

Bootcamp-Project 1

BUILD A CI/CD PIPELINE FOR A .NET WEB APPLICATION USING
AZURE DEVOPS

Mohanramrajan Erran Bothalraj

Table of Contents

Introduction	2
Project Objectives	2
Expected Outcome	2
Architecture Diagram	2
Pre-Requisite	3
Solution Steps	3
Part 1: Source Code Management (Azure Repos).....	3
Part 2: Continuous Integration (Build Pipeline)	4
Part 3: Continuous Deployment (Release Pipeline)	7
Part 4: Approval Gates Configuration	7
Part 5: Execution and Validation	8
Validation	8
Part 6: Azure Monitor	10

Introduction

In modern software development, **Continuous Integration (CI) and Continuous Deployment (CD)** play a crucial role in ensuring efficient, automated, and reliable application delivery. This project focuses on setting up a **CI/CD pipeline** in **Azure DevOps** for a .NET application, enabling seamless code integration, automated testing, and controlled deployments to Azure App Service.

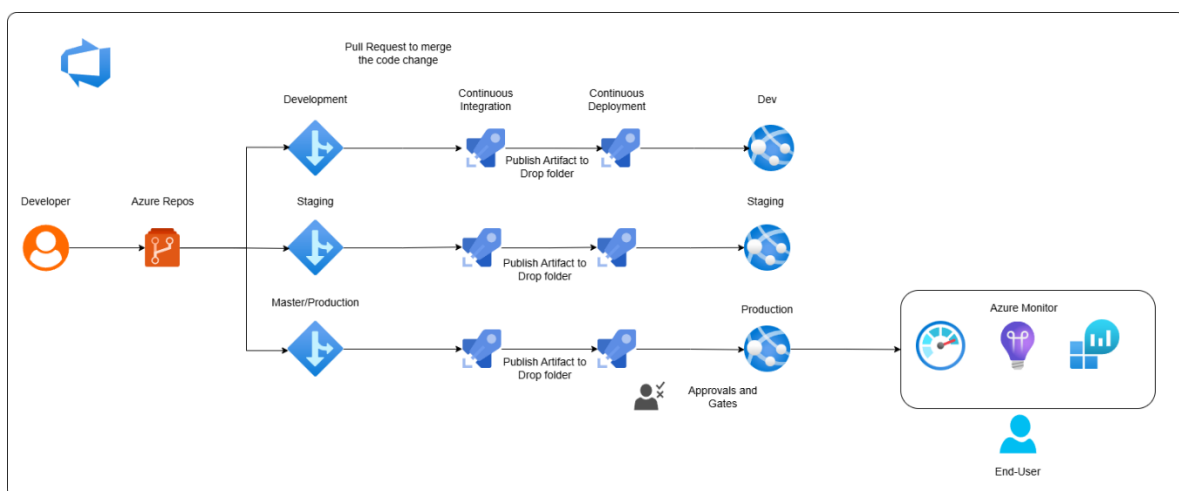
Project Objectives

- **Source Code Management:** Store and manage the application code in **Azure Repos (Git)**.
- **Automated Build Process:** Set up a **CI pipeline** to restore dependencies, build the application, and run unit tests.
- **Deployment Automation:** Configure a **CD pipeline** to deploy the application to different environments (Dev, Staging, Production).
- **Approval Workflows:** Implement **manual approval gates** for production deployments.
- **Application Monitoring:** Use **Azure Monitor and Application Insights** for real-time tracking of performance and errors.

Expected Outcome

By implementing this CI/CD pipeline, the project ensures **faster development cycles, high-quality releases, and efficient application monitoring**, enabling a **DevOps-driven** approach for smooth application deployment and maintenance.

Architecture Diagram



Pre-Requisite

1. GitHub repository with a .Net application with Unit tests:
<https://github.com/merranbo1989/newlocalrepo.git>
2. Create Web Apps for 3 environments (Dev, Staging, and Production) using Azure App Services

Home > App Services

Default Directory (merranbooutlook.onmicrosoft.com)


+ Create Manage Deleted Apps Manage view Refresh Export to CSV Open query Assign tags Start Restart Stop Delete

Filter for any field... Subscription equals all Resource group equals all Location equals all Add filter

Showing 1 to 3 of 3 records.

Name	Status	Location	Pricing Tier	App Service Plan	Subscription	App Type
DevEnv001	Running	Canada Central	Free	ASP-BootcampProject1-b320	Azure subscription 1	Web App
ProdEnv001	Running	Canada Central	Free	ASP-BootcampProject1-b320	Azure subscription 1	Web App
StageEnv001	Running	Canada Central	Free	ASP-BootcampProject1-b320	Azure subscription 1	Web App

3. Create a new Organization and Project in Azure DevOps environment.



Edit profile

Azure DevOps Organizations

Create new organization

dev.azure.com/merranbo1989 (Owner)

Projects

BootCamp-Project1

New project

Actions

Open in Visual Studio

Solution Steps

Part 1: Source Code Management (Azure Repos)

- A. Create a New DevOps Project under organization

Azure DevOps merranbo1989 / BootCamp-Project1 / Overview / Summary

Search

BootCamp-Project1

Private Invite

Overview Summary Dashboards Wiki Boards Repos

Project stats

- B. Import the .Net application code from the GitHub repository (Pre-Requisite step 1) into the Azure Repo (BootCamp-Project1)

Import a repository

Import

BootCamp-Project1 is empty. Add some code!

Clone to your computer

HTTPS SSH https://merranbo1989@dev.azure.com/merranbo1989/BootCamp-Pr OR Clone

Learn more about HTTPS

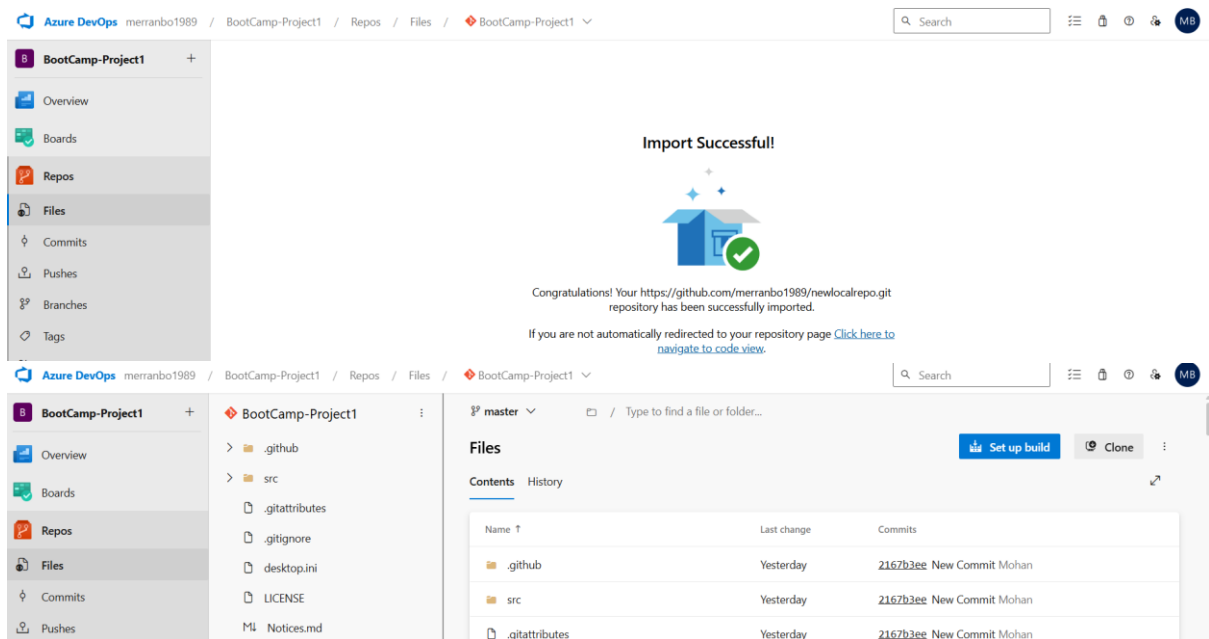
Generate Git Credentials

Import a Git repository

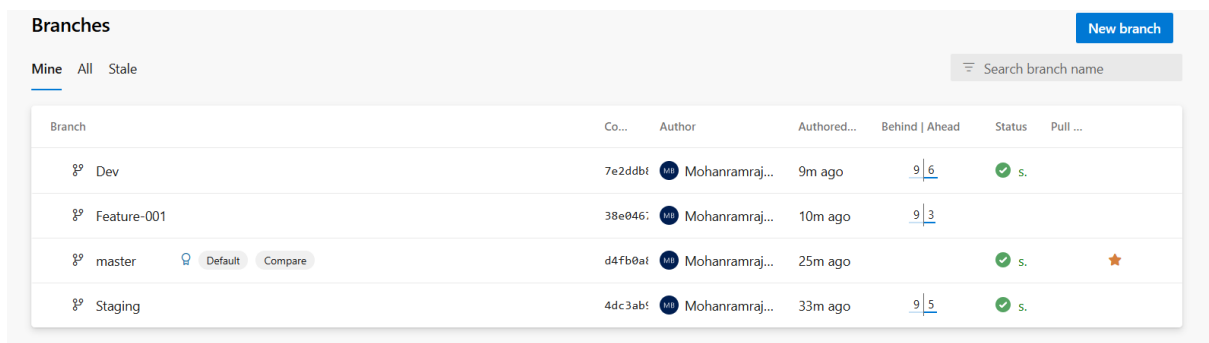
Repository type Git

Clone URL * https://github.com/merranbo1989/newlocalrepo.git

Requires Authentication

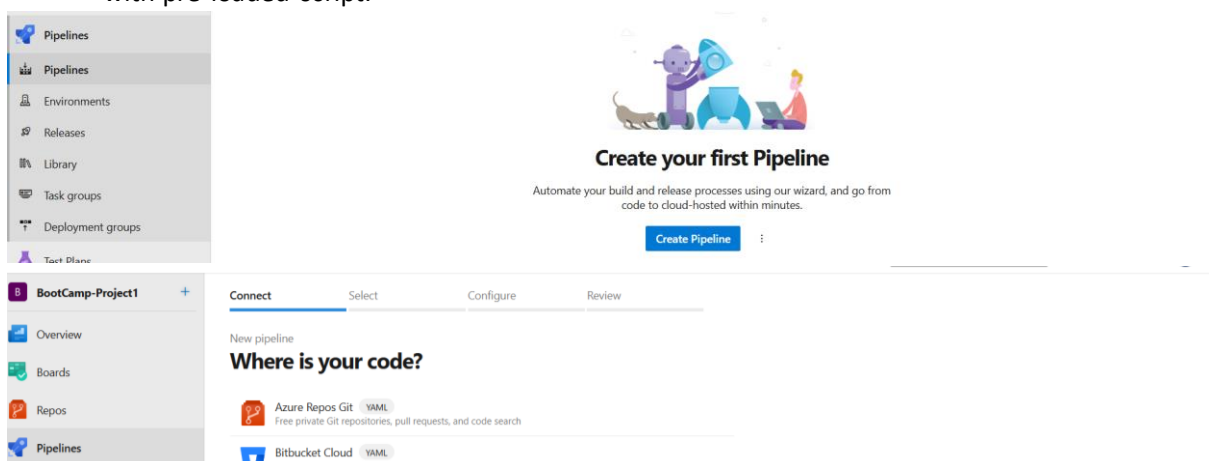


- C. Create the following branches in the Azure Repos for code management
- Master (default): used for stable code to be deployed in PROD environment
 - Dev: used by develops normally contains mirror image of PROD environment
 - Staging: used for testing or UAT purposes
 - Feature: this branch is used to update any new/existing feature in the code and then merged into Dev branch.



Part 2: Continuous Integration (Build Pipeline)

- A. Create an Azure Build pipeline using ASP.NET template. Where a YAML file will be created with pre-loaded script.




✓ Connect **Select** Configure Review

New pipeline

Select a repository

Filter by keywords BootCamp-Project1 ✕

 BootCamp-Project1

✓ Connect ✓ Select **Configure** Review

New pipeline

Configure your pipeline



ASP.NET
Build and test ASP.NET projects.



ASP.NET Core (.NET Framework)
Build and test ASP.NET Core projects targeting the full .NET Framework.



.NET Desktop
Build and run tests for .NET Desktop or Windows classic desktop solutions.

B. Update the Code in the YAML file attached here:

***** Code Starts Here *****

ASP.NET Core (.NET Framework)

Build and test ASP.NET Core projects targeting the full .NET Framework.

Add steps that publish symbols, save build artifacts, and more:

<https://docs.microsoft.com/azure/devops/pipelines/languages/dotnet-core>

trigger:

branches:

include:

- master*
- Dev*
- Staging*

pool:

vmImage: 'windows-latest'

variables:

*solution: '**/*.sln'*

buildPlatform: 'Any CPU'

buildConfiguration: 'Release'

steps:

- task: NuGetToolInstaller@1

displayName: NuGet Install'

```

- task: NuGetCommand@2
  displayName: 'Restore Dependencies'
  inputs:
    command: 'restore'
    restoreSolution: '$(solution)'

- task: VSBUILD@1
  displayName: 'Build'
  inputs:
    solution: '$(solution)'
    msbuildArgs: '/p:DeployOnBuild=true /p:WebPublishMethod=Package /p:PackageAsSingleFile=true
/p:SkipInvalidConfigurations=true
/p:DesktopBuildPackageLocation="$(build.artifactStagingDirectory)\WebApp.zip"
/p:DeployIisAppPath="Default Web Site"'
    platform: '$(buildPlatform)'
    configuration: '$(buildConfiguration)'

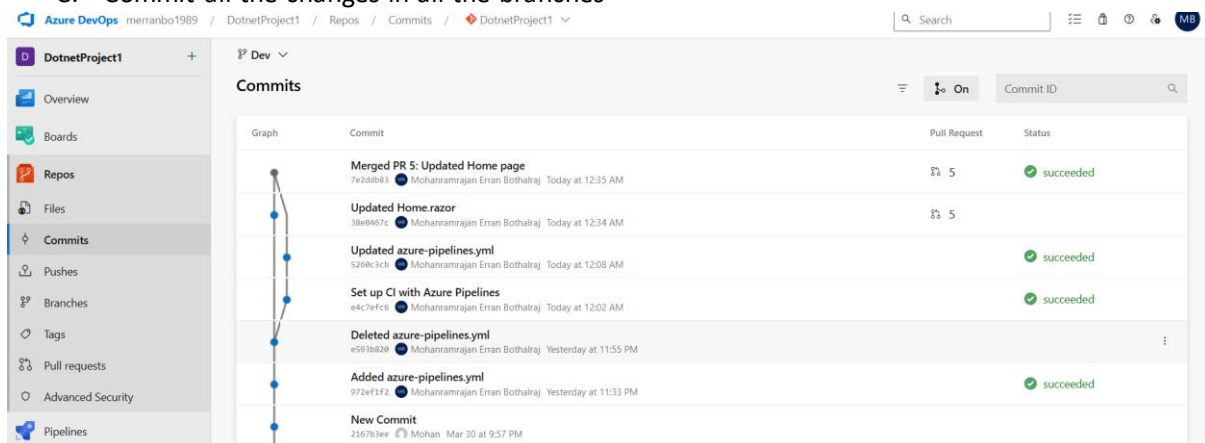
- task: DotNetCoreCLI@2
  displayName: 'Unit Test'
  inputs:
    command: 'test'
    projects: '**/*[Tt]est*/*.csproj'

- task: PublishBuildArtifacts@1
  displayName: 'Publish Artifact'
  inputs:
    PathToPublish: '$(Build.ArtifactStagingDirectory)'
    ArtifactName: 'drop'
    publishLocation: 'Container'

```

***** Code Ends Here *****

C. Commit all the changes in all the branches



Commit	Pull Request	Status
Merged PR 5: Updated Home page 7e2d0811 Mohanramrajan Erran Bothalraj Today at 12:35 AM	5	succeeded
Updated Home.razor 38e8667a Mohanramrajan Erran Bothalraj Today at 12:34 AM	5	succeeded
Updated azure-pipelines.yml 5268c3c1 Mohanramrajan Erran Bothalraj Today at 12:08 AM		succeeded
Set up CI with Azure Pipelines e4c7efcc Mohanramrajan Erran Bothalraj Today at 12:02 AM		succeeded
Deleted azure-pipelines.yml e593b82b Mohanramrajan Erran Bothalraj Yesterday at 11:55 PM		succeeded
Added azure-pipelines.yml 972ef1f2 Mohanramrajan Erran Bothalraj Yesterday at 11:33 PM		succeeded
New Commit 2167b3ee Mohan Mar 30 at 9:57 PM		

D. A new Pipeline will be created and will run automatically

Part 3: Continuous Deployment (Release Pipeline)

A. Create three separate pipelines for Dev, Staging, and PROD environments

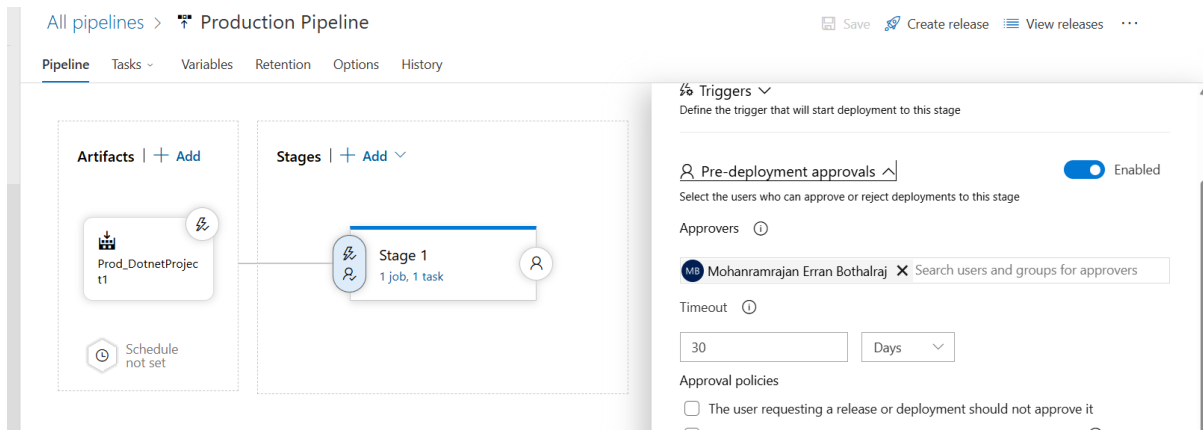
Name	Releases	Release pipeline	Created	Stages
<ul style="list-style-type: none"> Dev Pipeline Production Pipeline Staging Pipeline 	<ul style="list-style-type: none"> Release-2 (20250407.5) - Dev Release-1 (20250407.4) - master Release-1 (20250407....) - Staging Release-1 (20250407.2) - Dev 	<ul style="list-style-type: none"> Dev Pipeline Production Pipeline Staging Pipeline Dev Pipeline 	<ul style="list-style-type: none"> 4/7/2025, 12:39:20 AM 4/7/2025, 12:24:07 AM 4/7/2025, 12:17:29 AM 4/7/2025, 12:11:29 AM 	<ul style="list-style-type: none"> Stage 1 Stage 1 Stage 1 Stage 1

B. Configure the Dev pipeline and task to ensure the release pipeline is triggered automatically when a new build is triggered from Dev branch.

NOTE: Repeat the step for STAGING and PRODUCTION Environment

Part 4: Approval Gates Configuration

A. Implement Manual approval of a User for Pre-Deployment approvals for Production stage.



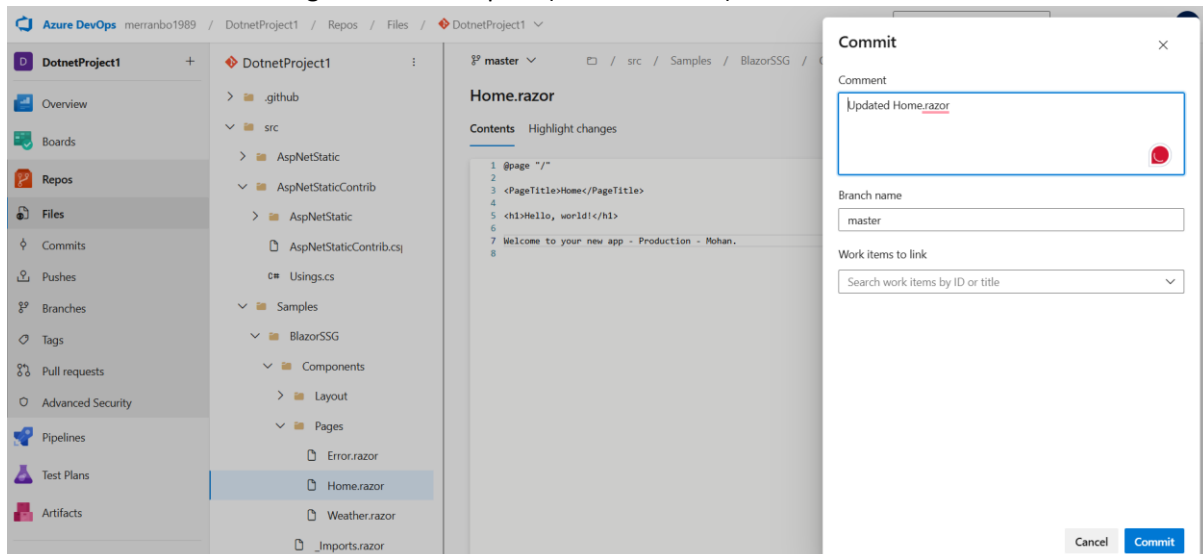
Part 5: Execution and Validation

Once the .Net application code in the “Azure Repos” are committed with any new change.

1. Azure Build Pipeline will be triggered automatically, which will then perform the tasks:
 - a. Restore Dependencies
 - b. Build
 - c. Unit Test
 - d. Publish the Artifact in Drop folder
2. After successful completion of the Build pipeline, Release pipeline will be triggered automatically, where the application will be deployed to “Development” environment followed by “Staging” Environment.
3. Finally, after fulfilling the Pre-Deployment checks, Gates, and approvals configured in “Part 4”. The application will be deployed to the “Production” environment.

Validation

A. Commit the changes in Azure Repos. (Master branch)



B. Build Pipeline is triggered

Overview

Boards

Repos

Pipelines

Pipelines

Environments

Releases

Library

Task groups

Deployment groups

Summary

Code Coverage

Triggered by

Mohanramrajan Erran Bothalraj

View change

Repository and version

DotnetProject1

master 8815b251

Time started and elapsed

Just now

14s

Related

0 work items

0 artifacts

Tests and coverage

Get started

Jobs

Name	Status	Duration
Job	Running	10s

DotnetProject1

Overview

Boards

Repos

Pipelines

Pipelines

Environments

Releases

Jobs in run #20250407.6

DotnetProject1

Jobs

Job	1m 43s
Initialize job	7s
Checkout DotnetProjec...	4s
NuGet Install	1s
Restore Dependencies	40s

Job

Pool: Azure Pipelines

Image: windows-latest

Agent: Hosted Agent

Started: Just now

Duration: 1m 43s

Job preparation parameters

Artifact produced

Job line console data:

Finishing: Job

View raw log

C. New Release pipeline is triggered

Boards

Repos

Pipelines

Pipelines

Environments

All pipelines

Dev Pipeline

Production Pipeline

Staging Pipeline

Pending approval on Prod stage.

Releases	Release pipeline	Created	Stages
Release-2 20250407.6 master	Production Pipeline	4/7/2025, 1:32:58 AM	Prod
Release-2 20250407.5 Dev	Dev Pipeline	4/7/2025, 12:39:20 AM	Stage 1

D. Go to Azure DevOps release pipeline to "Approve" the production deployment.

Search all pipelines

All pipelines

Dev Pipeline

Production Pipeline

Staging Pipeline

All pipelines

Releases

Pending approval on Prod stage.

Release-2	20250407.6	master
Release-2	20250407.5	Dev
Release-1	20250407.4	master
Release-1	20250407.3	Staging

Release-2 > Pending approval on Prod stage.

Pre-deployment approval pending

View logs

Approval pending for 1 minute

Waiting for all approvers to approve in sequence.

Timeout in 30d

Mohanramrajan Erran Bothalraj

Pending for 1 minute

Reassign

Comment

Pipeline Approved

Defer deployment for later

Approve

Reject

Deployment process

Queued

Pre-deployment approvals

Approved

Run on agent

Not started

Preparing an agent for the job

Waiting for the request to be queued

E. Validate the Production environment by launching the web page.

Pipeline

Tasks

Variables

Logs

Tests

Deploy

Cancel

Refresh

Download all logs

Edit

...

Deployment process

Succeeded

Pre-deployment approvals

Approved

Run on agent

Succeeded

Run on agent

Started: 4/7/2025, 1:38:26 AM

Pool: Hosted Windows 2019 with ...

Agent: Hosted Agent

1m 21s

Initialize job	succeeded	7s
Download artifact - Prod_DotnetProject1 - drop	succeeded	5s
Deploy Azure App Service	succeeded	1m 7s
Finalize Job	succeeded	<1s

ProdEnv001 Web App

Search

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Microsoft Defender for Cloud

Events (preview)

Recommended services (preview)

Log stream

Resource visualizer

Deployment

Essentials

Resource group (move) : [NCPL](#)

Status : Running

Location (move) : [Canada Central](#)

Subscription (move) : [Azure subscription 1](#)

Subscription ID : 3f48112c-a56d-44e0-9a80-1ed5fbc3aac

Tags (edit) : [Add tags](#)

Default domain : [prodenv001.azurewebsites.net](#)

App Service Plan : [ASP-NCPL-b28b \(B1: 1\)](#)

Operating System : Windows

Health Check : Not Configured

Properties

Monitoring

Logs

Capabilities

Notifications (0)

Recommendations

Web app

Name : ProdEnv001

Publishing model : Code

Deployment Center

Deployment logs

Last deployment : Successful on Monday, April 7, 01:39:47 AM

View logs

Refresh

prodenv001.azurewebsites.net

Home

Weather

Hello, world!

Welcome to your new app - Production - Mohan.

About

NOTE: Repeat the Steps for Dev, and Staging environments

Part 6: Azure Monitor

Track the application performance using Azure Monitor Application Insights.

A. Enable "Application Insights" in Prod environment.

ProdEnv001 Application Insights

Web App

Search

API

API Management

API definition

CORS

Monitoring

Alerts

Application Insights

Collect application monitoring data using Application Insights

Enable Disable

Feedback

Link to an Application Insights resource

Your app is connected to Application Insights resource: [ProdEnv001](#)

B. Create the resource Log Analytics Workspace

ProdEnv001 Application Dashboard

Application Insights

Search

Application Dashboard

Getting started

Search

Logs

Monitor resource group

Feedback

Favorites

Rename

Delete

Essentials

Resource group (move) : [NCPL](#)

Location : [Canada Central](#)

Subscription (move) : [Azure subscription 1](#)

Subscription ID : 3f48112c-a56d-44e0-9a80-1ed5fbc3aac

Tags (edit) : [Add tags](#)

Instrumentation Key : 462c918d-321c-4310-93e2-ab084ef0b951

Connection String : InstrumentationKey=462c918d-321c-4310-93e2-ab084ef0b951;Ingestio...

Workspace : [DefaultWorkspace-3f48112c-a56d-44e0-9a80-1ed5fbc3aac-CCAN](#)

Show data for last:

30 minutes 1 hour 6 hours 12 hours 1 day 3 days 7 days 30 days

Failed requests

Server response time

Server requests

Validation for PROD environment availability using the Alert rule created

Home > App Services > ProdEnv001 | Availability > Alert rules > app-availability-prodenv001

Alert rules

+ Create | Columns ...

Search

Name ↑↓

app-availability-prodenv001 ...

app-availability-prodenv001 | Action groups ☆ ...

Metric alert rule

Search

Manage action groups + Create action group

An action group invokes a defined set of notifications and actions when an alert is triggered. [Learn more](#)

Name	Contains actions
AppDowntime	1 Email

Overview

Activity log

Access control (IAM)

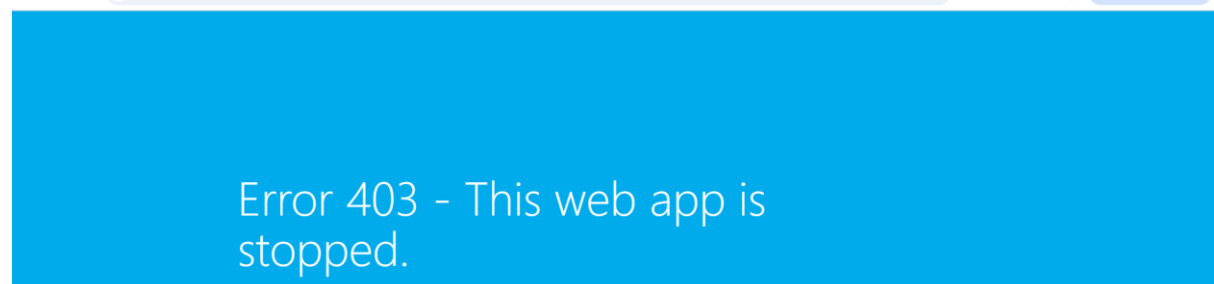
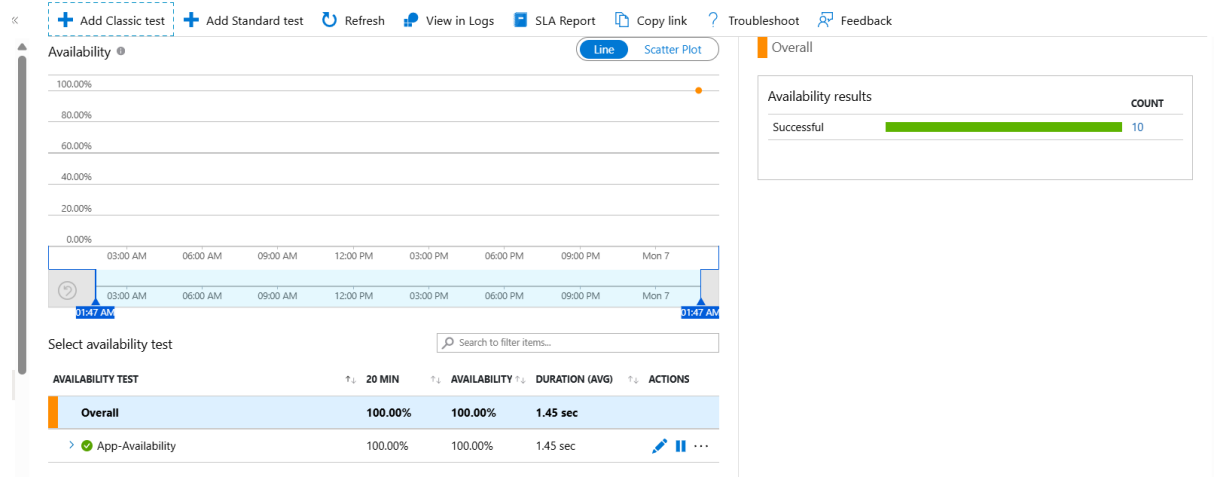
Tags

Diagnose and solve problems

History

Resource visualizer

Stop the “Prod” environment to check if the alert rule triggers the failure email to the user configured.



Check the Application Insights availability section and check the mailbox for failure email

Fired: Sev1 Azure Monitor Alert app-availability-prodenv001 on prodenv001 (microsoft.insights/components) at 4/7/2025 5:53:37 AM

Microsoft Azure <azure-noreply@microsoft.com> Unsubscribe

1:54 AM (0 minutes ago)

Fired: Sev1 Azure Monitor Alert app-availability-prodenv001 on prodenv001 (microsoft.insights/components) at 4/7/2025 5:53:37 AM

[View the alert in Azure Monitor >](#) [Investigate >](#)

Summary

Alert name	app-availability-prodenv001
Severity	Sev1

