Pipeline Architecture - FastCarz

Last updated by | Bhavana Singh | Jul 2, 2021 at 5:59 PM GMT+5:30

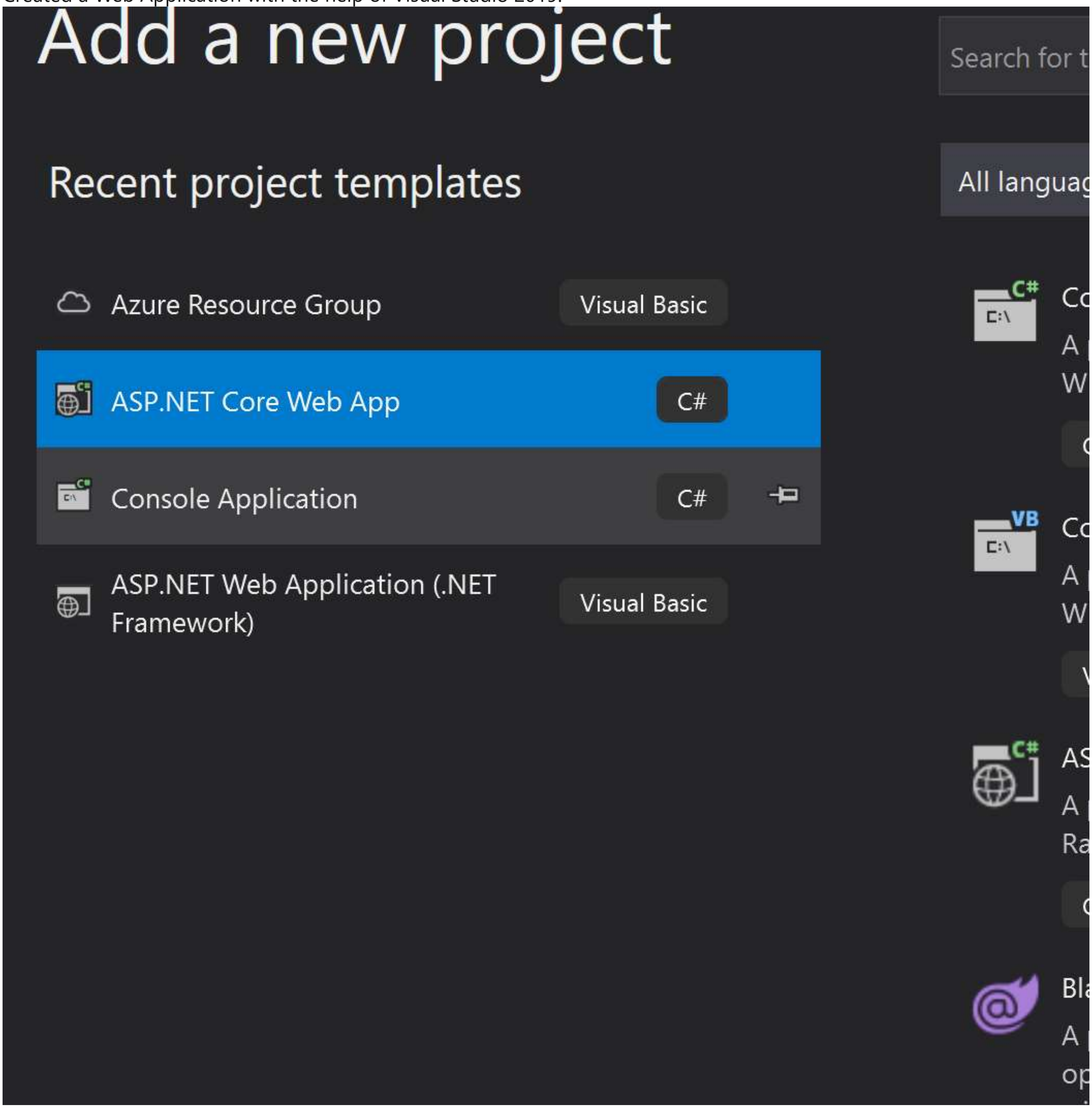
Requirement

1. Migrate on-premise .net Web Application and Web API to Azure Web App Service and Web API Service.
2. Use Azure Git
3. Copy Build Package to Drop Folder
4. Enable Continuous Integration Dev to Master branch. I.e. whenever PR is merged to master, the pipeline will be triggered automatically
5. There will be test projects which will create and maintained in the solution along the Web and API.
   The trigger should build all the 3 projects - Web, API and test.
   The build should not be successful if any test fails.
6. Automate creation of artifacts and code in the drop folder.
7. Use variable group for automating the deployment for Dev, QA and Prod environments. Store the environment variables in the variable group.
8. Use deployment gates for QA and Prod environment.

Steps:

1. Migrate on-premise .net Web Application and Web API to Azure Web App Service and Web API Service.

Created a Web Application with the help of Visual Studio 2019:



Add a new project

Search for t

Recent project templates

All languag

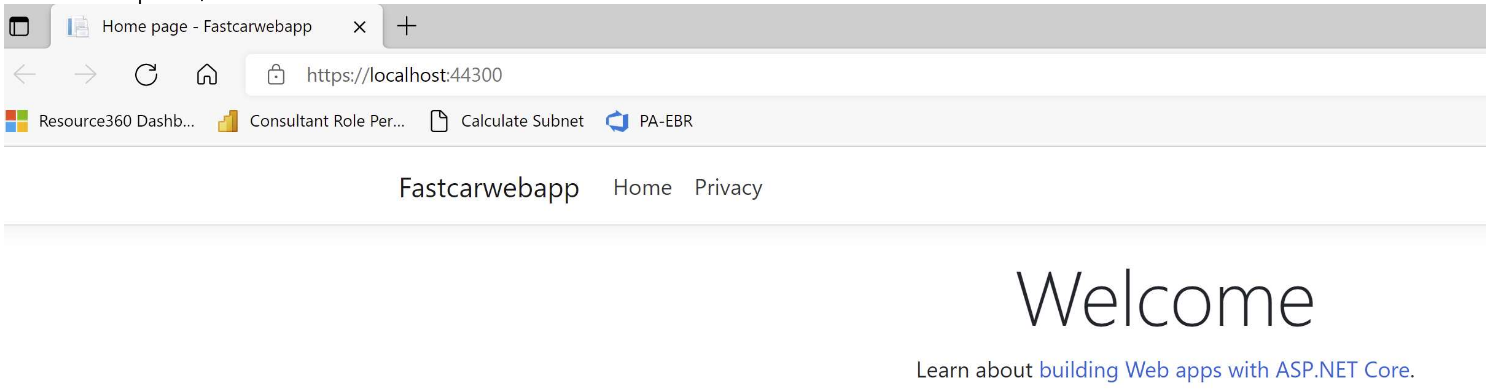
Azure Resource Group — Visual Basic

ASP.NET Core Web App — C#

Console Application — C#

ASP.NET Web Application (.NET Framework) — Visual Basic

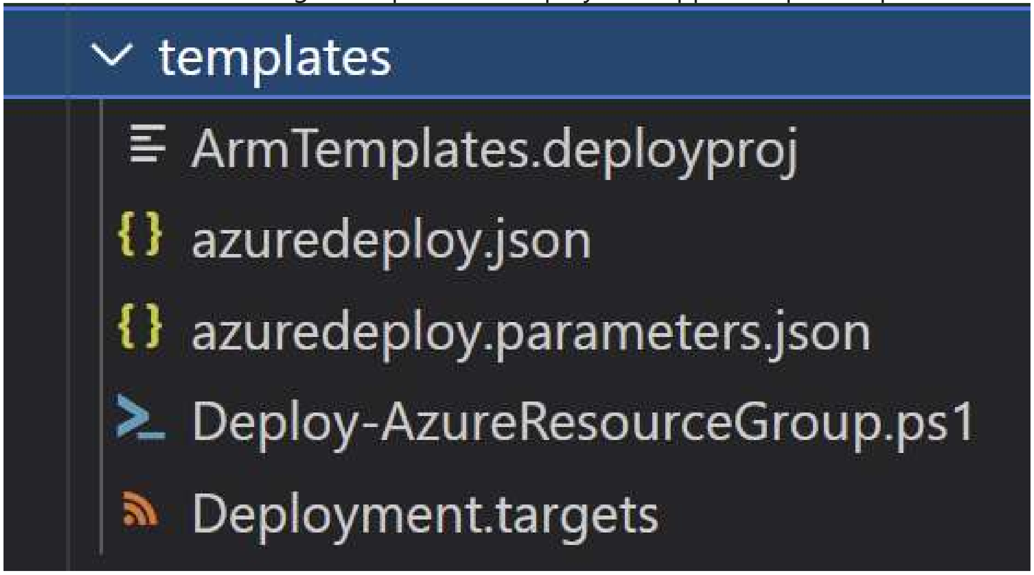
Used IIS express, the web was able to run



Welcome

Learn about building Web apps with ASP.NET Core.

Created Resource Manager Templates for deploy web app and updated parameters file



```
templates
    ArmTemplates.deployproj
    azuredeploy.json
    azuredeploy.parameters.json
    Deploy-AzureResourceGroup.ps1
    Deployment.targets
```

2. Use Azure Git

Used Azure Repo for storing the code

Used below command for uploading the code.

```
Git add .
Git commit -m "added fastcar app code"
Git push
```

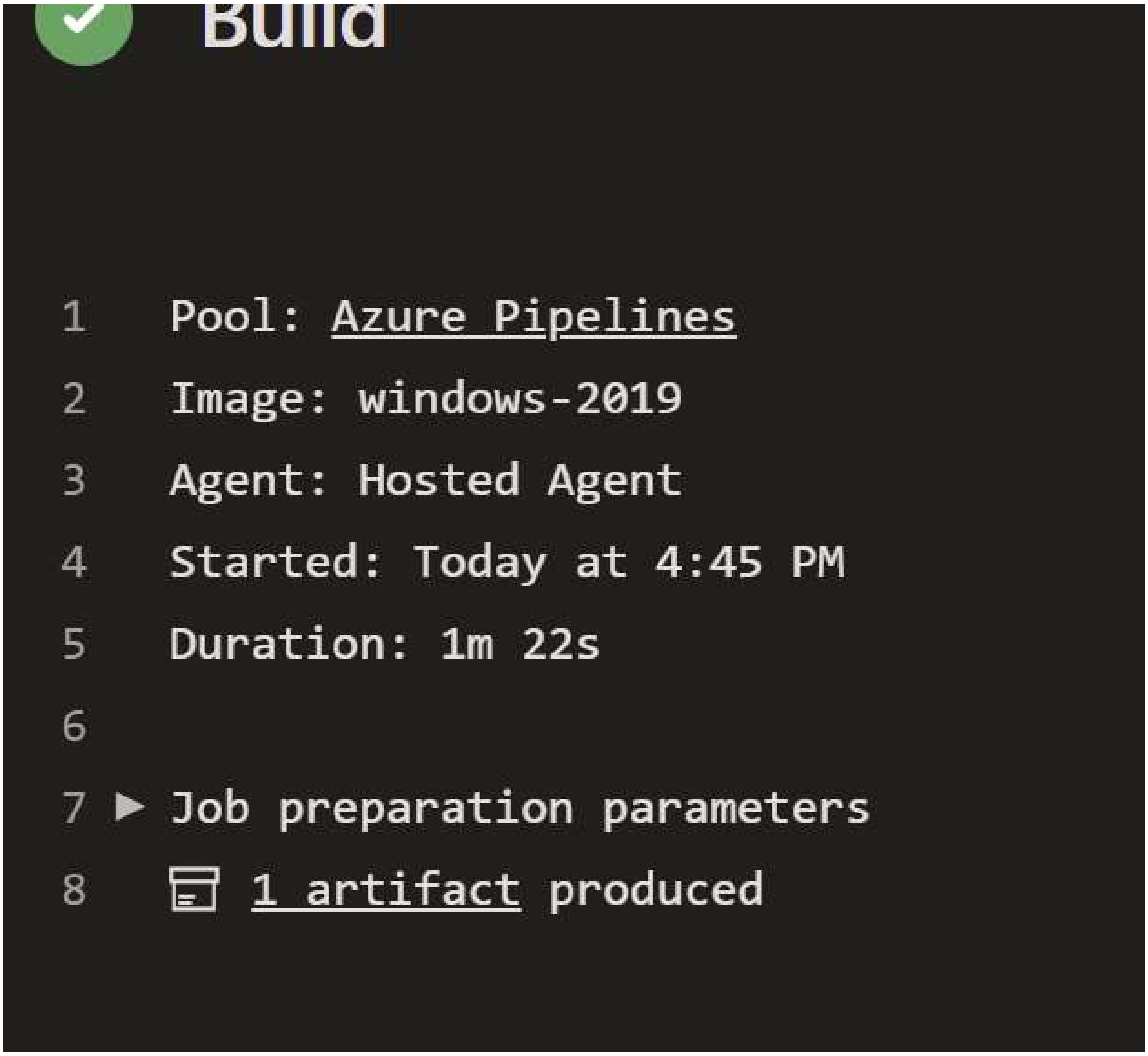
3. Copy Build Package to Drop Folder

Used this command for copy and publishing the package to drop folder.

- task: CopyFiles@2
  inputs:
  Contents: '**'
  TargetFolder: '$(build.artifactstagingdirectory)'

- task: PublishBuildArtifacts@1
  inputs:
  PathtoPublish: '$(Build.ArtifactStagingDirectory)'

```
    ArtifactName: 'drop'
    publishLocation: 'Container'
```



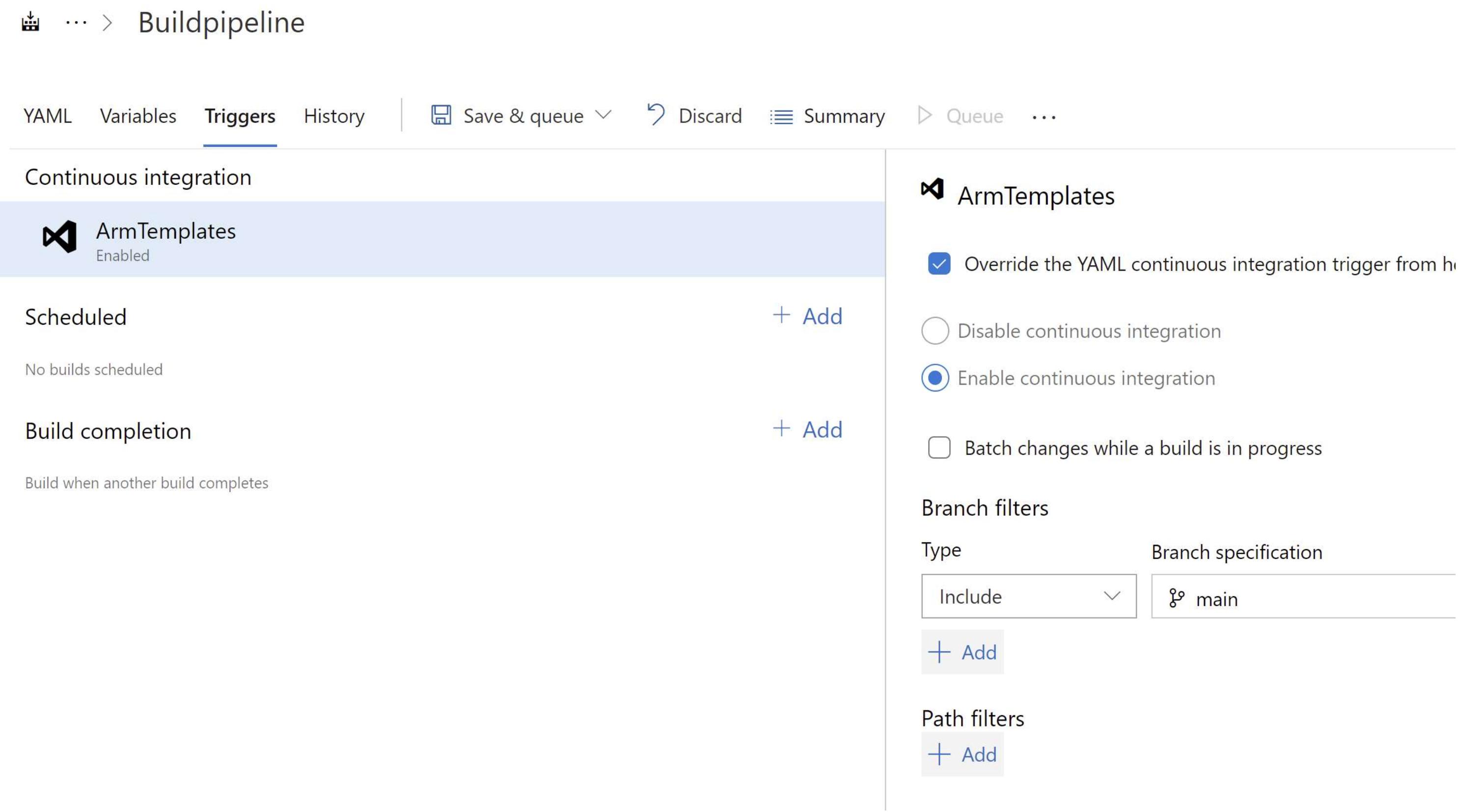
```
✓ Build

Pool: Azure Pipelines
Image: windows-2019
Agent: Hosted Agent
Started: Today at 4:45 PM
Duration: 1m 22s

▶ Job preparation parameters
1 artifact produced
```

4. Enable Continuous Integration Dev to Master branch. I.e. whenever PR is merged to master, the pipeline will be triggered automatically

CI enabled on the pipeline, so whenever there is a merge happens, the pipeline will run automatically.



Continuous integration

ArmTemplates
Enabled

Scheduled
+ Add

No builds scheduled

Build completion
+ Add

Build when another build completes

ArmTemplates

☑ Override the YAML continuous integration trigger from h

○ Disable continuous integration

● Enable continuous integration

☐ Batch changes while a build is in progress

Branch filters

Type	Branch specification
Include	main

+ Add

Path filters
+ Add

5. There will be test projects which will create and maintained in the solution along the Web and API.
The trigger should build all the 3 projects - Web, API and test.
The build should not be successful if any test fails. - which test we need to add, are we looking for

6. Automate creation of artifacts and code in the drop folder.

- Whenever there is any change in code, a new artifact will be produced automatically and will be published to drop folder.

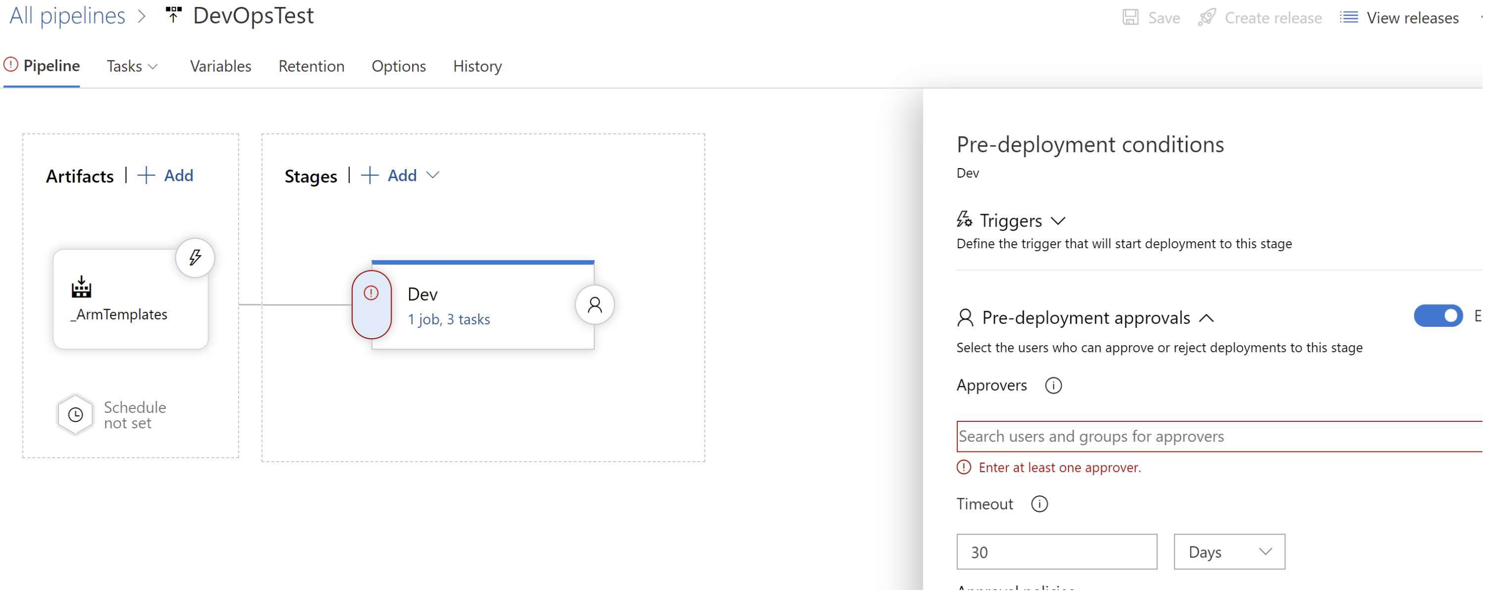
7. Use variable group for automating the deployment for Dev, Stag and Prod environments. Store the environment variables in the variable group.

- Created three different variable group for three different environments and added if condition in the code, so it will automatically picks the variable required for respective environment.

  $ifeq(variables['build.SourceBranchName'],'master') : value : prodgroup : armtemplates - prod$
  {{ if eq(variables['build.SourceBranchName'], 'dev') }}:
  value: dev
  group: armtemplates-dev
  ${{ if eq(variables['build.SourceBranchName'], 'qa') }}:
  value: staging
  group: armtemplates-qa

8. Use deployment gates for QA and Prod environment.

- Approvals is required pre-deployment or post-deployment conditions.



Pre-deployment conditions
Dev

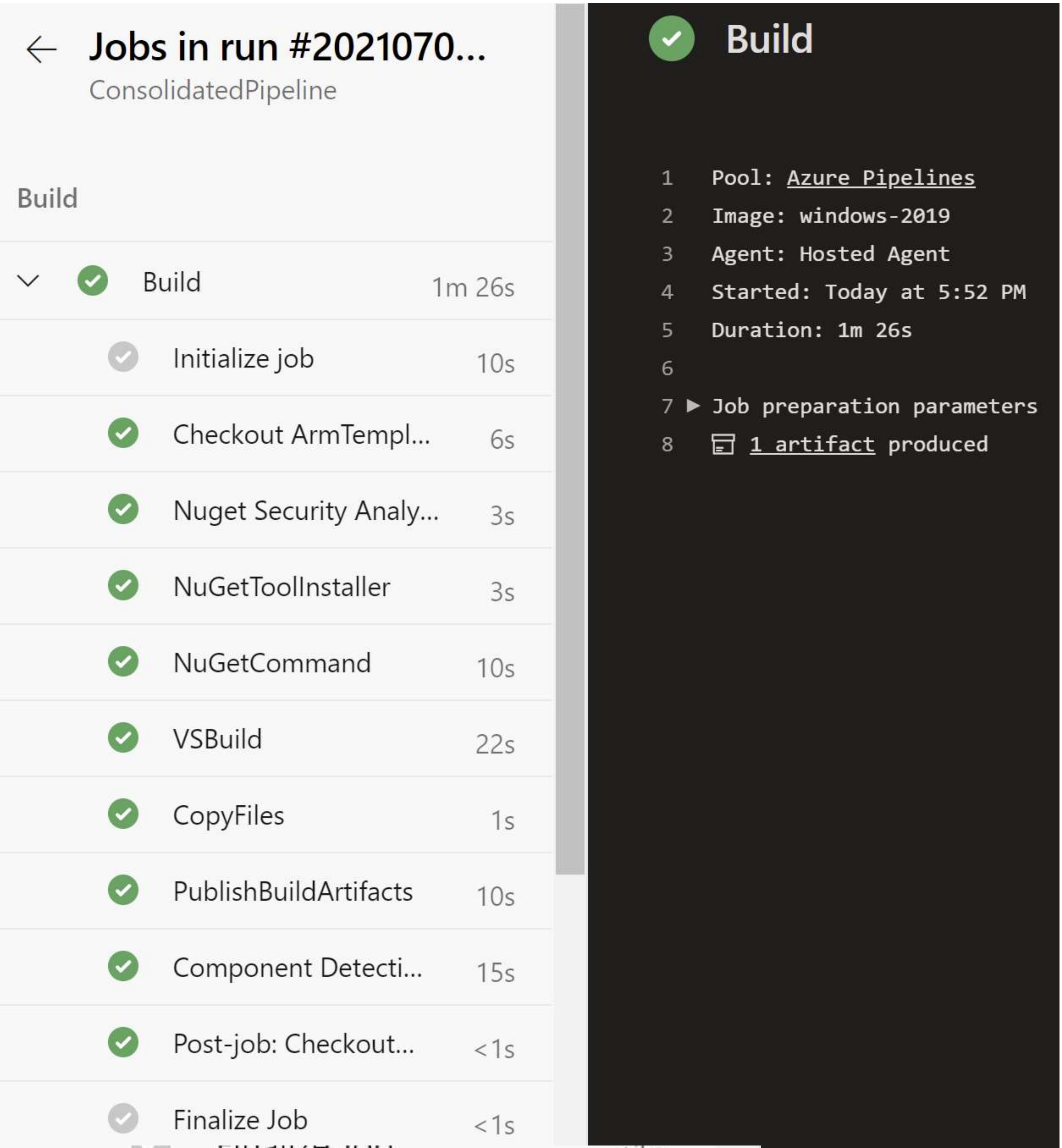
Triggers
Define the trigger that will start deployment to this stage

Pre-deployment approvals
Select the users who can approve or reject deployments to this stage

Approvers
Search users and groups for approvers
Enter at least one approver.

Timeout
30 Days

I have completed this deployment using Yaml pipeline by combining build and release pipeline:



Jobs in run #2021070...

ConsolidatedPipeline

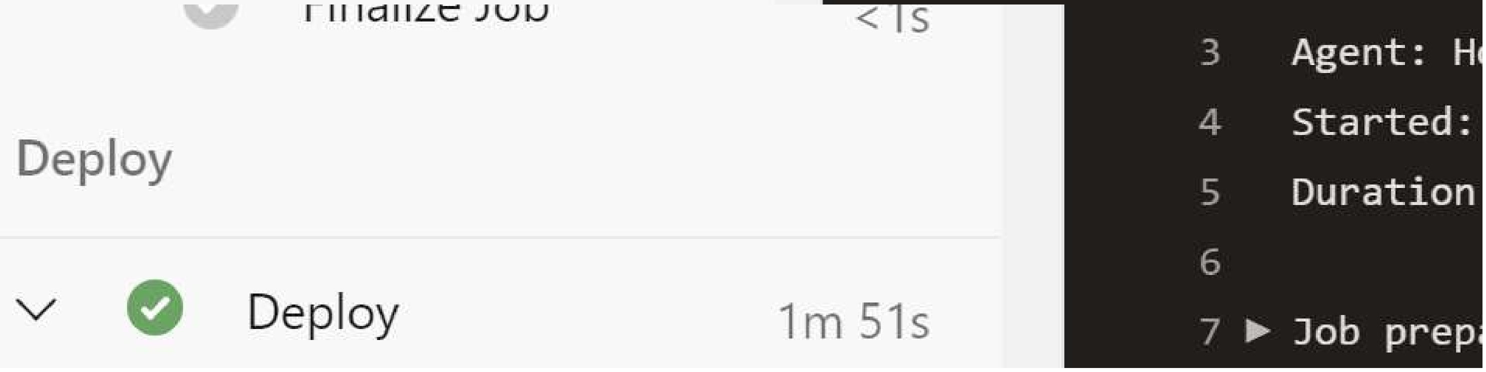
Build

Build	1m 26s
Initialize job	10s
Checkout ArmTempl...	6s
Nuget Security Analy...	3s
NuGetToolInstaller	3s
NuGetCommand	10s
VSBuild	22s
CopyFiles	1s
PublishBuildArtifacts	10s
Component Detecti...	15s
Post-job: Checkout...	<1s
Finalize Job	<1s

Build

```
Pool: Azure Pipelines
Image: windows-2019
Agent: Hosted Agent
Started: Today at 5:52 PM
Duration: 1m 26s

▶ Job preparation parameters
  1 artifact produced
```



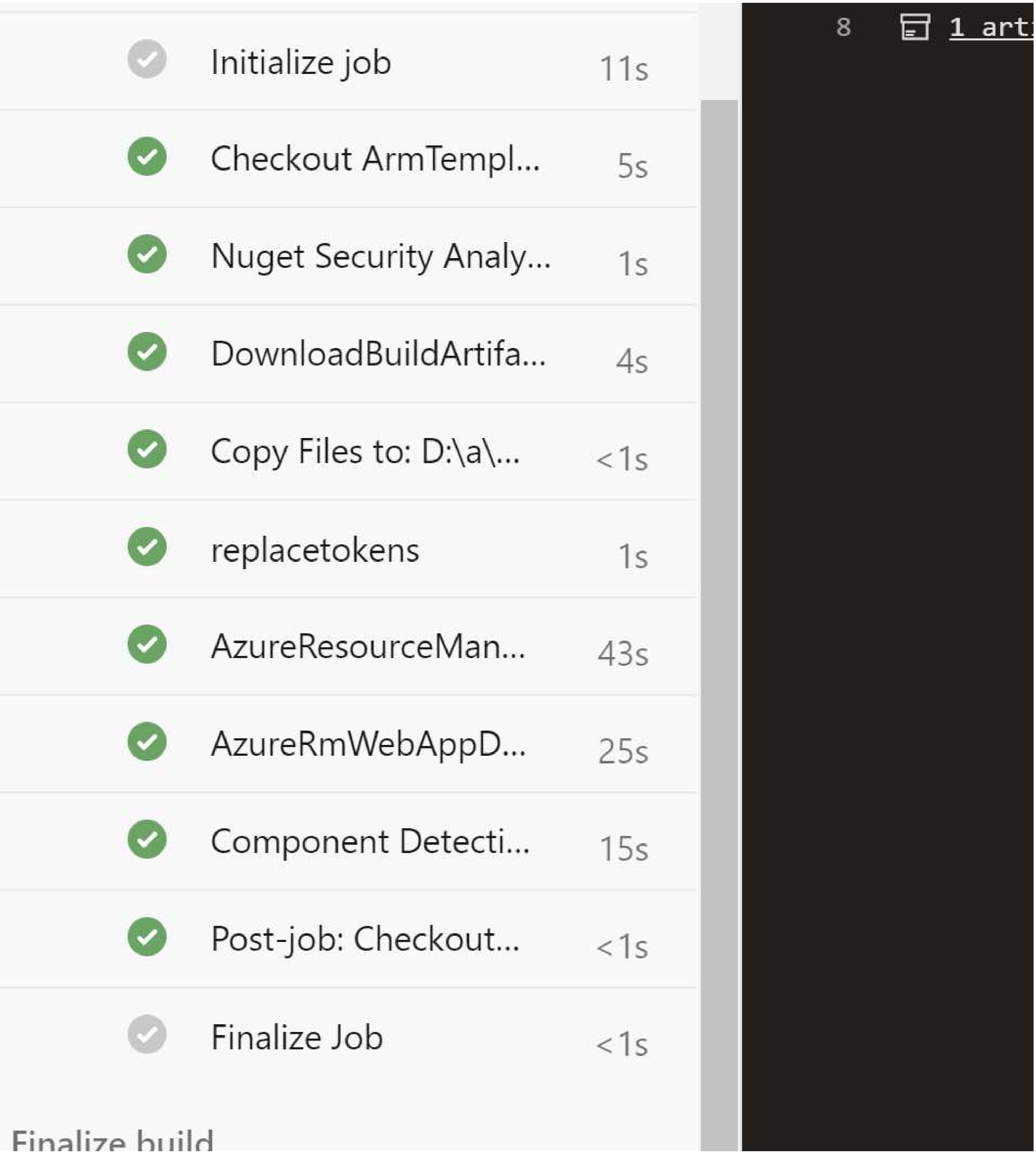
Finalize Job <1s

Deploy

Deploy	1m 51s

```
Agent: H
Started:
Duration

▶ Job prep
```



8 1 art

✓ Initialize job 11s

✓ Checkout ArmTempl... 5s

✓ Nuget Security Analy... 1s

✓ DownloadBuildArtifa... 4s

✓ Copy Files to: D:\a\... <1s

✓ replacetokens 1s

✓ AzureResourceMan... 43s

✓ AzureRmWebAppD... 25s

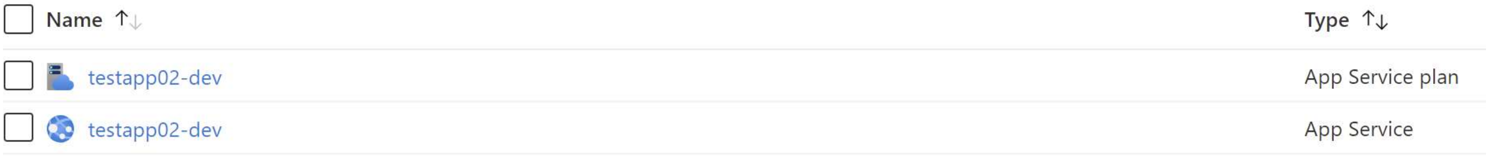
✓ Component Detecti... 15s

✓ Post-job: Checkout... <1s

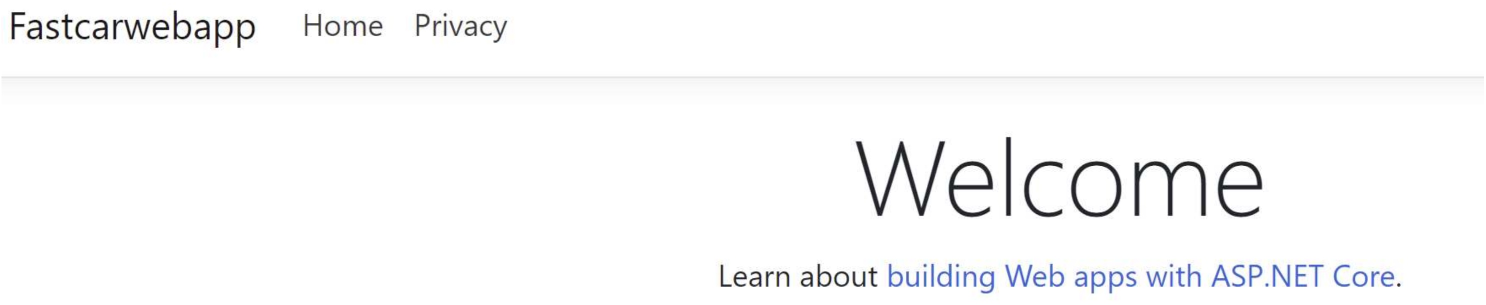
✓ Finalize Job <1s

Finalize build

This is the outcome:



□ Name ↑↓	Type ↑↓
□ testapp02-dev	App Service plan
□ testapp02-dev	App Service



Welcome

Learn about building Web apps with ASP.NET Core.