AppData\Local\Temp
TMP	C:\Users\moham\AppData\Local\Temp

New.. | Edit.. | Delete

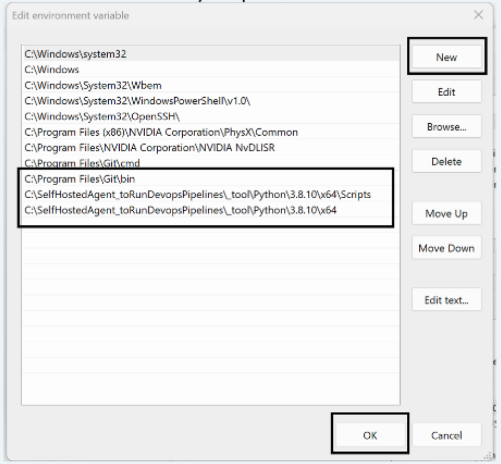
System variables

Variable	Value
ComSpec	C:\Windows\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS	20
OS	Windows_NT
Path	C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbe...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64
PROCESSOR_IDENTIFIER	Intel64 Family 6 Model 186 Stepping 2, GenuineIntel

New.. | Edit.. | Delete

OK | Cancel

3. Click on New - add your path - click Ok


Edit environment variable

```
C:\Windows\system32
C:\Windows
C:\Windows\System32\Wbem
C:\Windows\System32\WindowsPowerShell\v1.0\
C:\Windows\System32\OpenSSH\
C:\Program Files (x86)\NVIDIA Corporation\PhysX\Common
C:\Program Files\NVIDIA Corporation\NVIDIA NvDLISR
C:\Program Files\Git\cmd
C:\Program Files\Git\bin
C:\SelfHostedAgent_toRunDevopsPipelines\_tool\Python\3.8.10\x64\Scripts
C:\SelfHostedAgent_toRunDevopsPipelines\_tool\Python\3.8.10\x64
```

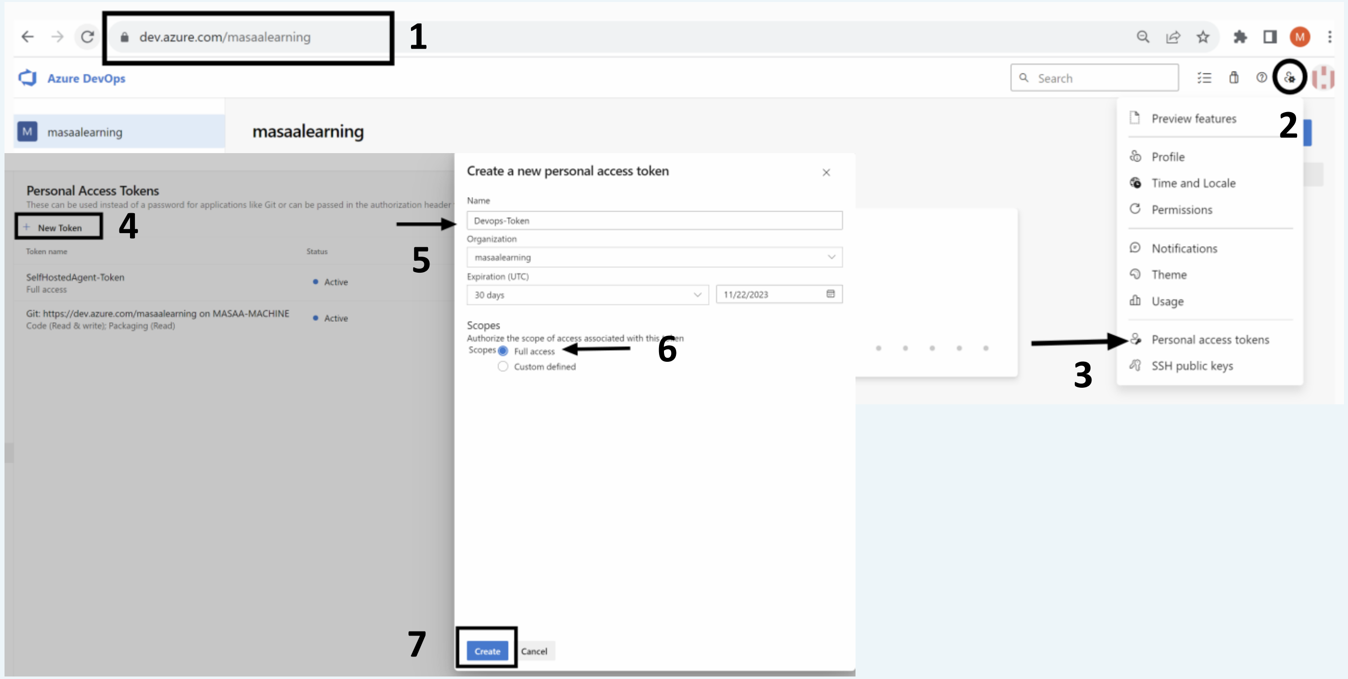
New | Edit | Browse... | Delete | Move Up | Move Down | Edit text...

OK | Cancel

Install Self-Hosted-Agent

1. Generate Personal Access Token (see here)
   - Goto your Devops Organization page(**Server URL**) https://dev.azure.com/masaalearning (it would be the URL of your Devops home page)
   - Go to **User Settings**
   - Click on **Personal access token**
   - Click on **New Token** – Give **Name** – Select **Full Access** – click on **Create**

   Note: copy token and store it you wouldn't be able to view it once you exit the page



Token name | Status
SelfHostedAgent-Token, Full access | Active
Git: https://dev.azure.com/masaalearning on MASAA-MACHINE, Code (Read & write); Packaging (Read) | Active

Create a new personal access token

Name: Devops-Token

Organization: masaalearning

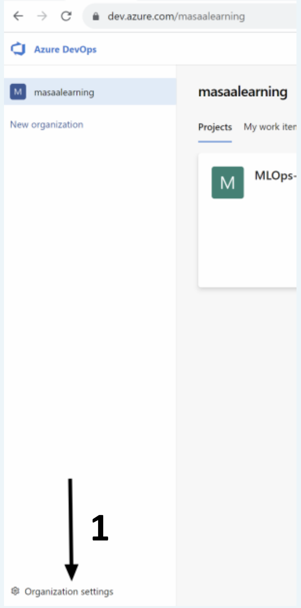
Expiration (UTC): 30 days, 11/22/2023

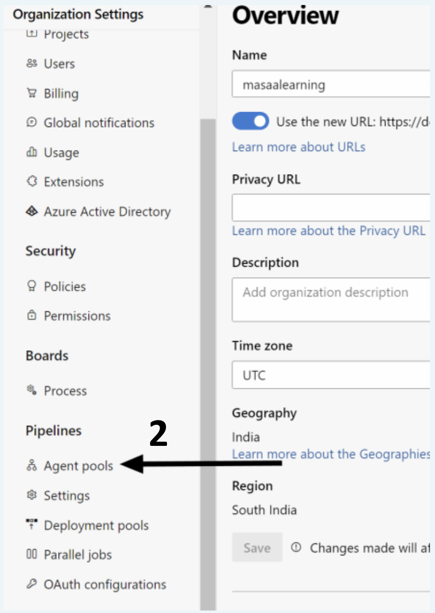
Scopes: Authorize the scope of access associated with this token

Scopes: [✓ Full access] Custom defined

2. Configure Self Hosted Agent

- Go to Organization Setting
- In Left Menu under Pipelines section – click **Agent pools**
- Select Default
- Click on New agent
- A window opens – download the agent and follow the instructions to create a agent
- During installation it prompts for
  - **Server URL:** your Devops home page url
  - **Personal Access Token (PAT) :** paste the token created in previous step
  - **Work folder:** any folder in your local system (laptop/desktop) where you want to save your Devops work
  - Keep all other prompts Default (just click Enter)





Name: masaalearning

Use the new URL: https://d

Learn more about URLs

Privacy URL:

Learn more about the Privacy URL

Description: Add organization description

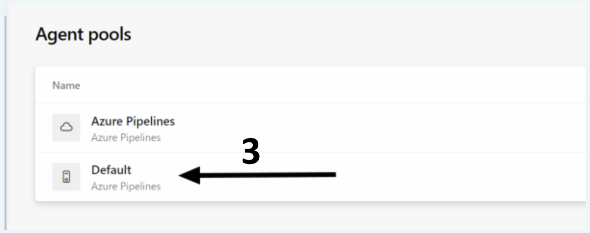
Time zone: UTC

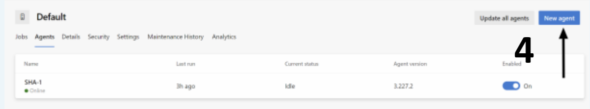
Geography: India

Learn more about the Geographies

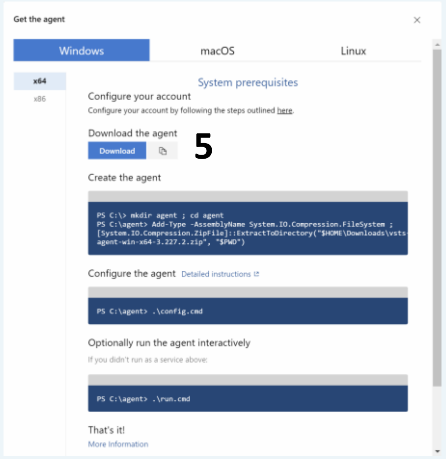
Region: South India

Save   Changes made will a





Name	Last run	Current status	Agent version	Enabled
SHA-1 Online	3h ago	Idle	3.227.2	On



Configure your account

Configure your account by following the steps outlined here.

Download the agent

Create the agent

```
PS C:\> mkdir agent ; cd agent
PS C:\agent> Add-Type -AssemblyName System.IO.Compression.FileSystem ;
[System.IO.Compression.ZipFile]::ExtractToDirectory("$HOME\Downloads\vsts-
agent-win-x64-3.227.2.zip", "$PWD")
```

Configure the agent Detailed instructions

```
PS C:\agent> .\config.cmd
```

Optionally run the agent interactively

If you didn't run as a service above:

```
PS C:\agent> .\run.cmd
```

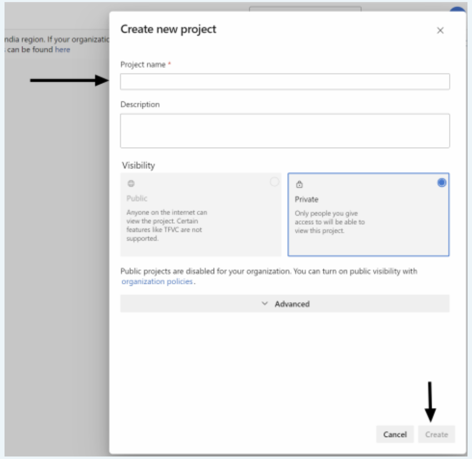
That's it!

More Information

Setting up Project

1. Create project
2. Get your Code into Azure Repos
   - If you code is in your local machine click on **Clone in VS Code**
     - It prompts you to open VS Code
     - It prompts you to select Destination Folder (select any folder where you want to save your code)
     - Copy your code in the Folder
     - Commit & Push your code
   - If your code is in GitHub repo click on **Import**
3. Create Service Connection to connect to external services from Azure Devops (See here)

1. Create Project



ndia region. If your organizatic
can be found here

Create new project

Project name *

Description

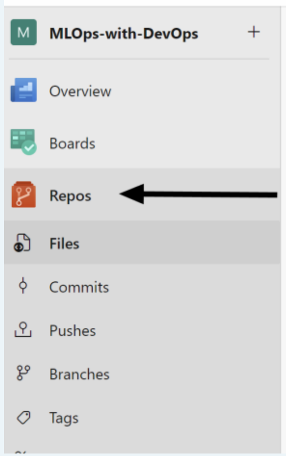
Visibility

Public
Anyone on the internet can view the project. Certain features like TFVC are not supported.

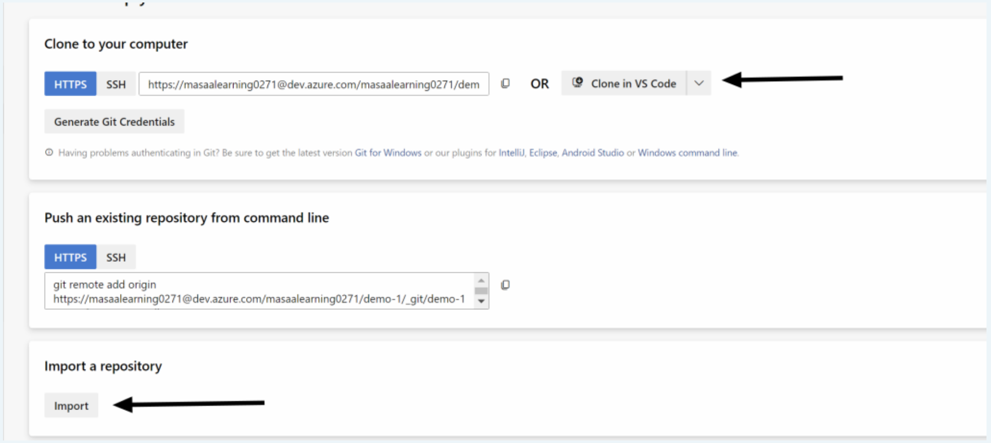
[✓ Private]
Only people you give access to will be able to view this project.

Public projects are disabled for your organization. You can turn on public visibility with organization policies.

2. Open your project – click on **Repos**



3. Clone your code to Repos either from your local system or GitHub



Clone to your computer

HTTPS SSH https://masaalearning0271@dev.azure.com/masaalearning0271/dem OR Clone in VS Code

Generate Git Credentials

Having problems authenticating in Git? Be sure to get the latest version Git for Windows or our plugins for IntelliJ, Eclipse, Android Studio or Windows command line.

Push an existing repository from command line

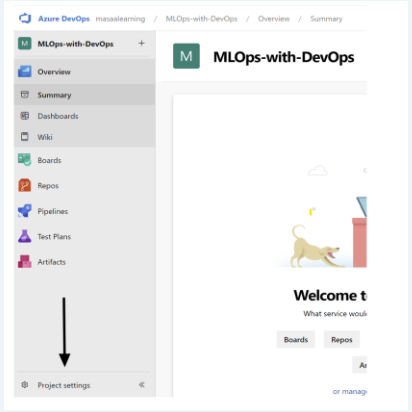
HTTPS SSH

```
git remote add origin
https://masaalearning0271@dev.azure.com/masaalearning0271/demo-1/_git/demo-1
```

Import a repository

Creating Service Connection

1. Go to **Project Setting**



Azure DevOps masaalearning / MLOps-with-DevOps / Overview / Summary

MLOps-with-DevOps +

- Overview
- Summary
- Dashboards
- Wiki
- Boards
- Repos
- Pipelines
- Test Plans
- Artifacts
- Project settings «

M MLOps-with-DevOps

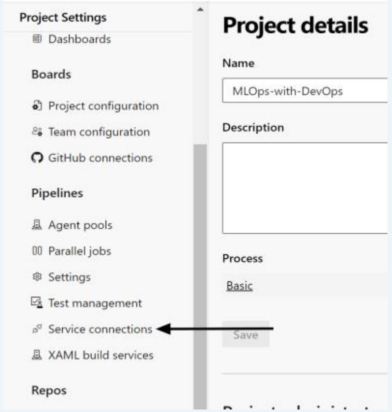
Welcome t

What service woul

Boards Repos

or manag

2. click **Service Connections**



Project Settings

- Dashboards

Boards

- Project configuration
- Team configuration
- GitHub connections

Pipelines

- Agent pools
- Parallel jobs
- Settings
- Test management
- Service connections
- XAML build services

Repos

Project details

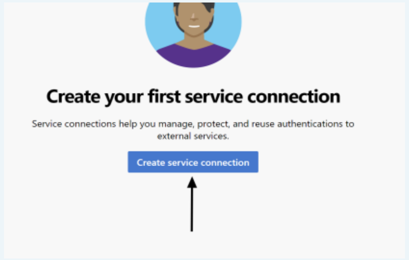
Name: MLOps-with-DevOps

Description

Process: Basic

Save

3. click **Create service connection**

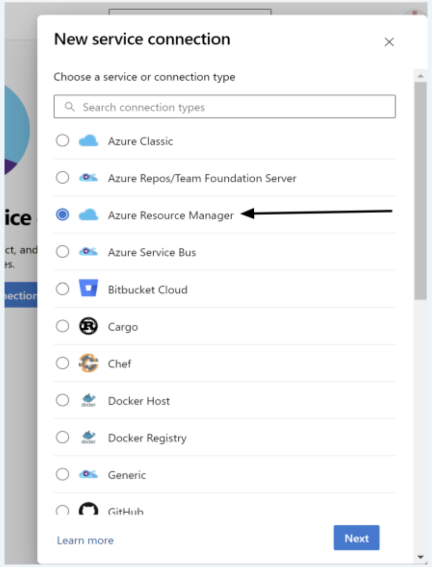


Create your first service connection

Service connections help you manage, protect, and reuse authentications to external services.

Create service connection

4. select **Azure Resource Manager**



New service connection ×

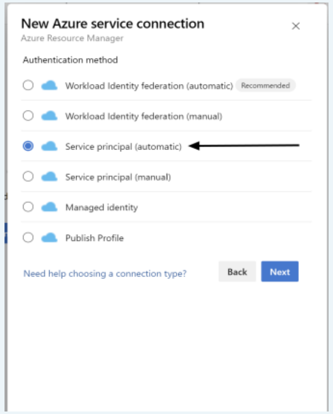
Choose a service or connection type

Search connection types

- ○ Azure Classic
- ○ Azure Repos/Team Foundation Server
- ● Azure Resource Manager
- ○ Azure Service Bus
- ○ Bitbucket Cloud
- ○ Cargo
- ○ Chef
- ○ Docker Host
- ○ Docker Registry
- ○ Generic
- ○ GitHub

Learn more Next

5. select **Service Principal (automatic)**



New Azure service connection ×

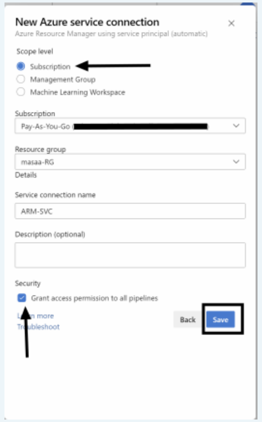
Azure Resource Manager

Authentication method

- ○ Workload Identity federation (automatic) Recommended
- ○ Workload Identity federation (manual)
- ● Service principal (automatic)
- ○ Service principal (manual)
- ○ Managed identity
- ○ Publish Profile

Need help choosing a connection type? Back Next

6. select **Subscription**



New Azure service connection ×

Azure Resource Manager using service principal (automatic)

Scope level

- ● Subscription
- ○ Management Group
- ○ Machine Learning Workspace

Subscription: Pay-As-You-Go [redacted]

Resource group: masaa-RG

Details

Service connection name: ARM-SVC

Description (optional)

Security

☑ Grant access permission to all pipelines

Learn more
Troubleshoot

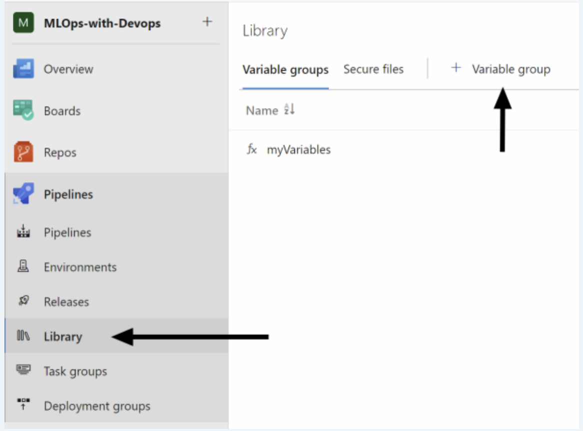
Back Save

Building Continuous Integration (CI) Pipeline

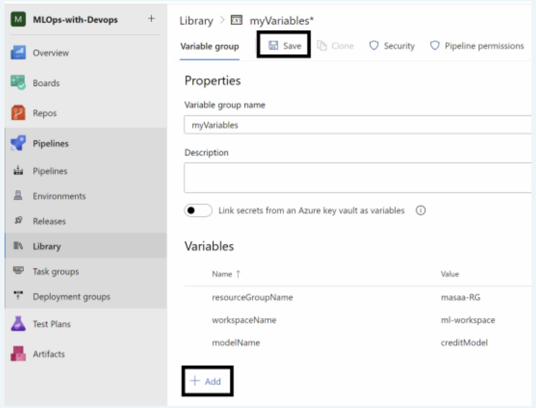
1. Create Variables

- Goto Library under Pipelines Section
- Create your variables & save it
- Link variables to your pipeline
  - Go to your CI pipeline
  - Under **Variables Tab**
  - Select **Variable groups**
  - Click on **Link variable group**
  - Select your variable group (myVariables)
  - Click on **Link**

1. Goto Library



2. Create your variables

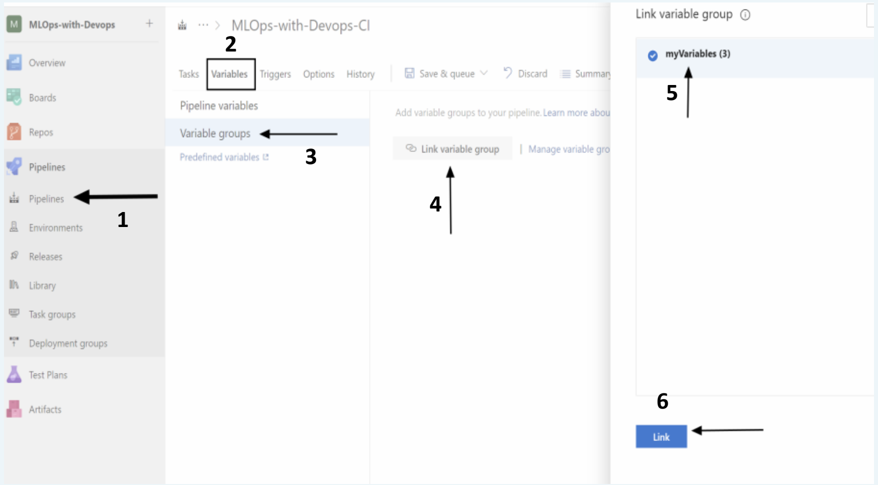


Variable group name: myVariables

Description:

Name	Value
resourceGroupName	masaa-RG
workspaceName	ml-workspace
modelName	creditModel

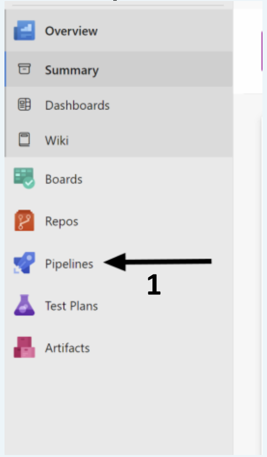
3. Link Variables to Pipeline



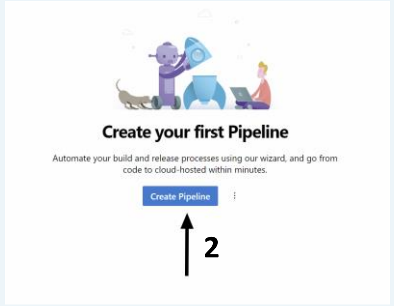
Masaa Learning

2. Building Continuous Integration (CI) Pipeline

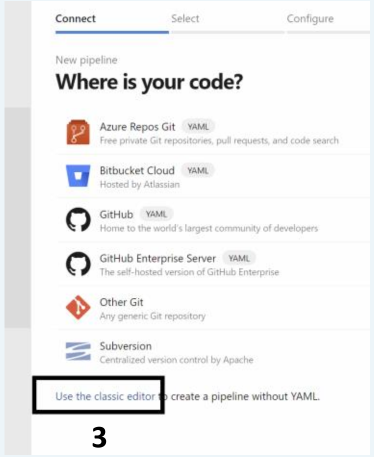
1. Goto **Pipelines**


Overview
Summary
Dashboards
Wiki
Boards
Repos
Pipelines
Test Plans
Artifacts
1

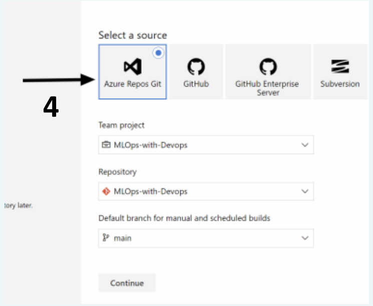
2. Create **pipeline**


Create your first Pipeline
Automate your build and release pipelines using our wizard, and go from code to cloud-hosted within minutes.
Create Pipeline
2

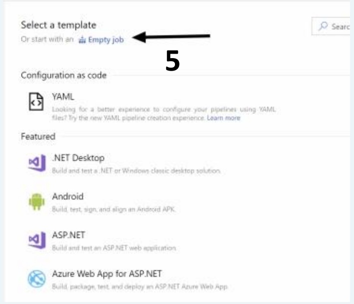
3. Click on **Use Classic Editor**


Connect Select Configure
New pipeline
Where is your code?
Azure Repos Git YAML
Free private Git repositories, pull requests, and code search
Bitbucket Cloud YAML
Hosted by Atlassian
GitHub YAML
Home to the world's largest community of developers
GitHub Enterprise Server YAML
The self-hosted version of GitHub Enterprise
Other Git
Any generic Git repository
Subversion
Centralized version control by Apache
Use the classic editor to create a pipeline without YAML.
3

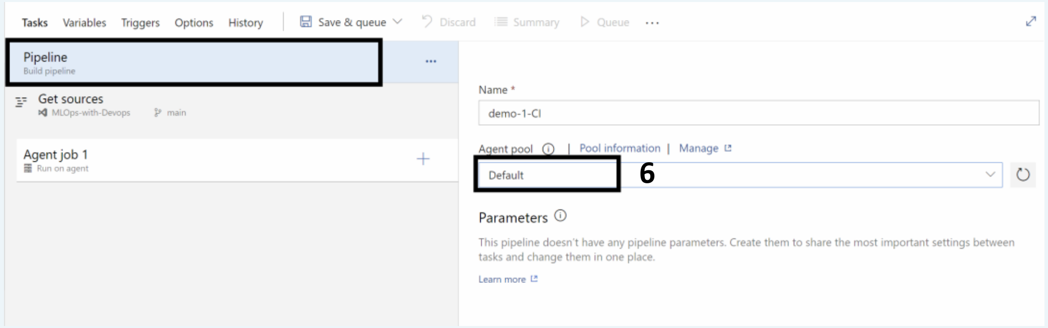
4. Select **Azure Repos Git** – click **Continue**


Select a source
Azure Repos Git | GitHub | GitHub Enterprise Server | Subversion
Team project: MLOps-with-Devops
Repository: MLOps-with-Devops
Default branch for manual and scheduled builds: main
Continue
4

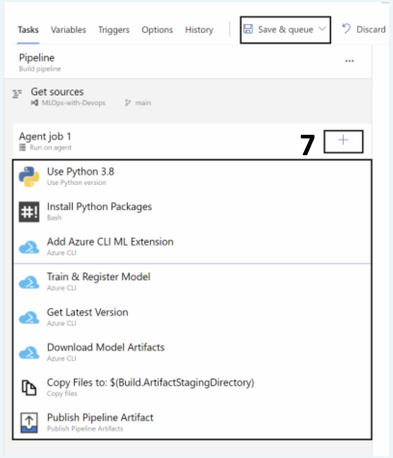
5. Select **Empty Job**


Select a template
Or start with an Empty job
5
Configuration as code
YAML
Looking for a better experience to configure your pipelines using YAML files? Try the new YAML pipeline creation experience. Learn more
Featured
.NET Desktop
Build and test a .NET or Windows classic desktop solution.
Android
Build, test, sign, and align an Android APK.
ASP.NET
Build and test an ASP.NET web application.
Azure Web App for ASP.NET
Build, package, test, and deploy an ASP.NET Azure Web App.

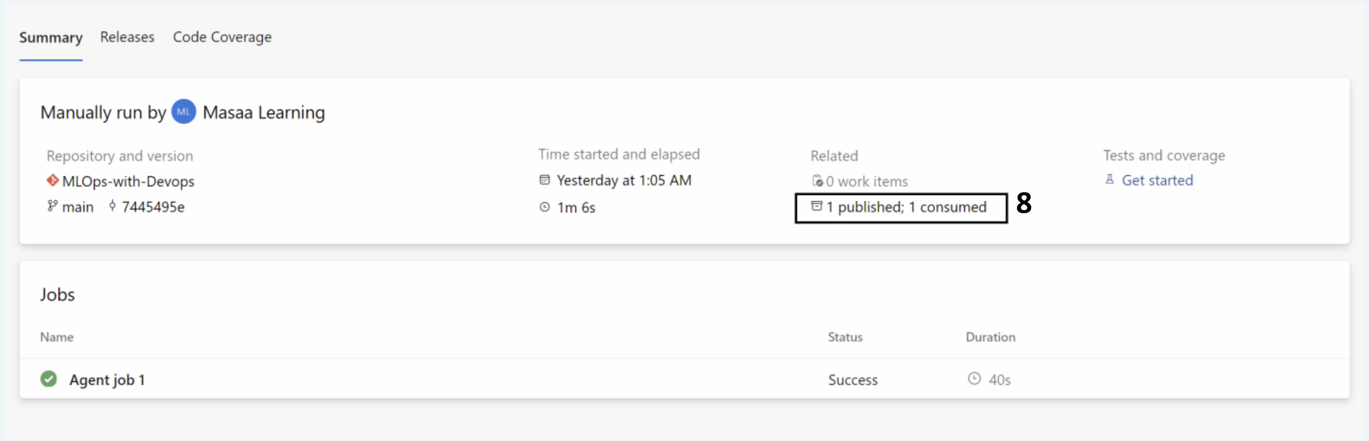
6. Select **Pipeline** – under **Agent pool** – Select **Default**


Tasks Variables Triggers Options History | Save & queue | Discard | Summary | Queue | ...
Pipeline
Build pipeline
Get sources
MLOps-with-Devops main
Agent job 1
Run on agent
Name *
demo-1-CI
Agent pool | Pool information | Manage
Default
6
Parameters
This pipeline doesn't have any pipeline parameters. Create them to share the most important settings between tasks and change them in one place.
Learn more

7. Click on **+ Sign** – create your pipeline Steps – click **Save & Queue**


Tasks Variables Triggers Options History | Save & queue | Discard
Pipeline
Build pipeline
Get sources
MLOps-with-Devops main
Agent job 1
Run on agent
7 +
Use Python 3.8
Use Python version
Install Python Packages
Bash
Add Azure CLI ML Extension
Azure CLI
Train & Register Model
Azure CLI
Get Latest Version
Azure CLI
Download Model Artifacts
Azure CLI
Copy Files to: $(Build.ArtifactStagingDirectory)
Copy files
Publish Pipeline Artifact
Publish Pipeline Artifacts

8. Once the pipeline runs successfully, view the published artifacts


Summary Releases Code Coverage

Manually run by ML Masaa Learning

Repository and version
MLOps-with-Devops
main 7445495e

Time started and elapsed
Yesterday at 1:05 AM
1m 6s

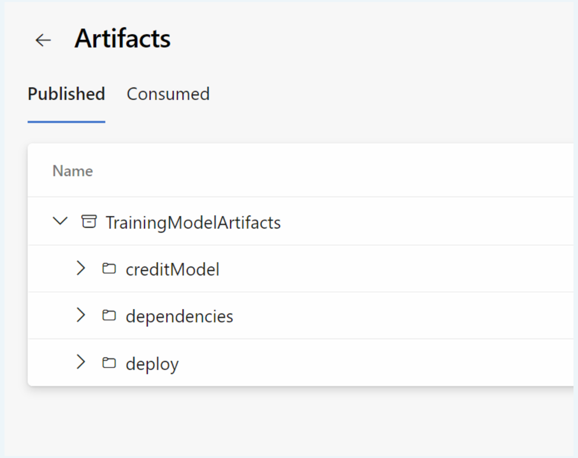
Related
0 work items
1 published; 1 consumed

8

Tests and coverage
Get started

Jobs

Name	Status	Duration
Agent job 1	Success	40s


Artifacts

Published Consumed

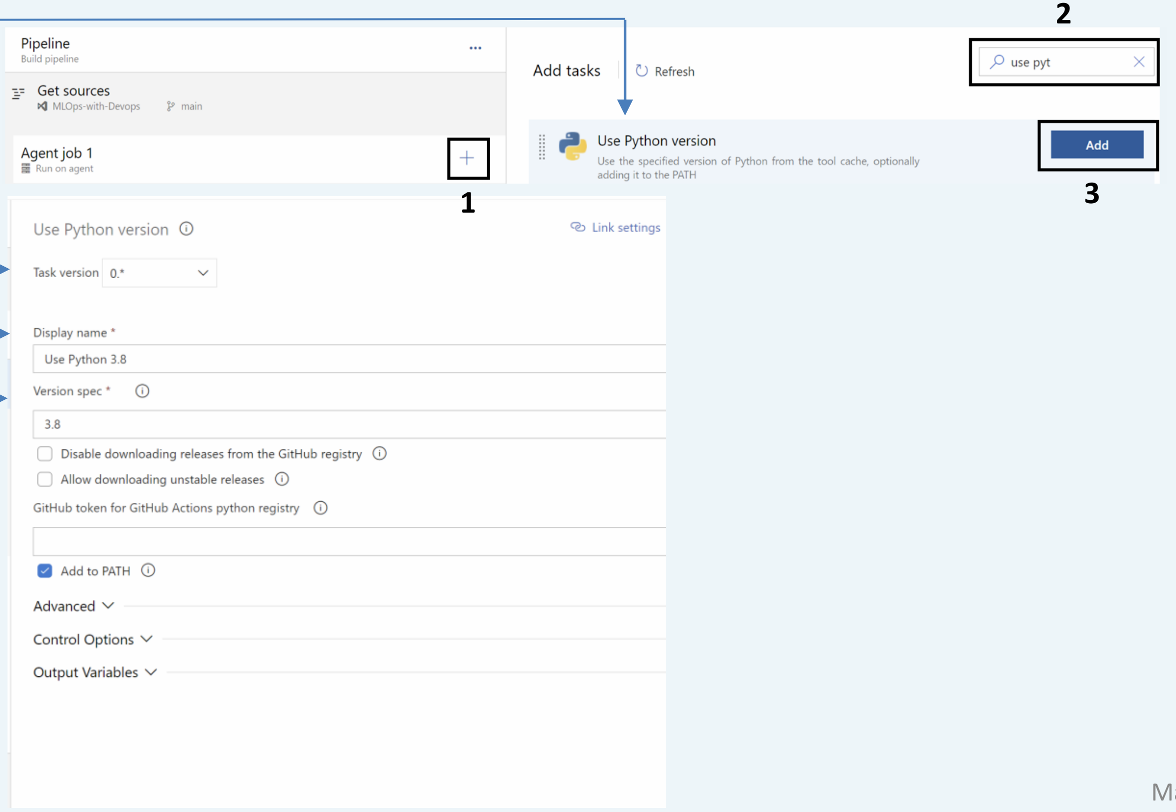
Name
TrainingModelArtifacts
creditModel
dependencies
deploy

We’ll use these artifacts in our deployment

2.1 Building Pipeline Steps : Examples

1. Click on + Sign to add Step
2. Search for your task & add
3. Give Inputs to the task

```
steps:
- task: UsePythonVersion@0
  displayName: 'Use Python 3.8’
  inputs:
    versionSpec: 3.8
```



Use Python version

Task version: 0.*

Display name *: Use Python 3.8

Version spec *: 3.8

☐ Disable downloading releases from the GitHub registry

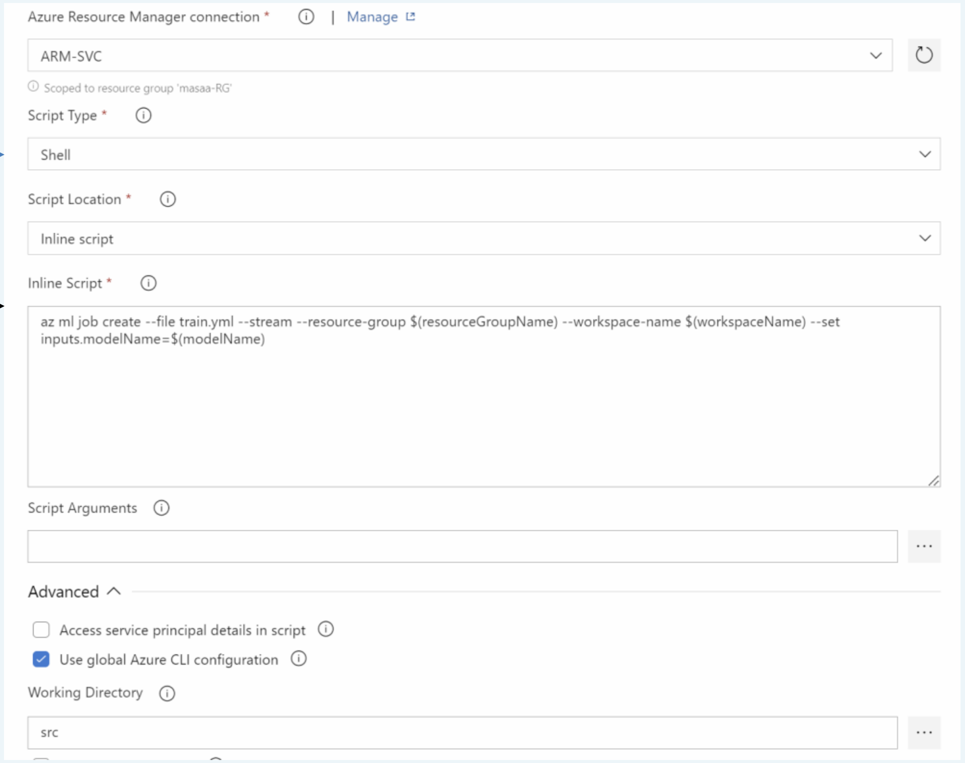
☐ Allow downloading unstable releases

GitHub token for GitHub Actions python registry:

☑ Add to PATH

Masaa Learning

```
steps:
- task: AzureCLI@2
  displayName: 'Train & Register Model'
  inputs:
    azureSubscription: 'ARM-SVC'
    scriptType: bash
    scriptLocation: inlineScript
    inlineScript: 'az ml job create --file train.yml --stream --resource-group
$(resourceGroupName) --workspace-name $(workspaceName) --set
inputs.modelName=$(modelName)'
    useGlobalConfig: true
    workingDirectory: src
  enabled: false
```



Azure Resource Manager connection * | Manage

ARM-SVC

Scoped to resource group 'masaa-RG'

Script Type *

Shell

Script Location *

Inline script

Inline Script *

```
az ml job create --file train.yml --stream --resource-group $(resourceGroupName) --workspace-name $(workspaceName) --set
inputs.modelName=$(modelName)
```

Script Arguments

Advanced

☐ Access service principal details in script

☑ Use global Azure CLI configuration

Working Directory

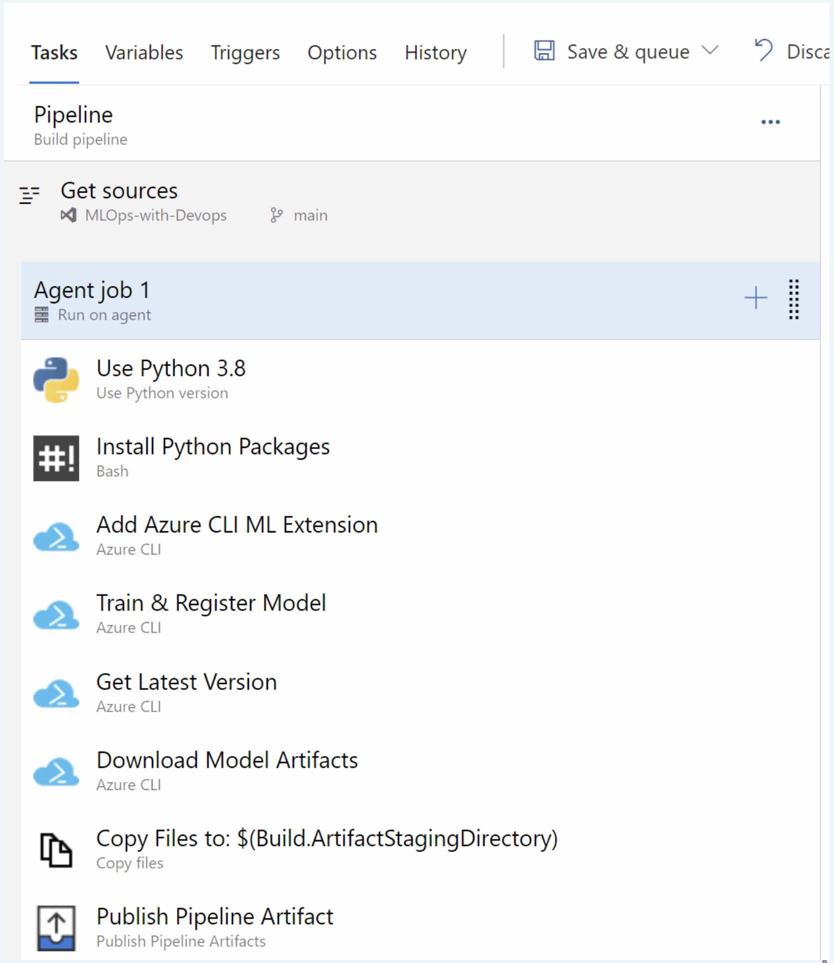
src

2.2 Pipeline Steps

```
 steps:
 - task: UsePythonVersion@0
   displayName: 'Use Python 3.8’
   inputs:
      versionSpec: 3.8
```

```
steps:
- task: Bash@3
  displayName: 'Install Python Packages’
  inputs:
    targetType: filePath
    filePath: './dependencies/install_requirements.sh’
    workingDirectory: dependencies
  enabled: false
```

```
steps:
- task: AzureCLI@2
  displayName: 'Add Azure CLI ML Extension'
  inputs:
    azureSubscription: 'ARM-SVC'
    scriptType: bash
    scriptLocation: inlineScript
    inlineScript: 'az extension add -n ml'
    useGlobalConfig: true
  enabled: false
```



Pipeline
Build pipeline

Get sources
MLOps-with-Devops main

Agent job 1
Run on agent

Use Python 3.8
Use Python version

Install Python Packages
Bash

Add Azure CLI ML Extension
Azure CLI

Train & Register Model
Azure CLI

Get Latest Version
Azure CLI

Download Model Artifacts
Azure CLI

Copy Files to: $(Build.ArtifactStagingDirectory)
Copy files

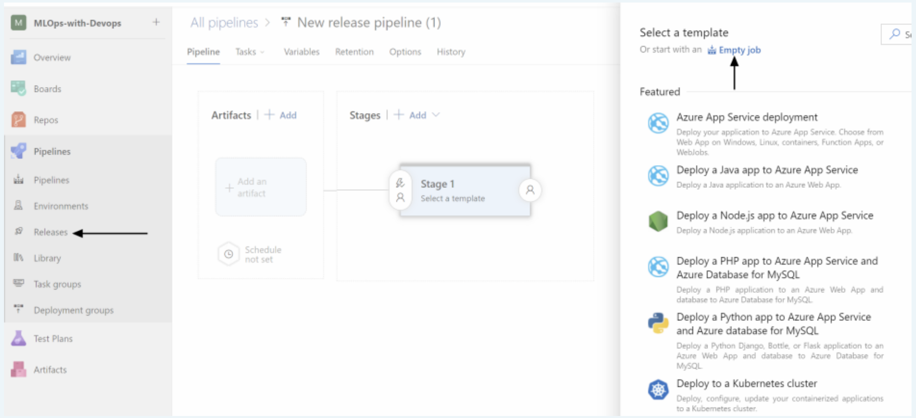
Publish Pipeline Artifact
Publish Pipeline Artifacts

```
steps:
- task: AzureCLI@2
  displayName: 'Train & Register Model'
  inputs:
    azureSubscription: 'ARM-SVC'
    scriptType: bash
    scriptLocation: inlineScript
    inlineScript: 'az ml job create --file train.yml --stream --resource-group $(resourceGroupName) --workspace-name $(workspaceName) --set
inputs.modelName=$(modelName)'
    useGlobalConfig: true
    workingDirectory: src
  enabled: false

steps:
- task: AzureCLI@2
  displayName: 'Get Latest Version'
  inputs:
    azureSubscription: 'ARM-SVC'
    scriptType: ps
    scriptLocation: inlineScript
    inlineScript: |
     $version=$(az ml model show --name $(modelName) --label latest --resource-group $(resourceGroupName) --workspace-name $(workspaceName)
--query version --output tsv)
     Write-Host "##vso[task.setvariable variable=version]$version"
    useGlobalConfig: true
```

```
steps:
- task: AzureCLI@2
  displayName: 'Download Model Artifacts'
  inputs:
    azureSubscription: 'ARM-SVC'
    scriptType: bash
    scriptLocation: inlineScript
    inlineScript: |
     az ml model download --name $(modelName) --version $(version) --download-path . --resource-group $(resourceGroupName) --workspace-name $(workspaceName)

    useGlobalConfig: true

steps:
- task: CopyFiles@2
  displayName: 'Copy Files to: $(Build.ArtifactStagingDirectory)'
  inputs:
    SourceFolder: '$(Build.SourcesDirectory)'
    Contents: |
     **/$(modelName)/*
      **/dependencies/*
      **/deploy/*
    TargetFolder: '$(Build.ArtifactStagingDirectory)'

steps:
- task: PublishPipelineArtifact@1
  displayName: 'Publish Pipeline Artifact'
  inputs:
    targetPath: '$(Build.ArtifactStagingDirectory)'
    artifact: TrainingModelArtifacts
```

Building Continuous Deployment (CD) Pipeline

1. Goto **Releases** under Pipelines Section – create **New Pipeline** – click on **Empty Job**



Select a template

Or start with an Empty job

Featured

Azure App Service deployment
Deploy your application to Azure App Service. Choose from Web App on Windows, Linux, containers, Function Apps, or WebJobs.

Deploy a Java app to Azure App Service
Deploy a Java application to an Azure Web App.

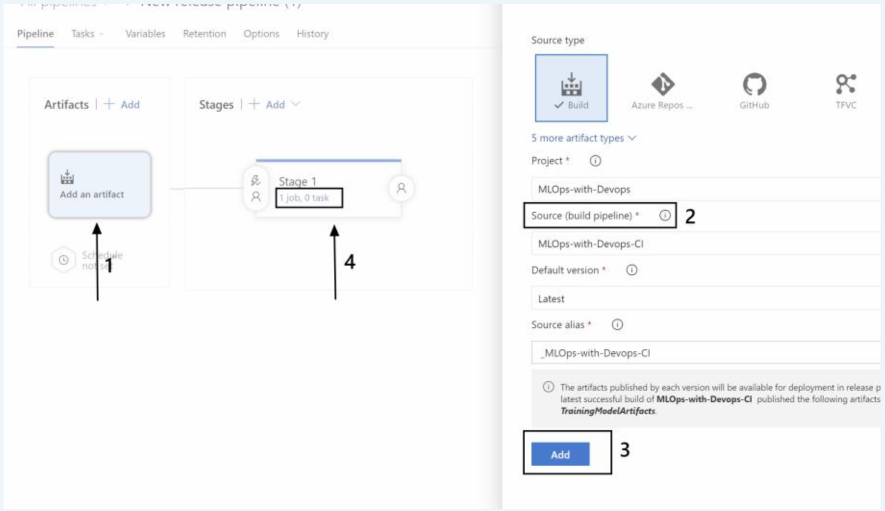
Deploy a Node.js app to Azure App Service
Deploy a Node.js application to an Azure Web App.

Deploy a PHP app to Azure App Service and Azure Database for MySQL
Deploy a PHP application to an Azure Web App and database to Azure Database for MySQL.

Deploy a Python app to Azure App Service and Azure database for MySQL
Deploy a Python Django, Bottle, or Flask application to an Azure Web App and database to Azure Database for MySQL.

Deploy to a Kubernetes cluster
Deploy, configure, update your containerized applications to a Kubernetes cluster.

2. Add **Artifacts** – create **Deployment pipeline steps**



Source type: Build

Project *: MLOps-with-Devops

Source (build pipeline) *: MLOps-with-Devops-CI

Default version *: Latest

Source alias *: _MLOps-with-Devops-CI

The artifacts published by each version will be available for deployment in release p latest successful build of **MLOps-with-Devops-CI** published the following artifacts ***TrainingModelArtifacts***.

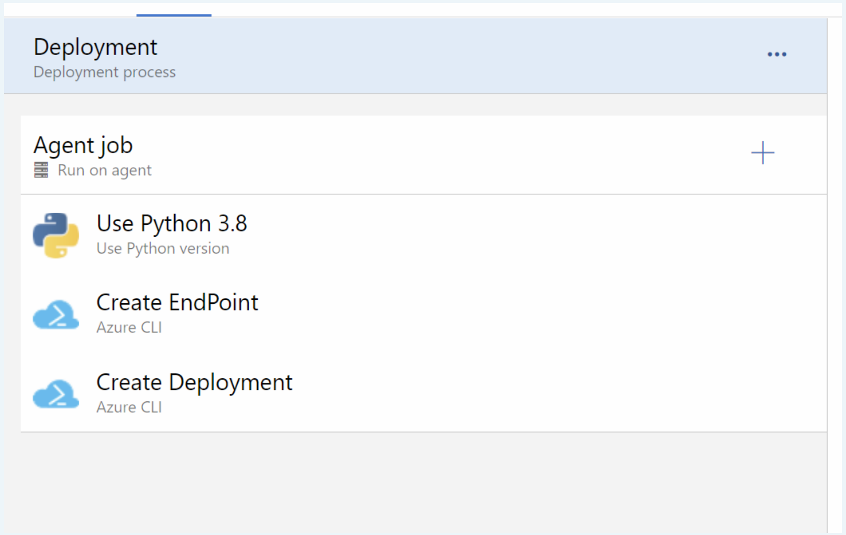
Masaa Learning

3.Deployment Steps

```
 steps:
 - task: UsePythonVersion@0
   displayName: 'Use Python 3.8'
   inputs:
     versionSpec: 3.8
```

```
steps:
- task: AzureCLI@2
  displayName: 'Create EndPoint'
  inputs:
    azureSubscription: 'ARM-SVC'
    scriptType: bash
    scriptLocation: inlineScript
    inlineScript: 'az ml online-endpoint create --file endpoint.yml --resource-group $(resourceGroupName)
--workspace-name $(workspaceName) '
    useGlobalConfig: true
    workingDirectory: '$(System.DefaultWorkingDirectory)/_MLOps-with-Devops-
CI/TrainingModelArtifacts/deploy'
```

```
steps:
- task: AzureCLI@2
  displayName: 'Create Deployment'
  inputs:
    azureSubscription: 'ARM-SVC'
    scriptType: bash
    scriptLocation: inlineScript
    inlineScript: 'az ml online-deployment create --file  deploy.yml --resource-group $(resourceGroupName)
--workspace-name $(workspaceName) --set instance_type=Standard_DS2_v2'
    useGlobalConfig: true
    workingDirectory: '$(System.DefaultWorkingDirectory)/_MLOps-with-Devops-
CI/TrainingModelArtifacts/deploy'
```



Deployment
Deployment process

Agent job
Run on agent

Use Python 3.8
Use Python version

Create EndPoint
Azure CLI

Create Deployment
Azure CLI