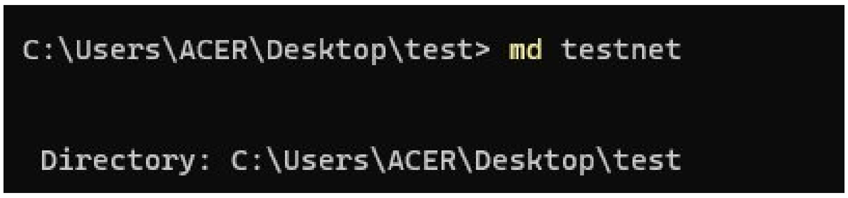
Deploying .NET Core Applications using Azure DevOps CI/CD Pipelines and Azure App Services

We will use Azure DevOps Project to set up continuous delivery (CD) and continuous integration (CI) pipelines in this project. The purpose is to quickly deploy an app to a Azure services, in this case App Service

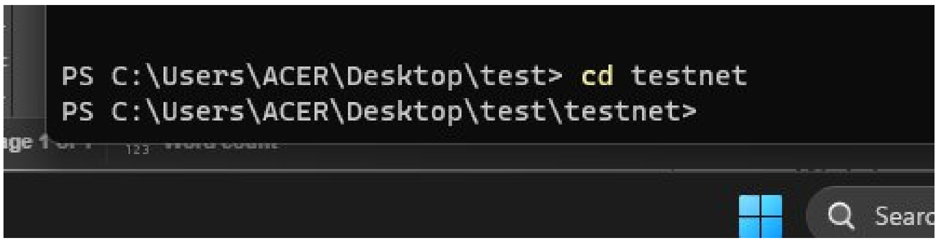
In this project, we will build an sample ASP.NET core sample code, explore the CI/CD pipelines, commit code changes and run CI/CD.

1. Setting up a sample .NET core project:

We wil first create a directory



```
C:\Users\ACER\Desktop\test> md testnet


 Directory: C:\Users\ACER\Desktop\test
```

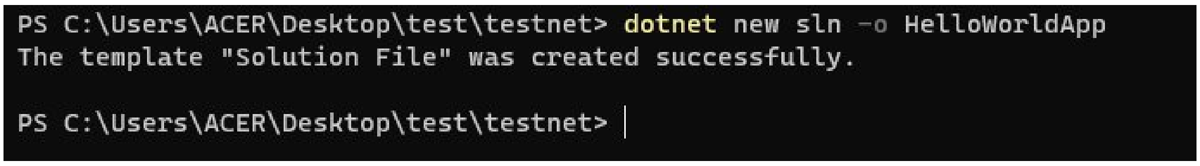


```
PS C:\Users\ACER\Desktop\test> cd testnet
PS C:\Users\ACER\Desktop\test\testnet>
```

After installing the .net SDK in the system, we will run the command:

dotnet new sln -o HelloWorldApp

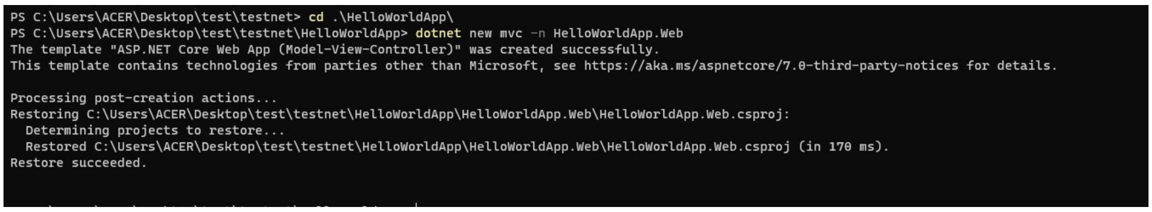
This will create a solution file



```
PS C:\Users\ACER\Desktop\test\testnet> dotnet new sln -o HelloWorldApp
The template "Solution File" was created successfully.

PS C:\Users\ACER\Desktop\test\testnet>
```

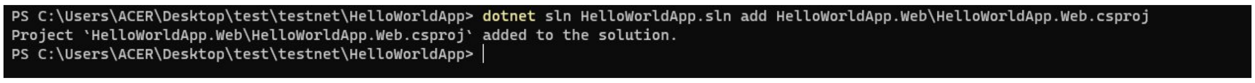
We wil go inside the the file and create the new mvc project:

dotnet new mvc -n HelloWorldApp.Web



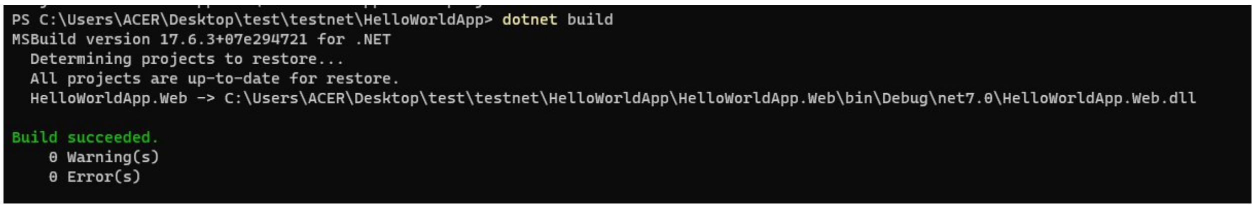
```
PS C:\Users\ACER\Desktop\test\testnet> cd .\HelloWorldApp\
PS C:\Users\ACER\Desktop\test\testnet\HelloWorldApp> dotnet new mvc -n HelloWorldApp.Web
The template "ASP.NET Core Web App (Model-View-Controller)" was created successfully.
This template contains technologies from parties other than Microsoft, see https://aka.ms/aspnetcore/7.0-third-party-notices for details.

Processing post-creation actions...
Restoring C:\Users\ACER\Desktop\test\testnet\HelloWorldApp\HelloWorldApp.Web\HelloWorldApp.Web.csproj:
  Determining projects to restore...
  Restored C:\Users\ACER\Desktop\test\testnet\HelloWorldApp\HelloWorldApp.Web\HelloWorldApp.Web.csproj (in 170 ms).
Restore succeeded.
```

Now we will add the project to the solution:



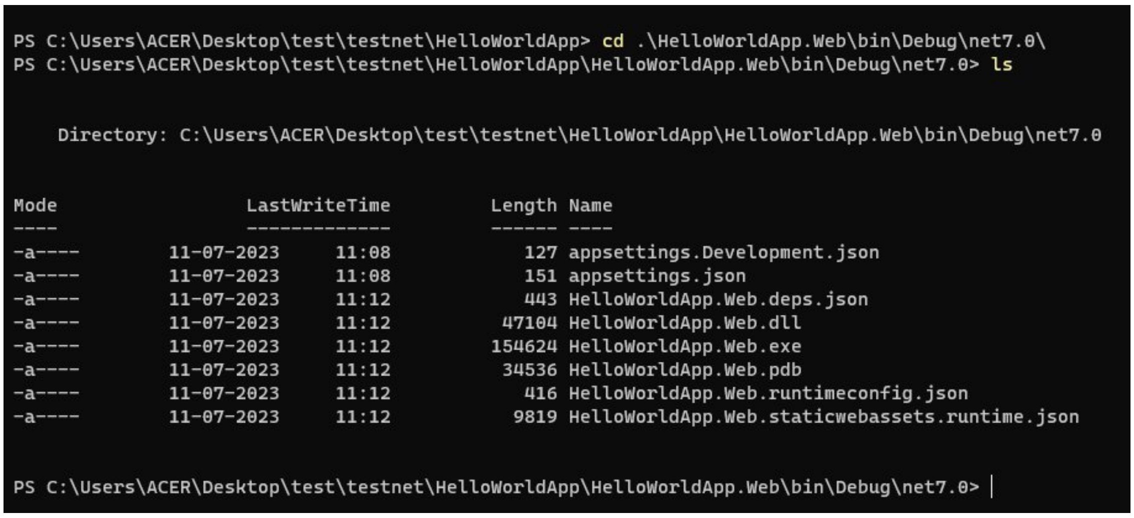
```
PS C:\Users\ACER\Desktop\test\testnet\HelloWorldApp> dotnet sln HelloWorldApp.sln add HelloWorldApp.Web\HelloWorldApp.Web.csproj
Project `HelloWorldApp.Web\HelloWorldApp.Web.csproj` added to the solution.
PS C:\Users\ACER\Desktop\test\testnet\HelloWorldApp> |
```

Now we will build in local machine to test



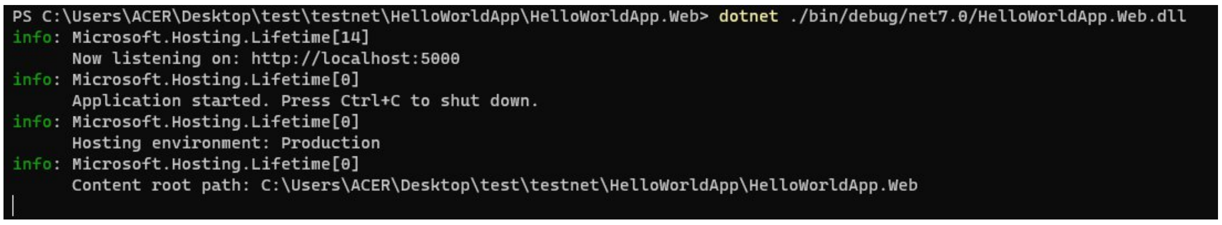
```
PS C:\Users\ACER\Desktop\test\testnet\HelloWorldApp> dotnet build
MSBuild version 17.6.3+07e294721 for .NET
  Determining projects to restore...
  All projects are up-to-date for restore.
  HelloWorldApp.Web -> C:\Users\ACER\Desktop\test\testnet\HelloWorldApp\HelloWorldApp.Web\bin\Debug\net7.0\HelloWorldApp.Web.dll

Build succeeded.
    0 Warning(s)
    0 Error(s)
```

We can see it has built without any error:



```
PS C:\Users\ACER\Desktop\test\testnet\HelloWorldApp> cd .\HelloWorldApp.Web\bin\Debug\net7.0\
PS C:\Users\ACER\Desktop\test\testnet\HelloWorldApp\HelloWorldApp.Web\bin\Debug\net7.0> ls


    Directory: C:\Users\ACER\Desktop\test\testnet\HelloWorldApp\HelloWorldApp.Web\bin\Debug\net7.0


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        11-07-2023     11:08            127 appsettings.Development.json
-a----        11-07-2023     11:08            151 appsettings.json
-a----        11-07-2023     11:12            443 HelloWorldApp.Web.deps.json
-a----        11-07-2023     11:12          47104 HelloWorldApp.Web.dll
-a----        11-07-2023     11:12         154624 HelloWorldApp.Web.exe
-a----        11-07-2023     11:12          34536 HelloWorldApp.Web.pdb
-a----        11-07-2023     11:12            416 HelloWorldApp.Web.runtimeconfig.json
-a----        11-07-2023     11:12           9819 HelloWorldApp.Web.staticwebassets.runtime.json


PS C:\Users\ACER\Desktop\test\testnet\HelloWorldApp\HelloWorldApp.Web\bin\Debug\net7.0> |
```

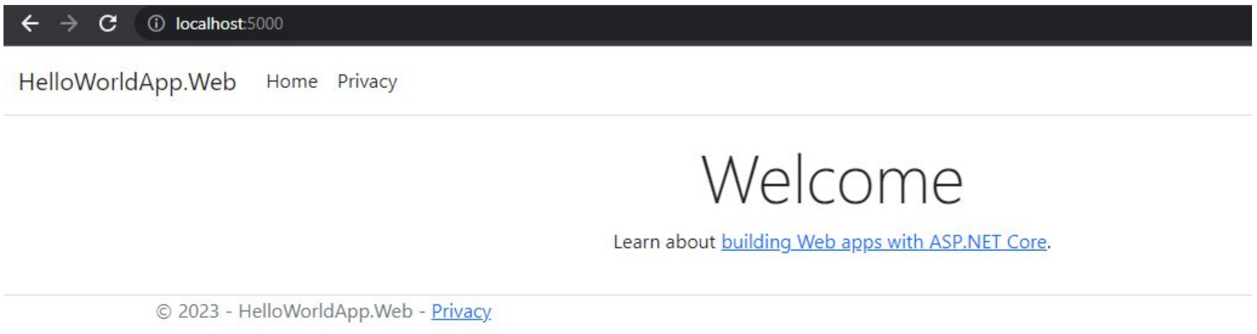
Now we will check out dll from our project folder



```
PS C:\Users\ACER\Desktop\test\testnet\HelloWorldApp\HelloWorldApp.Web> dotnet ./bin/debug/net7.0/HelloWorldApp.Web.dll
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5000
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Production
info: Microsoft.Hosting.Lifetime[0]
      Content root path: C:\Users\ACER\Desktop\test\testnet\HelloWorldApp\HelloWorldApp.Web
|
```

We see we can host it locally

Going to the http://localhost:5000 we can see our sameple .net core web all



Welcome

Learn about building Web apps with ASP.NET Core.

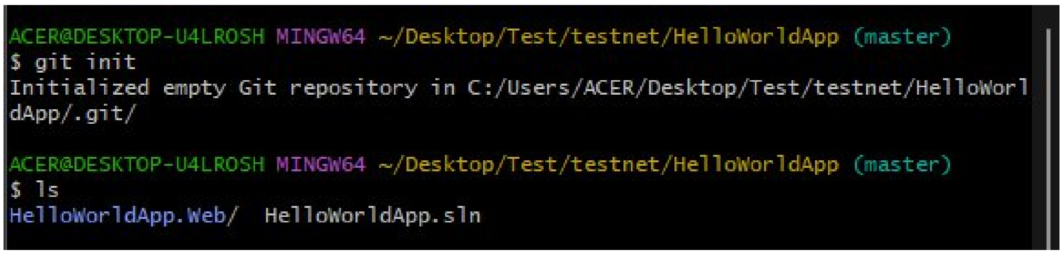
© 2023 - HelloWorldApp.Web - Privacy

2. Add the Code to the github:

As it works, we will create a new repository in github and push these there.

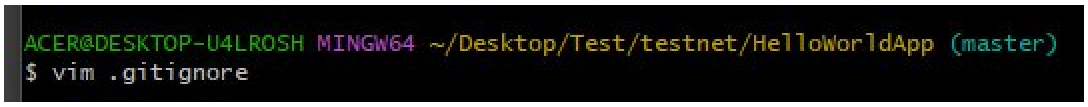
We will add this to Azure repo later.

First we will initialize the repo using git init in git bash

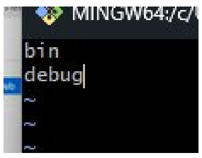


```
ACER@DESKTOP-U4LROSH MINGW64 ~/Desktop/Test/testnet/HelloWorldApp (master)
$ git init
Initialized empty Git repository in C:/Users/ACER/Desktop/Test/testnet/HelloWorl
dApp/.git/

ACER@DESKTOP-U4LROSH MINGW64 ~/Desktop/Test/testnet/HelloWorldApp (master)
$ ls
HelloWorldApp.Web/  HelloWorldApp.sln
```

Then we will add a gitignore file

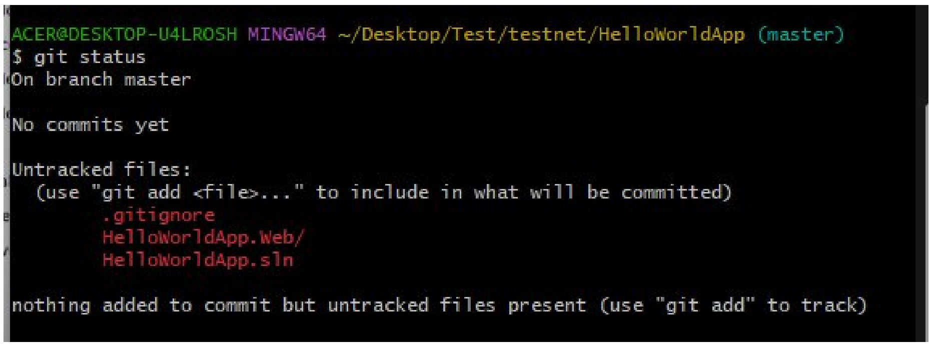


```
ACER@DESKTOP-U4LROSH MINGW64 ~/Desktop/Test/testnet/HelloWorldApp (master)
$ vim .gitignore
```

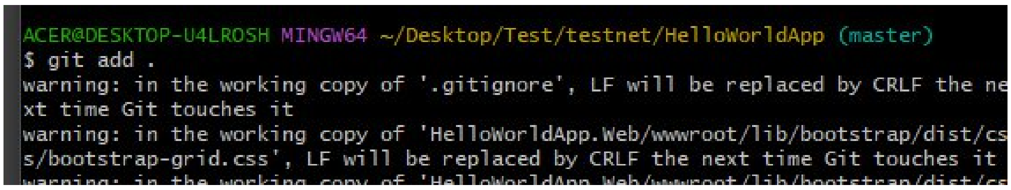


```
bin
debug
~
~
~
```

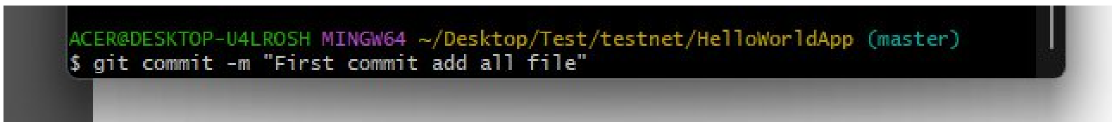
Now we will add and commit



```
ACER@DESKTOP-U4LROSH MINGW64 ~/Desktop/Test/testnet/HelloWorldApp (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
        HelloWorldApp.Web/
        HelloWorldApp.sln

nothing added to commit but untracked files present (use "git add" to track)
```

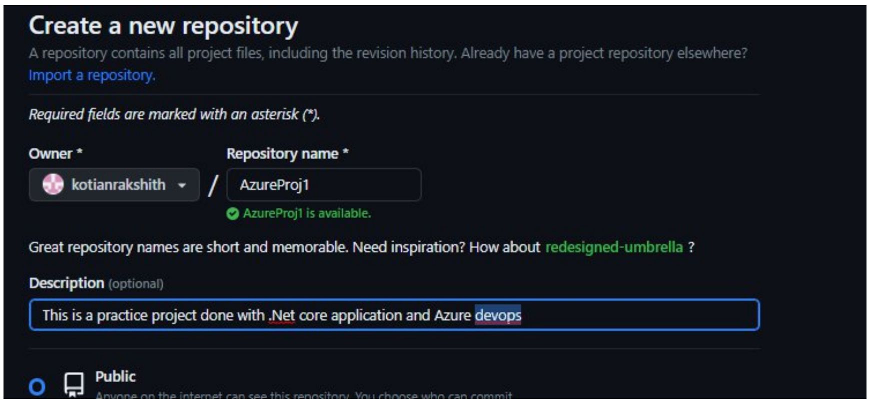


```
ACER@DESKTOP-U4LROSH MINGW64 ~/Desktop/Test/testnet/HelloWorldApp (master)
$ git add .
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the ne
xt time Git touches it
warning: in the working copy of 'HelloWorldApp.Web/wwwroot/lib/bootstrap/dist/cs
s/bootstrap-grid.css', LF will be replaced by CRLF the next time Git touches it
```



```
ACER@DESKTOP-U4LROSH MINGW64 ~/Desktop/Test/testnet/HelloWorldApp (master)
$ git commit -m "First commit add all file"
```

In github i have created new repository:



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Required fields are marked with an asterisk (*).

Owner * kotianrakshith / Repository name * AzureProj1

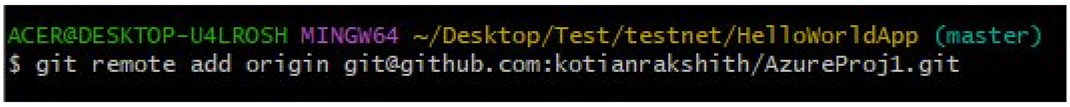
AzureProj1 is available.

Great repository names are short and memorable. Need inspiration? How about redesigned-umbrella ?

Description (optional): This is a practice project done with .Net core application and Azure devops

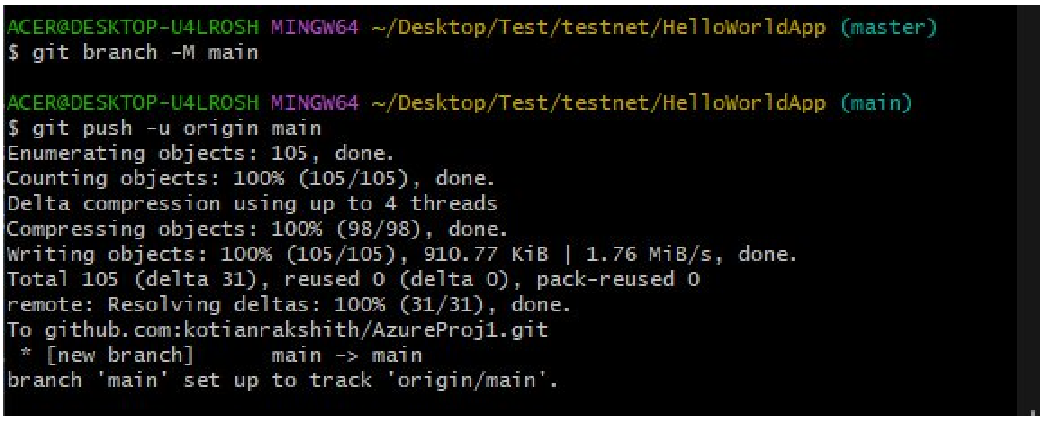
Public

Now we will add this as origin our git repo



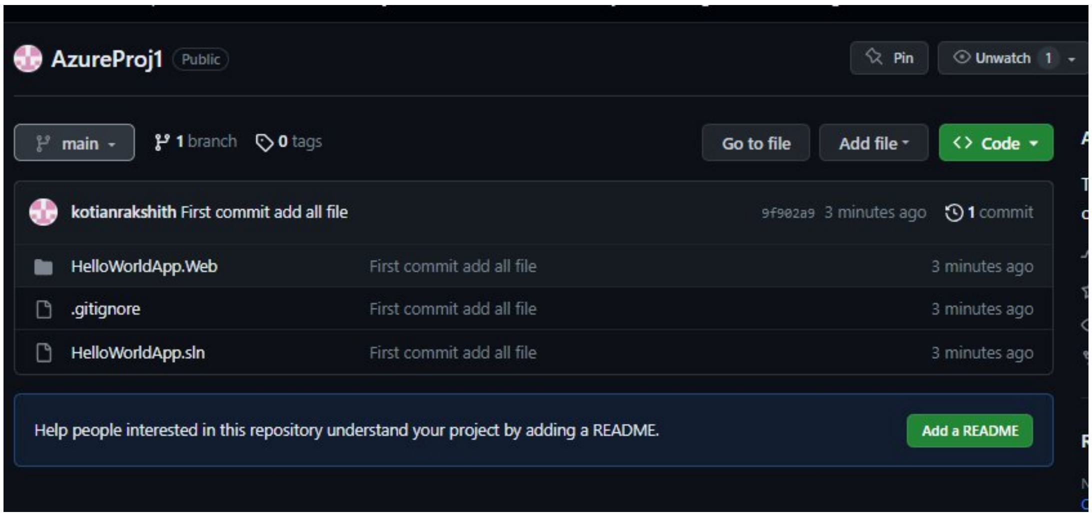
```
ACER@DESKTOP-U4LROSH MINGW64 ~/Desktop/Test/testnet/HelloWorldApp (master)
$ git remote add origin git@github.com:kotianrakshith/AzureProj1.git
```

Now we will rename and push all the files:



```
ACER@DESKTOP-U4LROSH MINGW64 ~/Desktop/Test/testnet/HelloWorldApp (master)
$ git branch -M main

ACER@DESKTOP-U4LROSH MINGW64 ~/Desktop/Test/testnet/HelloWorldApp (main)
$ git push -u origin main
Enumerating objects: 105, done.
Counting objects: 100% (105/105), done.
Delta compression using up to 4 threads
Compressing objects: 100% (98/98), done.
Writing objects: 100% (105/105), 910.77 KiB | 1.76 MiB/s, done.
Total 105 (delta 31), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (31/31), done.
To github.com:kotianrakshith/AzureProj1.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

Now we see that all files are in github



AzureProj1 Public

main · 1 branch · 0 tags

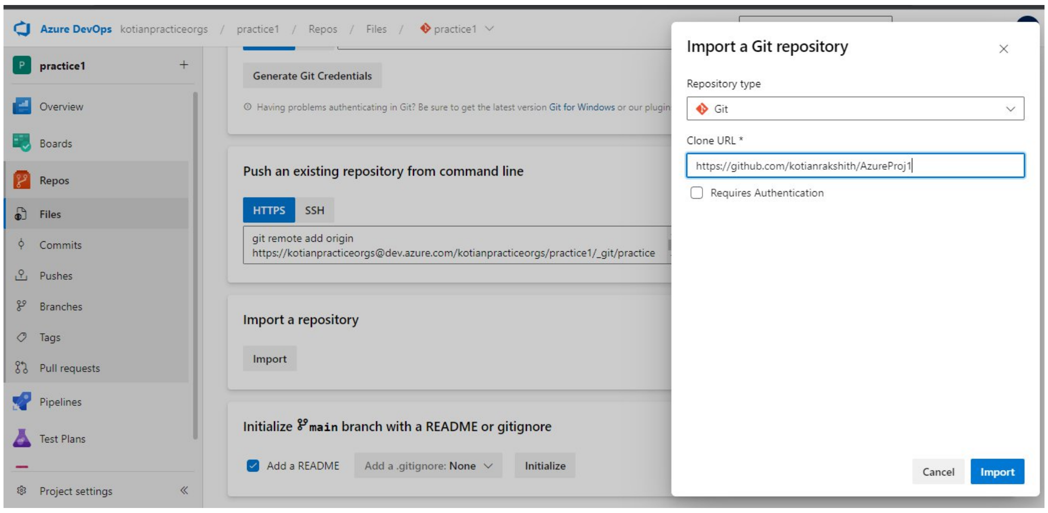
kotianrakshith First commit add all file · 9f902a9 3 minutes ago · 1 commit

HelloWorldApp.Web	First commit add all file	3 minutes ago
.gitignore	First commit add all file	3 minutes ago
HelloWorldApp.sln	First commit add all file	3 minutes ago

Help people interested in this repository understand your project by adding a README.

3. **Import the project to Azure Repo:**

   Go to Repos section of Azure DevOps and click on ‘Import a repository’:



Having problems authenticating in Git? Be sure to get the latest version Git for Windows or our plugin

Push an existing repository from command line

```
git remote add origin
https://kotianpracticeorgs@dev.azure.com/kotianpracticeorgs/practice1/_git/practice
```

Import a repository

Initialize main branch with a README or gitignore

Import a Git repository

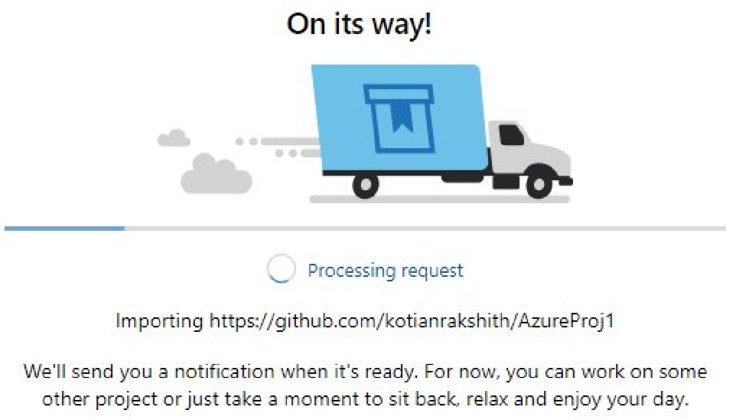
Repository type: Git

Clone URL *: https://github.com/kotianrakshith/AzureProj1

☐ Requires Authentication

Now paste your github link and import.

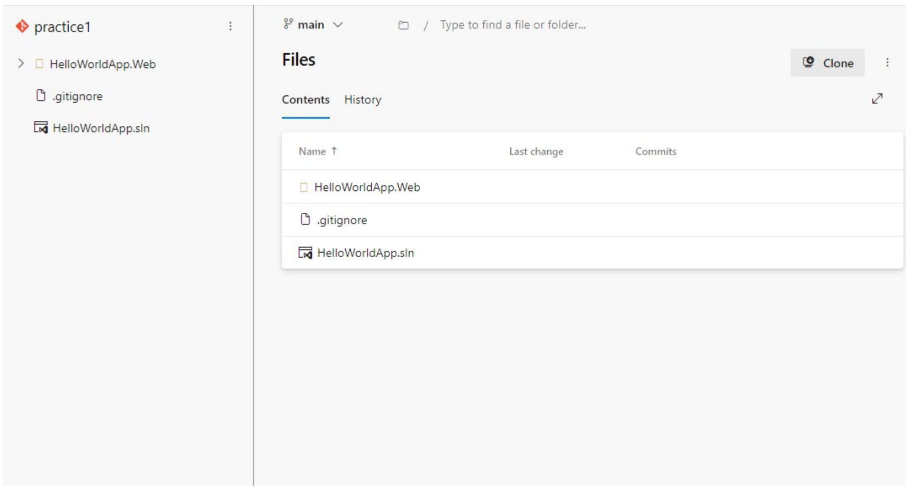
You can also push from your local repo is you want:


On its way!

Processing request

Importing https://github.com/kotianrakshith/AzureProj1

We'll send you a notification when it's ready. For now, you can work on some other project or just take a moment to sit back, relax and enjoy your day.

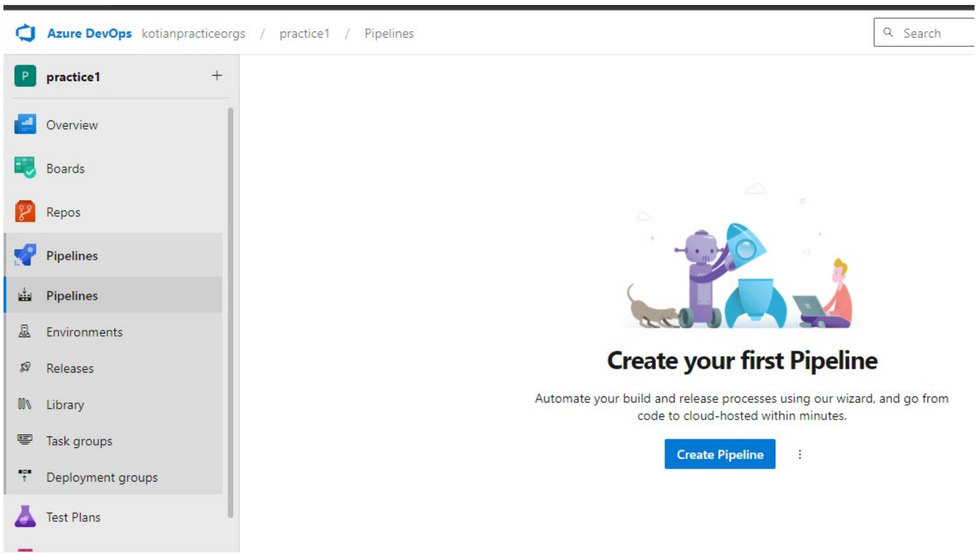


Name ↑	Last change	Commits
HelloWorldApp.Web		
.gitignore		
HelloWorldApp.sln		

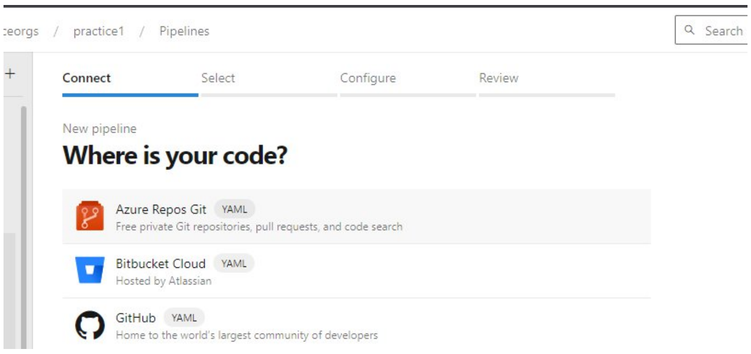
Once its imported you should be able to see your files

4. <u>Creating a pipeline</u>

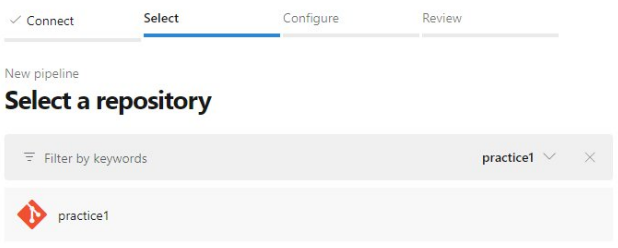
Go to pipline section of the Azure DevOps and click ‘create pipeline’


Azure DevOps kotianpracticeorgs / practice1 / Pipelines
Search
practice1
Overview
Boards
Repos
Pipelines
Pipelines
Environments
Releases
Library
Task groups
Deployment groups
Test Plans
Create your first Pipeline
Automate your build and release processes using our wizard, and go from code to cloud-hosted within minutes.
Create Pipeline

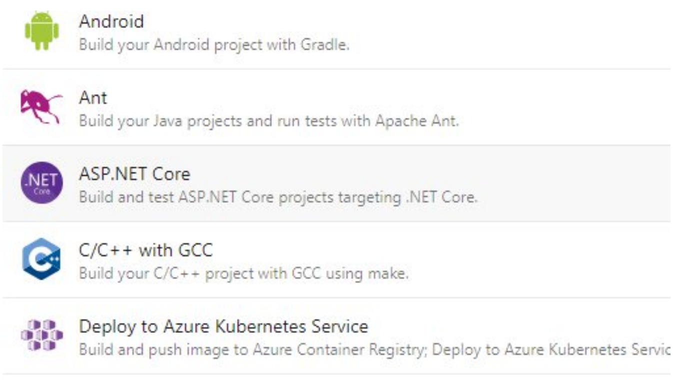
For the code chose Azure Repos Git:


eorgs / practice1 / Pipelines
Search
Connect
Select
Configure
Review
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

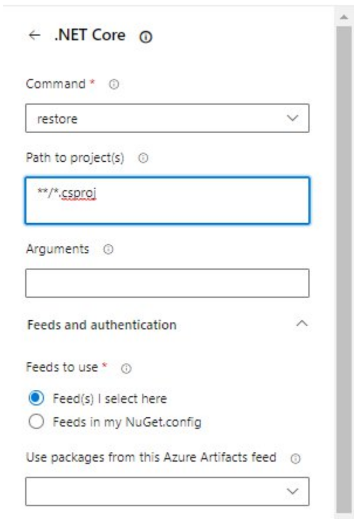
And chose your repo:



In the template configuraion we will chose ASP.NET core


Android
Build your Android project with Gradle.
Ant
Build your Java projects and run tests with Apache Ant.
ASP.NET Core
Build and test ASP.NET Core projects targeting .NET Core.
C/C++ with GCC
Build your C/C++ project with GCC using make.
Deploy to Azure Kubernetes Service
Build and push image to Azure Container Registry; Deploy to Azure Kubernetes Servic

You will see an yaml, we will edit this code to add restore,build,publish steps:


← .NET Core

Command * restore

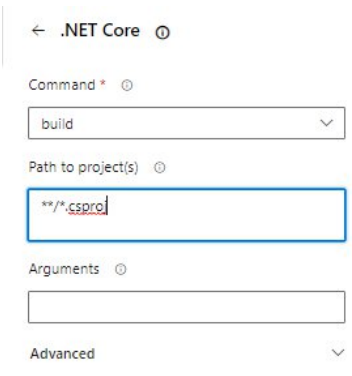
Path to project(s) **/*.csproj

Arguments

Feeds and authentication

Feeds to use *
- [✓ Feed(s) I select here]
- Feeds in my NuGet.config

Use packages from this Azure Artifacts feed

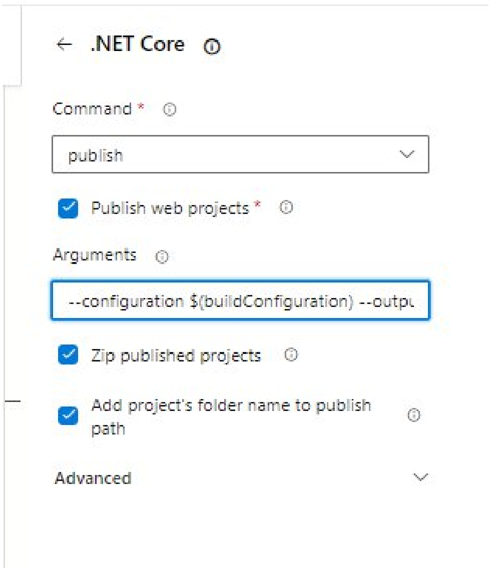

← .NET Core

Command * build

Path to project(s) **/*.csproj

Arguments

Advanced


← .NET Core

Command * publish

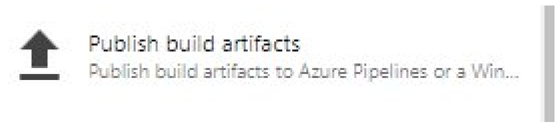
- [✓] Publish web projects *

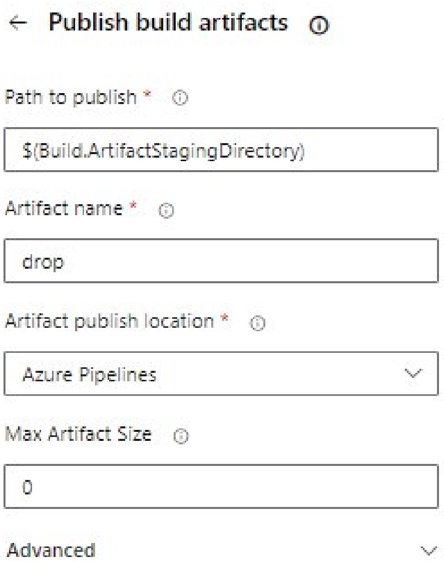
Arguments --configuration $(buildConfiguration) --outpu

- [✓] Zip published projects
- [✓] Add project's folder name to publish path

Advanced

We will also add publish build artifact task :


Publish build artifacts
Publish build artifacts to Azure Pipelines or a Win...


← Publish build artifacts

Path to publish * $(Build.ArtifactStagingDirectory)

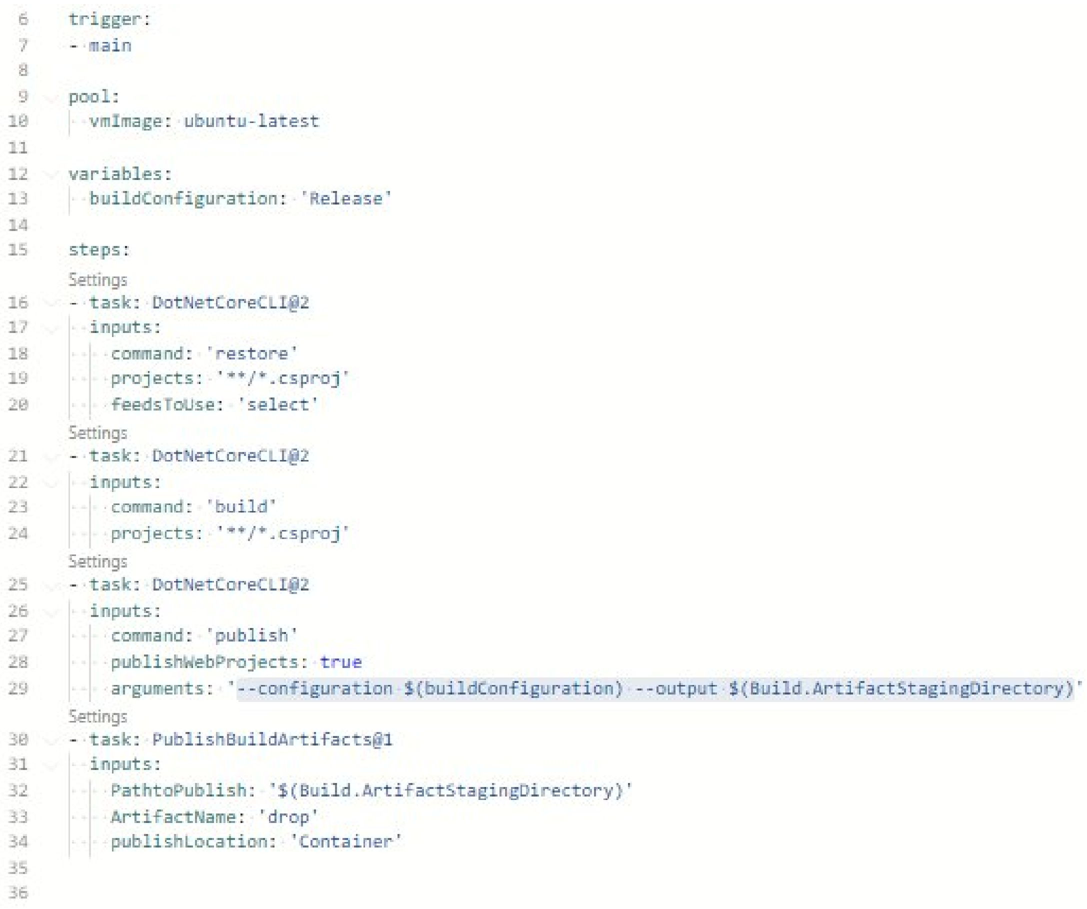
Artifact name * drop

Artifact publish location * Azure Pipelines

Max Artifact Size 0

Advanced

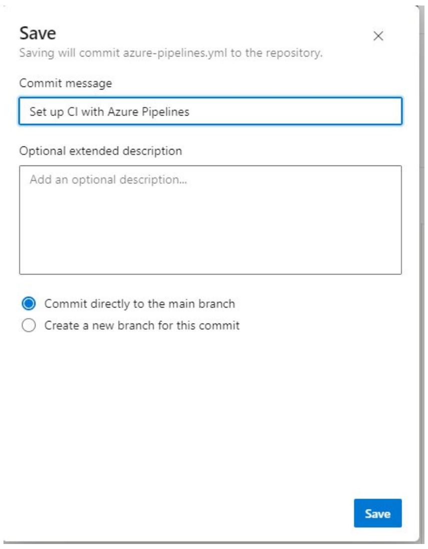
Finally we will have the whole code:



```
trigger:
- main

pool:
  vmImage: ubuntu-latest

variables:
  buildConfiguration: 'Release'

steps:
Settings
- task: DotNetCoreCLI@2
  inputs:
    command: 'restore'
    projects: '**/*.csproj'
    feedsToUse: 'select'
Settings
- task: DotNetCoreCLI@2
  inputs:
    command: 'build'
    projects: '**/*.csproj'
Settings
- task: DotNetCoreCLI@2
  inputs:
    command: 'publish'
    publishWebProjects: true
    arguments: '--configuration $(buildConfiguration) --output $(Build.ArtifactStagingDirectory)'
Settings
- task: PublishBuildArtifacts@1
  inputs:
    PathtoPublish: '$(Build.ArtifactStagingDirectory)'
    ArtifactName: 'drop'
    publishLocation: 'Container'
```

It will be saved in the Azure repo, but i will also add it in the github repo for future reference.

Now you can save:



Save

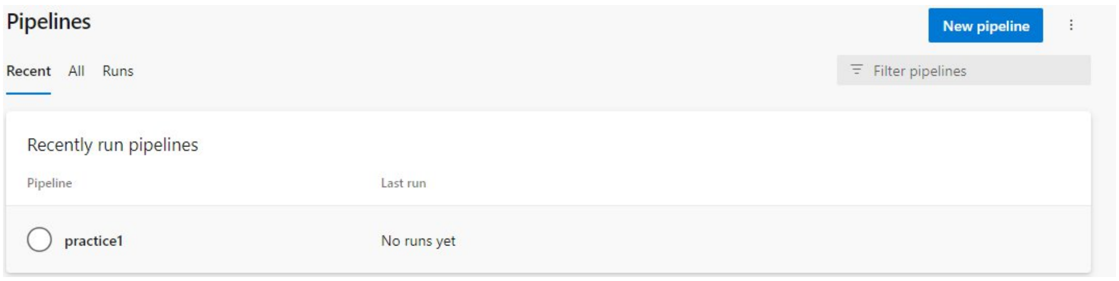
Saving will commit azure-pipelines.yml to the repository.

Commit message: Set up CI with Azure Pipelines

Optional extended description: Add an optional description...

[✓ Commit directly to the main branch]
Create a new branch for this commit

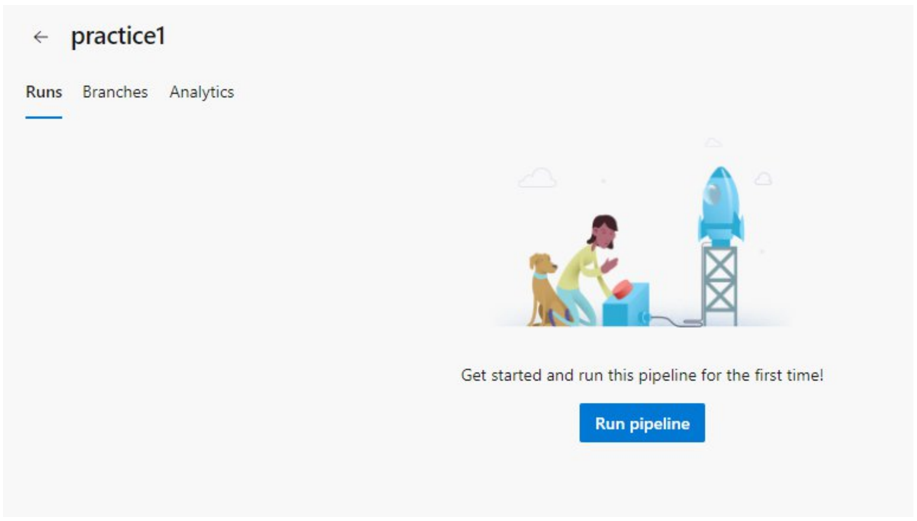
Now you can see a pipeline created:



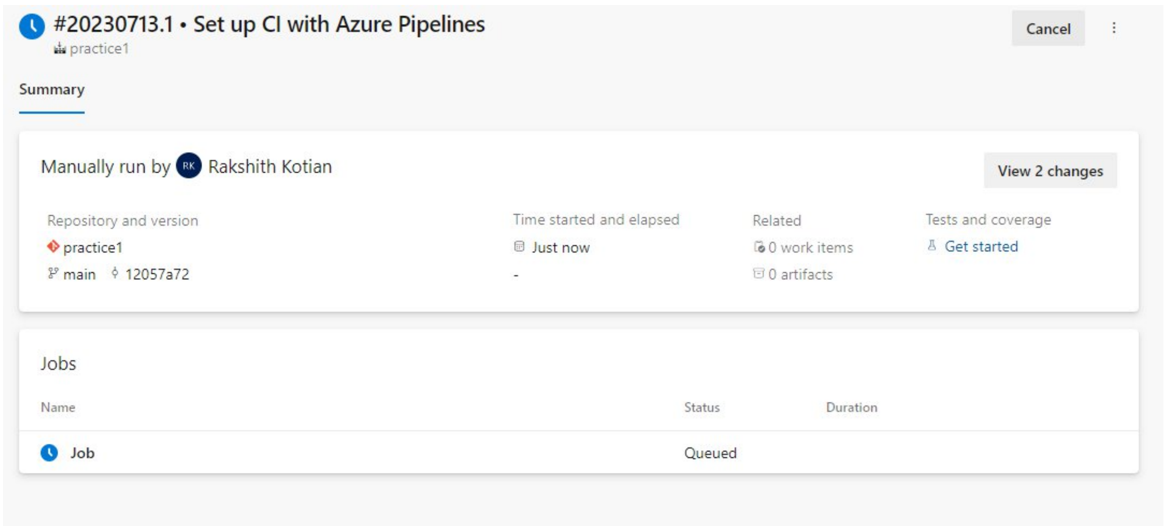
Recently run pipelines

Pipeline	Last run
practice1	No runs yet

Lets run the pipeline:



We can see the job is running:



#20230713.1 • Set up CI with Azure Pipelines
practice1

Manually run by RK Rakshith Kotian

Repository and version
practice1
main 12057a72

Time started and elapsed
Just now
-

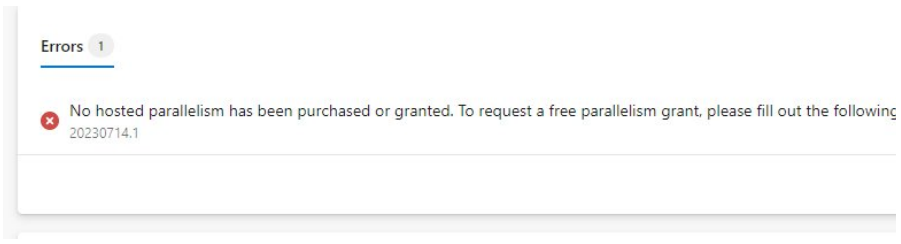
Related
0 work items
0 artifacts

Tests and coverage
Get started

Jobs

Name	Status	Duration
Job	Queued	

We got an error because in the free Azure we have setup we do not yet have access to run parallel job in Microsoft hosted machine.

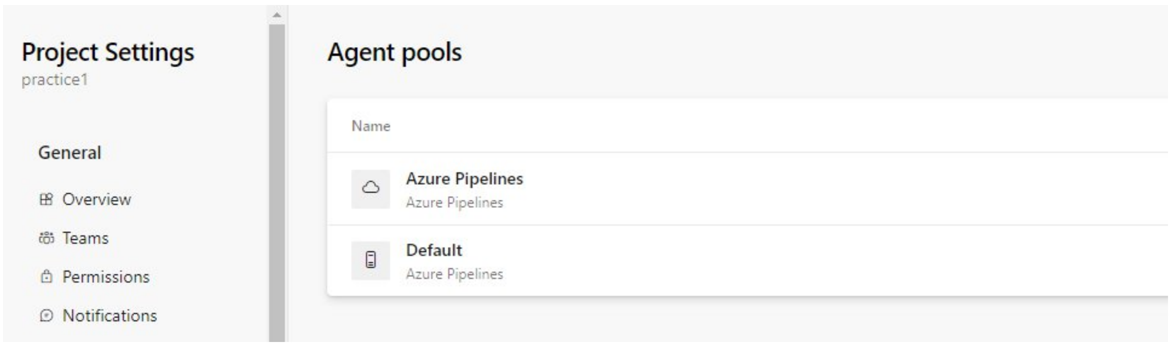


No hosted parallelism has been purchased or granted. To request a free parallelism grant, please fill out the following
20230714.1

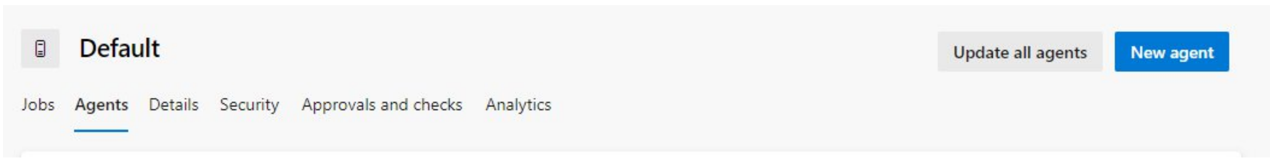
So we can run the build in the local machine:

We will first add our local machine as an agent:

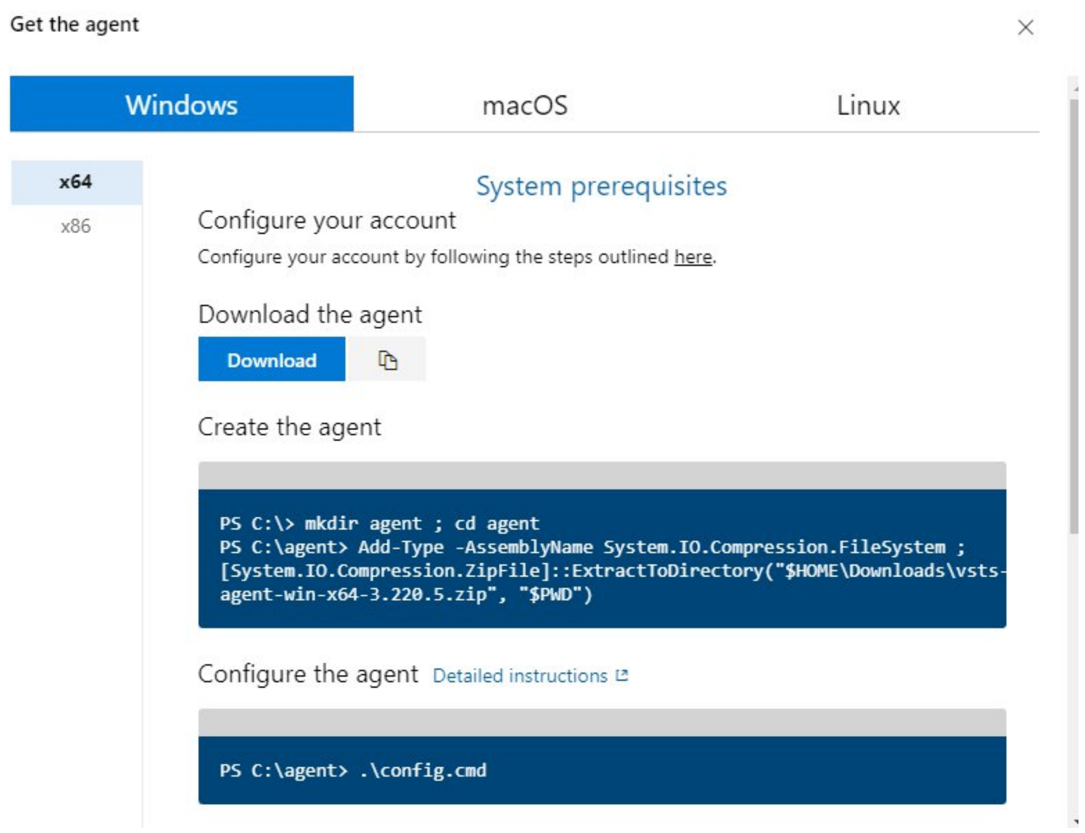
In project settings go to agent pools:



Here go to default and click New agent:



Steps are given in detail:



Configure your account

Configure your account by following the steps outlined here.

Download the agent

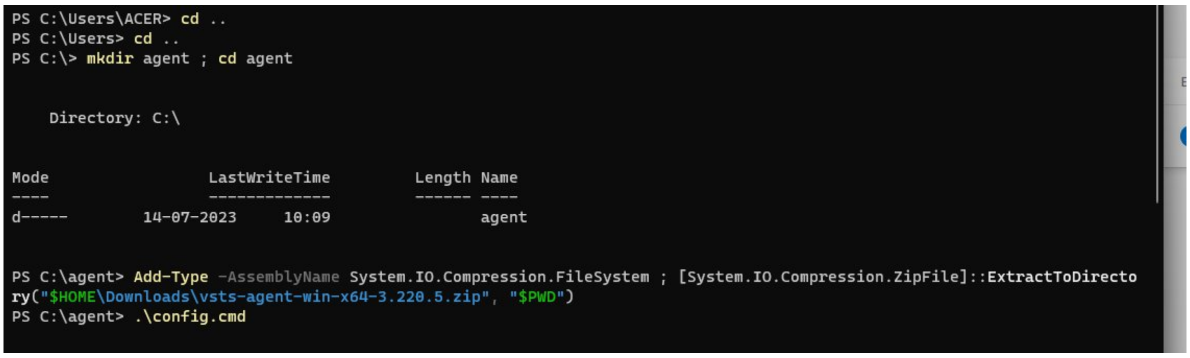
Create the agent

```
PS C:\> mkdir agent ; cd agent
PS C:\agent> Add-Type -AssemblyName System.IO.Compression.FileSystem ;
[System.IO.Compression.ZipFile]::ExtractToDirectory("$HOME\Downloads\vsts-
agent-win-x64-3.220.5.zip", "$PWD")
```

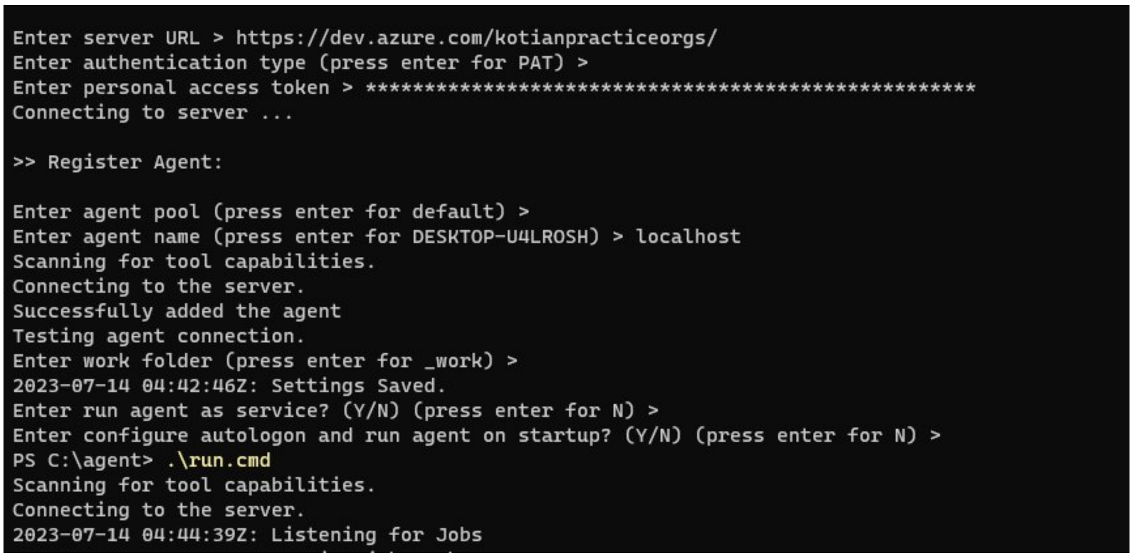
Configure the agent

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

We will follow the same step in windows local machine:

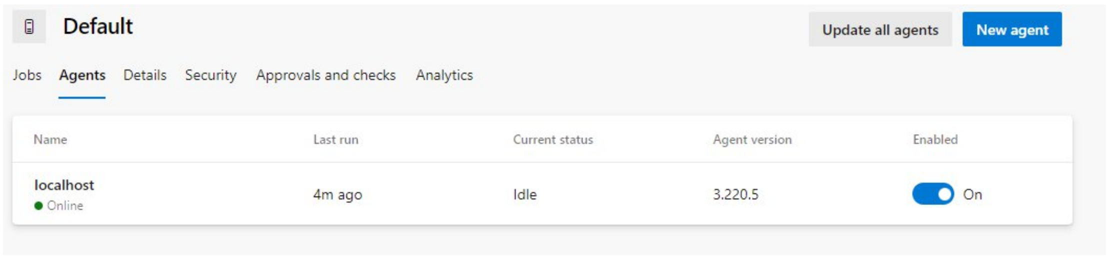


```
PS C:\Users\ACER> cd ..
PS C:\Users> cd ..
PS C:\> mkdir agent ; cd agent


    Directory: C:\


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        14-07-2023     10:09                agent


PS C:\agent> Add-Type -AssemblyName System.IO.Compression.FileSystem ; [System.IO.Compression.ZipFile]::ExtractToDirecto
ry("$HOME\Downloads\vsts-agent-win-x64-3.220.5.zip", "$PWD")
PS C:\agent> .\config.cmd
```



```
Enter server URL > https://dev.azure.com/kotianpracticeorgs/
Enter authentication type (press enter for PAT) >
Enter personal access token > ****************************************************
Connecting to server ...

>> Register Agent:

Enter agent pool (press enter for default) >
Enter agent name (press enter for DESKTOP-U4LROSH) > localhost
Scanning for tool capabilities.
Connecting to the server.
Successfully added the agent
Testing agent connection.
Enter work folder (press enter for _work) >
2023-07-14 04:42:46Z: Settings Saved.
Enter run agent as service? (Y/N) (press enter for N) >
Enter configure autologon and run agent on startup? (Y/N) (press enter for N) >
PS C:\agent> .\run.cmd
Scanning for tool capabilities.
Connecting to the server.
2023-07-14 04:44:39Z: Listening for Jobs
```

(Personal access token required for this can be created from User setting-> personal access tokens)



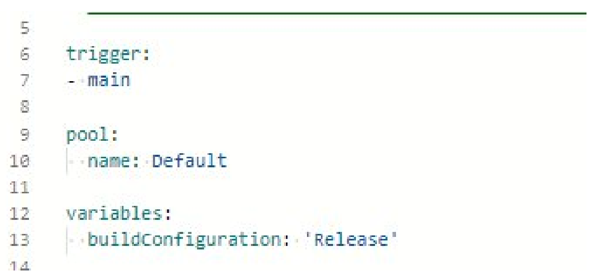
Name	Last run	Current status	Agent version	Enabled
localhost ● Online	4m ago	Idle	3.220.5	On

Now we can see one new agent is added.

Now to change the our build to run in localhost we will have to change the code:

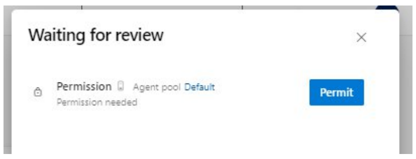
In the YAML under pool, delete the vmImage line and replace it with the following:

```
pool:

name: Default
```



```yaml

trigger:
- main

pool:
  name: Default

variables:
  buildConfiguration: 'Release'

```

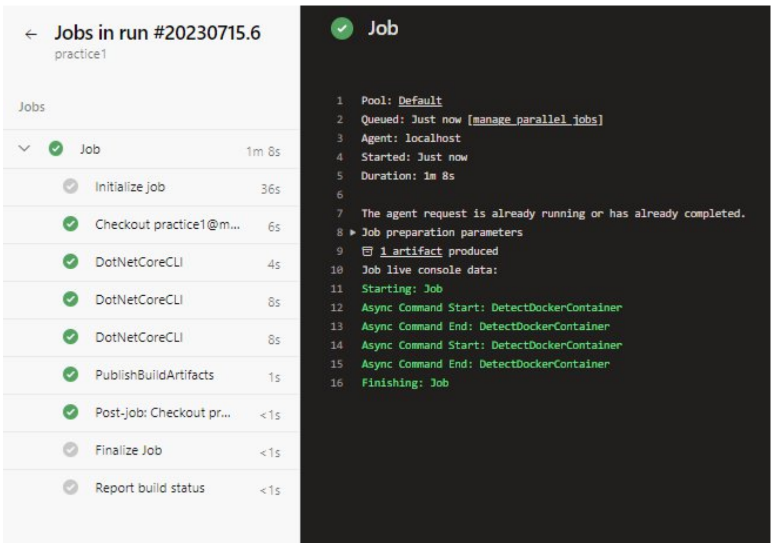
Once you save and run it will ask for permission to use the default pool, you can confirm.



Waiting for review

Permission Agent pool Default
Permission needed

Now if you see the build has completed



Jobs in run #20230715.6
practice1

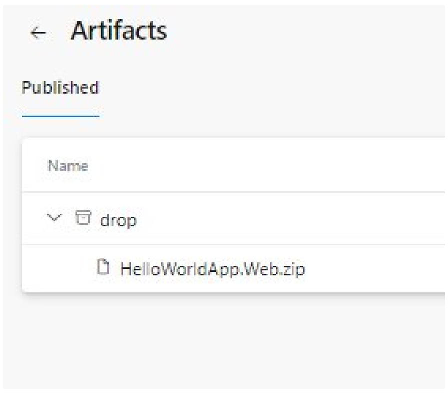
Jobs

Job	1m 8s
Initialize job	36s
Checkout practice1@m...	6s
DotNetCoreCLI	4s
DotNetCoreCLI	8s
DotNetCoreCLI	8s
PublishBuildArtifacts	1s
Post-job: Checkout pr...	<1s
Finalize Job	<1s
Report build status	<1s

Job

```
Pool: Default
Queued: Just now [manage parallel jobs]
Agent: localhost
Started: Just now
Duration: 1m 8s

The agent request is already running or has already completed.
▸ Job preparation parameters
1 artifact produced
Job live console data:
Starting: Job
Async Command Start: DetectDockerContainer
Async Command End: DetectDockerContainer
Async Command Start: DetectDockerContainer
Async Command End: DetectDockerContainer
Finishing: Job
```

You can go to the artifact created:



Artifacts

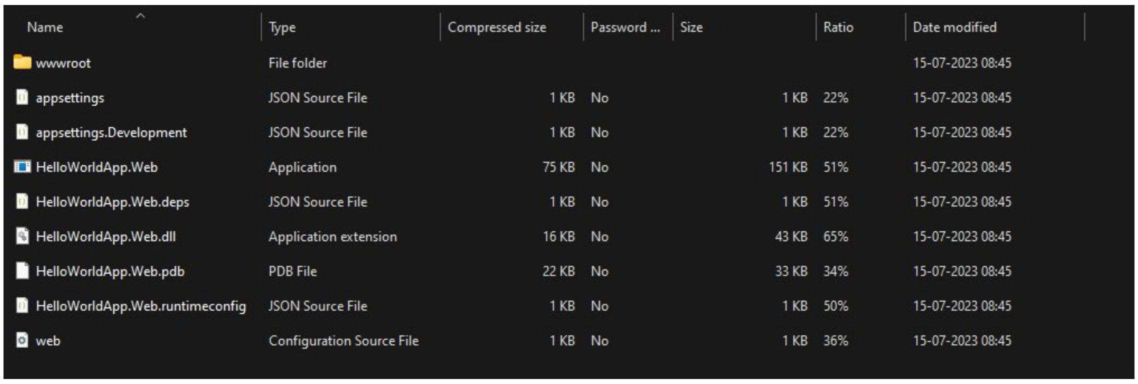
Published

Name

drop

HelloWorldApp.Web.zip

You will see the zip created and if you download and open you will see all the required files:

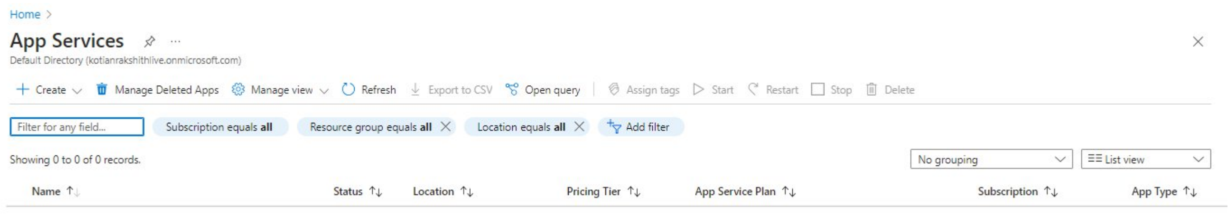


Name	Type	Compressed size	Password ...	Size	Ratio	Date modified
wwwroot	File folder					15-07-2023 08:45
appsettings	JSON Source File	1 KB	No	1 KB	22%	15-07-2023 08:45
appsettings.Development	JSON Source File	1 KB	No	1 KB	22%	15-07-2023 08:45
HelloWorldApp.Web	Application	75 KB	No	151 KB	51%	15-07-2023 08:45
HelloWorldApp.Web.deps	JSON Source File	1 KB	No	1 KB	51%	15-07-2023 08:45
HelloWorldApp.Web.dll	Application extension	16 KB	No	43 KB	65%	15-07-2023 08:45
HelloWorldApp.Web.pdb	PDB File	22 KB	No	33 KB	34%	15-07-2023 08:45
HelloWorldApp.Web.runtimeconfig	JSON Source File	1 KB	No	1 KB	50%	15-07-2023 08:45
web	Configuration Source File	1 KB	No	1 KB	36%	15-07-2023 08:45

5. **<u>Create an app service:</u>**

If we want to deploy our app to an app service we need to create an app service

We will go to Azure portal now and navigate to app service:



Home >

App Services

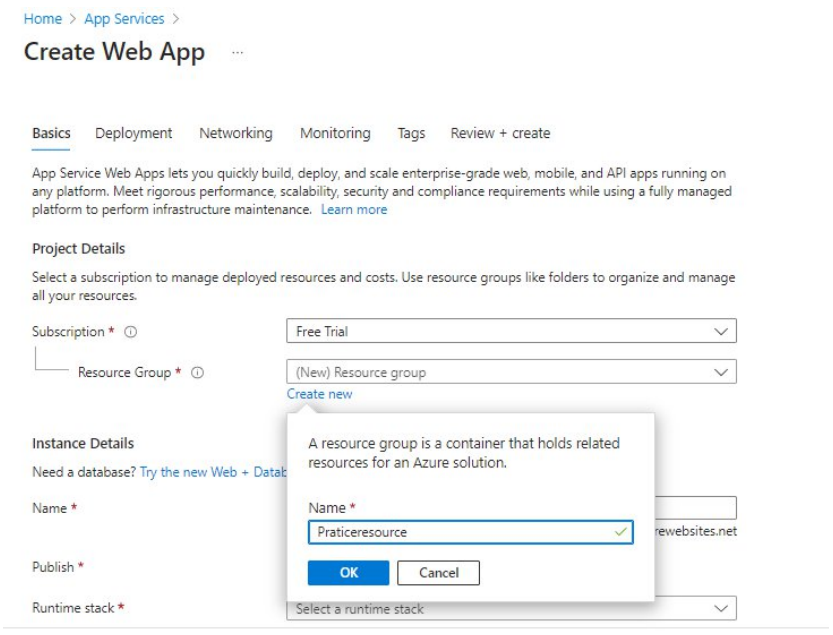
Default Directory (kotianrakshithlive.onmicrosoft.com)

Showing 0 to 0 of 0 records.

Name	Status	Location	Pricing Tier	App Service Plan	Subscription	App Type

Click create,

Then we will add new resource group



Home > App Services >

Create Web App

App Service Web Apps lets you quickly build, deploy, and scale enterprise-grade web, mobile, and API apps running on any platform. Meet rigorous performance, scalability, security and compliance requirements while using a fully managed platform to perform infrastructure maintenance. Learn more

Project Details

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *: Free Trial

Resource Group *: (New) Resource group

Create new

Instance Details

Need a database? Try the new Web + Datab

Name *

rewebsites.net

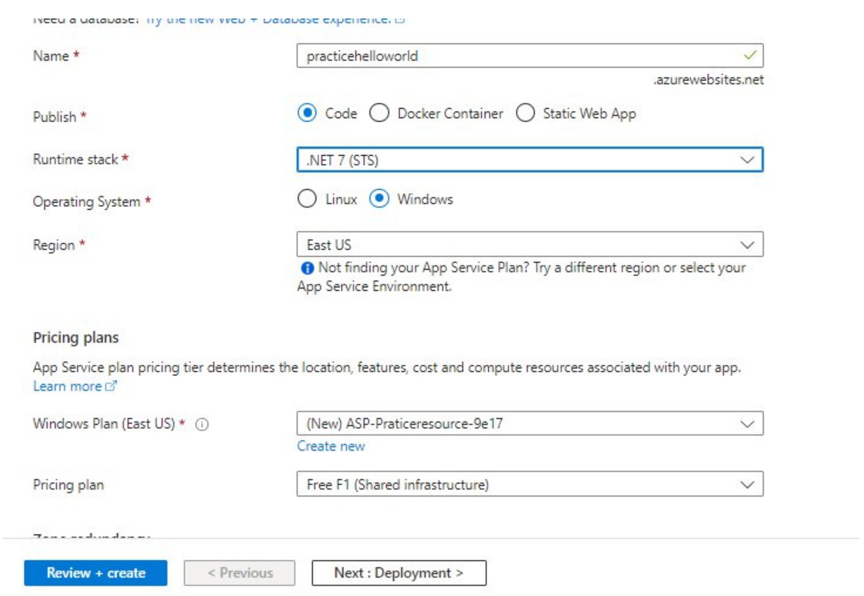
Publish *

Runtime stack *: Select a runtime stack

A resource group is a container that holds related resources for an Azure solution.

Name *: Praticeresource

Then give a name and .net runtime stack


Name * practicehelloworld .azurewebsites.net

Publish * [✓ Code] Docker Container Static Web App

Runtime stack * .NET 7 (STS)

Operating System * Linux [✓ Windows]

Region * East US

Not finding your App Service Plan? Try a different region or select your App Service Environment.

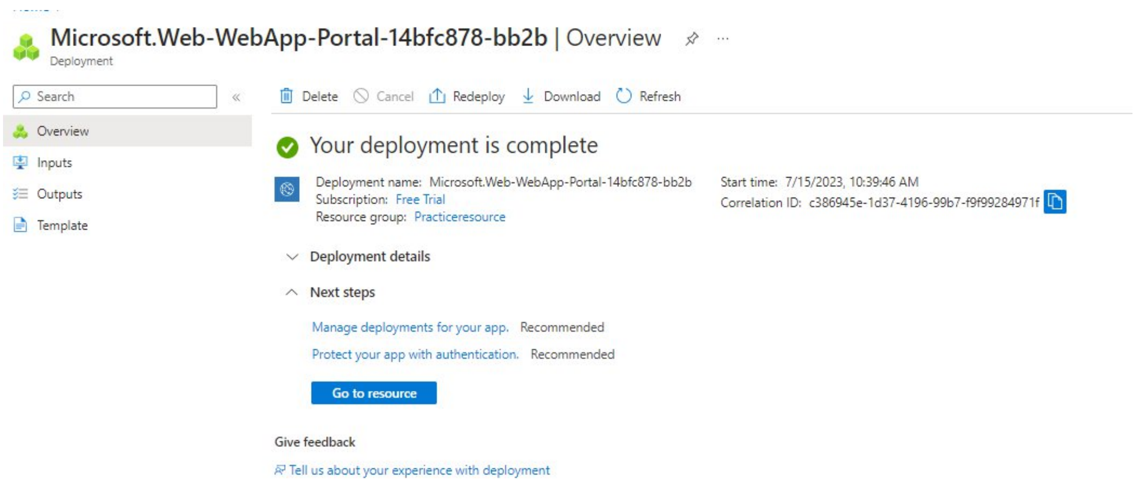
Pricing plans

App Service plan pricing tier determines the location, features, cost and compute resources associated with your app. Learn more

Windows Plan (East US) * (New) ASP-Praticeresource-9e17 Create new

Pricing plan Free F1 (Shared infrastructure)

Then you can review and create:


Microsoft.Web-WebApp-Portal-14bfc878-bb2b | Overview

Deployment

Your deployment is complete

Deployment name: Microsoft.Web-WebApp-Portal-14bfc878-bb2b
Subscription: Free Trial
Resource group: Practiceresource

Start time: 7/15/2023, 10:39:46 AM
Correlation ID: c386945e-1d37-4196-99b7-f9f99284971f

Deployment details

Next steps

Manage deployments for your app. Recommended

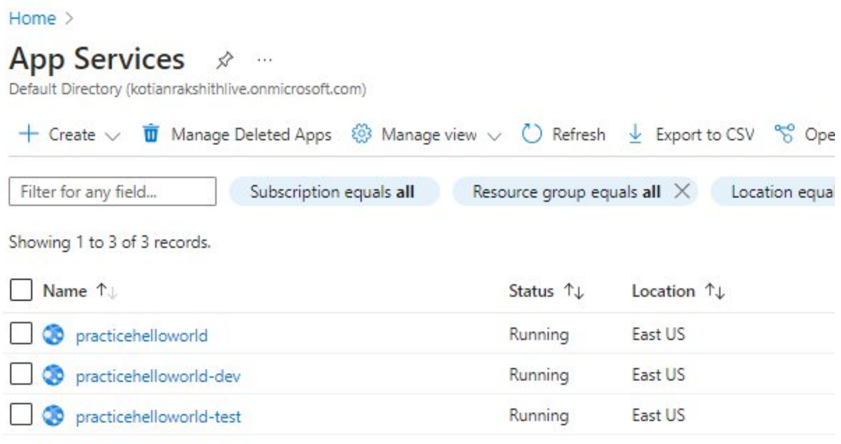
Protect your app with authentication. Recommended

Give feedback

Tell us about your experience with deployment

Simiarly we wil also create web app for dev and test environment.

You should finally have three web app like below:


Home >

App Services

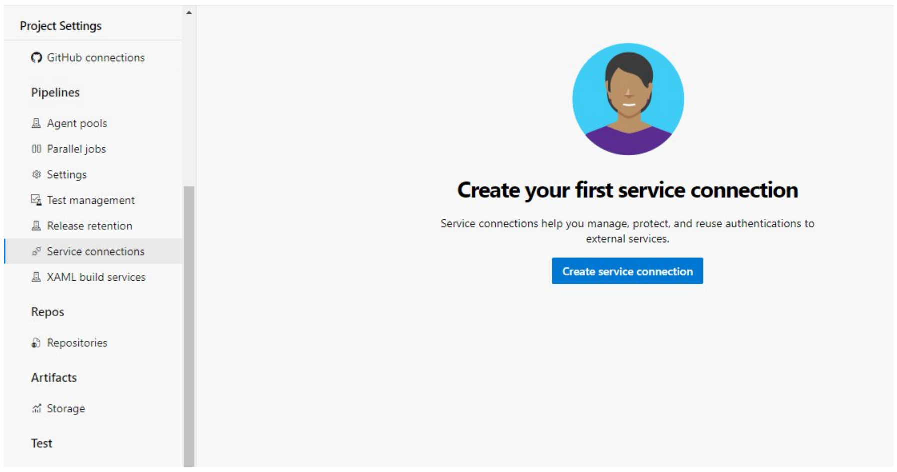
Default Directory (kotianrakshithlive.onmicrosoft.com)

Showing 1 to 3 of 3 records.

Name	Status	Location
practicehelloworld	Running	East US
practicehelloworld-dev	Running	East US
practicehelloworld-test	Running	East US

6. <u>Connect Azure DevOps with Azure Portal subscription using Azure AD:</u>

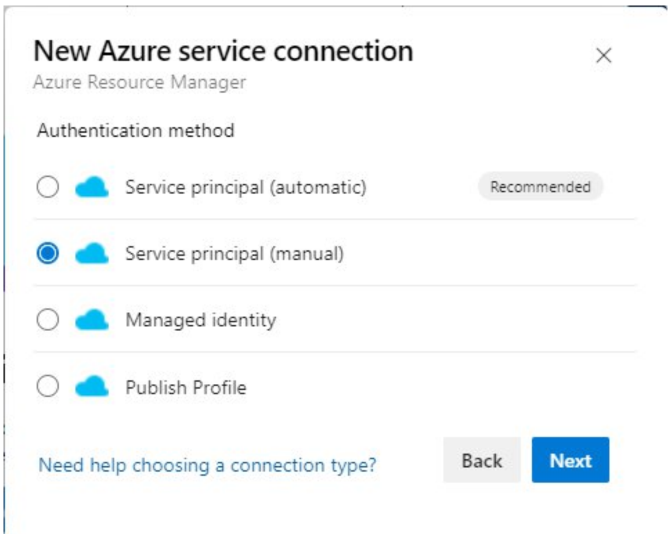
We will go to project settings in Azure DevOps and Service connections:



Create your first service connection

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

We will select service principal:



New Azure service connection

Azure Resource Manager

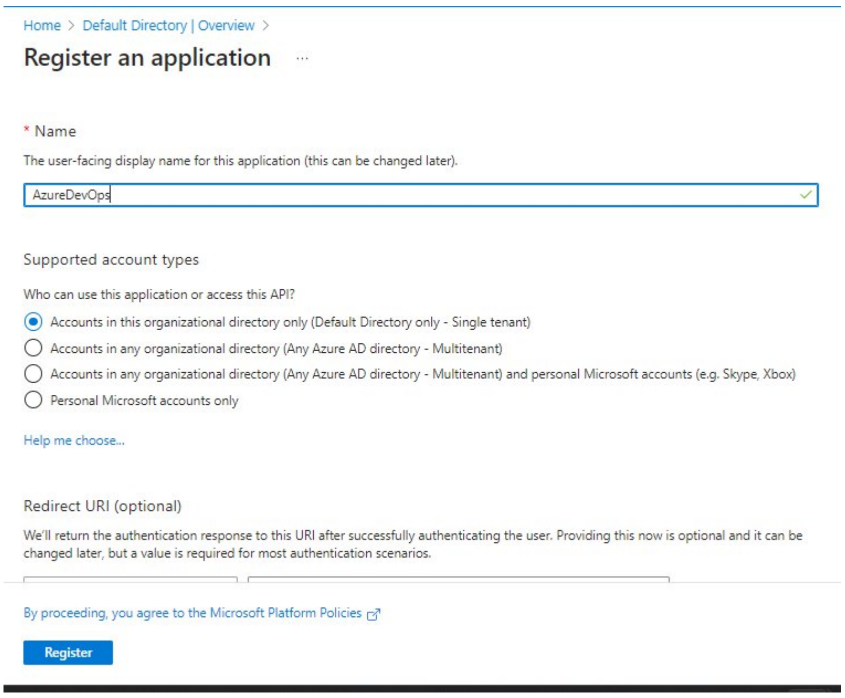
Authentication method

- Service principal (automatic) — Recommended
- [✓ Service principal (manual)]
- Managed identity
- Publish Profile

As we need some details in the next page to be filled, we will open Azure AD(soon will be renamed to Entra ID)

Add an app registration:

We will name it AzureDevOps and register


Home > Default Directory | Overview >

Register an application

* Name

The user-facing display name for this application (this can be changed later).

AzureDevOps

Supported account types

Who can use this application or access this API?

- [✓ Accounts in this organizational directory only (Default Directory only - Single tenant)]
- Accounts in any organizational directory (Any Azure AD directory - Multitenant)
- Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)
- Personal Microsoft accounts only

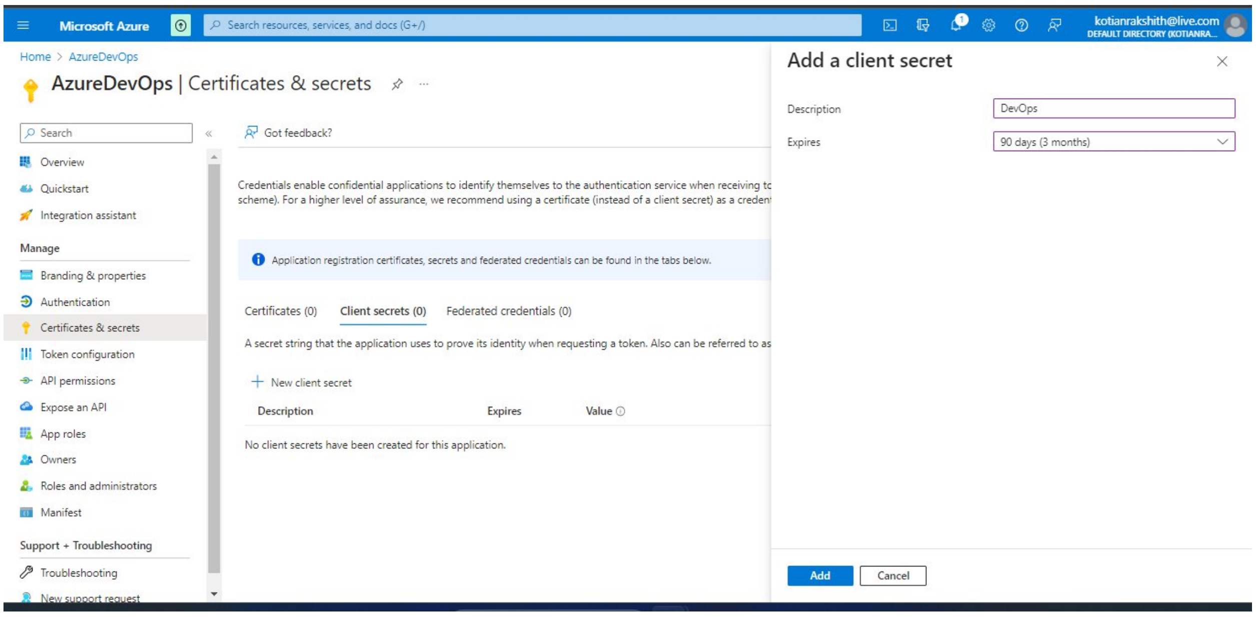
Help me choose...

Redirect URI (optional)

We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

By proceeding, you agree to the Microsoft Platform Policies

We will also go to clients and secrets and create a secret:



Home > AzureDevOps

AzureDevOps | Certificates & secrets

Credentials enable confidential applications to identify themselves to the authentication service when receiving to scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a creden

Application registration certificates, secrets and federated credentials can be found in the tabs below.

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as

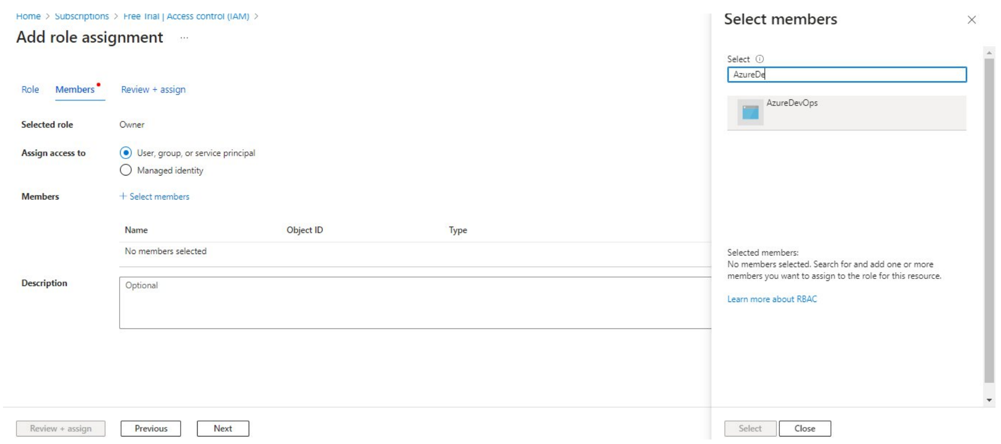
Description	Expires	Value

No client secrets have been created for this application.

Add a client secret

Description	DevOps
Expires	90 days (3 months)

We will also go to subscription and add a new role assignment to the app registration we just created.


Home > Subscriptions > Free Trial | Access control (IAM) >

Add role assignment

Selected role: Owner

Assign access to: [✓ User, group, or service principal] Managed identity

Members: + Select members

Name	Object ID	Type
No members selected		

Description: Optional

Select members

Select: AzureDe

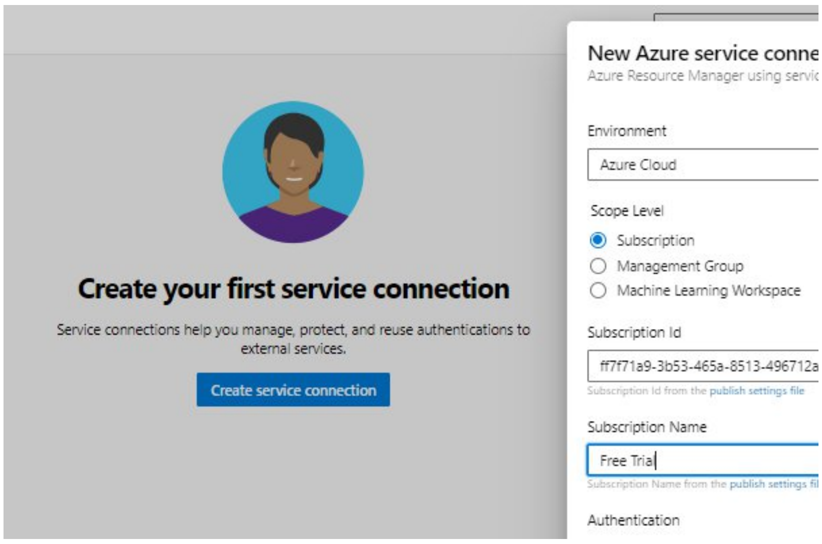
AzureDevOps

Selected members:
No members selected. Search for and add one or more members you want to assign to the role for this resource.

Learn more about RBAC

Now using all the info we will fill the Azure DevOps service connection fields:

First we will give subscription id and name


Create your first service connection
Service connections help you manage, protect, and reuse authentications to external services.
Create service connection

New Azure service conne

Azure Resource Manager using servic

Environment: Azure Cloud

Scope Level: [✓ Subscription] Management Group Machine Learning Workspace

Subscription Id: ff7f71a9-3b53-465a-8513-496712a

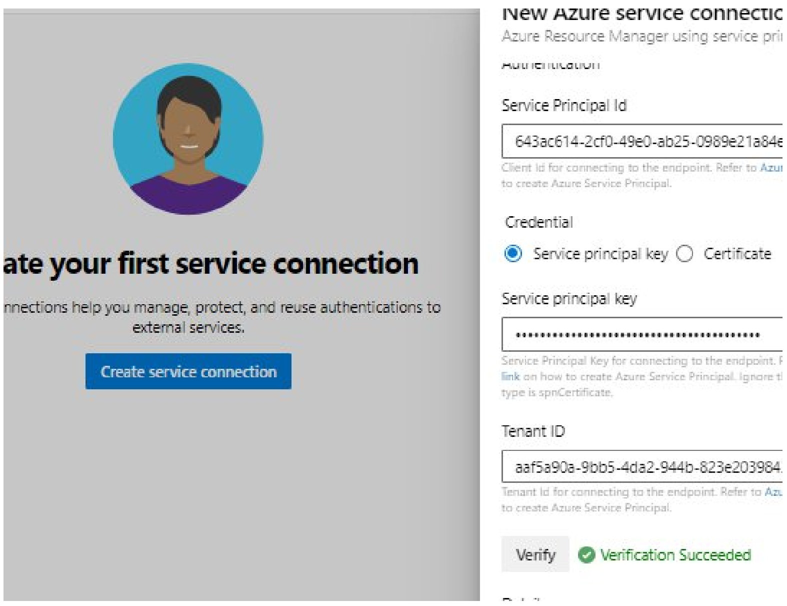
Subscription Id from the publish settings file

Subscription Name: Free Trial

Subscription Name from the publish settings fil

Authentication

Then we will give Service Principal Id(client ID) Service principal key(secret value), Tenant ID.


ate your first service connection
nnections help you manage, protect, and reuse authentications to external services.
Create service connection

New Azure service connectic

Azure Resource Manager using service pri

Authentication

Service Principal Id: 643ac614-2cf0-49e0-ab25-0989e21a84e

Client Id for connecting to the endpoint. Refer to Azur to create Azure Service Principal.

Credential: [✓ Service principal key] Certificate

Service principal key: ••

Service Principal Key for connecting to the endpoint. F link on how to create Azure Service Principal. Ignore t type is spnCertificate.

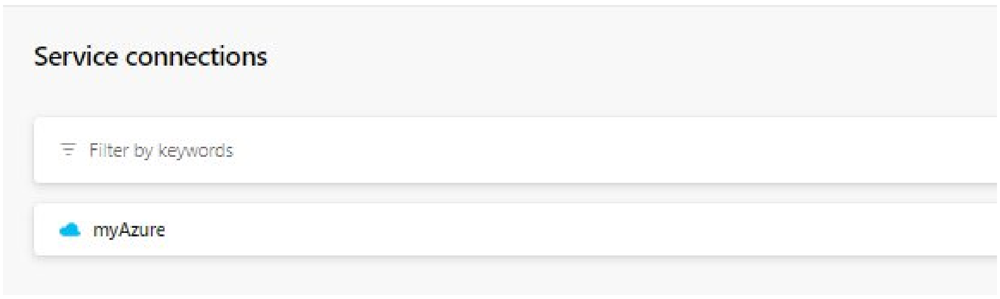
Tenant ID: aaf5a90a-9bb5-4da2-944b-823e203984

Tenant Id for connecting to the endpoint. Refer to Azu to create Azure Service Principal.

Verify ✓ Verification Succeeded

Then when we verify it should be successful otherwise it may be missing something.

When we verify and save a new service connection should be added:


Service connections

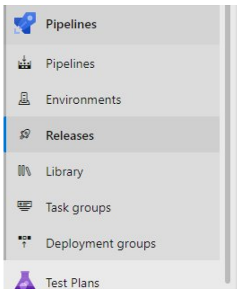
Filter by keywords

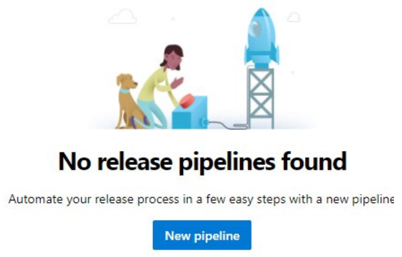
myAzure

This will help you to find the web app created in Azure portal in Azure DevOps portal.

7. **Create a release pipeline:**

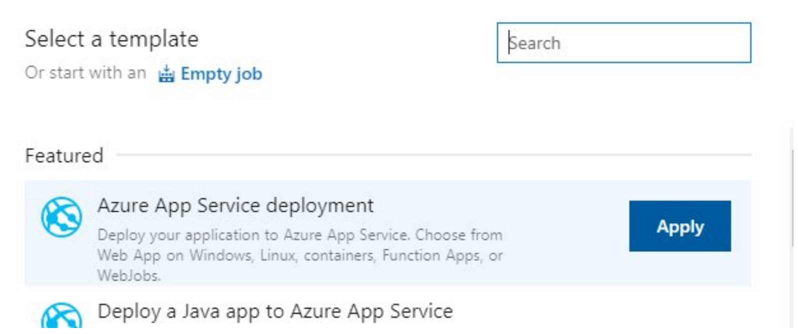
In the Azure DevOps go to pipeline and then releases and then click new pipeline:




No release pipelines found

Automate your release process in a few easy steps with a new pipeline

In template chose ‘Azure App Service Deployment’:


Select a template

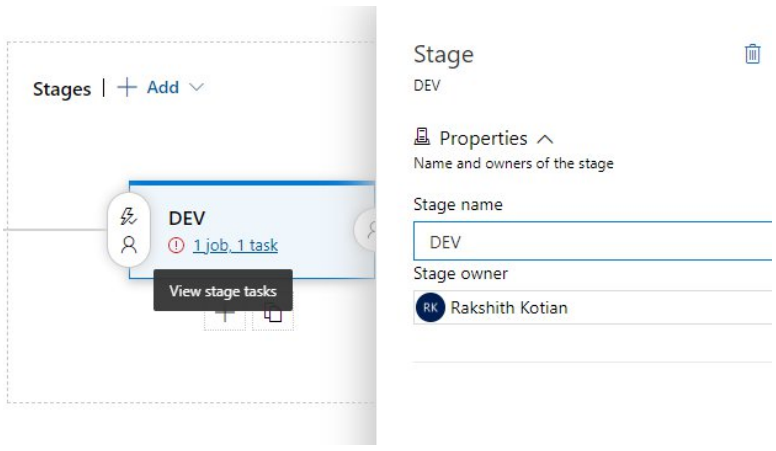
Or start with an Empty job

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

Name it and got to tasks:



Stage

DEV

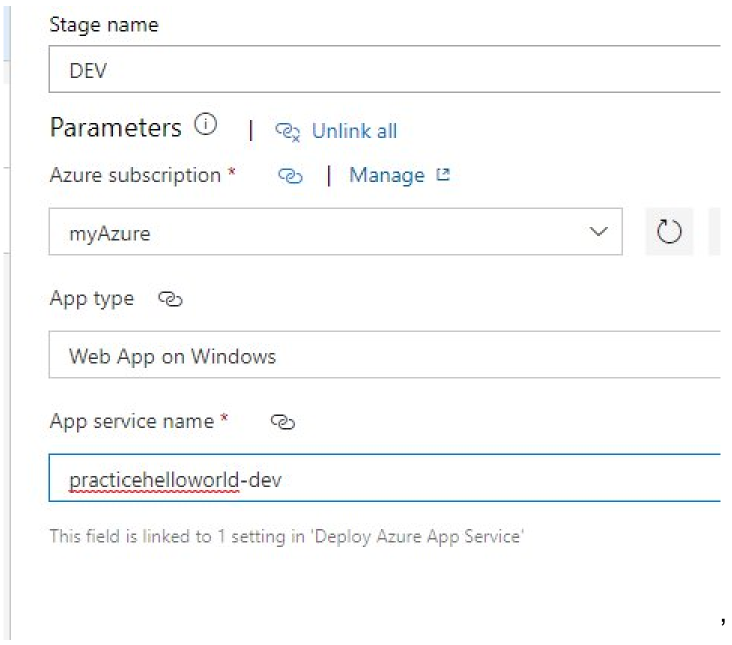
Properties

Name and owners of the stage

Stage name: DEV

Stage owner: Rakshith Kotian

Here chose your subscription and the webapp:



Stage name

DEV

Parameters ⓘ | Unlink all

Azure subscription * | Manage

myAzure

App type

Web App on Windows

App service name *

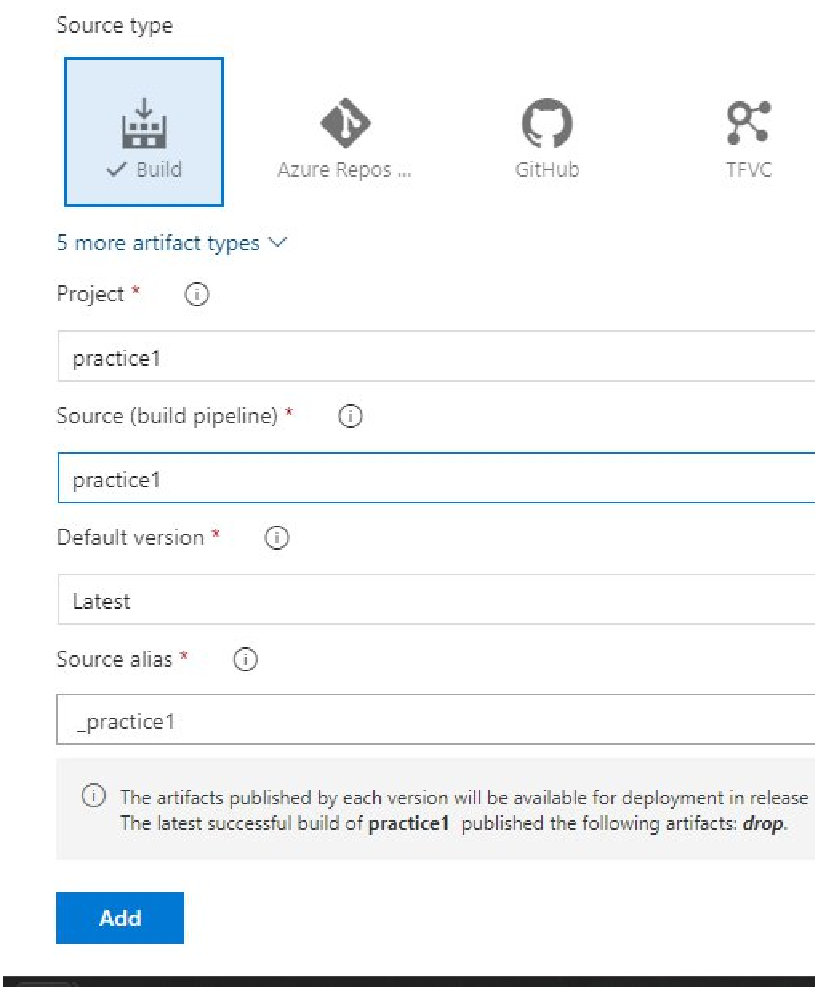
practicehelloworld-dev

This field is linked to 1 setting in 'Deploy Azure App Service'

,

Save it.

Now go to the pipeline and add artifact:



Source type

Project * ⓘ

practice1

Source (build pipeline) * ⓘ

practice1

Default version * ⓘ

Latest

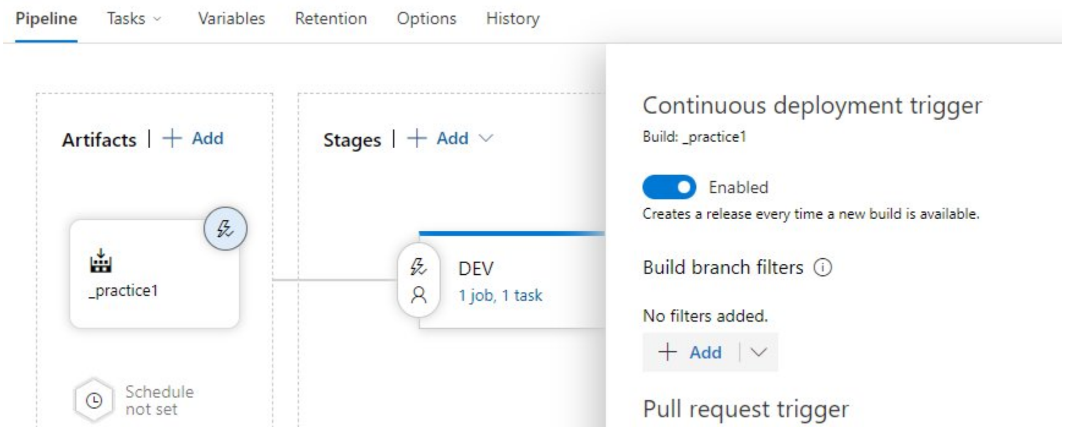
Source alias * ⓘ

_practice1

ⓘ The artifacts published by each version will be available for deployment in release
The latest successful build of **practice1** published the following artifacts: ***drop***.

You can select your project where build is done and click add

You can also enable continuous deployment using the trigger:



Continuous deployment trigger

Build: _practice1

Enabled

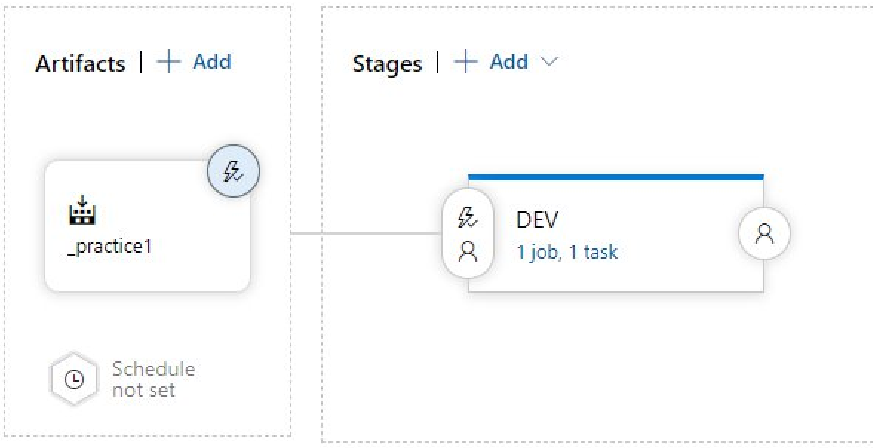
Creates a release every time a new build is available.

Build branch filters

No filters added.

Pull request trigger

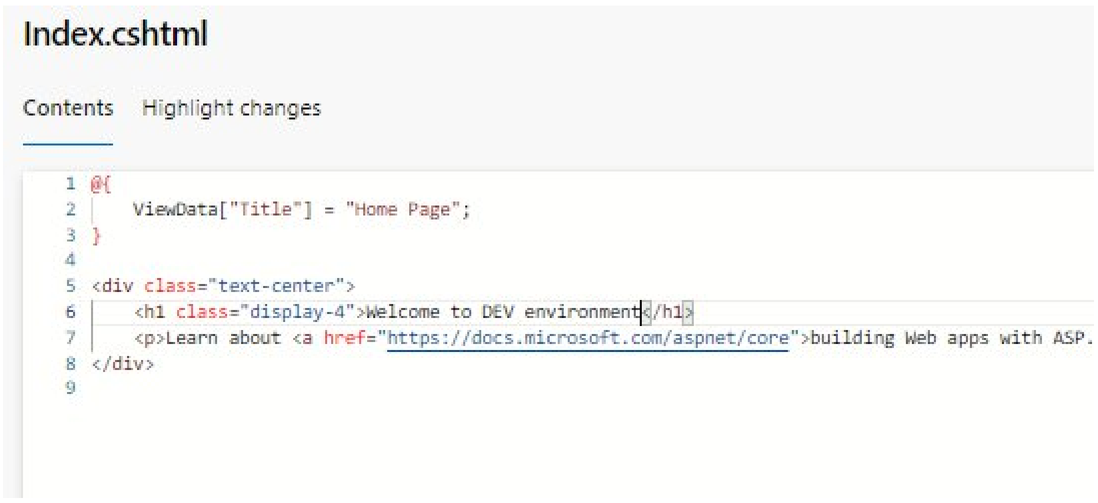
Now we have pipeline till Dev.



Either we can run this manually, but as there is automated CI/CD lets just change small code and it should build and deploy automatically:

We are editing HelloWorldApp.Web/Views/Home/Index.cshtml

(This is the index page of our web app)



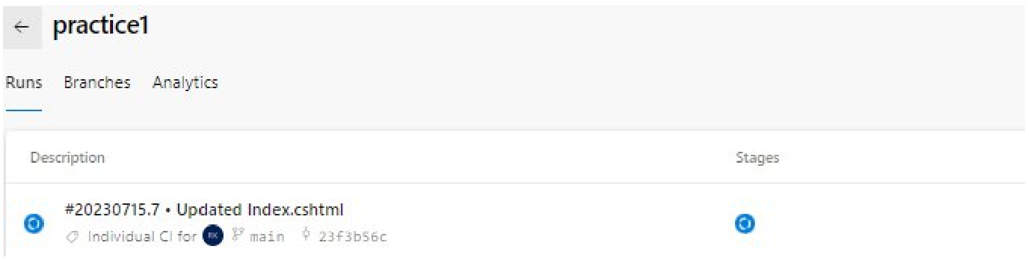
Index.cshtml

Contents   Highlight changes

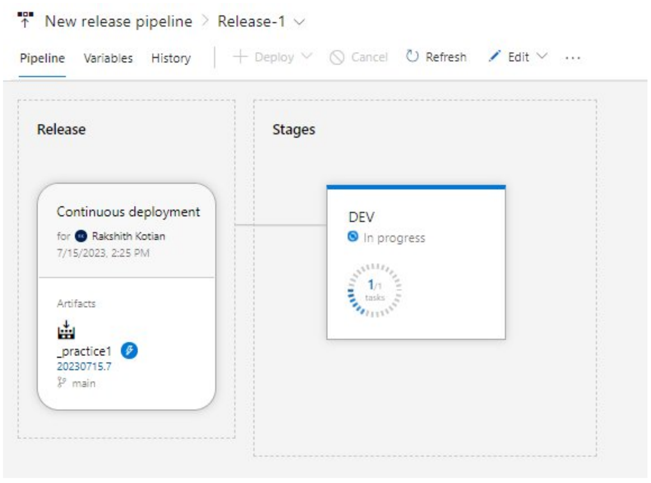
```
@{
    ViewData["Title"] = "Home Page";
}

<div class="text-center">
    <h1 class="display-4">Welcome to DEV environment</h1>
    <p>Learn about <a href="https://docs.microsoft.com/aspnet/core">building Web apps with ASP.
</div>

```

We will commit this.

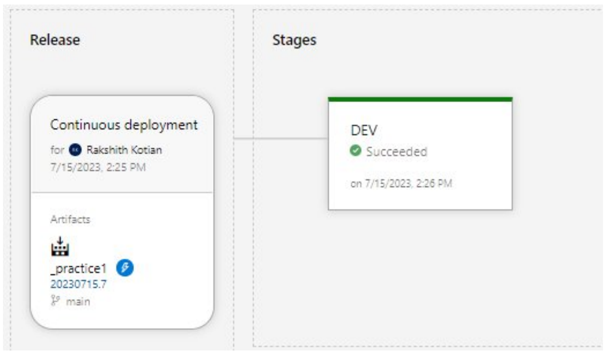
First we will see pipeline running:



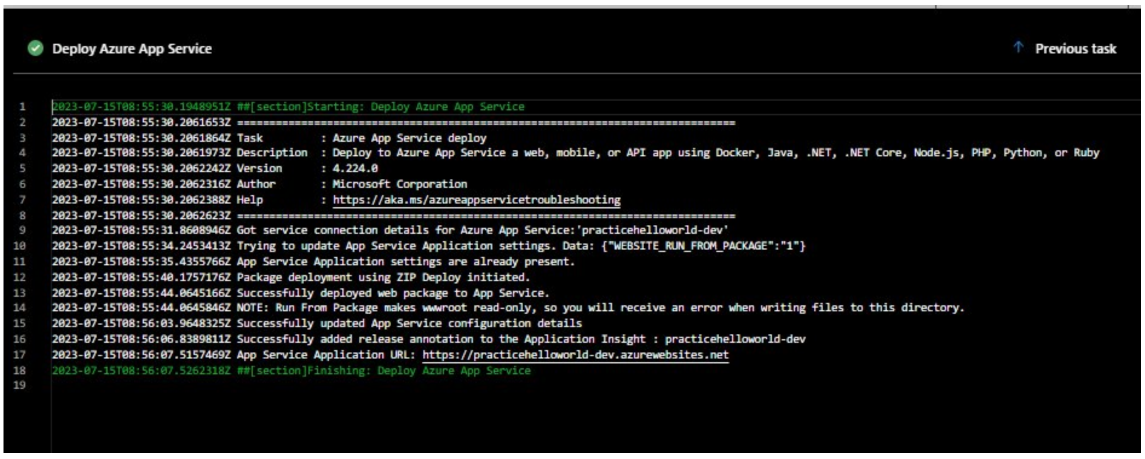
Once its successfull we will see release running:



Then you can see successful release pipline:

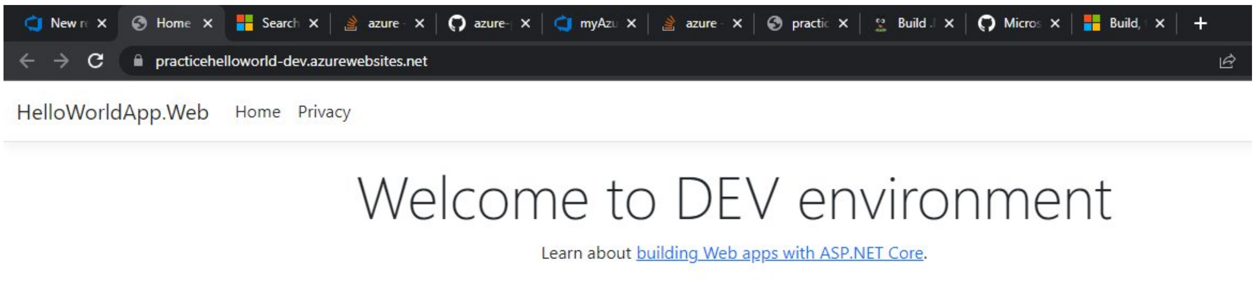


Then you can check its log and go to the url mentioned:



```
2023-07-15T08:55:30.1948951Z ##[section]Starting: Deploy Azure App Service
2023-07-15T08:55:30.2061653Z ==============================================================================
2023-07-15T08:55:30.2061864Z Task         : Azure App Service deploy
2023-07-15T08:55:30.2061973Z Description  : Deploy to Azure App Service a web, mobile, or API app using Docker, Java, .NET, .NET Core, Node.js, PHP, Python, or Ruby
2023-07-15T08:55:30.2062242Z Version      : 4.224.0
2023-07-15T08:55:30.2062316Z Author       : Microsoft Corporation
2023-07-15T08:55:30.2062388Z Help         : https://aka.ms/azureappservicetroubleshooting
2023-07-15T08:55:30.2062623Z ==============================================================================
2023-07-15T08:55:31.8608946Z Got service connection details for Azure App Service:'practicehelloworld-dev'
2023-07-15T08:55:34.2453413Z Trying to update App Service Application settings. Data: {"WEBSITE_RUN_FROM_PACKAGE":"1"}
2023-07-15T08:55:35.4355766Z App Service Application settings are already present.
2023-07-15T08:55:40.1757176Z Package deployment using ZIP Deploy initiated.
2023-07-15T08:55:44.0645166Z Successfully deployed web package to App Service.
2023-07-15T08:55:44.0645846Z NOTE: Run From Package makes wwwroot read-only, so you will receive an error when writing files to this directory.
2023-07-15T08:56:03.9648325Z Successfully updated App Service configuration details
2023-07-15T08:56:06.8389811Z Successfully added release annotation to the Application Insight : practicehelloworld-dev
2023-07-15T08:56:07.5157469Z App Service Application URL: https://practicehelloworld-dev.azurewebsites.net
2023-07-15T08:56:07.5262318Z ##[section]Finishing: Deploy Azure App Service
```

We see that it is successfully deployed and available in the web app url:



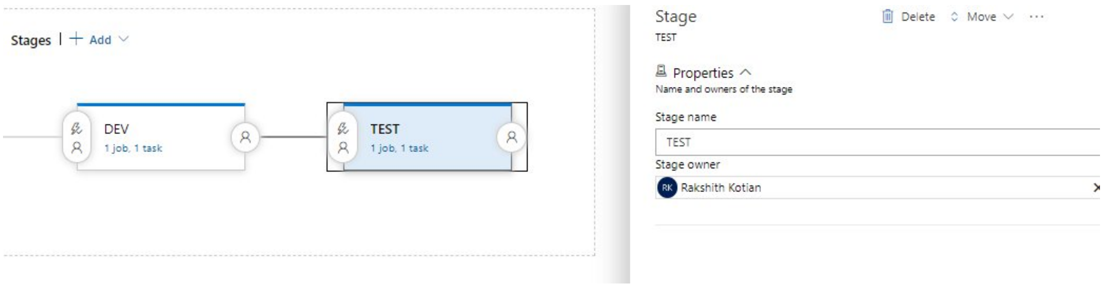
Welcome to DEV environment

Learn about building Web apps with ASP.NET Core.

8. **Deploy till Production:**

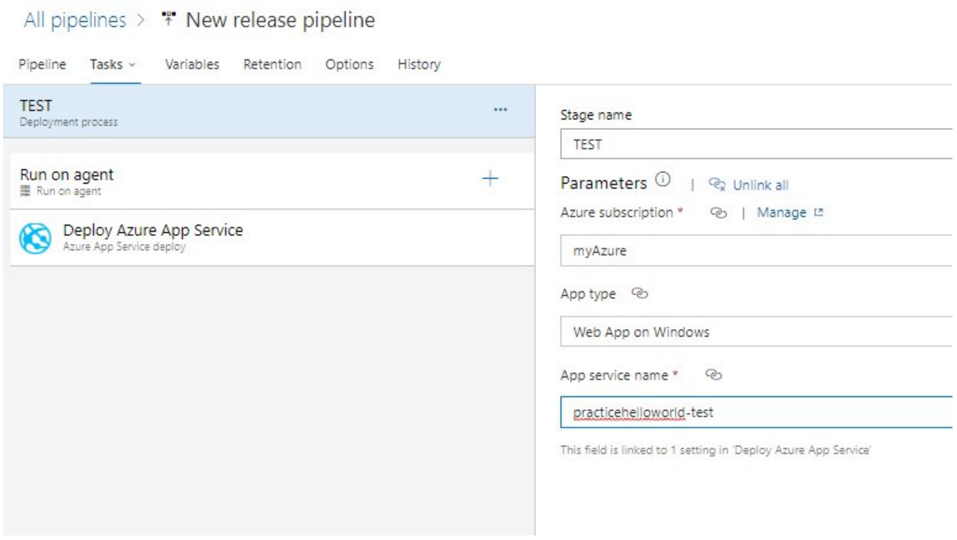
Till now we deployed till dev environment. Lets add test environment and also production environment.

First let us just clone DEV step and rename it to TEST and change the web app:



Stage name: TEST

Stage owner: Rakshith Kotian



Stage name: TEST

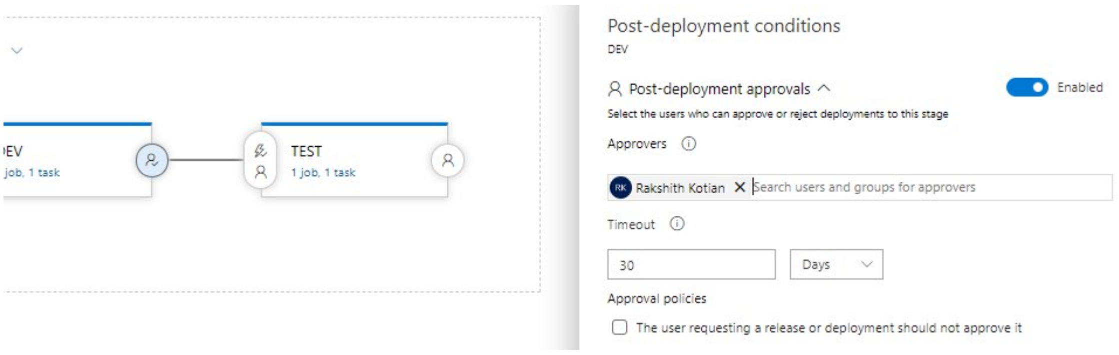
Azure subscription *: myAzure

App type: Web App on Windows

App service name *: practicehelloworld-test

This field is linked to 1 setting in 'Deploy Azure App Service'

We will also add the post deployment condition to require approval between the stages:

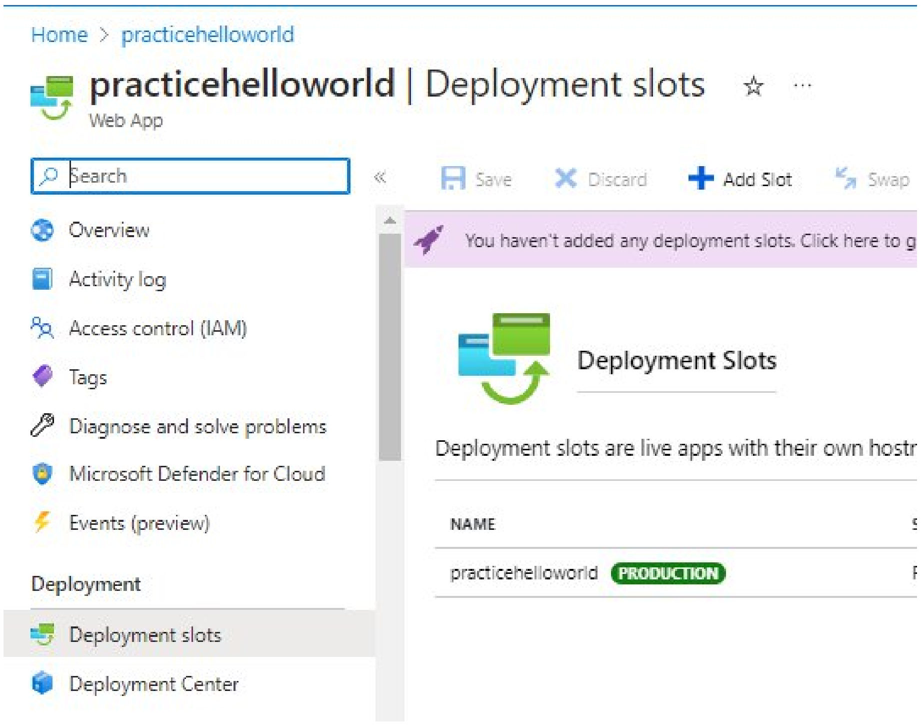


Post-deployment conditions
DEV
Post-deployment approvals
Enabled
Select the users who can approve or reject deployments to this stage
Approvers
Rakshith Kotlan
Search users and groups for approvers
Timeout
30 Days
Approval policies
The user requesting a release or deployment should not approve it

Now before we add production let us add a deployment slot in production web app so we can use the swap between two slots.

(we have to upgrade webapp which provides staging slots)

Now go to the app service and then 'Deploment slots' and click Add slot:

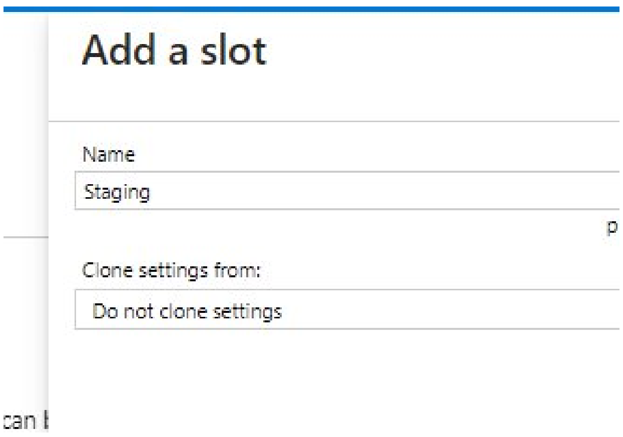


You haven't added any deployment slots. Click here to g

Deployment slots are live apps with their own hostr

NAME
practicehelloworld PRODUCTION

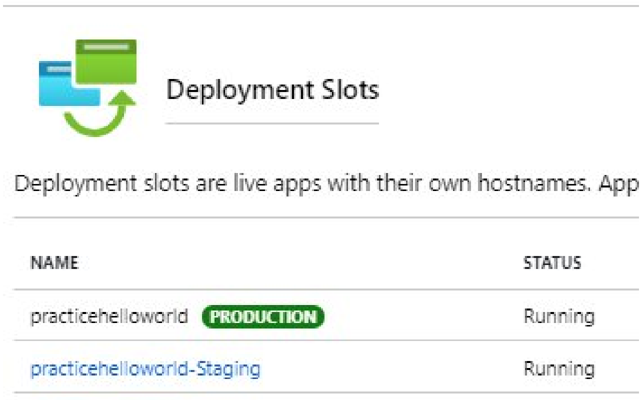
We will name it as staging and add it :



Name
Staging

Clone settings from:
Do not clone settings

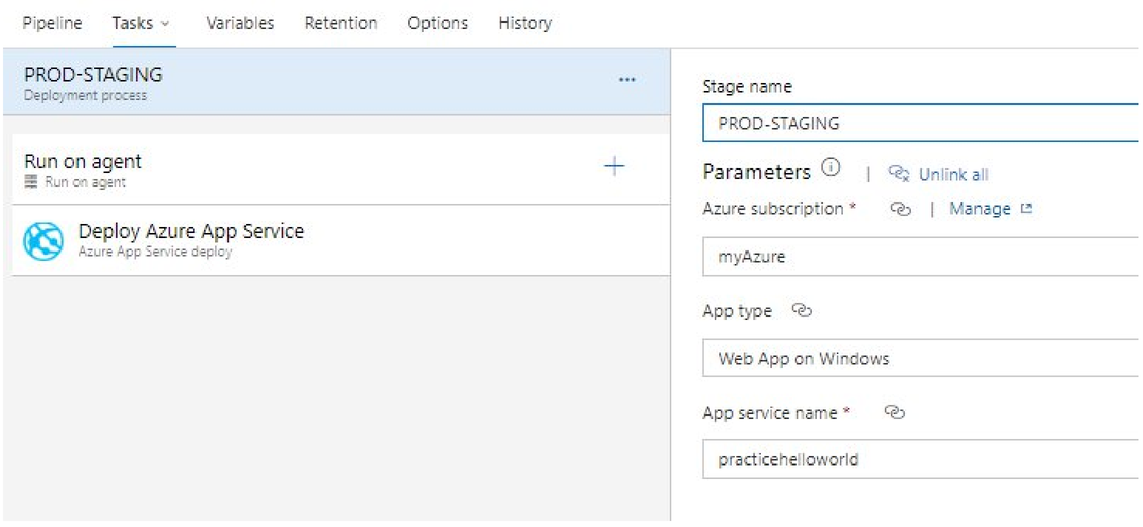
Now you will be abe to see two deployment slots:


Deployment Slots

Deployment slots are live apps with their own hostnames. App

NAME	STATUS
practicehelloworld PRODUCTION	Running
practicehelloworld-Staging	Running

Now lets add new stage for production in release pipeline:



PROD-STAGING
Deployment process

Run on agent
Run on agent

Deploy Azure App Service
Azure App Service deploy

Stage name: PROD-STAGING

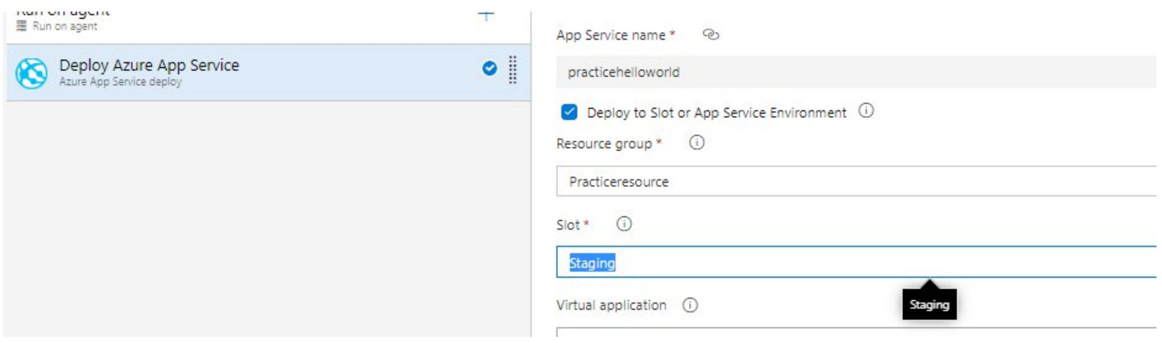
Parameters | Unlink all

Azure subscription * | Manage: myAzure

App type: Web App on Windows

App service name *: practicehelloworld

In the task change deployment to staging slot:


Run on agent
Run on agent

Deploy Azure App Service
Azure App Service deploy

App Service name *: practicehelloworld

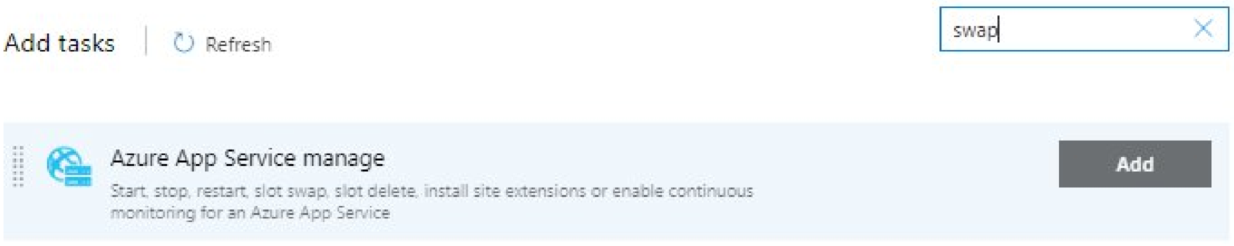
Deploy to Slot or App Service Environment

Resource group *: Practiceresource

Slot *: Staging

Virtual application

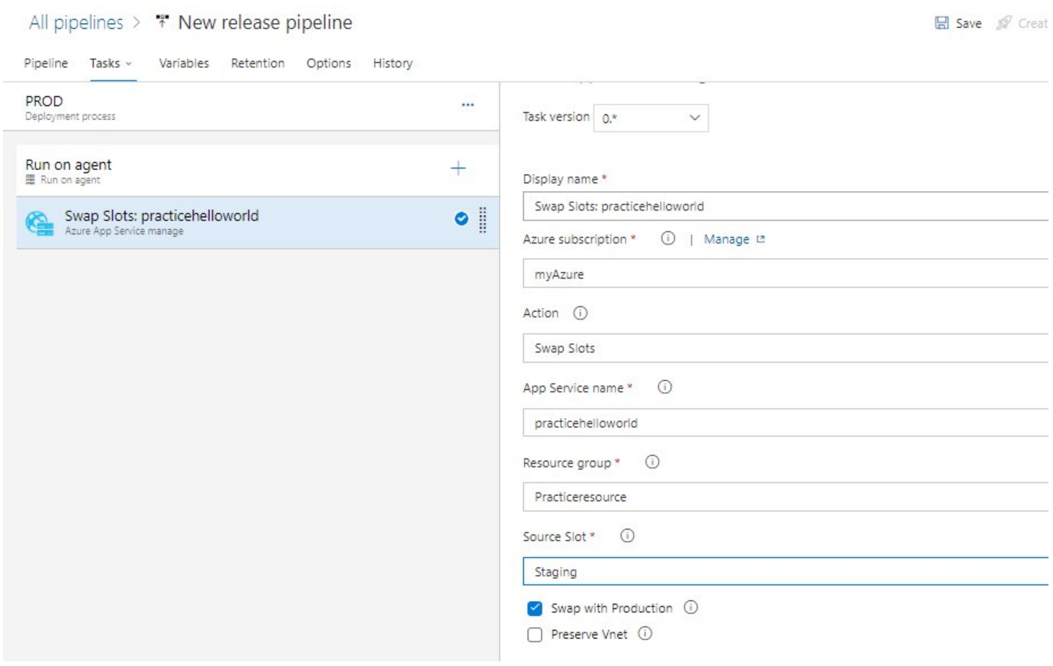
Then add the new step to swap the slots(Azure App Service Manage):



Azure App Service manage

Start, stop, restart, slot swap, slot delete, install site extensions or enable continuous monitoring for an Azure App Service

Now fill the required details:


All pipelines > New release pipeline

Save Creat

Pipeline Tasks Variables Retention Options History

PROD
Deployment process

Run on agent
Run on agent

Swap Slots: practicehelloworld
Azure App Service manage

Task version: 0.*

Display name *: Swap Slots: practicehelloworld

Azure subscription * | Manage: myAzure

Action: Swap Slots

App Service name *: practicehelloworld

Resource group *: Practiceresource

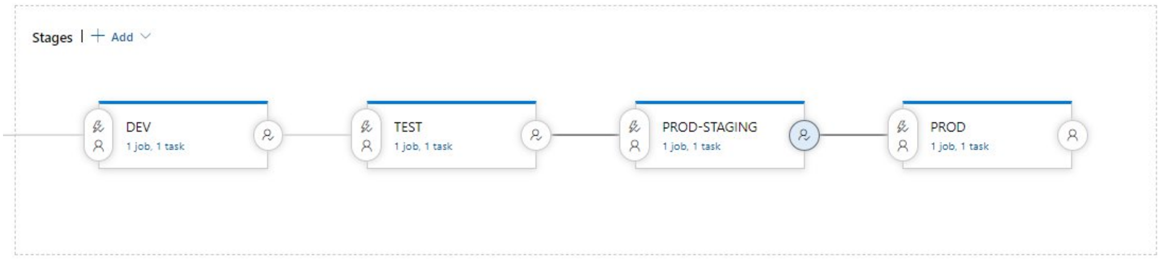
Source Slot *: Staging

[✓] Swap with Production

[ ] Preserve Vnet

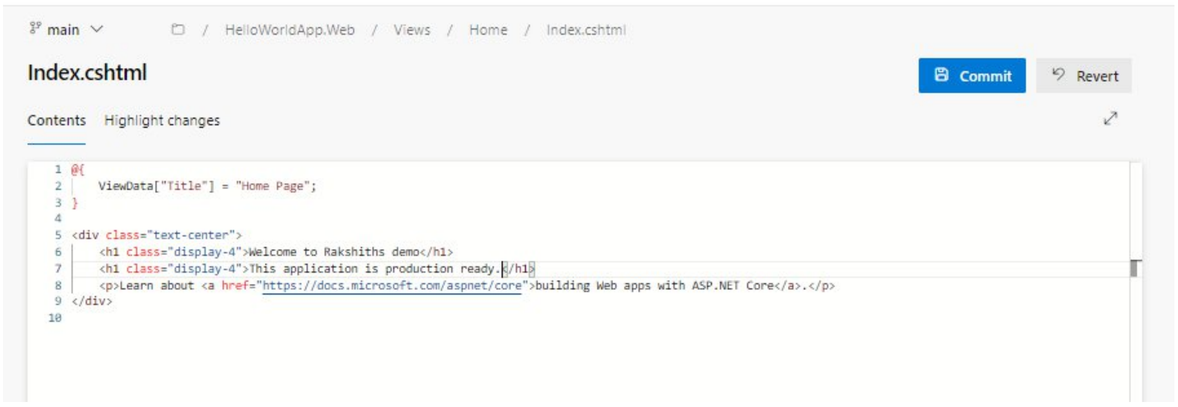
Also make sure you have post deployment conditon for approval in every step.

Now your pipeline stages is complete:



9. <u>**Make final changes and trigger CI/CD:**</u>

Now lets make changes in the index file as we did before:


main / HelloWorldApp.Web / Views / Home / Index.cshtml

Index.cshtml

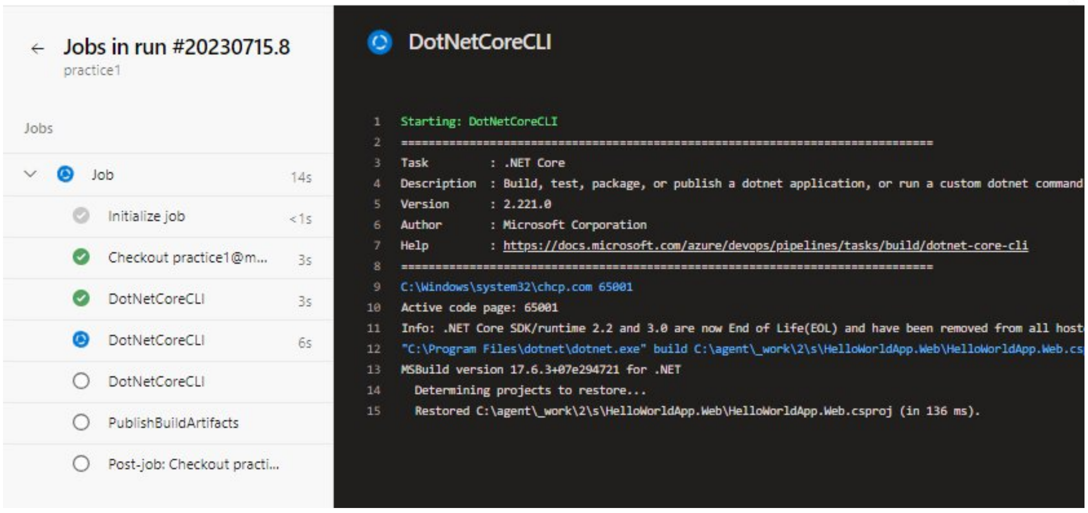
Commit Revert

Contents Highlight changes

```
@{
    ViewData["Title"] = "Home Page";
}

<div class="text-center">
    <h1 class="display-4">Welcome to Rakshiths demo</h1>
    <h1 class="display-4">This application is production ready.</h1>
    <p>Learn about <a href="https://docs.microsoft.com/aspnet/core">building Web apps with ASP.NET Core</a>.</p>
</div>
```

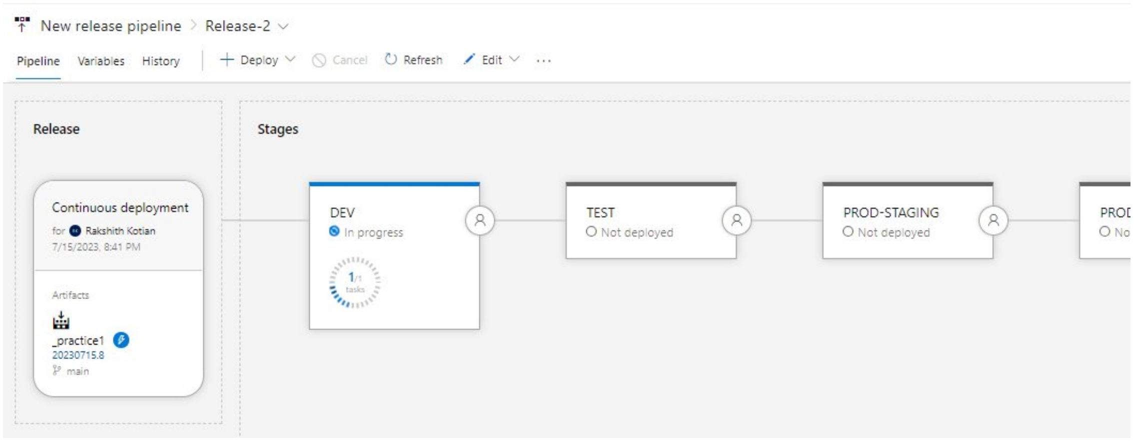
When we commit it it will trigger the CI/CD pipeline:

First build pipeline will run:

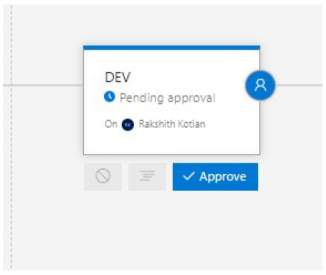


```
Starting: DotNetCoreCLI
==============================================================================
Task         : .NET Core
Description  : Build, test, package, or publish a dotnet application, or run a custom dotnet command
Version      : 2.221.0
Author       : Microsoft Corporation
Help         : https://docs.microsoft.com/azure/devops/pipelines/tasks/build/dotnet-core-cli
==============================================================================
C:\Windows\system32\chcp.com 65001
Active code page: 65001
Info: .NET Core SDK/runtime 2.2 and 3.0 are now End of Life(EOL) and have been removed from all host
"C:\Program Files\dotnet\dotnet.exe" build C:\agent\_work\2\s\HelloWorldApp.Web\HelloWorldApp.Web.cs
MSBuild version 17.6.3+07e294721 for .NET
  Determining projects to restore...
  Restored C:\agent\_work\2\s\HelloWorldApp.Web\HelloWorldApp.Web.csproj (in 136 ms).
```

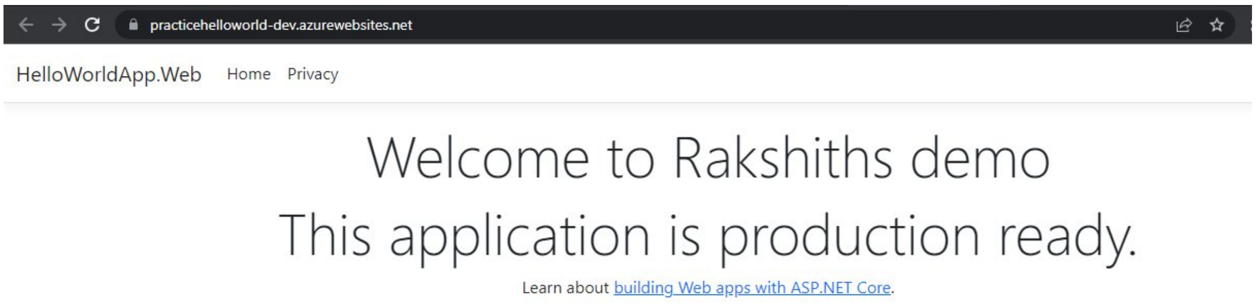
Then release will begin:



Once DEV stage is complete it will wait for approval to proceed to next stage:



When we check the dev app service link we should see our changes reflected:

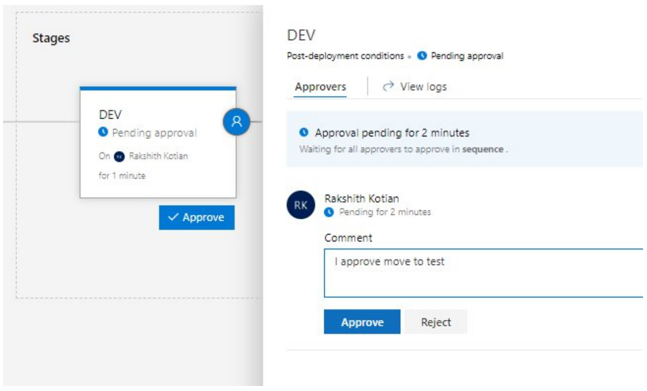

practicehelloworld-dev.azurewebsites.net
HelloWorldApp.Web Home Privacy

Welcome to Rakshiths demo

This application is production ready.

Learn about building Web apps with ASP.NET Core.

Now let us approve:



DEV

Post-deployment conditions • Pending approval

Approvers | View logs

Approval pending for 2 minutes

Waiting for all approvers to approve in **sequence** .

Rakshith Kotian

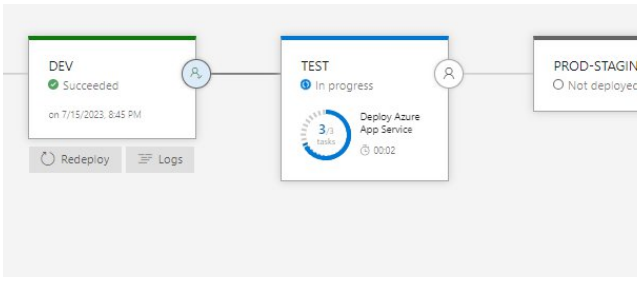
Pending for 2 minutes

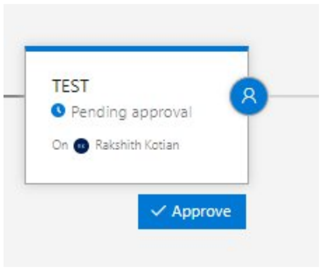
Comment

I approve move to test

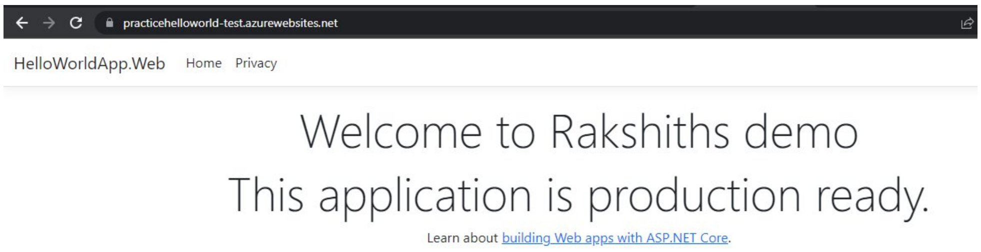
Approve Reject

Now test stage will start:





Similarly we will check the test link:



practicehelloworld-test.azurewebsites.net

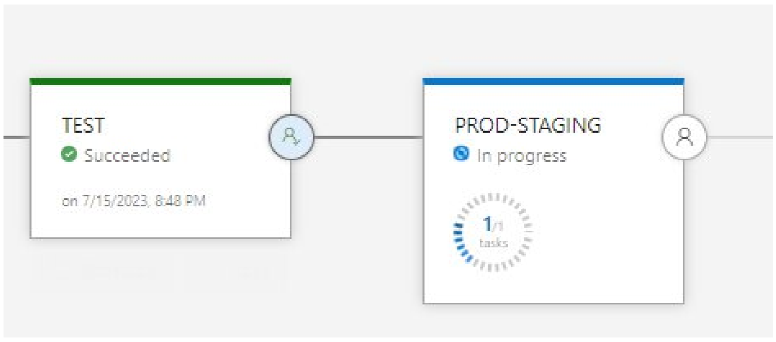
HelloWorldApp.Web Home Privacy

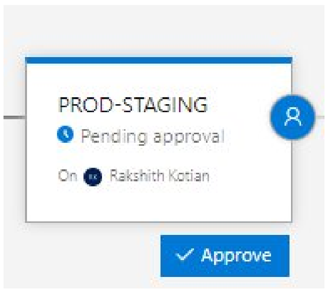
Welcome to Rakshiths demo

This application is production ready.

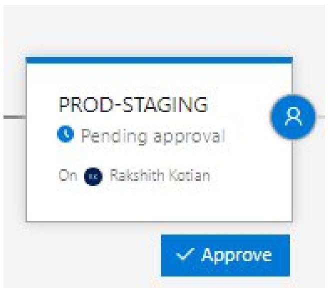
Learn about building Web apps with ASP.NET Core.

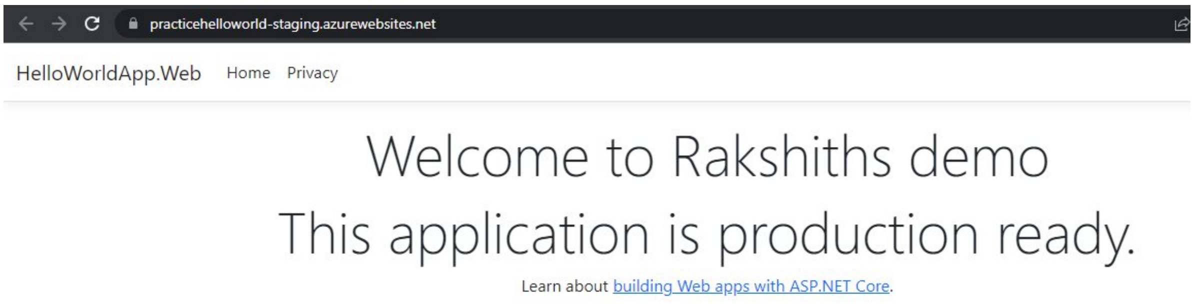
Now after approving again it will deploy to staging slot:





We will now check the staging link:



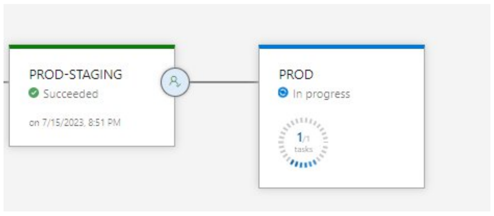


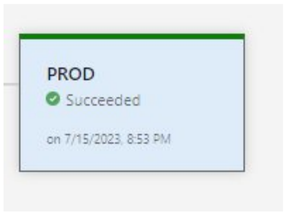
Welcome to Rakshiths demo

This application is production ready.

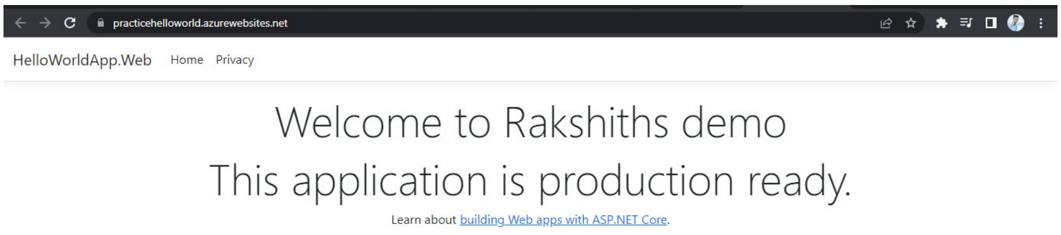
Learn about building Web apps with ASP.NET Core.

Finally we approve the swapping:





Now let us go to our final production link:



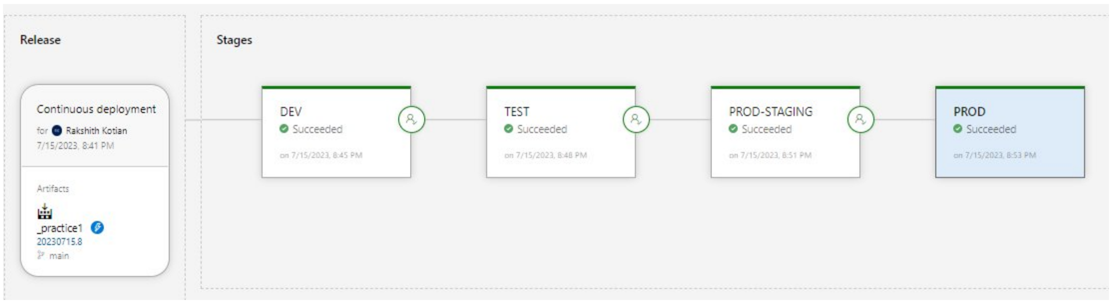
Welcome to Rakshiths demo

This application is production ready.

Learn about building Web apps with ASP.NET Core.

We see that it is successfully deployed.

We can see the whole pipeline completed:



That completes this detailed deployment of our application through Azure DevOps.