

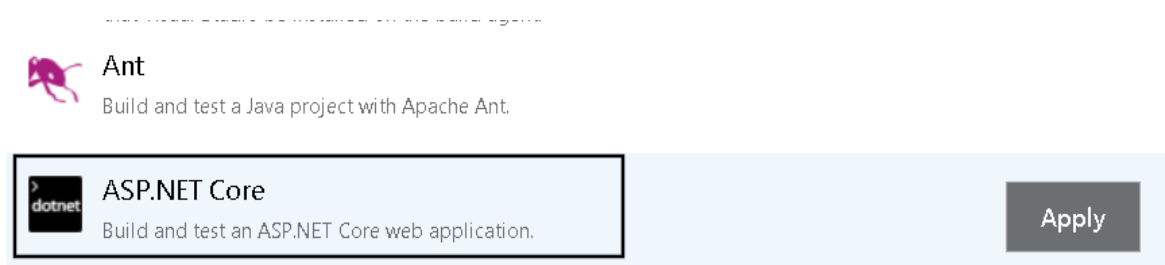
#Deploy .Net app using Pipelines

What is .Net?

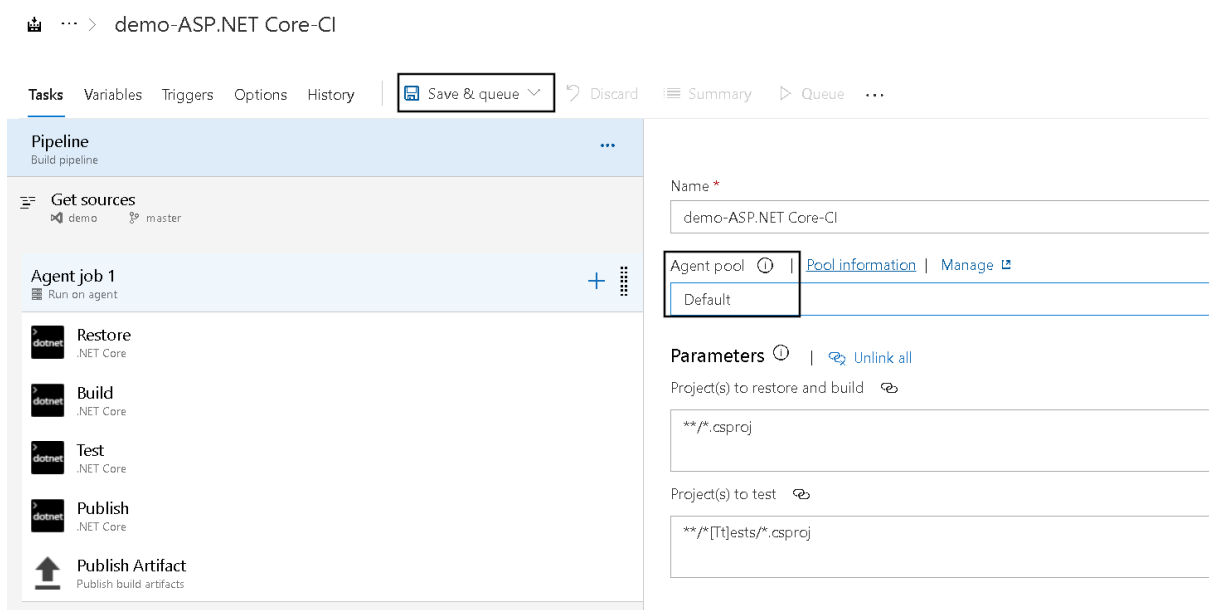
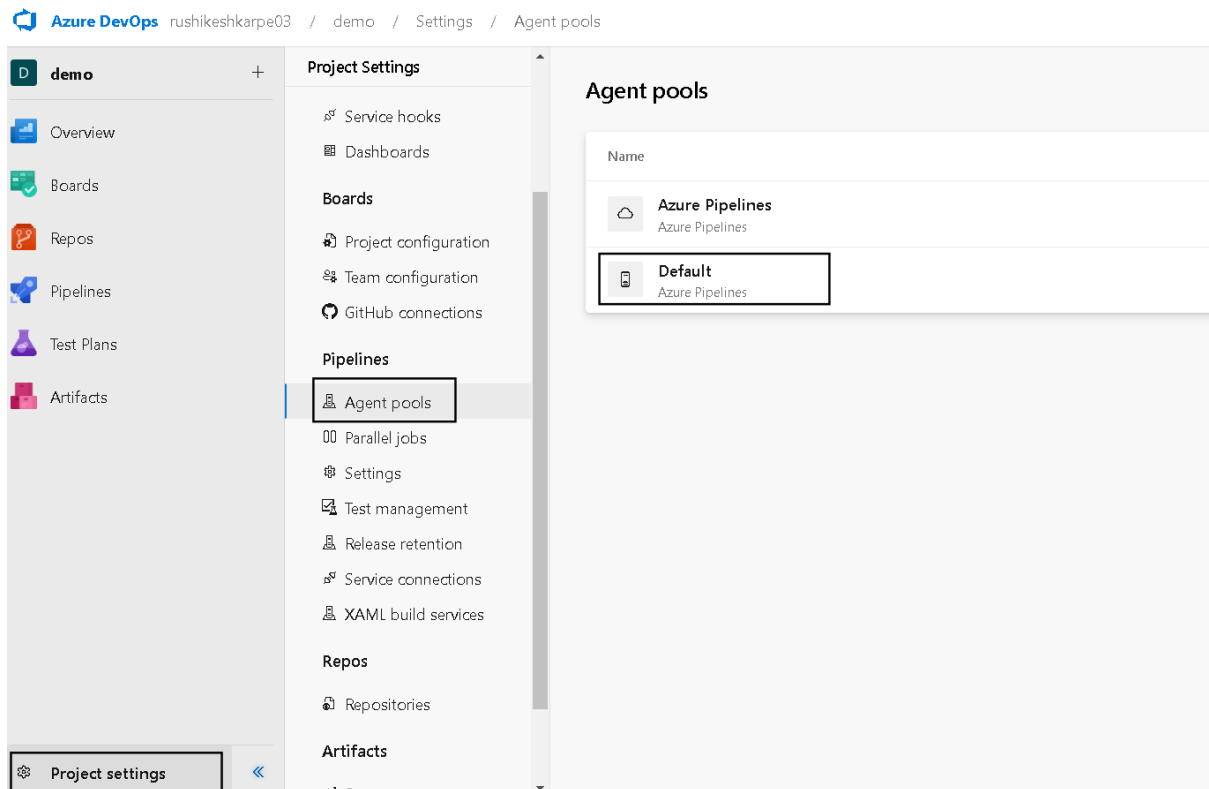
.NET is a free, cross-platform, [open-source developer platform](#) for building [many kinds of applications](#). It can run programs written in [multiple languages](#), with [C#](#) being the most popular. It relies on a [high-performance](#) runtime that is used in production by many [high-scale apps](#).

Steps—

1) Create a .net mvc using visual studio → push that on repo → active agent → go to pipeline → now enable classic editor option → go to pipeline settings → disable creation pipeline → enable that both options → got to Organization settings → disable creation of classic build pipeline → disable below also → go to project → create pipeline → click on use the classic editor button → select azure git → provide repo name and all → continue → search ASP .Net → click on that → Apply → click inside agent pool → default → Click on Save and queue button → provide comments(its optional) → save → click on agent job1 → there should be executing pipelines → its building → so we have to deploy project so we have to create a web app → duplicate the web app in Azure.



Click on Apply button to add the .Net core.



Wait for some time till all the process performed completely. Then onwards it will show you, your executing pipelines.

← Jobs in run #20250115.1

demo-ASP.NET Core-CI

Jobs

▼	Agent job 1	20s
✓	Initialize job	<1s
✓	Checkout demo@mast...	2s
✓	Restore	2s
✓	Build	8s
✓	Test	<1s
✓	Publish	4s
🔄	Publish Artifact	1s
○	Post-job: Checkout dem...	

Agent job 1

```

1 Pool: Default
2 Queued: Just now [manage_parallel_jobs]
3 Agent: DESKTOP-TDS2G8R
4 Started: Just now
5
6 The agent request is already running or has already completed.
7 ▶ Job preparation parameters
29 ▶ fx 3 queue time variables used
34 Job live console data:
35 Starting: Agent job 1
36 Async Command Start: WindowsPreinstalledGitTelemetry
37 Async Command End: WindowsPreinstalledGitTelemetry
38 Async Command Start: WindowsPreinstalledGitTelemetry
39 Async Command End: WindowsPreinstalledGitTelemetry
40 Async Command Start: WindowsPreinstalledGitTelemetry
41 Async Command End: WindowsPreinstalledGitTelemetry
42 Async Command Start: WindowsPreinstalledGitTelemetry
43 Async Command End: WindowsPreinstalledGitTelemetry
44 Finishing: Agent job 1

```

Steps—

Create a RG → Search webapp → click on app services → create → webapp → select RG → provide domain name → select .Net 8 → select region → select OS as windows → review+create → create → if error then delete a automatically created resource group → change region and try again

Resource groups

Default Directory (rushikeshkarpe03gmail.onmicrosoft.com)

[+ Create](#)
[⚙️ Manage view](#)
[↻ Refresh](#)
[⬇️ Export to CSV](#)
[🔗 Open query](#)
[🏷️ Assign tags](#)

Filter for any field...

Subscription equals **all**

Location equals **all** ✕

[+🔍 Add filter](#)

Showing 1 to 1 of 1 records.

☐ Name ↑↓

☐ 🔄 deploy

If it shows error after creating the Webapp change the region of resource group and create RG again. Delete the previous one.

Create Web App ...

Subscription * ⓘ Free Trial ▼

Resource Group * ⓘ deploy ▼
[Create new](#)

Instance Details

Name web1214 ✓
-cxd9bzegakgucweu.australiaeast-01.azurewebsites.net

☒ Secure unique default hostname on. [More about this update](#) ↗

Publish * ☒ Code ☐ Container

Runtime stack * .NET 8 (LTS) ▼

Operating System * ☐ Linux ☒ Windows

Region * Australia East ▼

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

Steps—

After creating web app → go to pipelines tab → click on release under pipeline → new pipeline → select Azure app service deployment on right side → click → click on three dots → save as template → provide name → there should be name New release pipeline → we have to do mapping of pipelines → click on → Add button in front of artifacts → select pipeline from dropdown → add → click on stages one job one task → click on azure subscription → click on free tier → authorized → login if required → click web app from azure service app.



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

Deploy a Java application to an Azure Web App.

Pipeline Tasks Variables Retention Options History

ges | + Add

Stage 1
1 job, 1 task

Stage
Stage 1

Delete Move ...

Properties
Name and owners of the stage

Stage name
Stage 1

Stage owner
RK Rishikesh Karpe

Source (build pipeline) * ⓘ
demo-ASP.NET Core-CI

Default version * ⓘ
Latest

Source alias * ⓘ
_demo-ASP.NET Core-CI



ⓘ The artifacts published by each version will be available for deployment in release pipelines. The latest successful build of **demo-ASP.NET Core-CI** published the following artifacts: **drop**.




Add


For selecting the Azure subscription, select the free tier subscription. You have to login if required. Here we have created a CI/CD pipeline.


Steps—


Click on run on agent → select default agent → click on create release


Azure subscription *  | [Manage](#) 



Free Trial (f336f6a1-20e3-48eb-8cb9-096c)    New

 Scoped to subscription 'Free Trial'

App type 



Web App on Windows 

App service name * 


web1214  


Agent selection 

Agent pool  | [Pool information](#) | [Manage](#) 

Default  

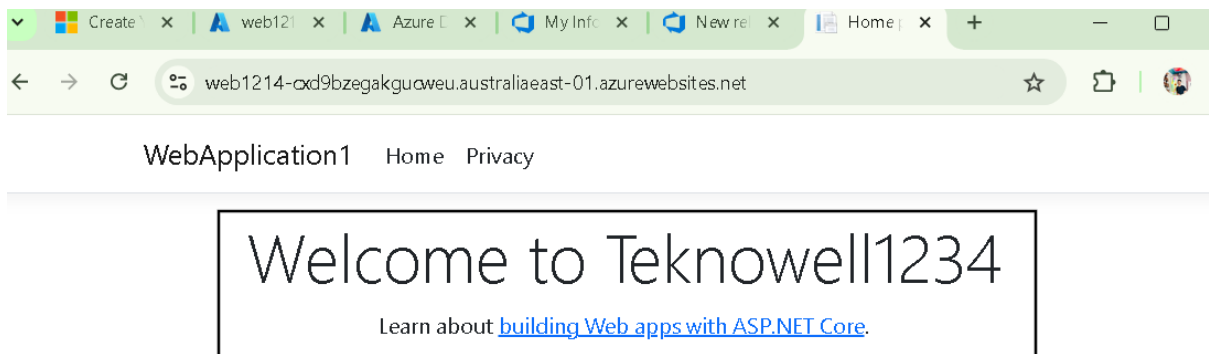
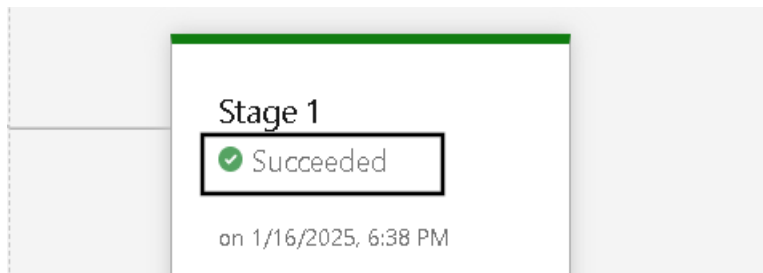
Click on release-1 from yellow bar. So the deployment has started after executing click on succeed.

 Release **Release-1** has been created

Pipeline **Tasks**  Variables Retention Options History

It takes some time to succeed. For this it is mandatory to run the agent and the status of the pipeline should be Online.

Go to Webapp copy the domain name and paste it in new tab. There should be running project.



Go to repos, change something in view of index.html and run the pipeline again