

[Open in app](#)

Jack Roper

[Follow](#)

13 Followers

[About](#)

Building an Azure DevOps CI/CD Pipeline for Terraform (Part 1 — Creating the Pipeline)

[Jack Roper](#) Oct 29, 2020 · 3 min read

This post will detail how to create a pipeline for Terraform using Azure DevOps! In [part 2](#), I'll show you how to create the 'release' stage.

Knowing how to build a CI/CD (Continuous integration / Continuous development) pipeline is a core function of the modern DevOps Engineer. A CI/CD pipeline allows you to automate your builds and deployments so you spend less time with the nuts and bolts and more time being creative.

The goal of creating a CI/CD pipeline is to have the Terraform code be deployed automatically once new code is detected in the master. Checking code in to a branch in the repository will not trigger a release or deployment. This way the code can be amended in a branch, then a pull request can be created, approved and completed, at which point the release and deployment process will be triggered.

Azure Pipelines is a service within Azure DevOps which includes subsections called 'pipelines' and 'releases'. Without further ado, get logged on to Azure DevOps and head to the 'Pipelines' section and create a new pipeline:

Connect

Select

Configure

Review

New pipeline

Where is your code?

[Open in app](#)

 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.](#)


My code is in an Azure DevOps repo, so select that then select the project. Select the starter pipeline, then save.


Head back to the pipelines section, click on your new pipeline, then Edit.


We are aiming to build a pipeline that will end up with multiple stages, as per the screenshot below. You'll need to add in each stage as required to match.


Pipeline


Build pipeline


 Get sources
Networking master


 Agent job 1
Run on agent


 Azure CLI
Azure CLI

 Download secure file
Download secure file

 Download secure file init
Download secure file

 Replace tokens in `**/*.tf`
Replace Tokens

 Terraform init
Run Terraform

 Terraform plan
Run Terraform

[Open in app](#)

Publish Artifact: drop
Publish build artifacts

As an overview, moving down each stage of the screenshot above, this pipeline does the following:

1. Gives the pipeline a name, specifies where it is run and on what type of agent.

Name *

Agent pool * ⓘ | Pool information | Manage ⓘ

Azure Pipelines

Agent Specification *

vs2017-win2016

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](#) ⓘ

Select a source

 Azure Repos Git

 GitHub

 GitHub Enterprise Server

 Subversion

 Bitbucket Cloud

 Other Git

Team project

Repository

Default branch for manual and scheduled builds

master

Clean ⓘ

false

Tag sources ⓘ

☒ Never

☐ On success

[Open in app](#)☒ Report build status ⓘ☐ Checkout submodules ⓘ☐ Checkout files from LFS ⓘ

Agent job ⓘ

[View YAML](#) [Remove](#)

Display name *

Agent job 1

Agent selection ^

Agent pool ⓘ | Manage ⓘ

<inherit from pipeline> ▾ ⓘ

Demands ⓘ

Name	Condition	Value
azureps	exists	

[+ Add](#)

Execution plan ^

Parallelism ⓘ

☒ None ☐ Multi-configuration ☐ Multi-agent

Timeout * ⓘ

0

Job cancel timeout * ⓘ

0

4. The Azure CLI section is added to create a resource group, storage account and container in the Azure subscription so that Terraform can use it as it's back-end to store the state file. The variables in the inline script are specified in the pipeline variable file (see near the end of this post for an example screenshot).

If you are using Terraform cloud or Terraform Enterprise this would not be necessary as the state would be stored there instead, however that would affect the later stages of the pipeline.

Select your Azure subscription here and use commands similar to the below:

Azure CLI ⓘ

[Link settings](#) [View YAML](#) [Remove](#)☐ Task version 1.* ▾

[Open in app](#)

Azure subscription * ⓘ | Manage ⓘ



Script Location * ⓘ

Inline script



Inline Script * ⓘ

```
call az group create --location $(location) --name $(rgname) --tags managedBy=$(managedBy) solutionOwner=$(solutionOwner)
activityName=$(activityName) dataClassification=$(dataClassification) automation=$(automation) costCenter1=$(costCenter1)
costCenter2=$(costCenter1) costCenter3=$(costCenter1)

call az storage account create --name $(statestorage) --resource-group $(rgname) --location $(location) --sku Standard_ZRS --https-only true --
encryption-services blob file queue table

call az storage container create --name $(statecontainer) --account-name $(statestorage)
```

Arguments ⓘ



Advanced ^

- ☐ Access service principal details in script ⓘ
- ☐ Use global Azure CLI configuration ⓘ

Display name *

Download secure file

Secure File * ⓘ

spn.tfvars



Retry Count ⓘ

5

Control Options ^

- ☒ Enabled
- ☐ Continue on error

Timeout * ⓘ

0

Run this task ⓘ

Only when all previous tasks have succeeded



Output Variables ^

Reference name ⓘ

spn

Variables list

spn.secureFilePath ⓘ

Display name *

Download secure file init

[Open in app](#)

Retry Count ⓘ

5

Control Options ^

☒ Enabled☐ Continue on error

Timeout * ⓘ

0

Run this task ⓘ

Only when all previous tasks have succeeded

Output Variables ^

Reference name ⓘ

init

Variables list

init.secureFilePath ⓘ

Run Terraform ⓘ

[Link settings](#)[View YAML](#)[Remove](#)

Task version

2.*

Display name *

Terraform init

Terraform template path * ⓘ

tf

Terraform arguments * ⓘ

init -backend-config="\$(init.secureFilePath)"

Plan file path ⓘ

☒ Install terraform ⓘ

Terraform version ⓘ

0.12.26

☐ Use Azure service principal endpoint ⓘ

Control Options ^

☒ Enabled☐ Continue on error

Timeout * ⓘ

0

Run Terraform ⓘ

[Link settings](#)[View YAML](#)[Remove](#)

[Open in app](#)

Terraform plan

Terraform template path * ⓘ

tf

Terraform arguments * ⓘ

plan -out=tfplan -var-file="\$(spn.secureFilePath)"

Plan file path ⓘ

☒ Install terraform ⓘ

Terraform version ⓘ

0.12.26

☒ Use Azure service principal endpoint ⓘ

Azure Connection Type * ⓘ

Azure Resource Manager

Azure Subscription * ⓘ | Manage ⓘ

☐ Init state in Azure storage ⓘ

Copy files ⓘ

[Link settings](#) [View YAML](#) [Remove](#)

Task version 2.*

Display name *

Copy Files to: \$(Build.ArtifactStagingDirectory)/Networking

Source Folder ⓘ

Contents * ⓘ

**

Target Folder * ⓘ

\$(Build.ArtifactStagingDirectory)/Networking

Advanced ^

- ☐ Clean Target Folder ⓘ
- ☐ Overwrite ⓘ
- ☐ Flatten Folders ⓘ
- ☐ Preserve Target Timestamp ⓘ

Control Options ^

- ☒ Enabled
- ☐ Continue on error

Publish build artifacts ⓘ

[Link settings](#) [View YAML](#) [Remove](#)

Publish Artifact: drop

Path to publish * ⓘ

\$(Build.ArtifactStagingDirectory)

Artifact name * ⓘ

drop

Artifact publish location * ⓘ

Azure Pipelines

Advanced ^

Control Options ^

☒ Enabled

☐ Continue on error

Timeout * ⓘ

0

Run this task ⓘ

Only when all previous tasks have succeeded

TasksVariablesTriggersOptionsRetentionHistorySave & queueDiscardSummaryQueue

Pipeline variables

Variable groups

Predefined variables 12

Name ↑	Value	Settable at queue time
activityName		
automation	CI/CD	
costCenter1		
dataClassification		
key		
location	uksouth	
managedBy		
rgname		
solutionOwner		
statecontainer	tfstate	
statestorage		
system.collectionId		
system.debug	false	<input checked="" type="checkbox"/>
system.definitionId		
system.teamProject		
+ Add		

TasksVariablesTriggersOptionsRetentionHistorySave & queueDiscardSummaryQueue

Continuous integration

Networking

Scheduled

No builds scheduled

Build completion

Networking

☒ Enable continuous integration

☐ Batch changes while a build is in progress

Branch filters

Type

Include

Branch specification

*/master

[Open in app](#)

Originally published at <https://azurelynnot.blogspot.com> on October 29, 2020.

[Azure](#)[Azure Devops](#)[Terraform](#)[Infrastructure As Code](#)[Azure Devops Pipeline](#)[About](#) [Help](#) [Legal](#)

Get the Medium app

