FIT5032 CI/CD of .NET 5.0 application GitHub Actions and Azure App Service

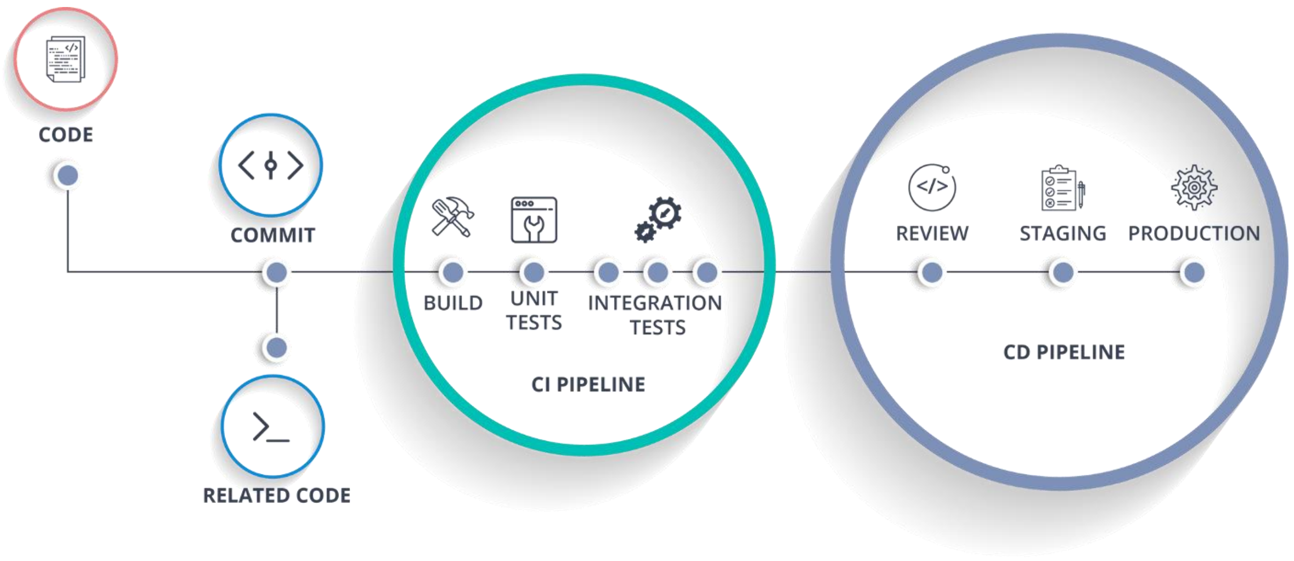
Author: Roshan Thirikkott

POST-CLASS ACTIVITIES

Introduction

Continuous Integration (**CI**) is the practice where developers regularly merge their code changes into a central repository, after which automated builds and tests are run. When a team of developers are working on a shared repository, frequent code updates make it easier to merge changes from other team members. CI helps to identify bugs quicker, improves quality, and reduce the time it takes to validate and release new software updates.

Continuous Deployment (**CD**) refers to the practice of using automation to publish and deploy software updates. Continuous Integration along with Continuous Deployment process helps us to build, test, and publish software updates as soon as code changes are made.

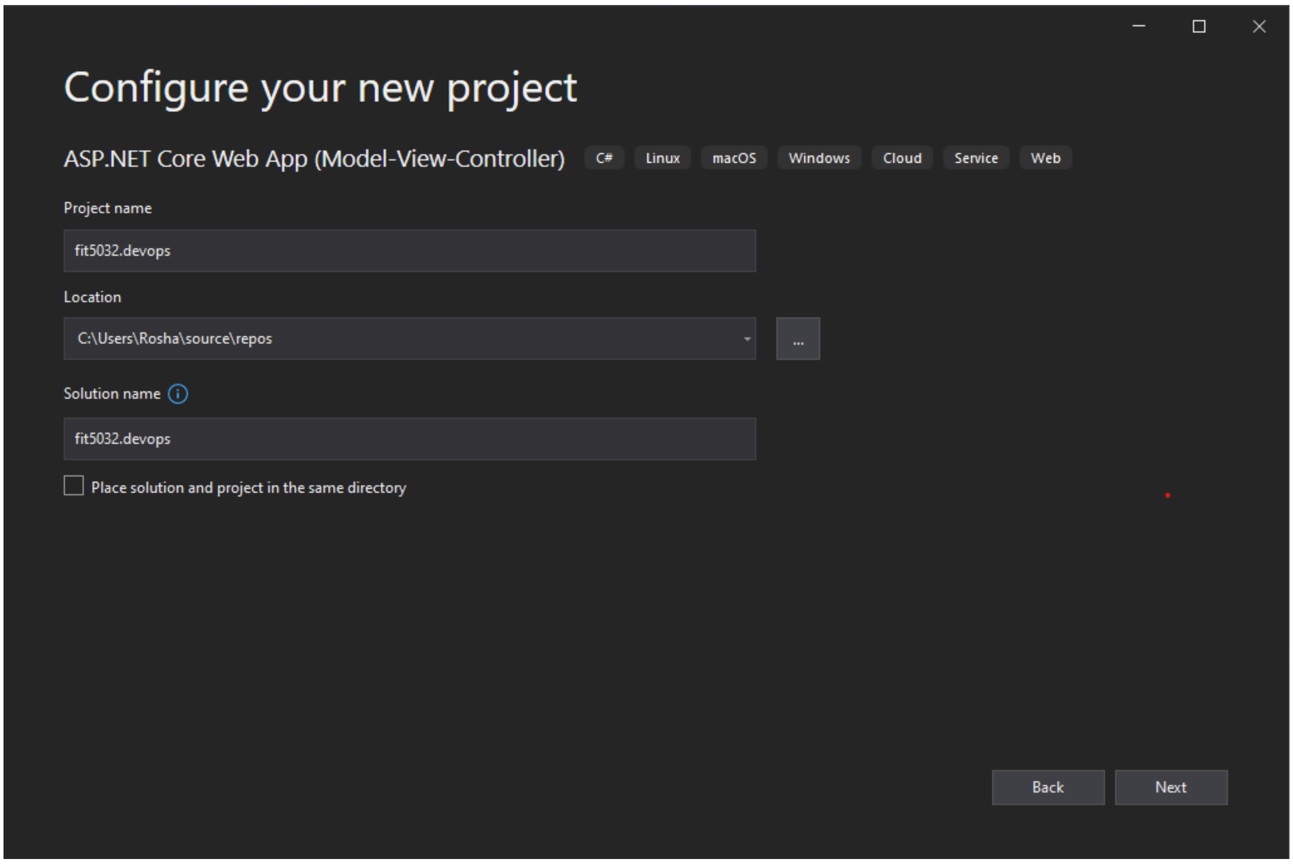


GitHub Actions

GitHub Actions by Github is used to automate software workflows. GitHub actions comes with integrated CI/CD. Using GitHub triggers like merging or commiting code changes to your GitHub repo, we can leverage CI/CD pipelines from GitHub Actions to continously build, test and deploy our code.

Step 1:

Create a new .NET Core (.NET 5.0) web application. You can use any Web API or MVC or other .NET core web application templates. Here we will be using a .NET 5.0 MVC application.



Configure your new project

ASP.NET Core Web App (Model-View-Controller)

Project name

fit5032.devops

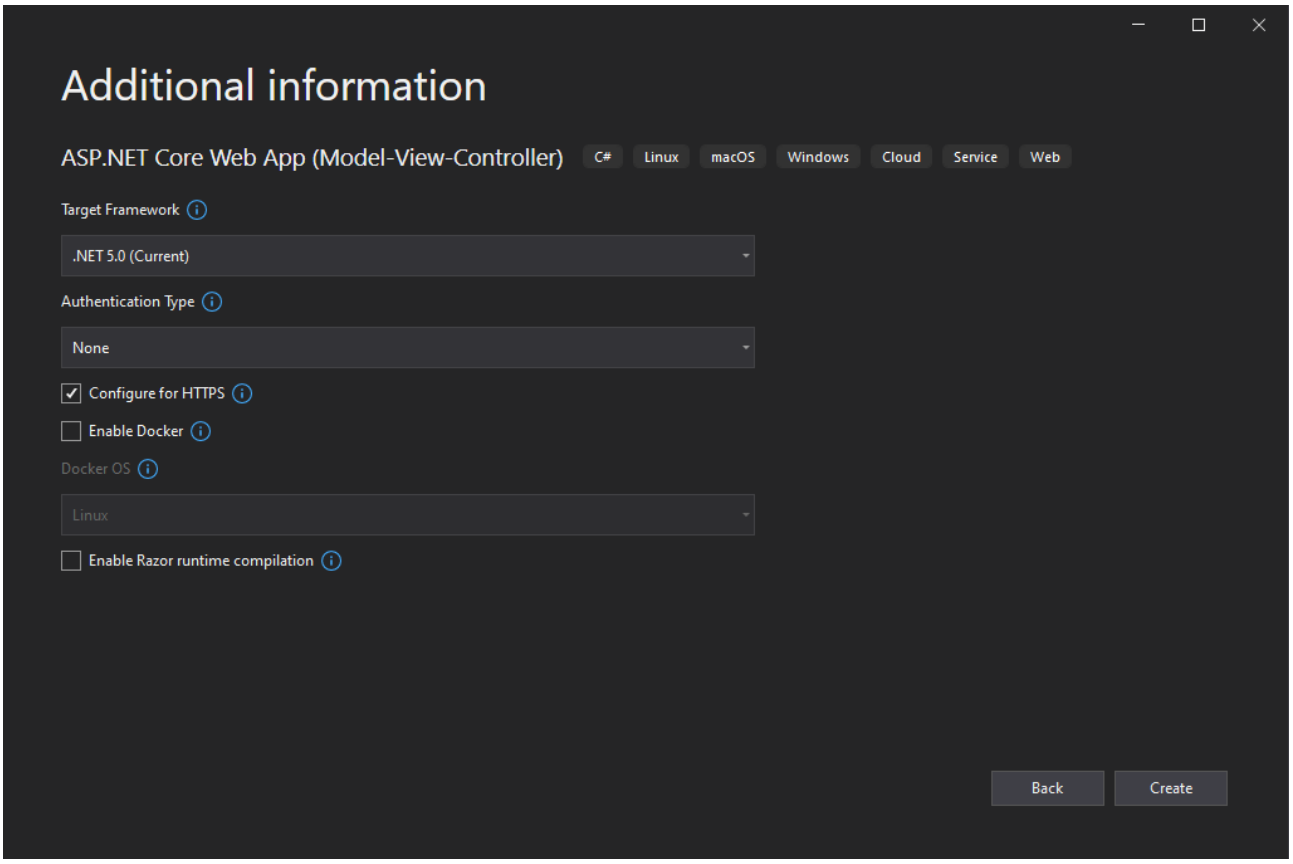
Location

C:\Users\Rosha\source\repos

Solution name

fit5032.devops

☐ Place solution and project in the same directory



Additional information

ASP.NET Core Web App (Model-View-Controller)

Target Framework

.NET 5.0 (Current)

Authentication Type

None

☑ Configure for HTTPS

☐ Enable Docker

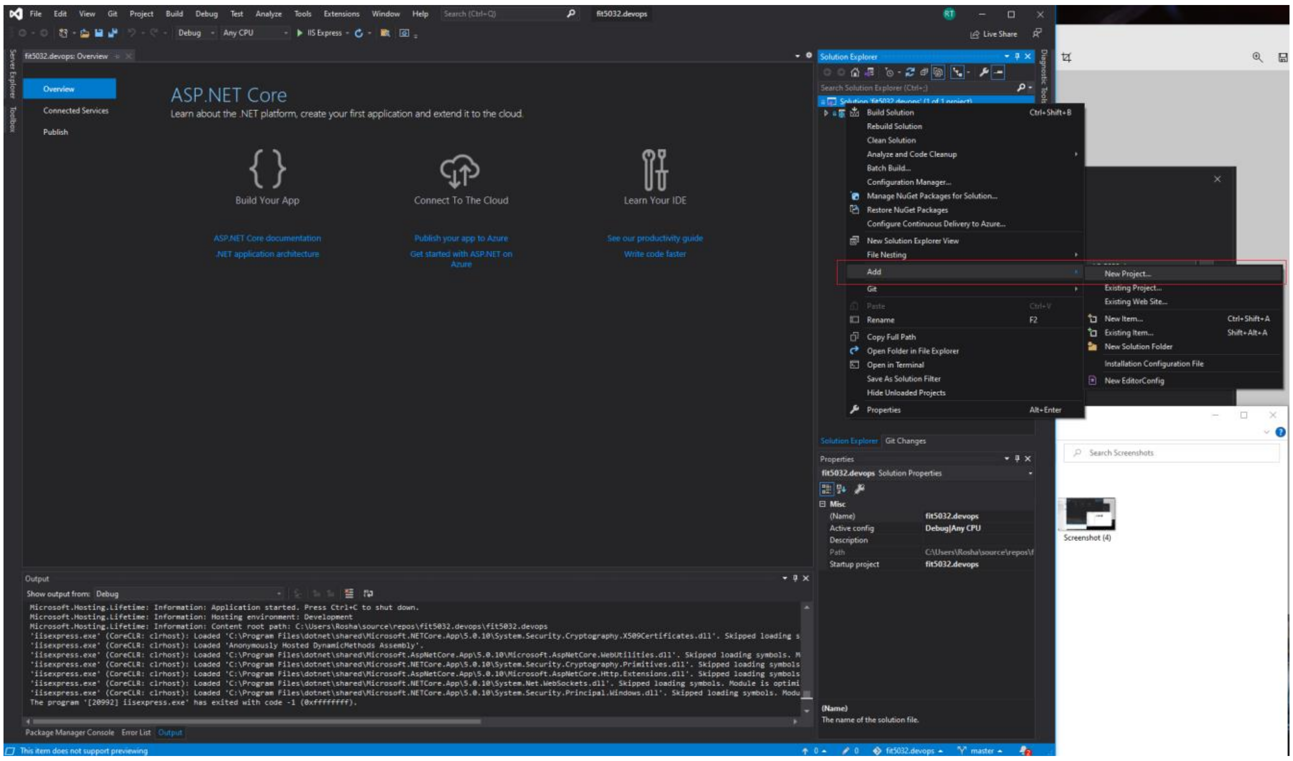
Docker OS

Linux

☐ Enable Razor runtime compilation

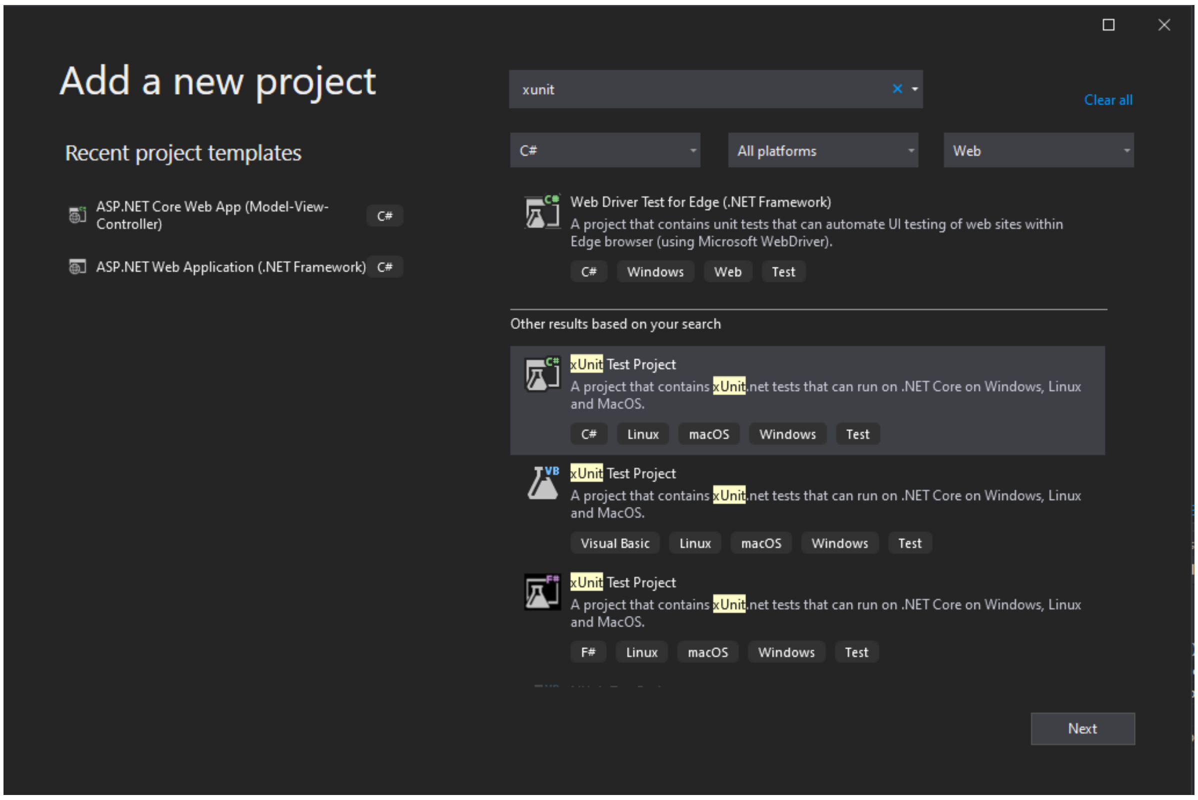
Step 2:

Let us also add a Unit Test project for running the Tests for the Web App that we have created. Right click on the solution and add a new project(Add → New Project).



```
Microsoft.Hosting.Lifetime: Information: Application started. Press Ctrl+C to shut down.
Microsoft.Hosting.Lifetime: Information: Hosting environment: Development
Microsoft.Hosting.Lifetime: Information: Content root path: C:\Users\Rosha\source\repos\fit5032.devops\fit5032.devops
'iisexpress.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\5.0.10\System.Security.Cryptography.X509Certificates.dll'. Skipped loading s
'iisexpress.exe' (CoreCLR: clrhost): Loaded 'Anonymously Hosted DynamicMethods Assembly'.
'iisexpress.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.AspNetCore.App\5.0.10\Microsoft.AspNetCore.WebUtilities.dll'. Skipped loading symbols. M
'iisexpress.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\5.0.10\System.Security.Cryptography.Primitives.dll'. Skipped loading symbols
'iisexpress.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.AspNetCore.App\5.0.10\Microsoft.AspNetCore.Http.Extensions.dll'. Skipped loading symbols
'iisexpress.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\5.0.10\System.Net.WebSockets.dll'. Skipped loading symbols. Module is optimi
'iisexpress.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\5.0.10\System.Security.Principal.Windows.dll'. Skipped loading symbols. Modu
The program '[20992] iisexpress.exe' has exited with code -1 (0xffffffff).
```

Select xUnit Test project for .NET Core


Add a new project
Recent project templates
ASP.NET Core Web App (Model-View-Controller) C#
ASP.NET Web Application (.NET Framework) C#
xunit
Clear all
C#
All platforms
Web
Next

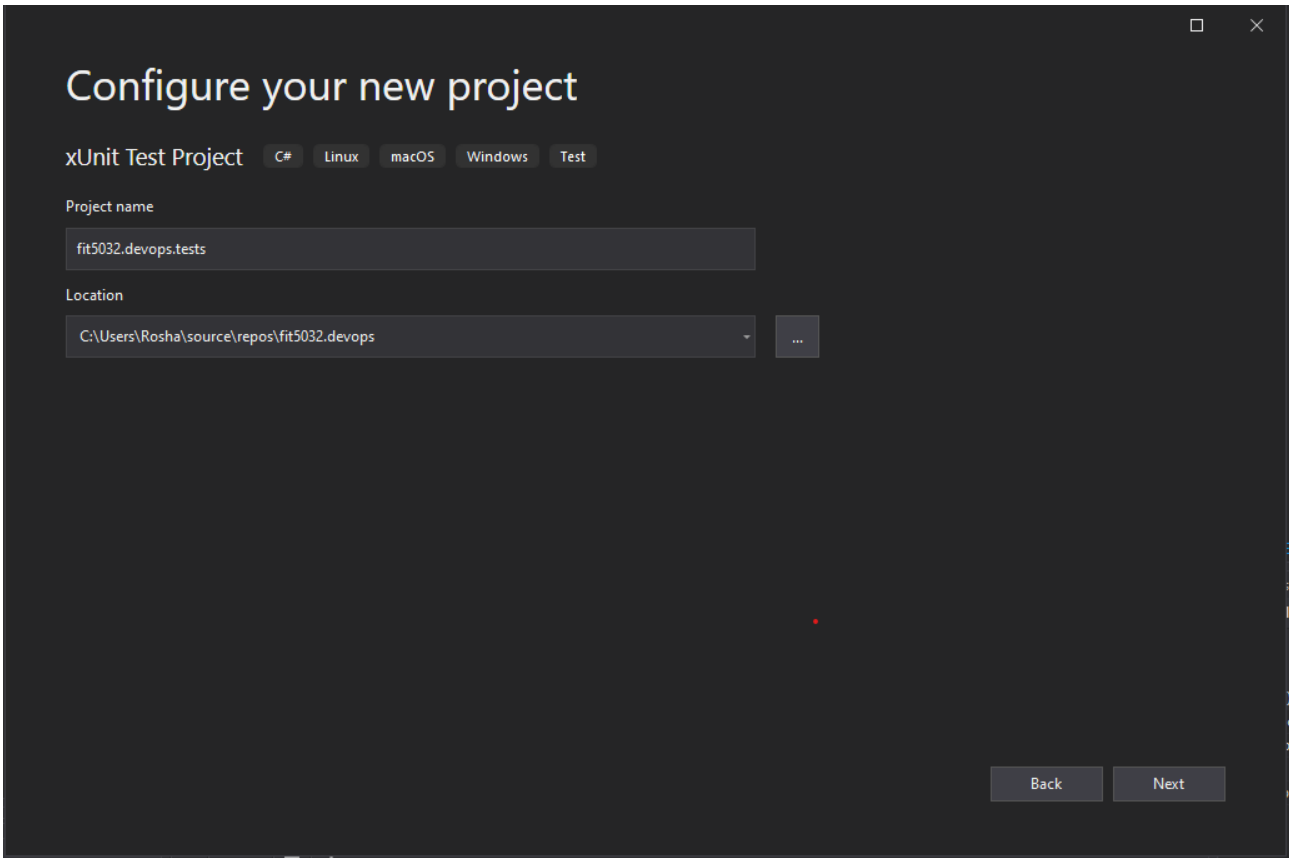
Web Driver Test for Edge (.NET Framework)
A project that contains unit tests that can automate UI testing of web sites within Edge browser (using Microsoft WebDriver).
C# Windows Web Test

Other results based on your search

xUnit Test Project
A project that contains xUnit.net tests that can run on .NET Core on Windows, Linux and MacOS.
C# Linux macOS Windows Test

xUnit Test Project
A project that contains xUnit.net tests that can run on .NET Core on Windows, Linux and MacOS.
Visual Basic Linux macOS Windows Test

xUnit Test Project
A project that contains xUnit.net tests that can run on .NET Core on Windows, Linux and MacOS.
F# Linux macOS Windows Test



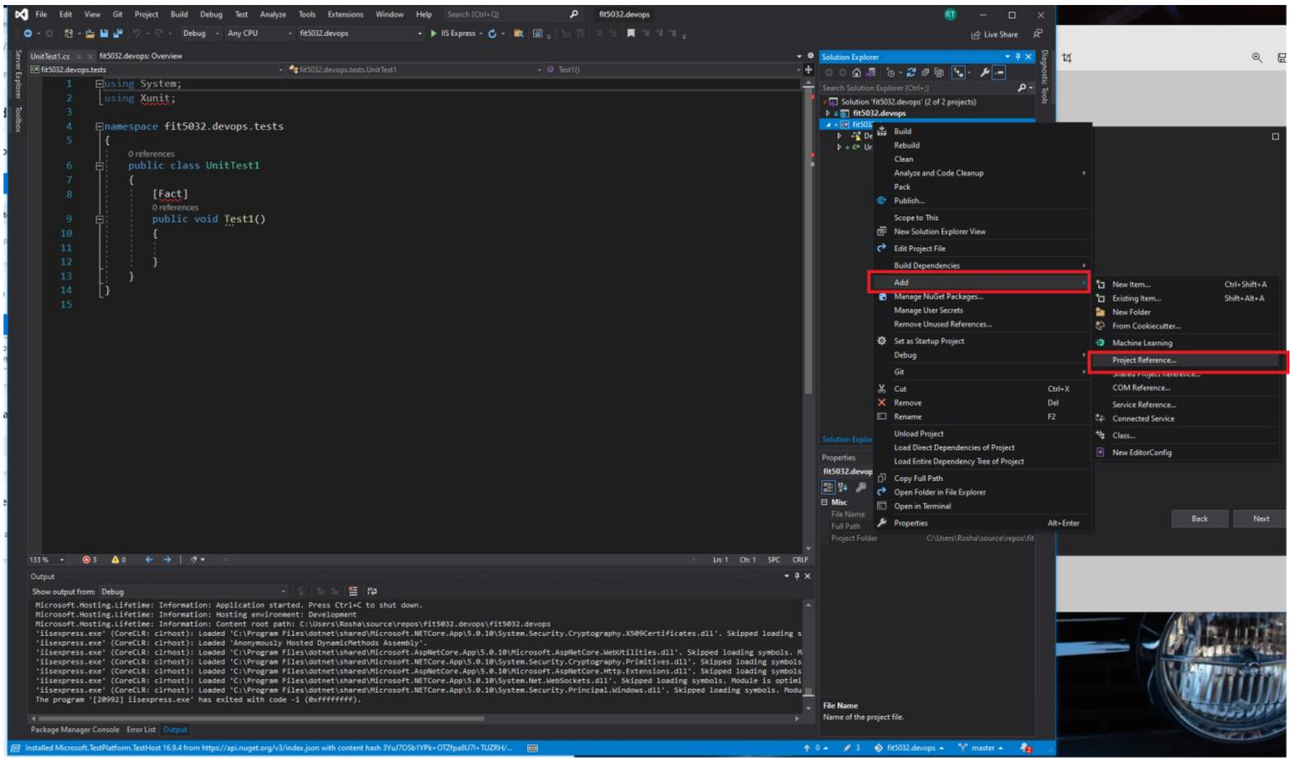
Configure your new project

xUnit Test Project — C# · Linux · macOS · Windows · Test

Project name: fit5032.devops.tests

Location: C:\Users\Rosha\source\repos\fit5032.devops

Once the xUnit Test project is created let's add the reference to the WebApp we created in Step 1 to the Unit Test project. To do this, right click on the Unit Test project and add project reference (Add → Project Reference)

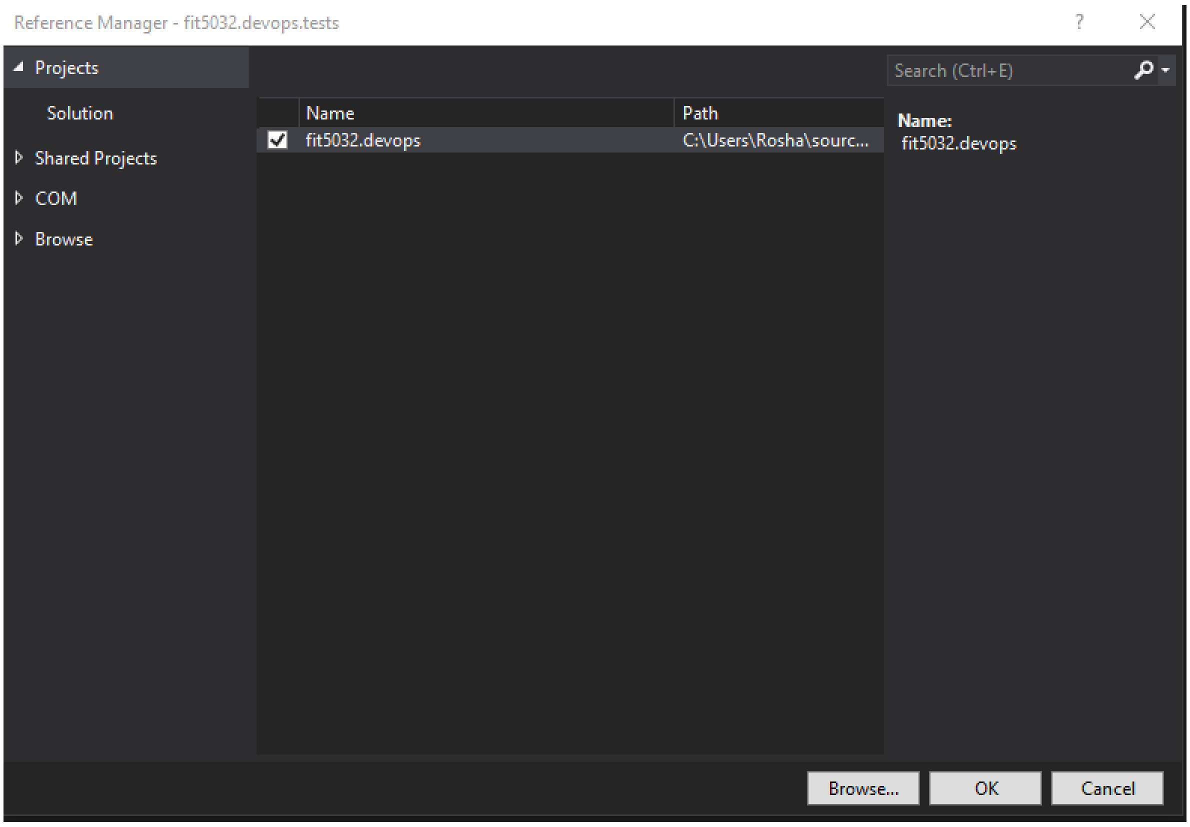


```
using System;
using Xunit;

namespace fit5032.devops.tests
{
    0 references
    public class UnitTest1
    {
        [Fact]
        0 references
        public void Test1()
        {

        }
    }
}
```

```
Microsoft.Hosting.Lifetime: Information: Application started. Press Ctrl+C to shut down.
Microsoft.Hosting.Lifetime: Information: Hosting environment: Development
Microsoft.Hosting.Lifetime: Information: Content root path: C:\Users\Rosha\source\repos\fit5032.devops\fit5032.devops
'iisexpress.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\5.0.10\System.Security.Cryptography.X509Certificates.dll'. Skipped loading s
'iisexpress.exe' (CoreCLR: clrhost): Loaded 'Anonymously Hosted DynamicMethods Assembly'.
'iisexpress.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.AspNetCore.App\5.0.10\Microsoft.AspNetCore.WebUtilities.dll'. Skipped loading symbols. M
'iisexpress.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\5.0.10\System.Security.Cryptography.Primitives.dll'. Skipped loading symbols
'iisexpress.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.AspNetCore.App\5.0.10\Microsoft.AspNetCore.Http.Extensions.dll'. Skipped loading symbols
'iisexpress.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\5.0.10\System.Net.WebSockets.dll'. Skipped loading symbols. Module is optimi
'iisexpress.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\5.0.10\System.Security.Principal.Windows.dll'. Skipped loading symbols. Modu
The program '[20992] iisexpress.exe' has exited with code -1 (0xffffffff).
```

Installed Microsoft.TestPlatform.TestHost 16.9.4 from https://api.nuget.org/v3/index.json with content hash 3YuJ7OSb1YPk+OTZfpa8U7I+TUZRH/...

Add reference to the MVC web app we created in Step 1 by ticking the select box.



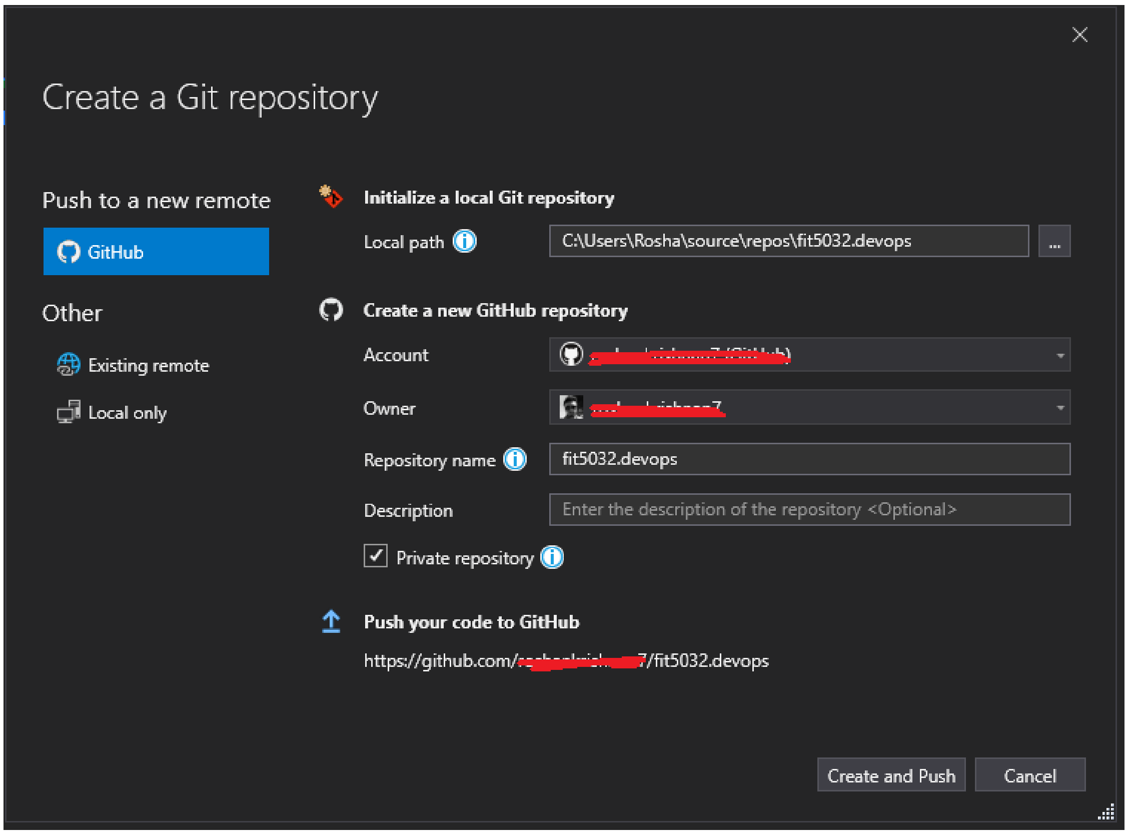
Reference Manager - fit5032.devops.tests

	Name	Path
✓	fit5032.devops	C:\Users\Rosha\sourc...

Name: fit5032.devops

Step 3:

Create a Git repository and push your code changes to GitHub



Create a Git repository

Initialize a local Git repository

Local path: C:\Users\Rosha\source\repos\fit5032.devops

Create a new GitHub repository

Account: [redacted]

Owner: [redacted]

Repository name: fit5032.devops

Description: Enter the description of the repository <Optional>

✓ Private repository

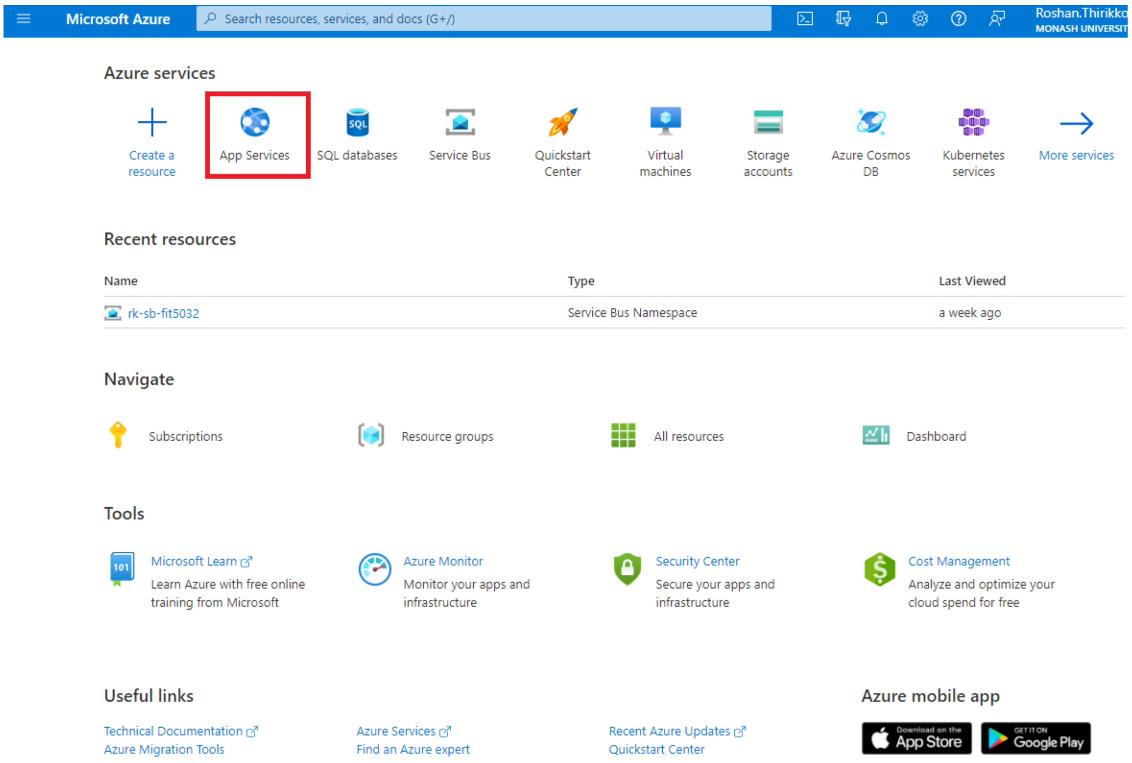
Push your code to GitHub

https://github.com/[redacted]/fit5032.devops

Step 3 –

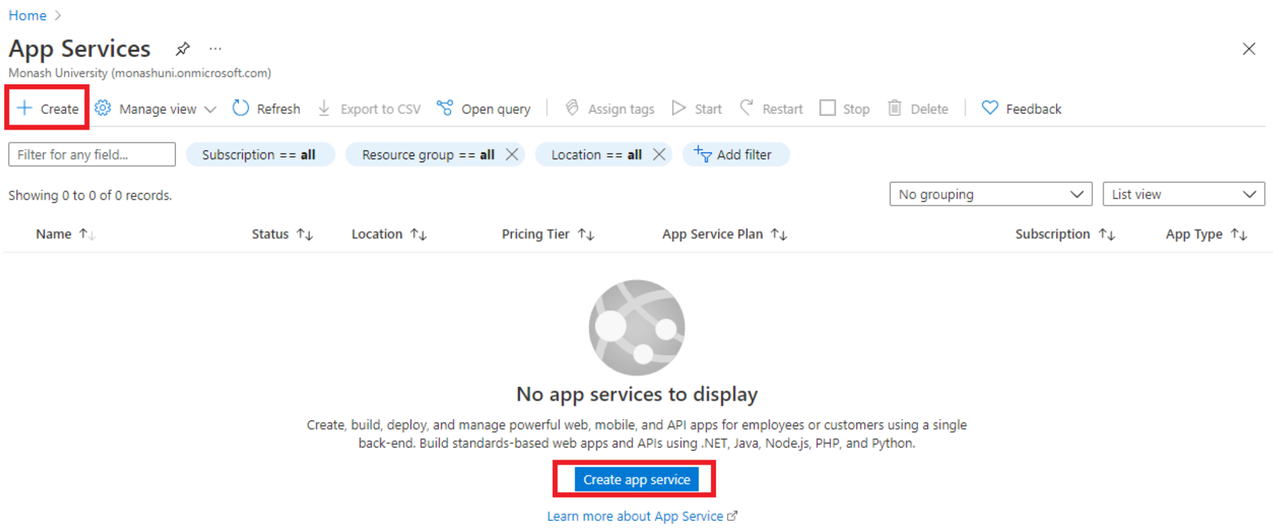
We will be using Azure App Service to host our Web application. If you do not have an Azure account, please go to Azure Portal and sign up using your Student Account. Monash student account enables you to $200 worth of free credits that you can use on various Azure resources.

Once you account is setup, let us go ahead and create an App Service. You can use the search bar on top to find out App Services.



Name	Type	Last Viewed
rk-sb-fit5032	Service Bus Namespace	a week ago

Create a new App Service by using either the Create button or the Create app service button.

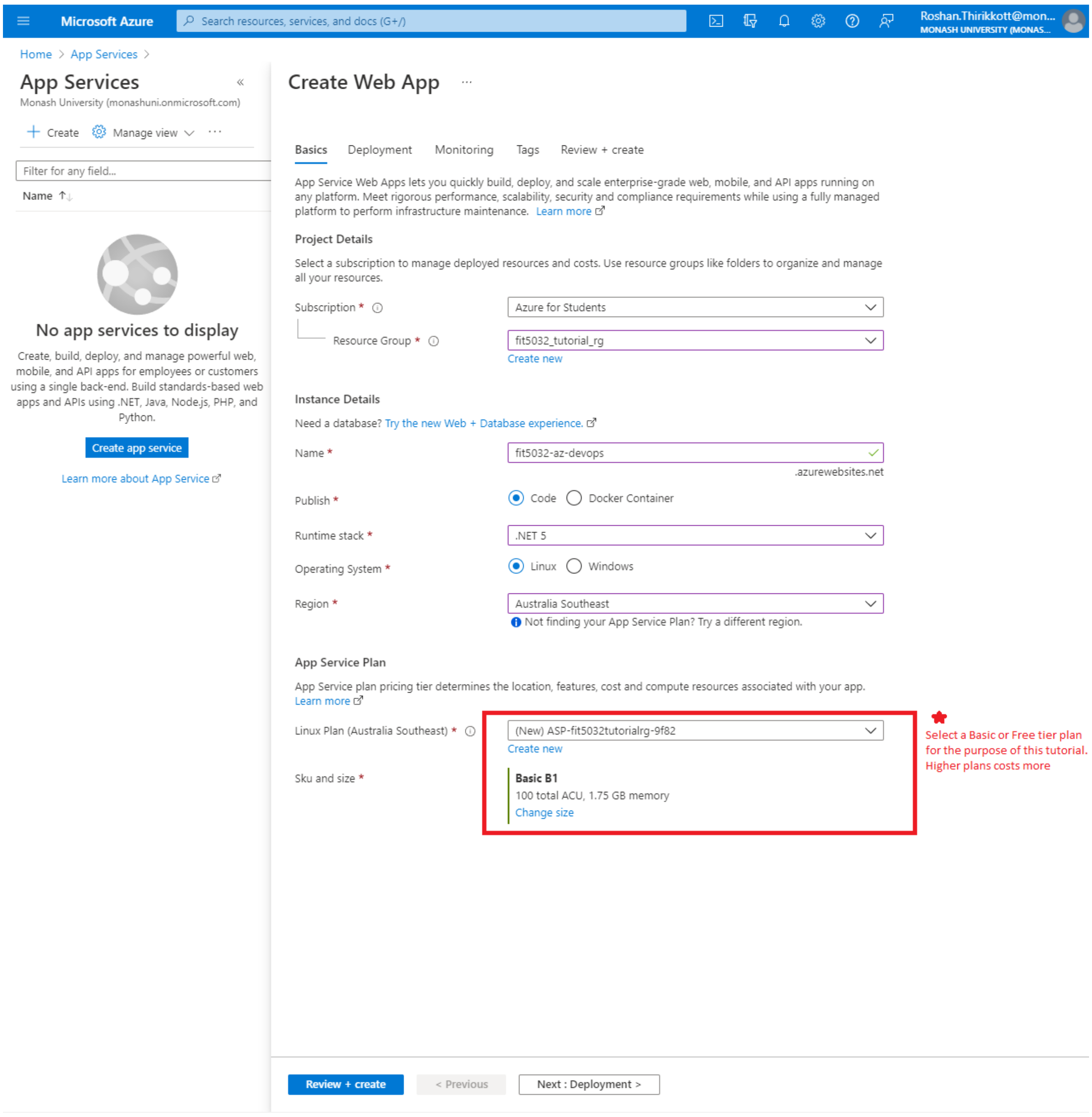


Create, build, deploy, and manage powerful web, mobile, and API apps for employees or customers using a single back-end. Build standards-based web apps and APIs using .NET, Java, Node.js, PHP, and Python.

Fill out the details in the new popup window.

Note: You can use the Create New under the Resource Group to create a new Resource Group if you do not already have one. (A resource group is a container that holds related resources for an Azure solution). You might need to provide a unique name for your web app instance. Make sure to select the appropriate run time stack (Here we will be using .NET 5). Since .NET 5.0 is cross platform compatible, we can choose either Linux or Windows as the Operating System for the App Service. Choose the appropriate App Service Plan. (Try to select one of the Basic or Free tier App service plan for the purpose of this task, as the other App Service plans will charge more, and you will end up exhausting your free credits sooner).

Once all the input details are filled in Click on Review + Create
Note down the name of the app service that you have created as you will be able to view your web apps using that url (https://**<name you have chosen>**.azurewebsites.net)



Create, build, deploy, and manage powerful web, mobile, and API apps for employees or customers using a single back-end. Build standards-based web apps and APIs using .NET, Java, Node.js, PHP, and Python.

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 * : Azure for Students

Resource Group * : fit5032_tutorial_rg
Create new

Instance Details

Need a database? Try the new Web + Database experience.

Name * : fit5032-az-devops .azurewebsites.net

Publish * : [✓ Code] Docker Container

Runtime stack * : .NET 5

Operating System * : [✓ Linux] Windows

Region * : Australia Southeast
Not finding your App Service Plan? Try a different region.

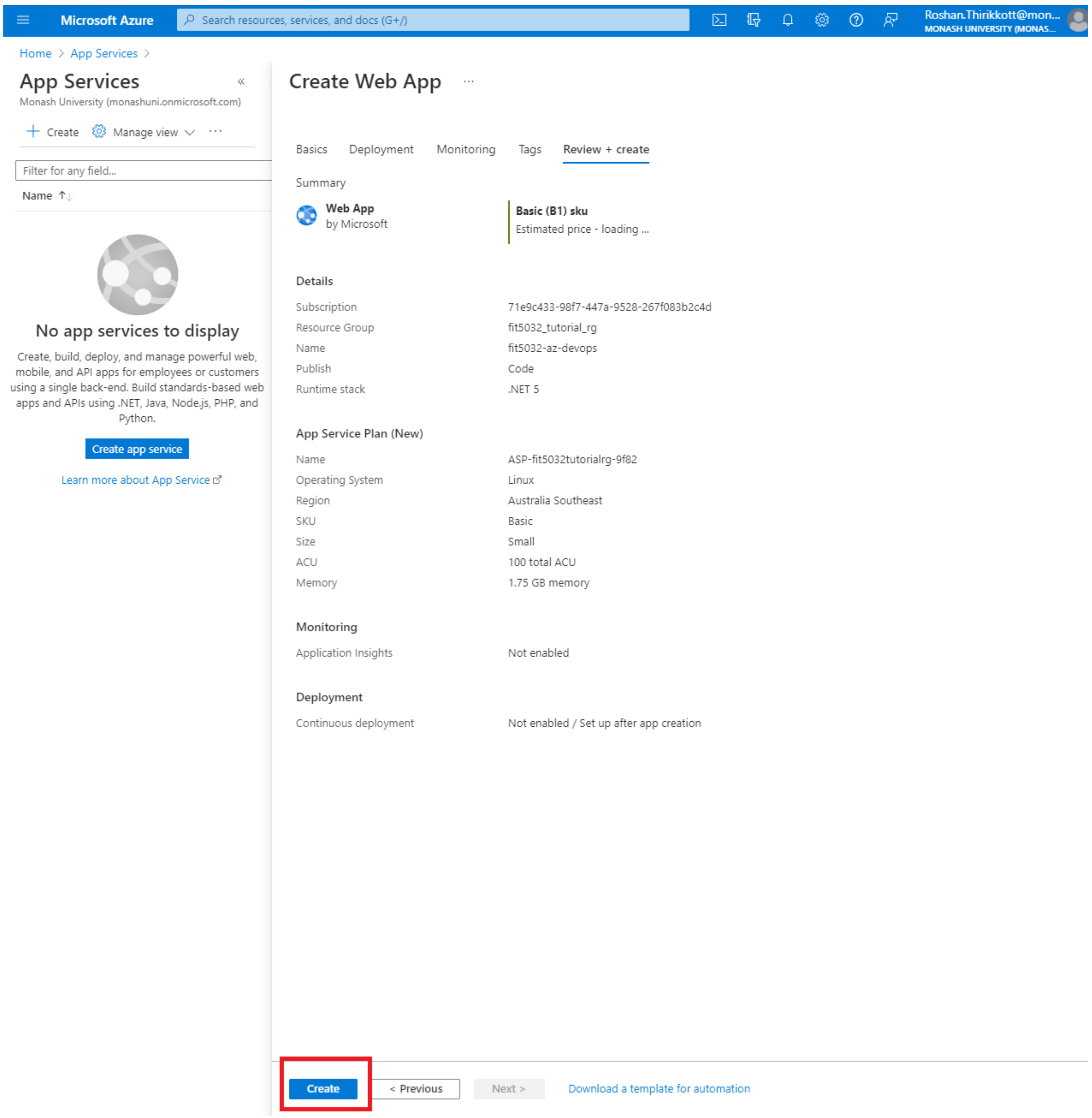
App Service Plan

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

Linux Plan (Australia Southeast) * : (New) ASP-fit5032tutorialrg-9f82
Create new

Sku and size * : **Basic B1**
100 total ACU, 1.75 GB memory
Change size

Select a Basic or Free tier plan for the purpose of this tutorial. Higher plans costs more


Microsoft Azure | Search resources, services, and docs (G+/) | Roshan.Thirikkott@mon... MONASH UNIVERSITY (MONAS...

Home > App Services >

App Services
Monash University (monashuni.onmicrosoft.com)

+ Create | Manage view | ···

Filter for any field...

Name ↑↓

No app services to display

Create, build, deploy, and manage powerful web, mobile, and API apps for employees or customers using a single back-end. Build standards-based web apps and APIs using .NET, Java, Node.js, PHP, and Python.

Create app service

Learn more about App Service

Create Web App ···

Basics | Deployment | Monitoring | Tags | Review + create

Summary

Web App
by Microsoft

Basic (B1) sku
Estimated price - loading ...

Details

Subscription	71e9c433-98f7-447a-9528-267f083b2c4d
Resource Group	fit5032_tutorial_rg
Name	fit5032-az-devops
Publish	Code
Runtime stack	.NET 5

App Service Plan (New)

Name	ASP-fit5032tutorialrg-9f82
Operating System	Linux
Region	Australia Southeast
SKU	Basic
Size	Small
ACU	100 total ACU
Memory	1.75 GB memory

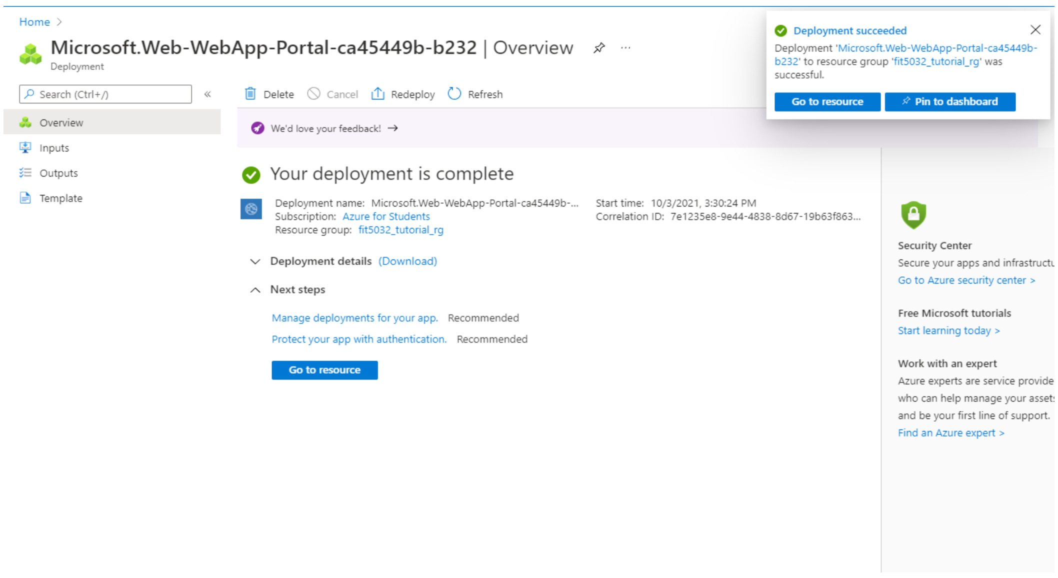
Monitoring

Application Insights	Not enabled

Deployment

Continuous deployment	Not enabled / Set up after app creation

Create | < Previous | Next > | Download a template for automation


Home >

Microsoft.Web-WebApp-Portal-ca45449b-b232 | Overview
Deployment

Search (Ctrl+/)

Overview
Inputs
Outputs
Template

Delete | Cancel | Redeploy | Refresh

We'd love your feedback! →

Deployment succeeded
Deployment 'Microsoft.Web-WebApp-Portal-ca45449b-b232' to resource group 'fit5032_tutorial_rg' was successful.

Go to resource | Pin to dashboard

Your deployment is complete

Deployment name: Microsoft.Web-WebApp-Portal-ca45449b-...
Subscription: Azure for Students
Resource group: fit5032_tutorial_rg

Start time: 10/3/2021, 3:30:24 PM
Correlation ID: 7e1235e8-9e44-4838-8d67-19b63f863...

Deployment details (Download)

Next steps

Manage deployments for your app. Recommended
Protect your app with authentication. Recommended

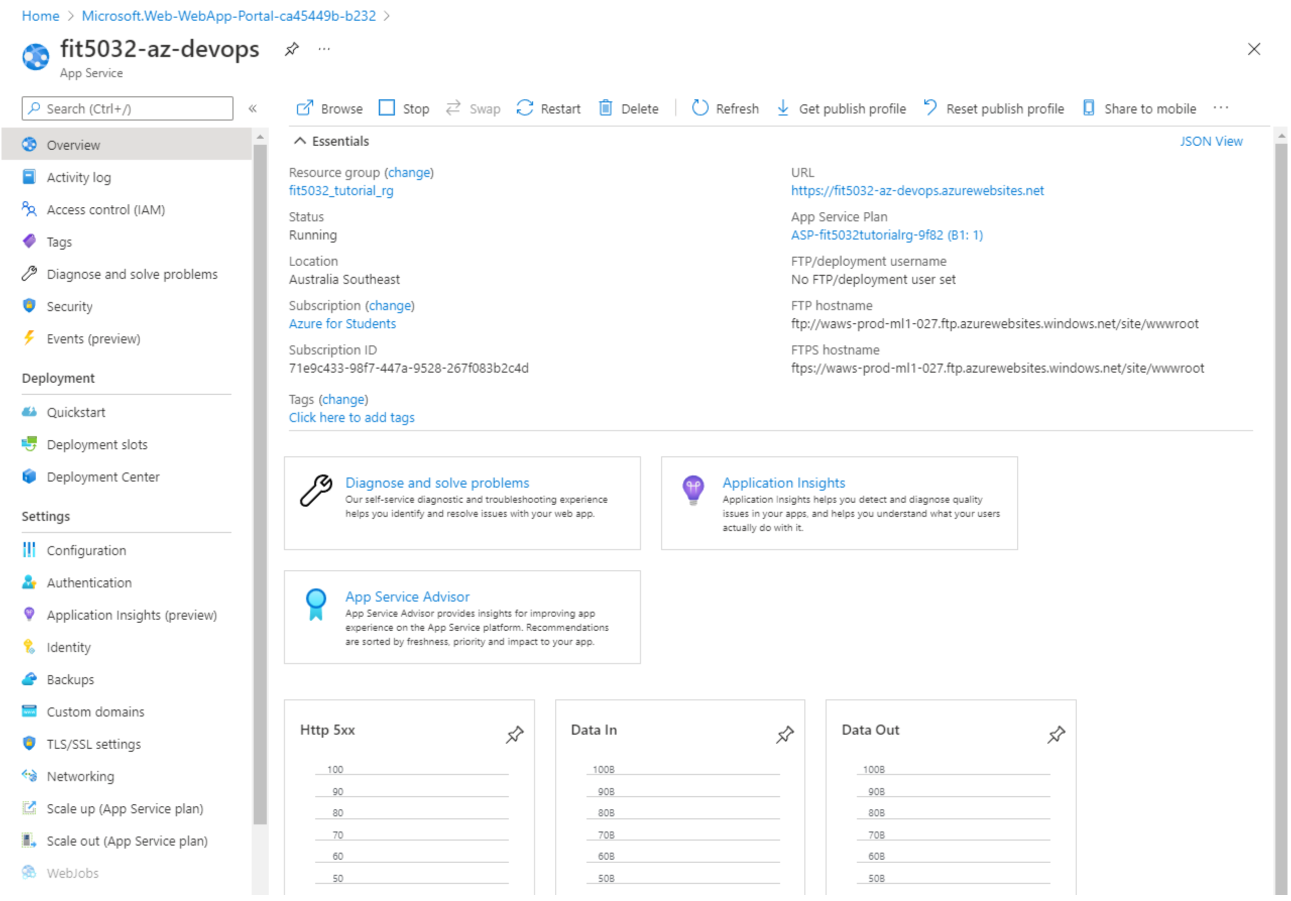
Go to resource

Security Center
Secure your apps and infrastructu
Go to Azure security center >

Free Microsoft tutorials
Start learning today >

Work with an expert
Azure experts are service provide
who can help manage your asset
and be your first line of support.
Find an Azure expert >

Once the deployment is completed, click on Go to resource to view the App Service that we created



Resource group (change)
fit5032_tutorial_rg

Status
Running

Location
Australia Southeast

Subscription (change)
Azure for Students

Subscription ID
71e9c433-98f7-447a-9528-267f083b2c4d

Tags (change)
Click here to add tags

URL
https://fit5032-az-devops.azurewebsites.net

App Service Plan
ASP-fit5032tutorialrg-9f82 (B1: 1)

FTP/deployment username
No FTP/deployment user set

FTP hostname
ftp://waws-prod-ml1-027.ftp.azurewebsites.windows.net/site/wwwroot

FTPS hostname
ftps://waws-prod-ml1-027.ftp.azurewebsites.windows.net/site/wwwroot

Diagnose and solve problems
Our self-service diagnostic and troubleshooting experience helps you identify and resolve issues with your web app.

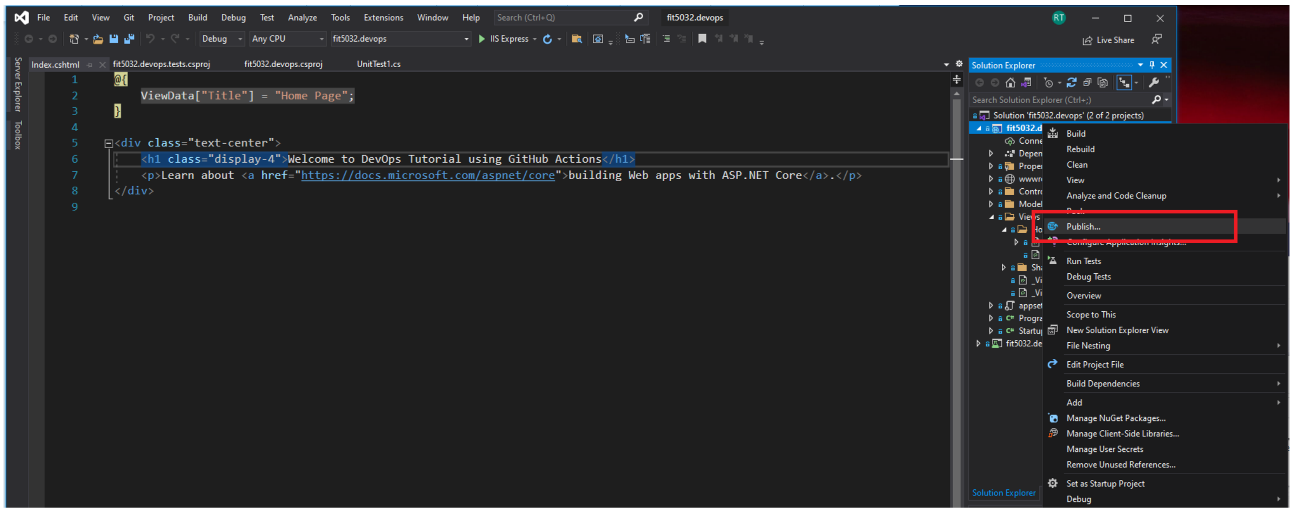
Application Insights
Application Insights helps you detect and diagnose quality issues in your apps, and helps you understand what your users actually do with it.

App Service Advisor
App Service Advisor provides insights for improving app experience on the App Service platform. Recommendations are sorted by freshness, priority and impact to your app.

Step 4:

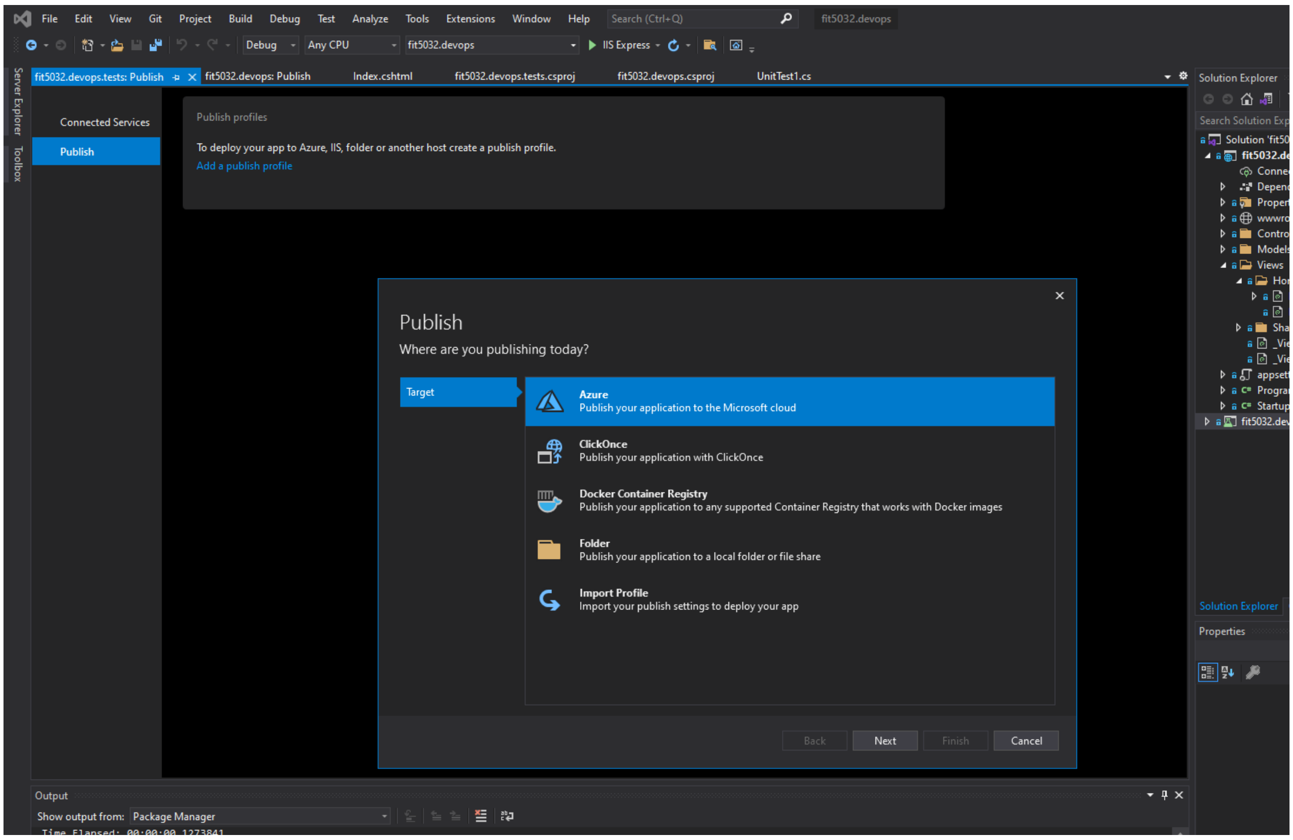
Set up CI/CD using GitHub Actions

Visual Studio helps us to set up the CI/CD pipeline using GitHub Actions in few steps. Right click on the web app project and select Publish option.



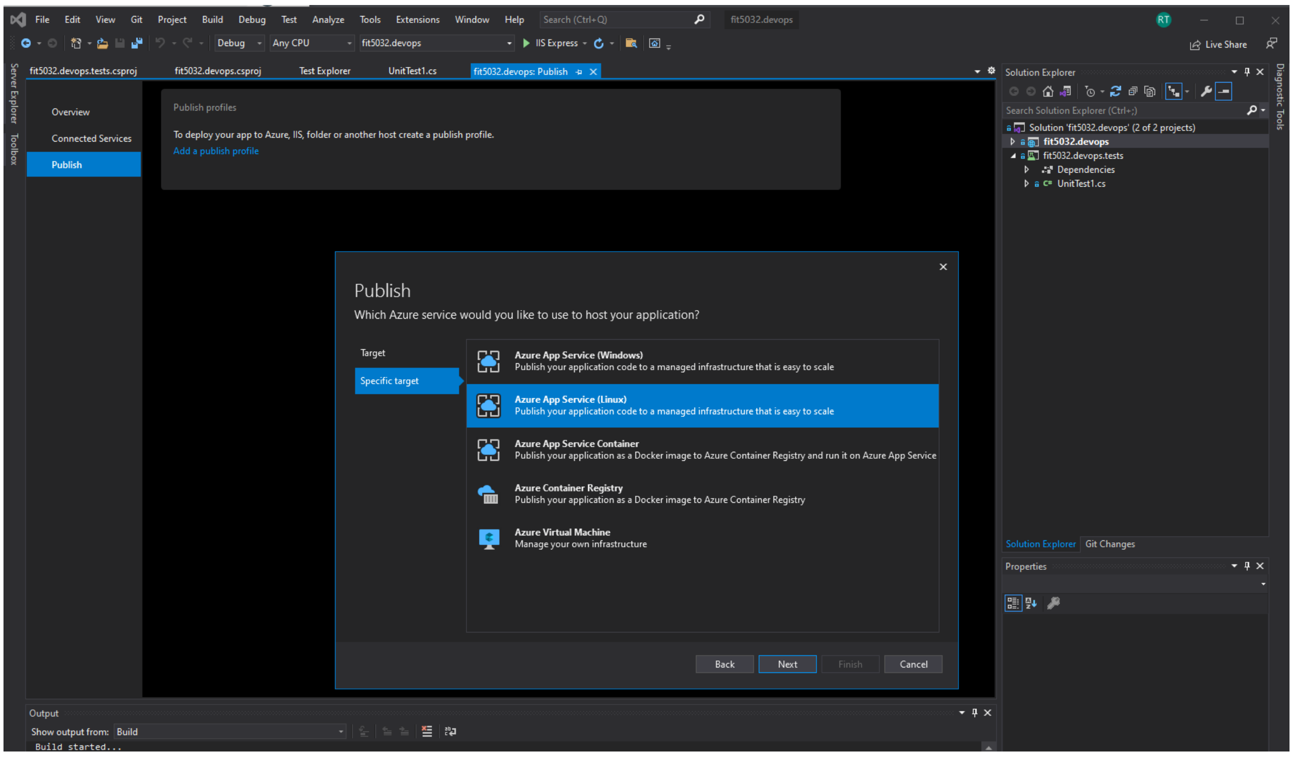
```
@{
    ViewData["Title"] = "Home Page";
}

<div class="text-center">
    <h1 class="display-4">Welcome to DevOps Tutorial using GitHub Actions</h1>
    <p>Learn about <a href="https://docs.microsoft.com/aspnet/core">building Web apps with ASP.NET Core</a>.</p>
</div>
```

Select Azure as the Target

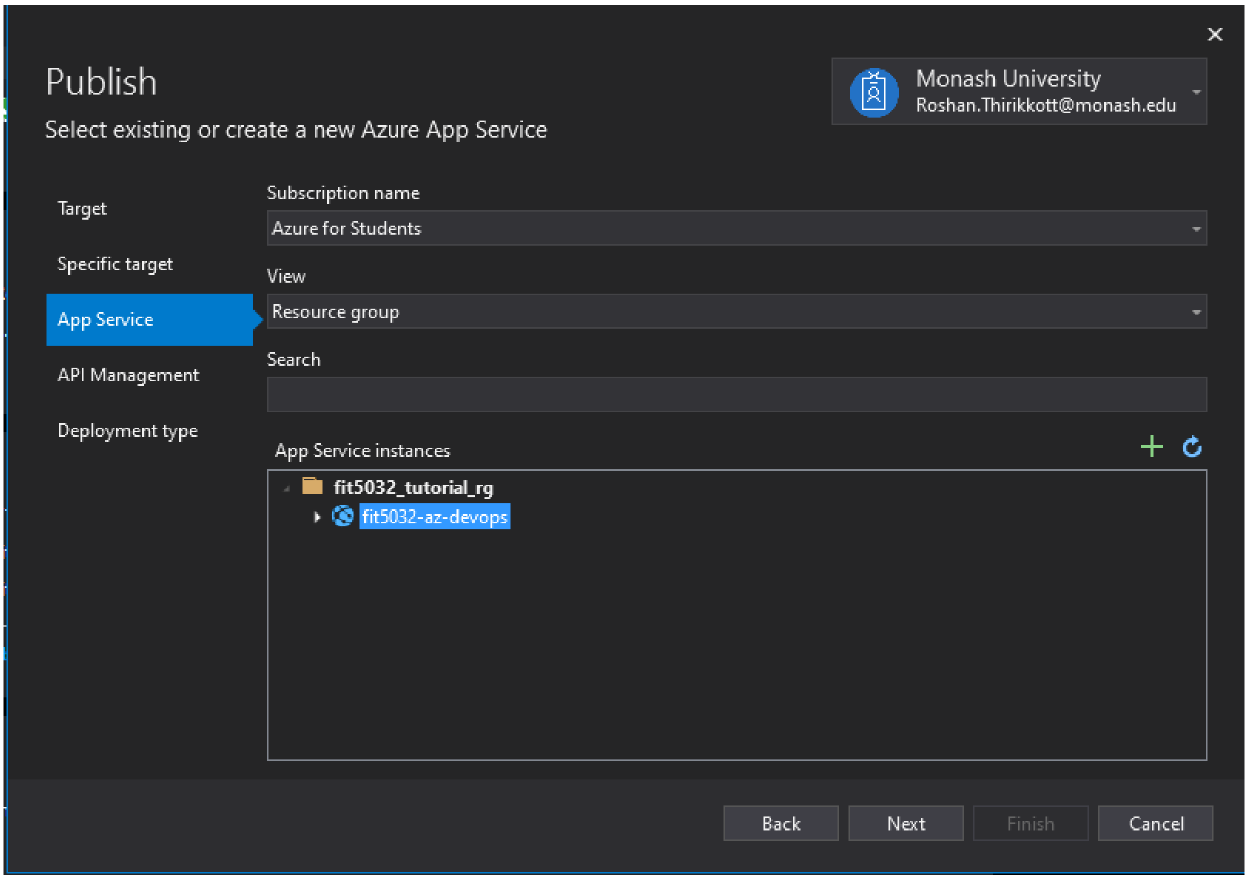

Publish
Where are you publishing today?
Target
Azure
Publish your application to the Microsoft cloud
ClickOnce
Publish your application with ClickOnce
Docker Container Registry
Publish your application to any supported Container Registry that works with Docker images
Folder
Publish your application to a local folder or file share
Import Profile
Import your publish settings to deploy your app
Back
Next
Finish
Cancel

Select the option Specific Target and choose the option Azure App Service (Linux).

Note: Choose Azure App Service (Windows) if you had created the App Service to run on Windows


Publish
Which Azure service would you like to use to host your application?
Target
Specific target
Azure App Service (Windows)
Publish your application code to a managed infrastructure that is easy to scale
Azure App Service (Linux)
Publish your application code to a managed infrastructure that is easy to scale
Azure App Service Container
Publish your application as a Docker image to Azure Container Registry and run it on Azure App Service
Azure Container Registry
Publish your application as a Docker image to Azure Container Registry
Azure Virtual Machine
Manage your own infrastructure
Back
Next
Finish
Cancel

Select the App Service that you had created in the previous step.


Publish

Select existing or create a new Azure App Service

Monash University
Roshan.Thirikkott@monash.edu

Subscription name: Azure for Students

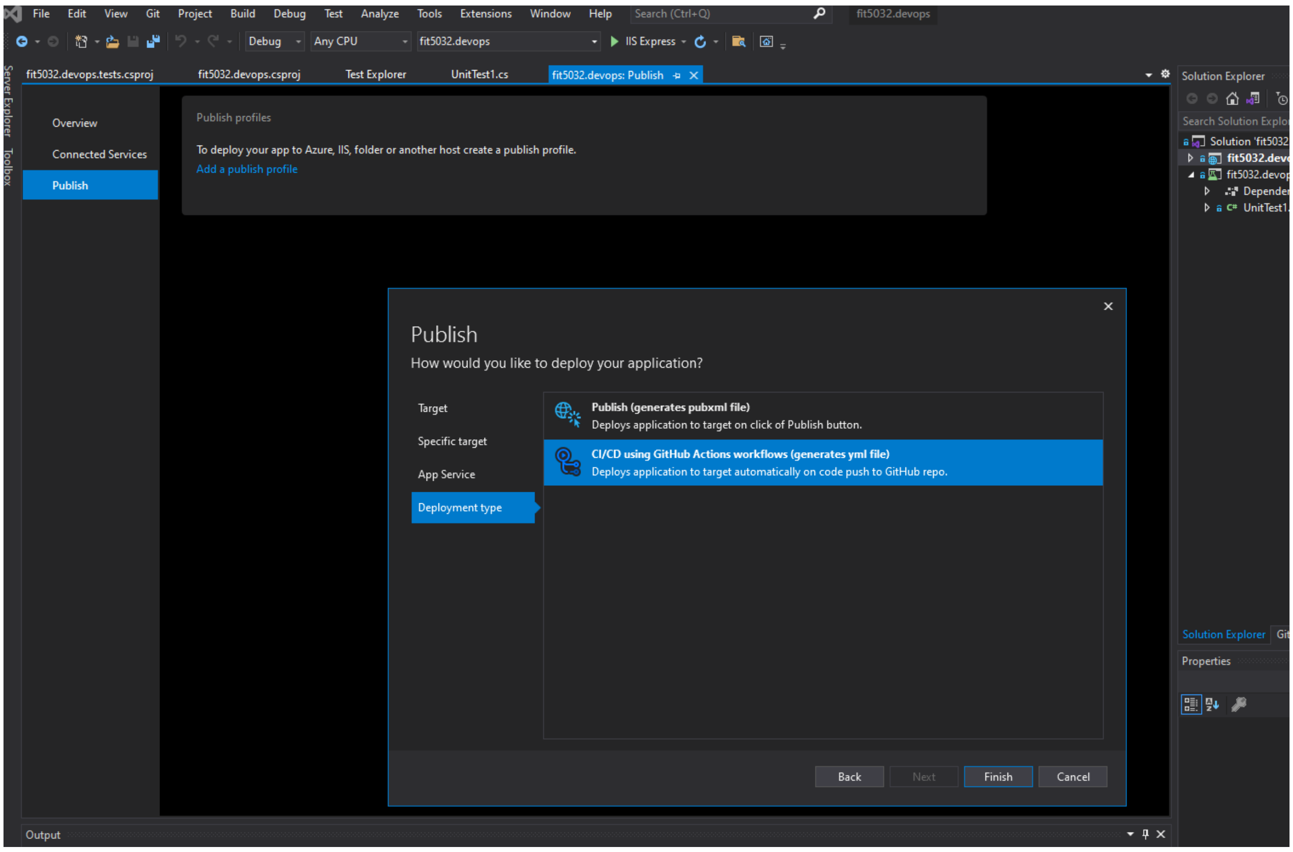
View: Resource group

Search:

App Service instances

fit5032_tutorial_rg
fit5032-az-devops

Deployment Type -> Choose option CI/CD using GitHub Actions workflows


Publish profiles

To deploy your app to Azure, IIS, folder or another host create a publish profile.
Add a publish profile

Publish

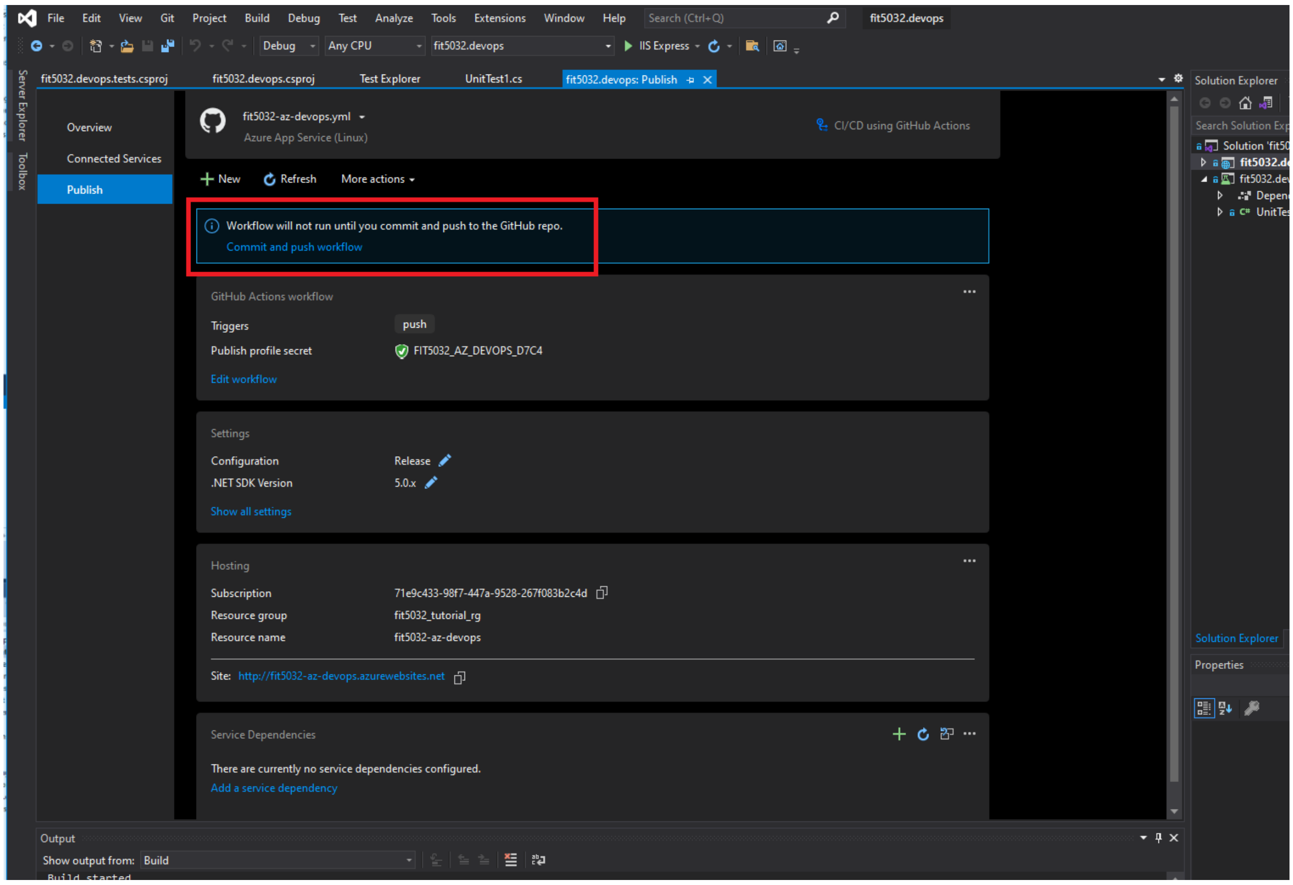
How would you like to deploy your application?

Publish (generates pubxml file)
Deploys application to target on click of Publish button.

CI/CD using GitHub Actions workflows (generates yml file)
Deploys application to target automatically on code push to GitHub repo.

The previous steps will build your deployment template for you. The triggers we select here will determine how the workflow runs. Here the default trigger is push, which means that as soon as you push changes to your master branch in your GitHub repo, it will trigger a GitHub Actions CI/CD pipeline run. You can setup more triggers by clicking on Edit Workflow. For now, let us go with the default workflow.

Now we need to commit the changes that were created.



fit5032-az-devops.yml

Azure App Service (Linux)

CI/CD using GitHub Actions

Workflow will not run until you commit and push to the GitHub repo.

Commit and push workflow

GitHub Actions workflow

Triggers	push
Publish profile secret	FIT5032_AZ_DEVOPS_D7C4

Edit workflow

Settings

Configuration	Release
.NET SDK Version	5.0.x

Show all settings

Hosting

Subscription	71e9c433-98f7-447a-9528-267f083b2c4d
Resource group	fit5032_tutorial_rg
Resource name	fit5032-az-devops

Site: http://fit5032-az-devops.azurewebsites.net

Service Dependencies

There are currently no service dependencies configured.

Add a service dependency

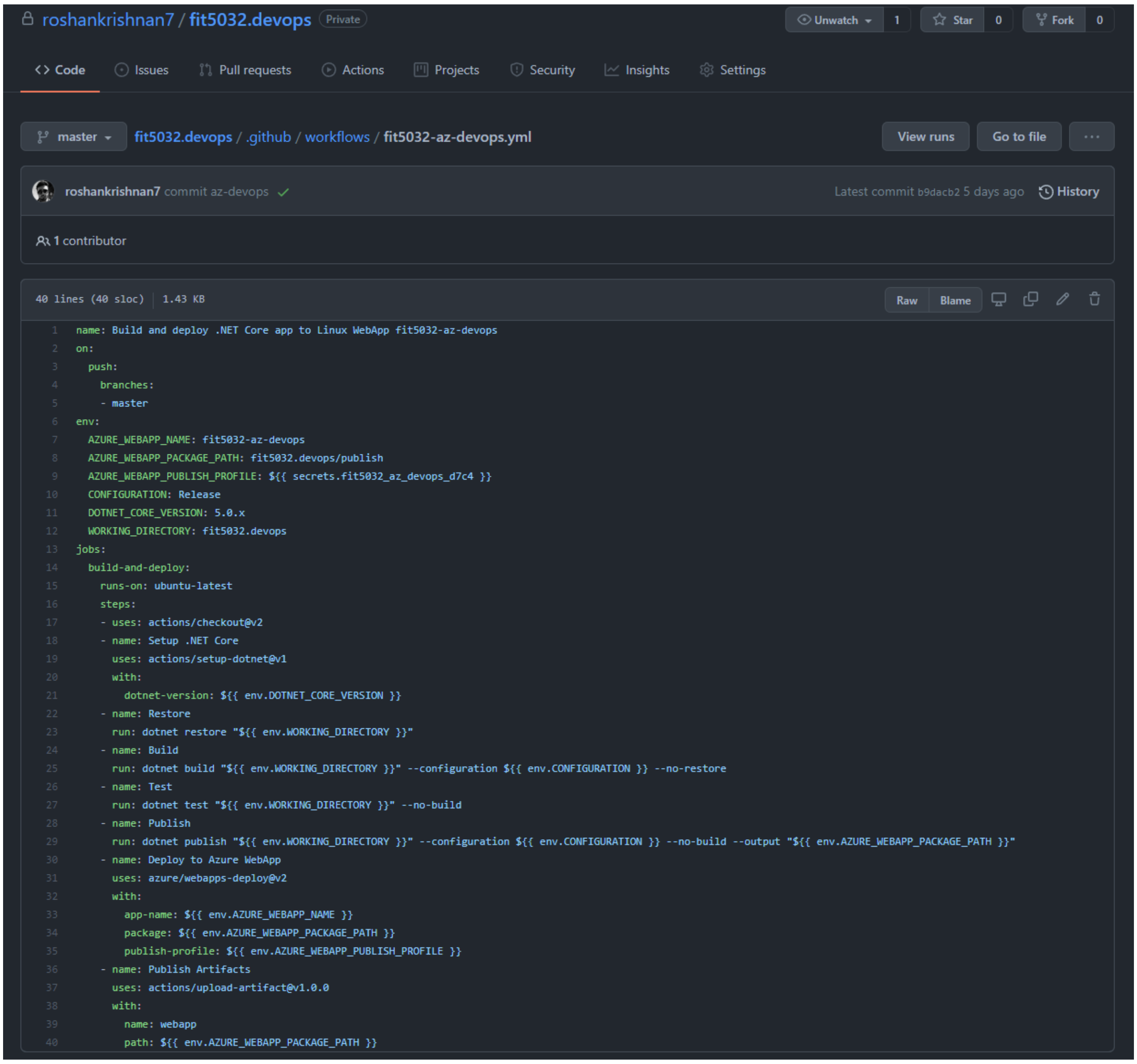
```
Build started
```

The template file used for deployment that was generated in your previous steps is located inside your repository as follows:

.github.workflows/<name of your project>.yml

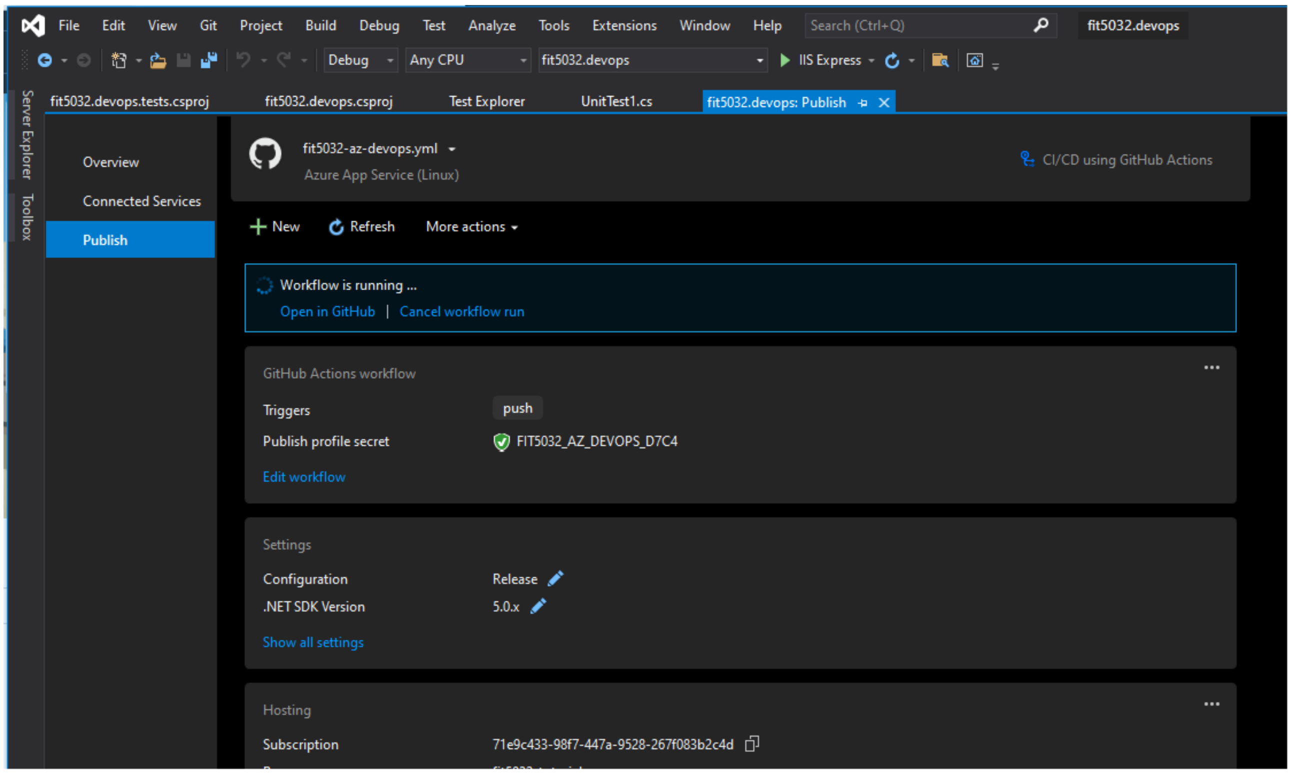
Note: This is a YAML file. YAML is a serialization language that is used often for configuration files.

This file contains all the steps that are involved in the CI/CD pipeline


roshankrishnan7 / fit5032.devops Private
Unwatch 1 Star 0 Fork 0
Code Issues Pull requests Actions Projects Security Insights Settings
master
fit5032.devops / .github / workflows / fit5032-az-devops.yml
View runs Go to file
roshankrishnan7 commit az-devops
Latest commit b9dacb2 5 days ago History
1 contributor
40 lines (40 sloc) 1.43 KB
Raw Blame

```yaml
name: Build and deploy .NET Core app to Linux WebApp fit5032-az-devops
on:
  push:
    branches:
    - master
env:
  AZURE_WEBAPP_NAME: fit5032-az-devops
  AZURE_WEBAPP_PACKAGE_PATH: fit5032.devops/publish
  AZURE_WEBAPP_PUBLISH_PROFILE: ${{ secrets.fit5032_az_devops_d7c4 }}
  CONFIGURATION: Release
  DOTNET_CORE_VERSION: 5.0.x
  WORKING_DIRECTORY: fit5032.devops
jobs:
  build-and-deploy:
    runs-on: ubuntu-latest
    steps:
    - uses: actions/checkout@v2
    - name: Setup .NET Core
      uses: actions/setup-dotnet@v1
      with:
        dotnet-version: ${{ env.DOTNET_CORE_VERSION }}
    - name: Restore
      run: dotnet restore "${{ env.WORKING_DIRECTORY }}"
    - name: Build
      run: dotnet build "${{ env.WORKING_DIRECTORY }}" --configuration ${{ env.CONFIGURATION }} --no-restore
    - name: Test
      run: dotnet test "${{ env.WORKING_DIRECTORY }}" --no-build
    - name: Publish
      run: dotnet publish "${{ env.WORKING_DIRECTORY }}" --configuration ${{ env.CONFIGURATION }} --no-build --output "${{ env.AZURE_WEBAPP_PACKAGE_PATH }}"
    - name: Deploy to Azure WebApp
      uses: azure/webapps-deploy@v2
      with:
        app-name: ${{ env.AZURE_WEBAPP_NAME }}
        package: ${{ env.AZURE_WEBAPP_PACKAGE_PATH }}
        publish-profile: ${{ env.AZURE_WEBAPP_PUBLISH_PROFILE }}
    - name: Publish Artifacts
      uses: actions/upload-artifact@v1.0.0
      with:
        name: webapp
        path: ${{ env.AZURE_WEBAPP_PACKAGE_PATH }}
```

In Visual Studio, it would show that the workflow is running



fit5032-az-devops.yml
Azure App Service (Linux)
CI/CD using GitHub Actions

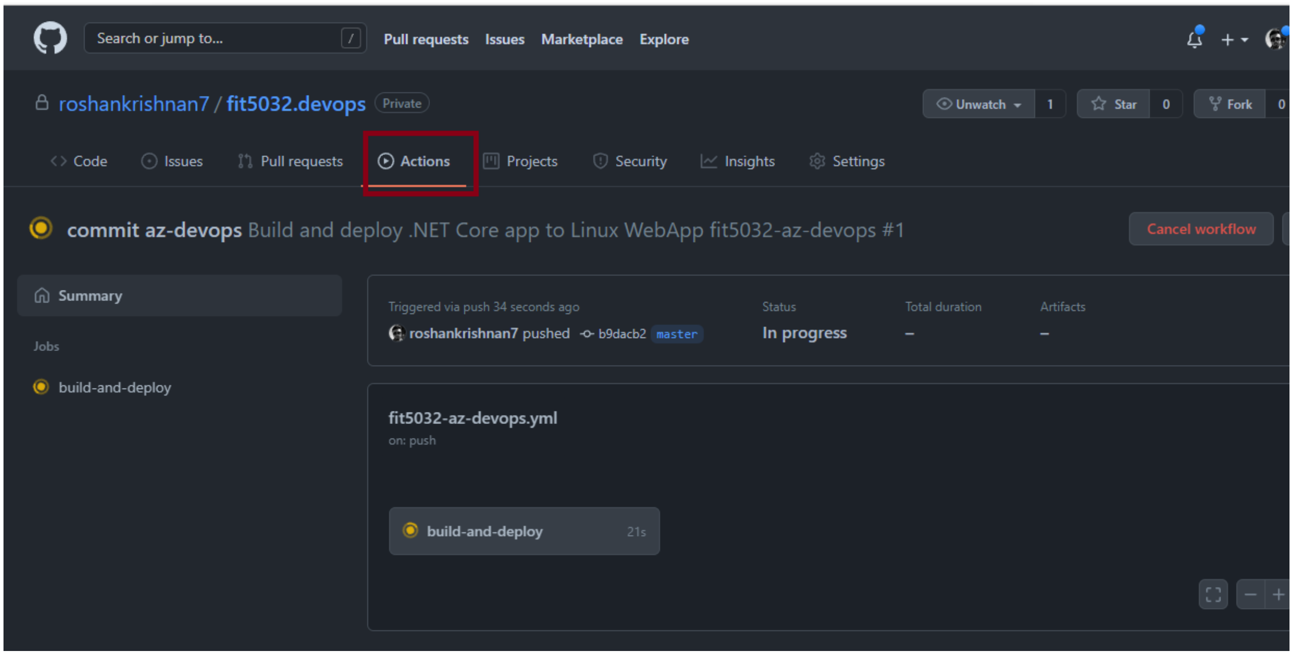
Workflow is running ...
Open in GitHub | Cancel workflow run

GitHub Actions workflow
Triggers: push
Publish profile secret: FIT5032_AZ_DEVOPS_D7C4
Edit workflow

Settings
Configuration: Release
.NET SDK Version: 5.0.x
Show all settings

Hosting
Subscription: 71e9c433-98f7-447a-9528-267f083b2c4d

Go to your GitHub repo and Look under the Actions tab, you should see that the CI/CD pipeline is now running



roshankrishnan7 / fit5032.devops Private

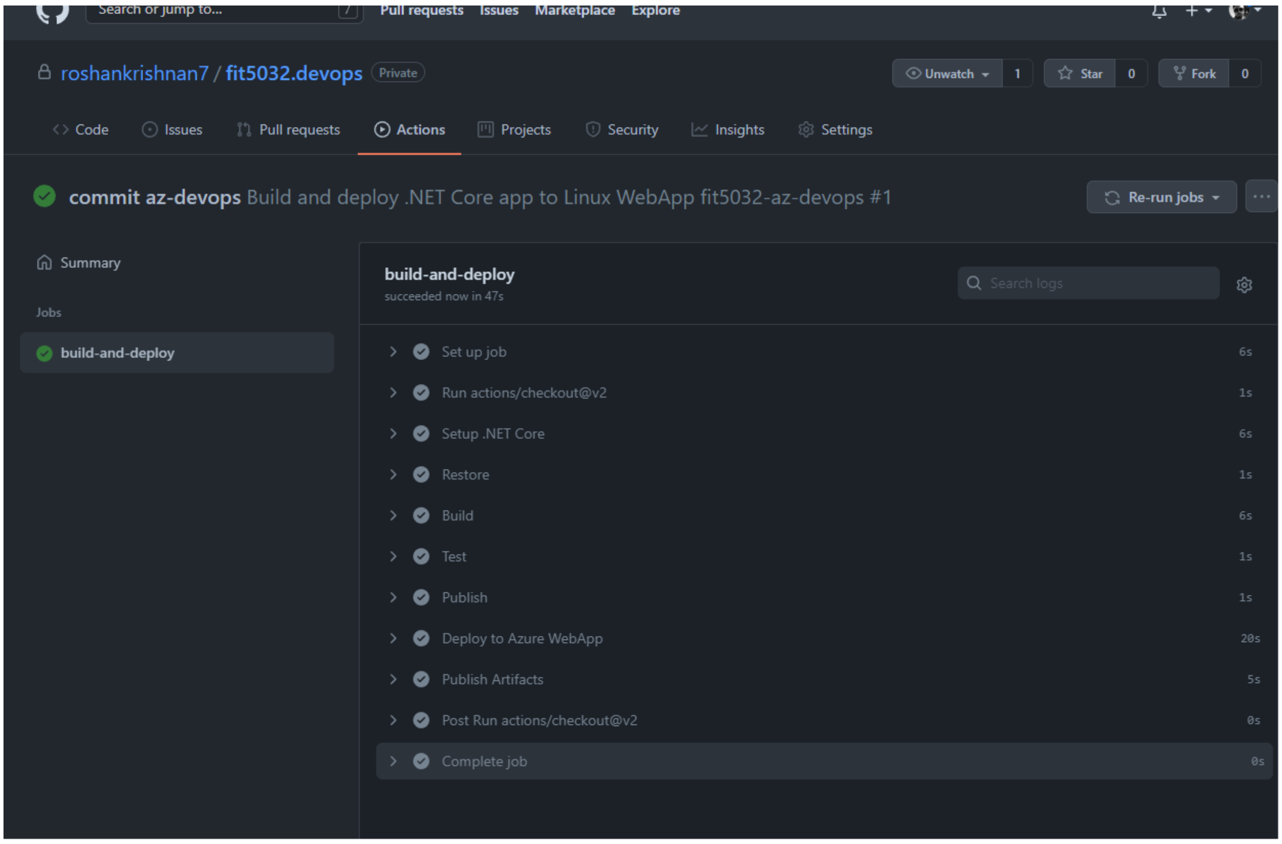
commit az-devops Build and deploy .NET Core app to Linux WebApp fit5032-az-devops #1

Triggered via push 34 seconds ago
roshankrishnan7 pushed b9dacb2 master
Status: In progress
Total duration: –
Artifacts: –

build-and-deploy

fit5032-az-devops.yml
on: push
build-and-deploy 21s

Once the run is successfully completed, you should see the various steps that the CI/CD pipeline executed, including Build, Test and Deploy to Azure Web App.

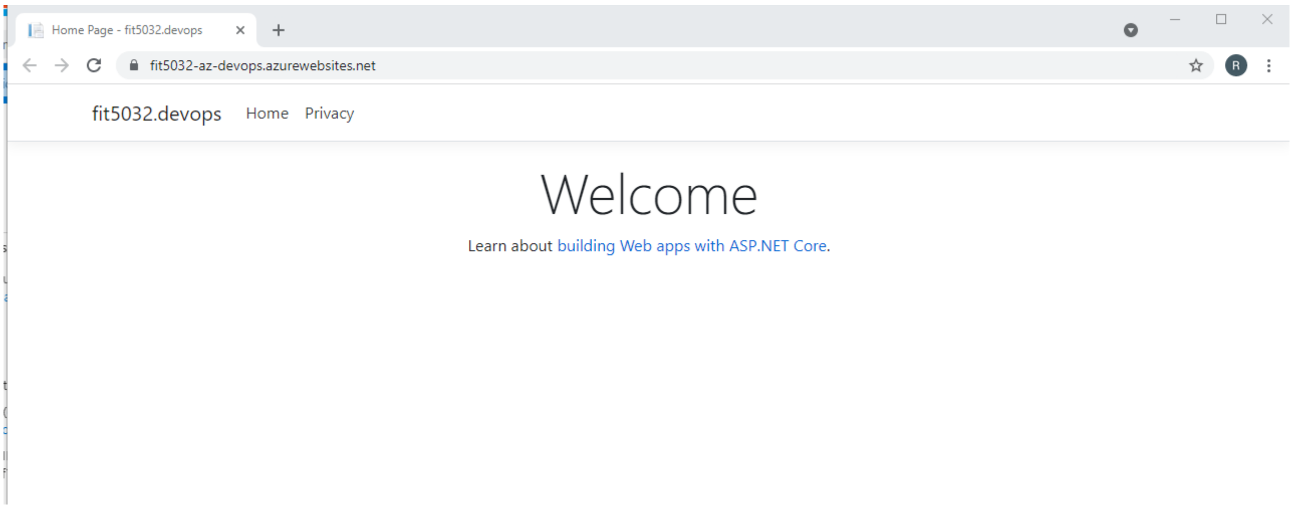


commit az-devops Build and deploy .NET Core app to Linux WebApp fit5032-az-devops #1

build-and-deploy
succeeded now in 47s

Set up job	6s
Run actions/checkout@v2	1s
Setup .NET Core	6s
Restore	1s
Build	6s
Test	1s
Publish	1s
Deploy to Azure WebApp	20s
Publish Artifacts	5s
Post Run actions/checkout@v2	0s
Complete job	0s

Go to the URL for the app service that you have created (https://**<name of app service>**.azurewebsites.net). You should see your MVC web application hosted there.


Home Page - fit5032.devops
fit5032-az-devops.azurewebsites.net
fit5032.devops
Home
Privacy

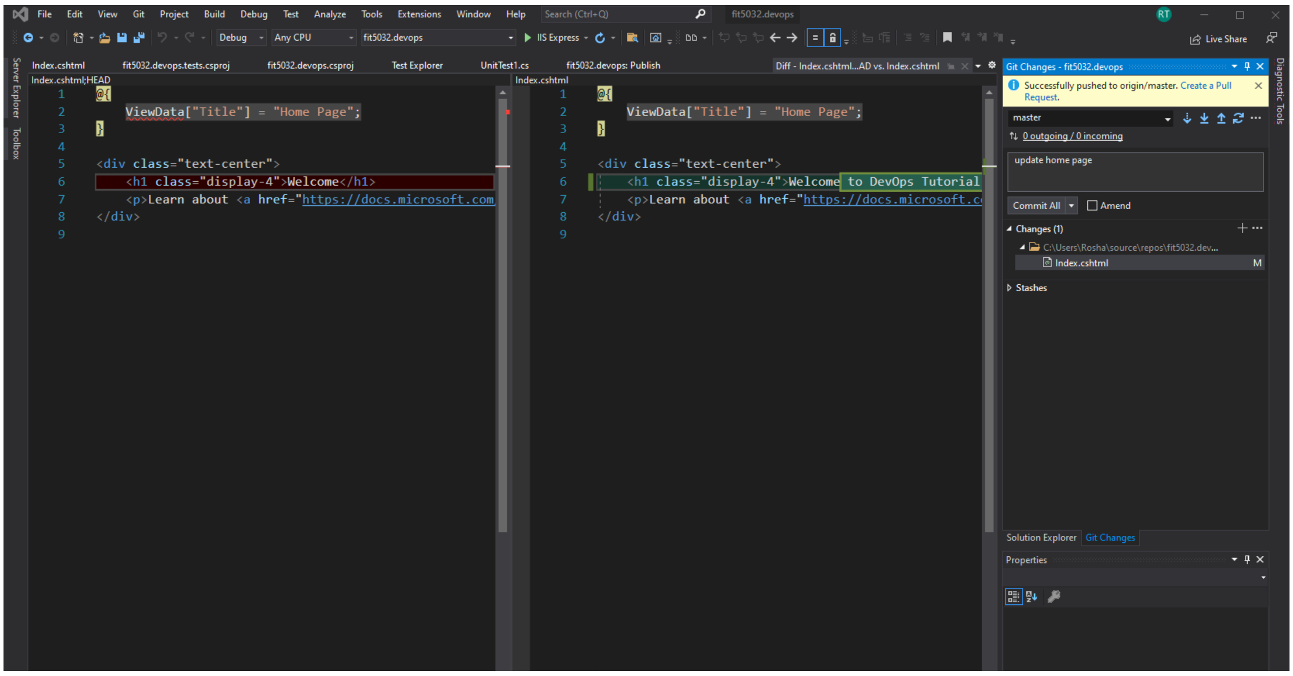
Welcome

Learn about building Web apps with ASP.NET Core.

Step 5:

CI/CD in Action

Now let us see how automated CI/CD pipeline works as soon as you make code changes to your repository. Let us go ahead and make some changes to your code. Let us modify the text displayed in our Default landing page. Commit and push the changes to GitHub repository.



Index.cshtml;HEAD

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

<div class="text-center">
    <h1 class="display-4">Welcome</h1>
    <p>Learn about <a href="https://docs.microsoft.com
</div>
```

Index.cshtml

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

<div class="text-center">
    <h1 class="display-4">Welcome to DevOps Tutorial
    <p>Learn about <a href="https://docs.microsoft.c
</div>
```

Git Changes - fit5032.devops

Successfully pushed to origin/master. Create a Pull Request.

master

0 outgoing / 0 incoming

update home page

Commit All ☐ Amend

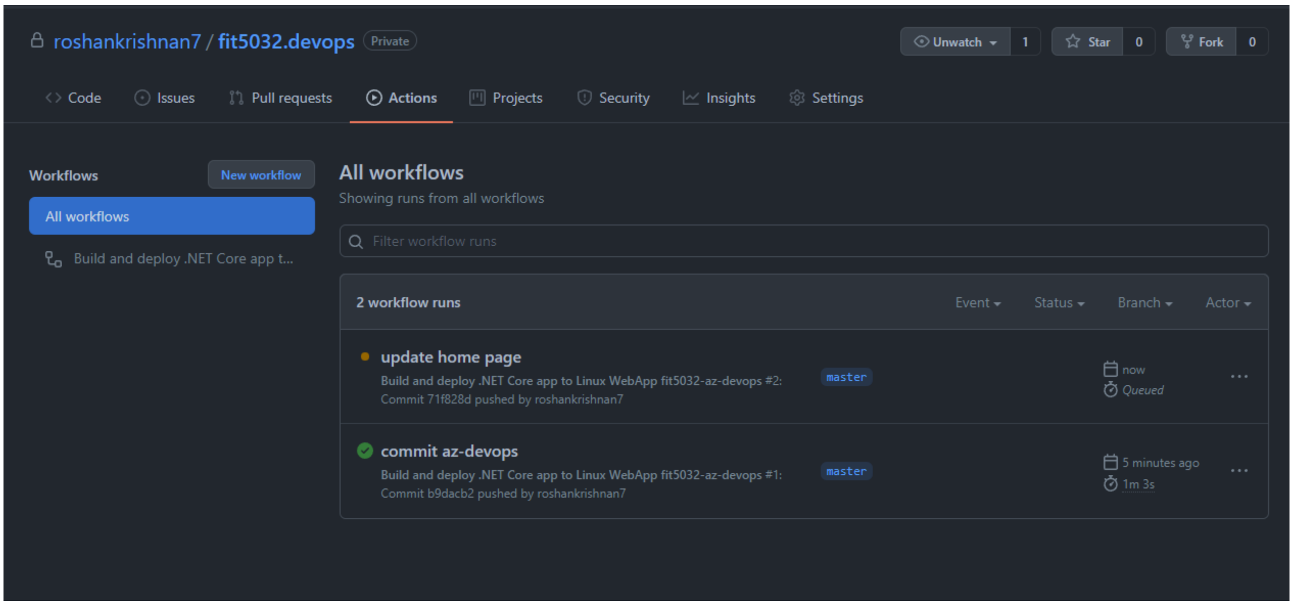
Changes (1)

C:\Users\Rosha\source\repos\fit5032.dev...

Index.cshtml M

Stashes

Go to GitHub and look under Actions tab. You should now see a new workflow running. This was triggered by our commit and push to the repository.



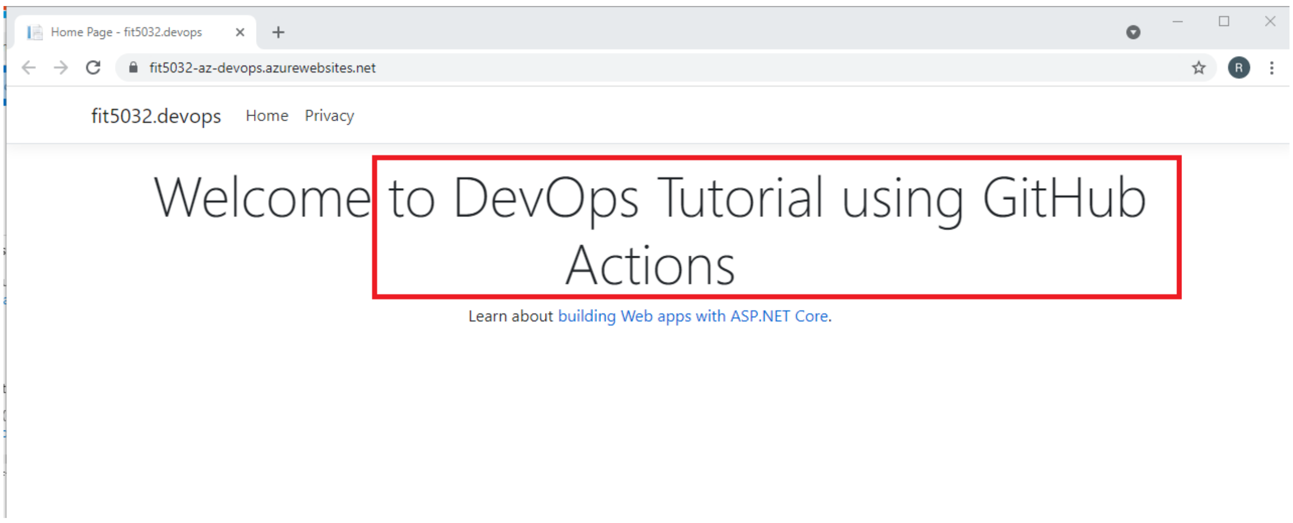
All workflows

Showing runs from all workflows

2 workflow runs

update home page Build and deploy .NET Core app to Linux WebApp fit5032-az-devops #2: Commit 71f828d pushed by roshankrishnan7	master	now Queued	...
commit az-devops Build and deploy .NET Core app to Linux WebApp fit5032-az-devops #1: Commit b9dacb2 pushed by roshankrishnan7	master	5 minutes ago 1m 3s	...

Visit the web page again once the new workflow finishes executing. You should see the changes that you made


Home Page - fit5032.devops
fit5032-az-devops.azurewebsites.net
fit5032.devops
Home
Privacy

Welcome to DevOps Tutorial using GitHub Actions

Learn about building Web apps with ASP.NET Core.

By now, you would have setup a CI/CD pipeline using GitHub actions to continuously build, test and deploy your code changes to your web application hosted in Azure App Service.

REFERENCES:

https://docs.github.com/en/actions

https://azure.microsoft.com/en-au/services/app-service/#overview

https://www.c-sharpcorner.com/article/github-actions-azu-continuous-deployment-of-asp-net-core-with-dotvvm-applic/