

# **Jack Roper**



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 <u>part</u> 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:

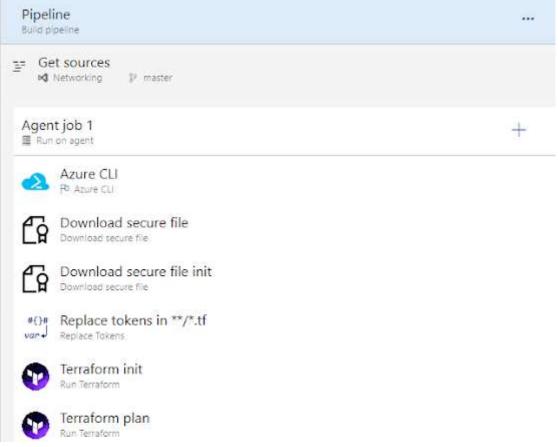




My code is in an Azure DevOps repo, so select that then select the project. Select the starter pipeline, then save.

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.

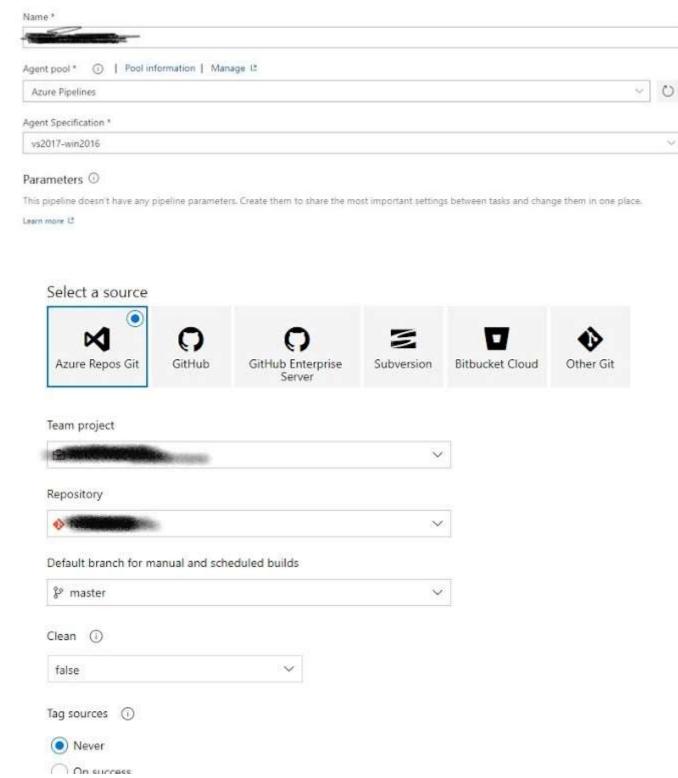


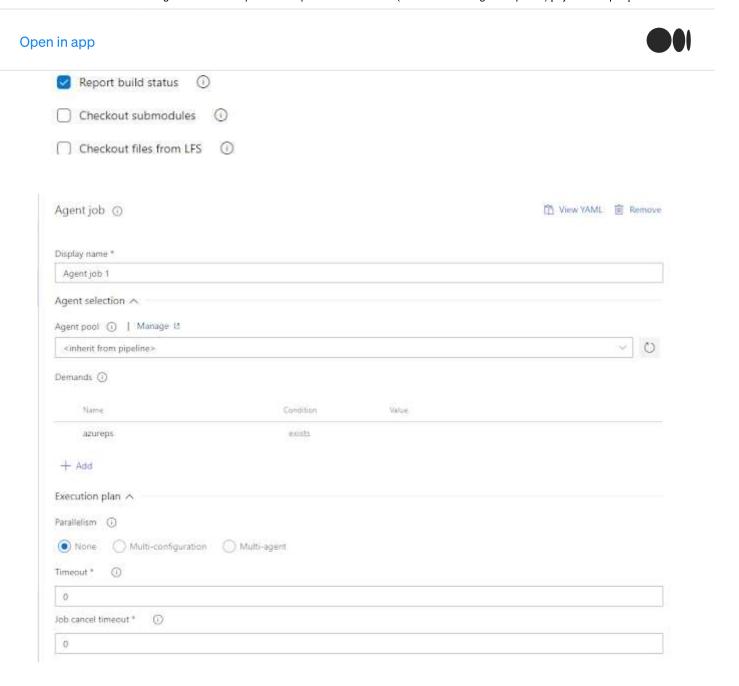




As an overview, moving down each stage of the screenshot above, this pipeline does the following:

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





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:



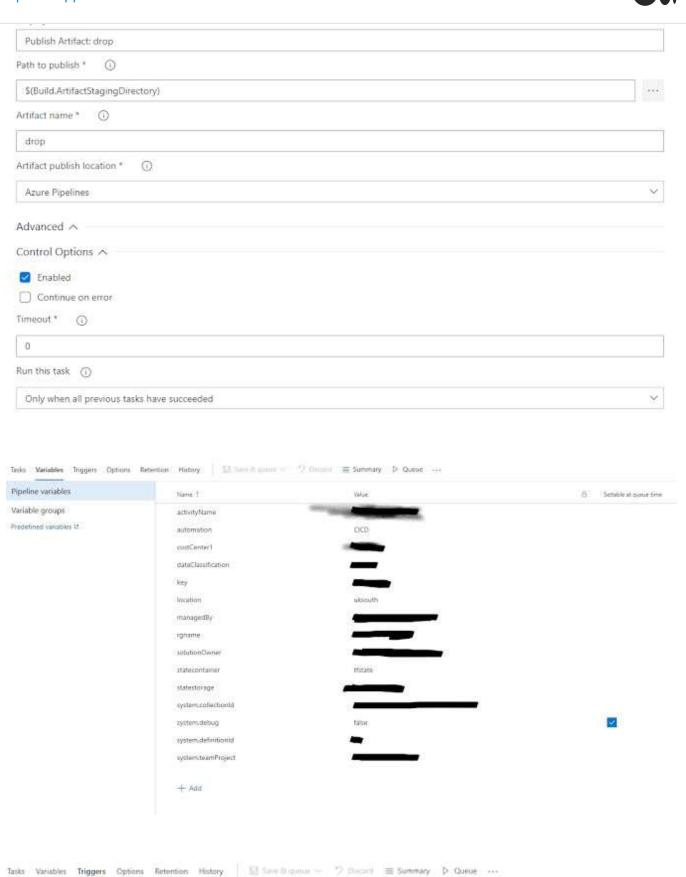
## Open in app Azure subscription -Script Location \* Inline script Inline Script \* 0 call az group create --location \${location} --name \${rgname} --tags managedBy=\${managedBy} solutionOwner=\${solutionOwner} $activity Name = \$(activity Name) \ data Classification = \$(data Classification) \ automation = \$(automation) \ costCenter1 = \$(costCenter1)$ 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 (1) Advanced ^ Access service principal details in script (1) Use global Azure CLI configuration (1) Display name \* Download secure file Secure File \* 0 spn.tfvars Retry Count (1) 5 Control Options ^ Enabled Continue on error Timeout \* (1) 0 Run this task (1) Only when all previous tasks have succeeded Output Variables ^ Reference name (1) spn Variables list 0 spn.secureFilePath Display name \* Download secure file init

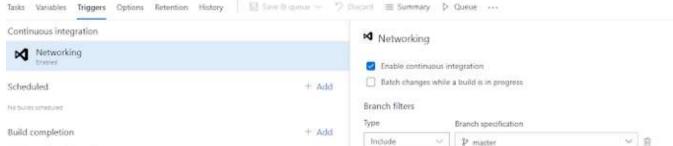
## Open in app Retry Count ① 5 Control Options ^ Enabled Continue on error Timeout \* 0 0 Run this task 🕦 Only when all previous tasks have succeeded Output Variables ^ Reference name (2) init Variables list init.secureFilePath D Link settings T View YAML Remove Run Terraform ① Task version 2,\* Display name \* Terraform init Terraform template path \* tf Terraform arguments \* 0 init -backend-config="\$(init.secureFilePath)" Plan file path (1) Install terraform (1) Terraform version (1) 0.12.26 Use Azure service principal endpoint 3 Control Options ^ Enabled Continue on error Timeout \* 0 0 Co Link settings T View YAML Remove Run Terraform ①



Terraform plan	
Ferraform template path * ①	
SECTION CONTRACTOR OF THE CONT	1
tf	
Terraform arguments * ①	
plan -out=tfplan -var-file="\$(spn.secureFilePath)"	
Plan file path ①	
☑ Install terraform ①	
Terraform version ①	
0.12.26	
Use Azure service principal endpoint ①	
Azure Connection Type * ①	
Azure Resource Manager	
Azure Subscription* ①   Manage @	
	× 1
☐ Init state in Azure storage ③	***!
Copy files ⊙	② Link settings
Copy files ⊙	② Link settings
	♥ Link settings 🔯 View YAML 📵 Remov
Task version 2.5	© Link settings ☐ View YAML
Task version 2.*   Display name *	S Link settings 🛅 View YAML 🗐 Remov
Task version 2.*  Display name *  Copy Files to: \$(Build.ArtifactStagingDirectory)/Networking	Co Link settings (1) View YAML (2) Remov
Task version 2.*  Display name *  Copy Files to: \$(Build.ArtifactStagingDirectory)/Networking	
Task version 2.*  Display name *  Copy Files to: \$(Build.ArtifactStagingDirectory)/Networking  Source Folder ①	C Link settings ( View YAML ( Remov
Task version 2.*  Display name *  Copy Files to: \$(Build.ArtifactStagingDirectory)/Networking  Source Folder ①  Contents * ①	
Task version 2.*  Display name *  Copy Files to: \$(Build.ArtifactStagingDirectory)/Networking  Source Folder ①	
Task version 2.*  Display name *  Copy Files to: S(Build.ArtifactStagingDirectory)/Networking  Source Folder ①  Contents * ①	
Task version 2.*  Display name *  Copy Files to: \$(Build.ArtifactStagingDirectory)/Networking  Source Folder ①  Contents * ①	
Task version 2.*  Display name *  Copy Files to; \${Build.ArtifactStagingDirectory}/Networking}  Source Folder ①  Contents * ①  **  Target Folder * ①  \${Build.ArtifactStagingDirectory}/Networking}	
Task version 2.*  Display name *  Copy Files to: \$(Build,ArtifactStagingDirectory)/Networking  Source Folder ①  Contents * ①  **  Target Folder * ①  \$(Build,ArtifactStagingDirectory)/Networking  Advanced ^	
Task version 2.*  Display name *  Copy Files to: \$(Build.ArtifactStagingDirectory)/Networking  Source Folder ①  Contents * ①  **  Target Folder * ①  \$(Build.ArtifactStagingDirectory)/Networking  Advanced ^  Clean Target Folder ①	
Task version 2.*  Display name *  Copy Files to: \$(Build,ArtifactStagingDirectory)/Networking  Source Folder ①  Contents * ①  **  Target Folder * ①  \$(Build,ArtifactStagingDirectory)/Networking  Advanced ^	
Task version 2.*  Display name *  Copy Files to: \$(Build.ArtifactStagingDirectory)/Networking  Source Folder ①  Contents * ①  **  Target Folder * ①  \$(Build.ArtifactStagingDirectory)/Networking  Advanced ^  Clean Target Folder ①  Overwrite ①	
Display name *  Copy Files to: \${Build.ArtifactStagingDirectory}/Networking  Source Folder ①  Contents * ①  **  Target Folder * ①  \${Build.ArtifactStagingDirectory}/Networking}  Advanced ^  Clean Target Folder ①  Overwrite ①  Flatten Folders ①	
Display name *  Copy Files to: \$(Build.ArtifactStagingDirectory)/Networking  Source Folder ①  Contents * ①  **  Target Folder * ①  \$(Build.ArtifactStagingDirectory)/Networking  Advanced ^  Clean Target Folder ①  Overwrite ①  Flatten Folders ①  Preserve Target Timestamp ①	









Originally published at <a href="https://azurelynot.blogspot.com">https://azurelynot.blogspot.com</a> on October 29, 2020.

Azure Devops Terraform Infrastructure As Code Azure Devops Pipeline

About Help Legal

Get the Medium app



