*DevOps CI/CD Azure Runbook*

*Version: 1.5*

Table of Contents

1. Set up a Development Environment in Azure3
2. Resource group4
3. Azure Active Directory6
4. Storage Account 9
5. VNET 10
6. Key Vault 10
7. App Service 10
8. Redis 11
9. Azure Service bus 12
10. AKS 13
11. Azure Monitor 17
12. About Subscription and Pricing Information 18

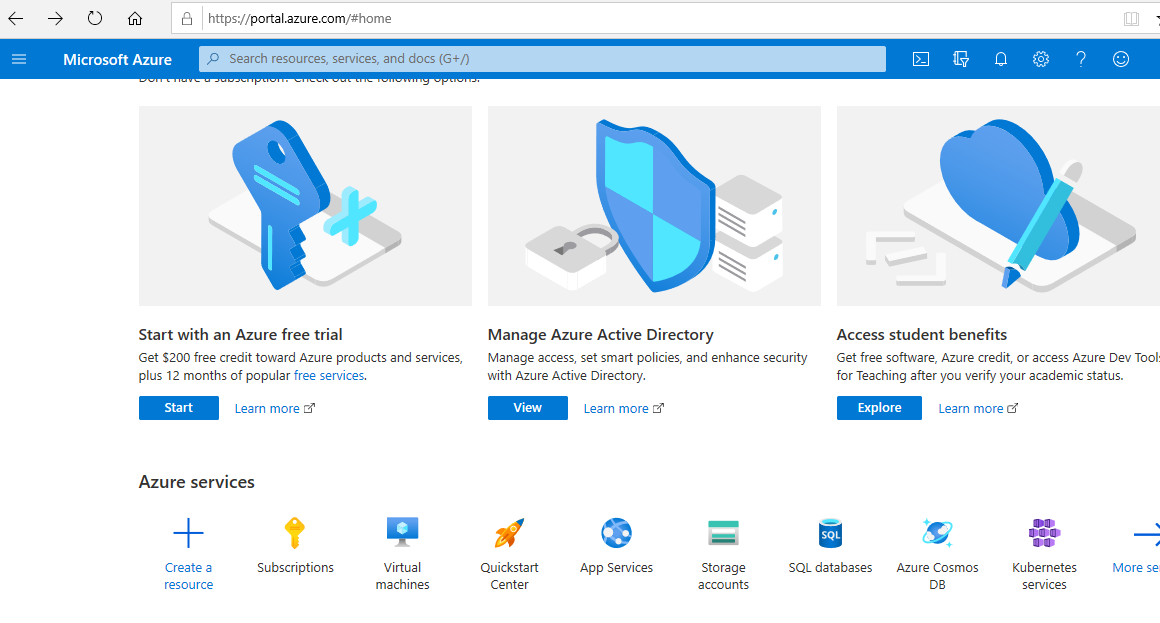
**1.Set up a Development Environment in Azure**

We will develop .NET based applications for Azure by using a Windows 10/Linux.

In the first mode, we can create apps resources in Azure by using manually /Azure Resource Manager templates.

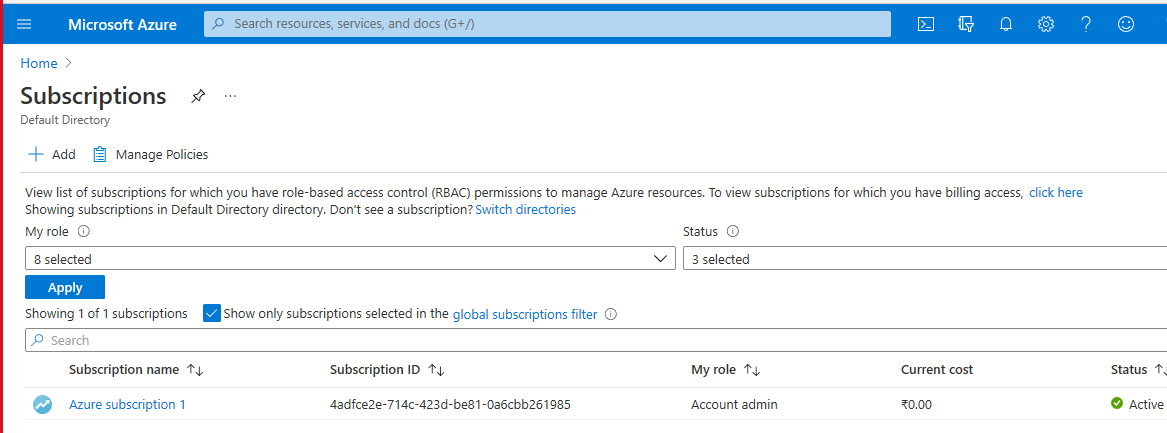
* 1. **Login to azure portal**

Login to the azure portal url: [www.portal.azure.com](http://www.portal.azure.com) by giving credentials

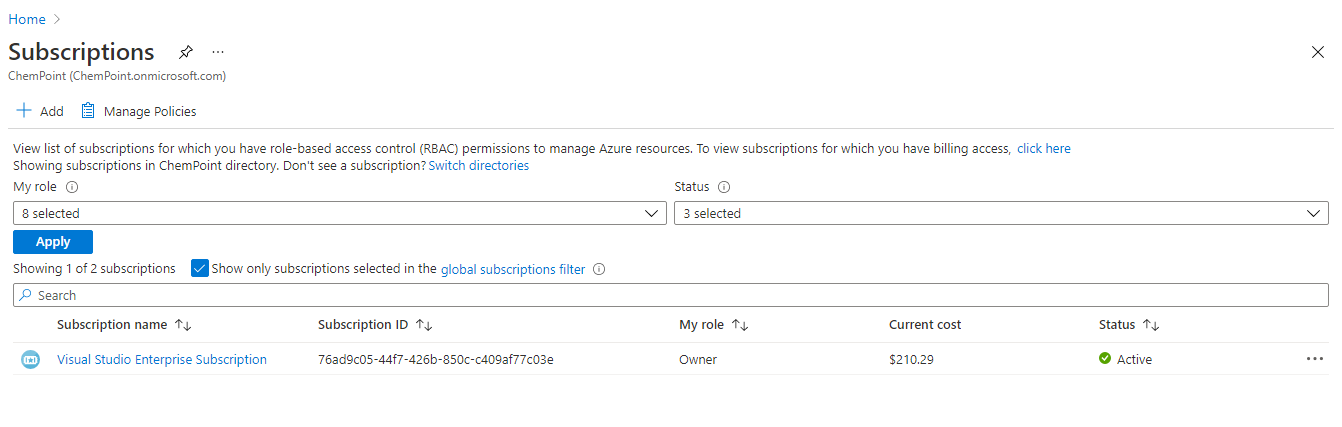


**1.2 Subscription**

* Go to the Azure Services & click on subscription
  + The primary purpose of a **subscription** is to provide a common billing paradigm for use of **Azure** services.
  + We can select Subscription like Developer and Standard
    - Developer for Non-prod Environments and Standard for Production Environment.
  + After selecting the subscription here we can see the details like subscription name, Subscription Id, role and billing details.



Click on the Subscription Name you will see the below details.

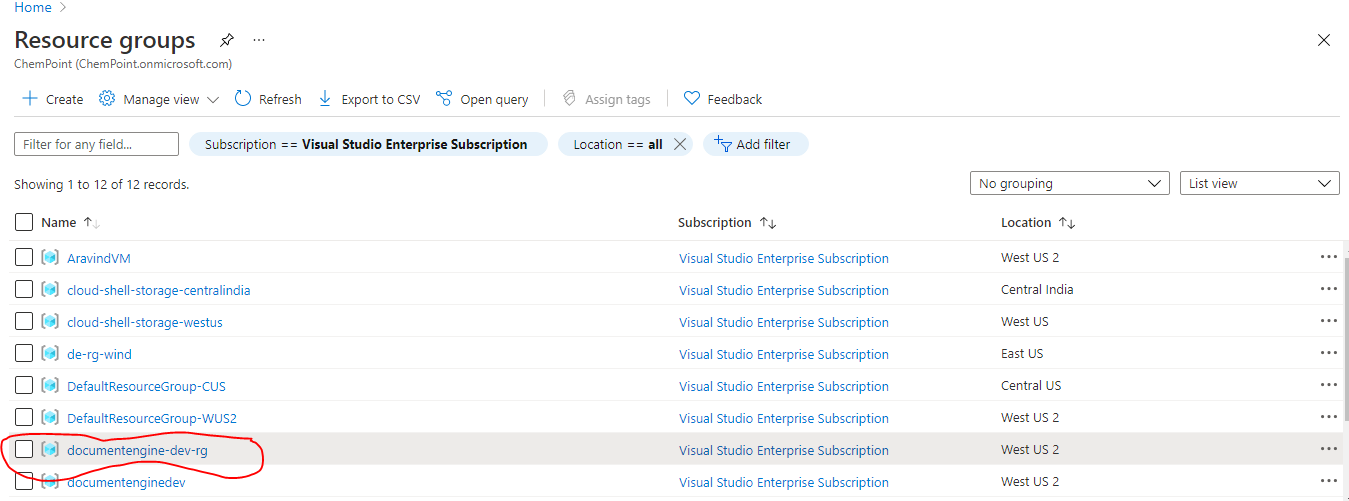


**2. Resource group**

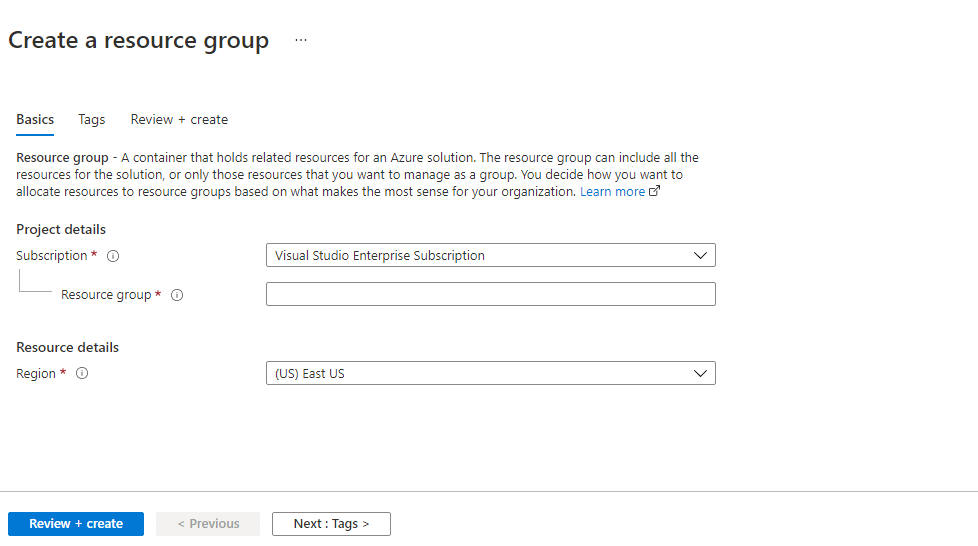
Azure Resources Groups are logical collections of virtual machines, storage accounts, virtual networks, web apps, databases, and/or database servers. Typically, users will group related resources for an application, divided into groups for production and non-production Environments.

Steps to create resource group

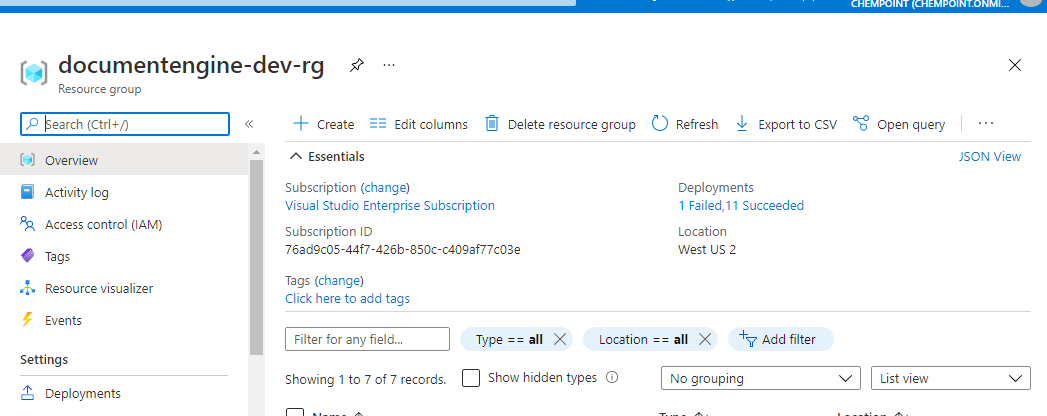
2.1 Go to Azure Services and click on resource group:



2.2 Click on new button to create a new resource group

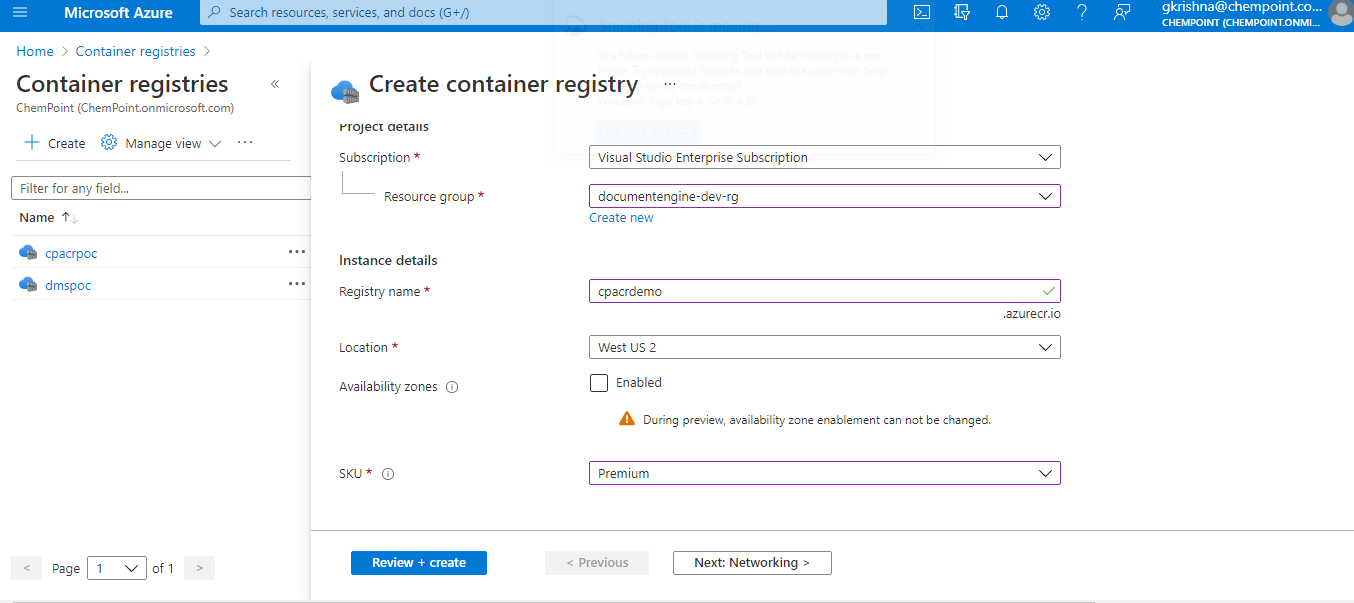


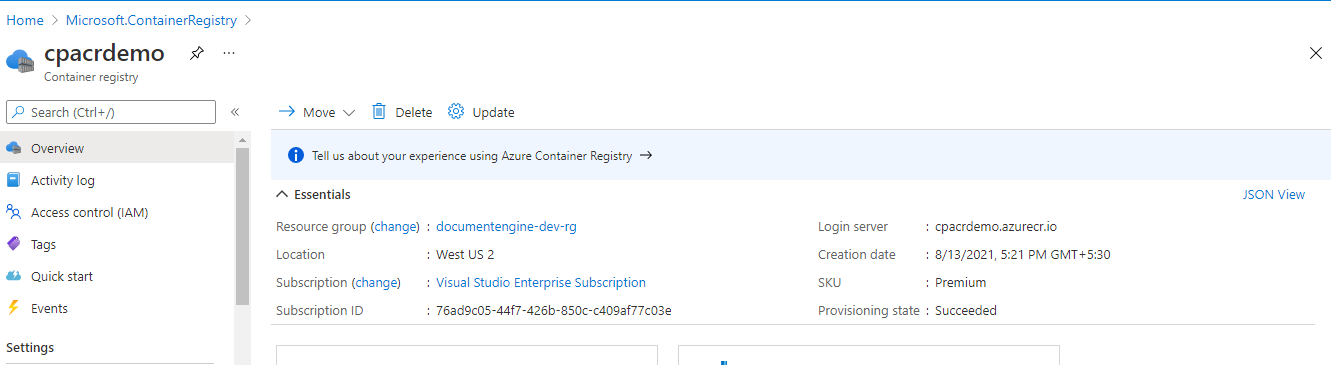
2.3 You can find the below resource group details which got created

.

**3.Azure Container registry**

Go to Azure Services and create ACR

****

****

**4.. AKS**

Azure Kubernetes Service (AKS) offers serverless Kubernetes, an integrated continuous integration and continuous delivery (CI/CD) experience and enterprise-grade security and governance. Deploy web apps, function apps, logic apps, API management and Event Grid on Kubernetes clusters and run anywhere.

Each cluster has one master node connected to one or more worker nodes. The worker nodes are responsible for running groups of containerized applications

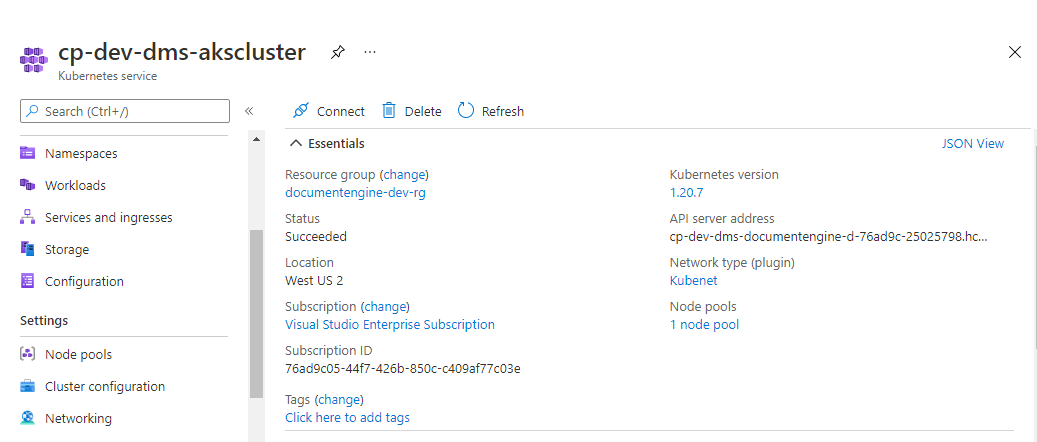
We need to create deployment.yaml and service.yaml to deploy our Applications to Kubernetes Cluster.

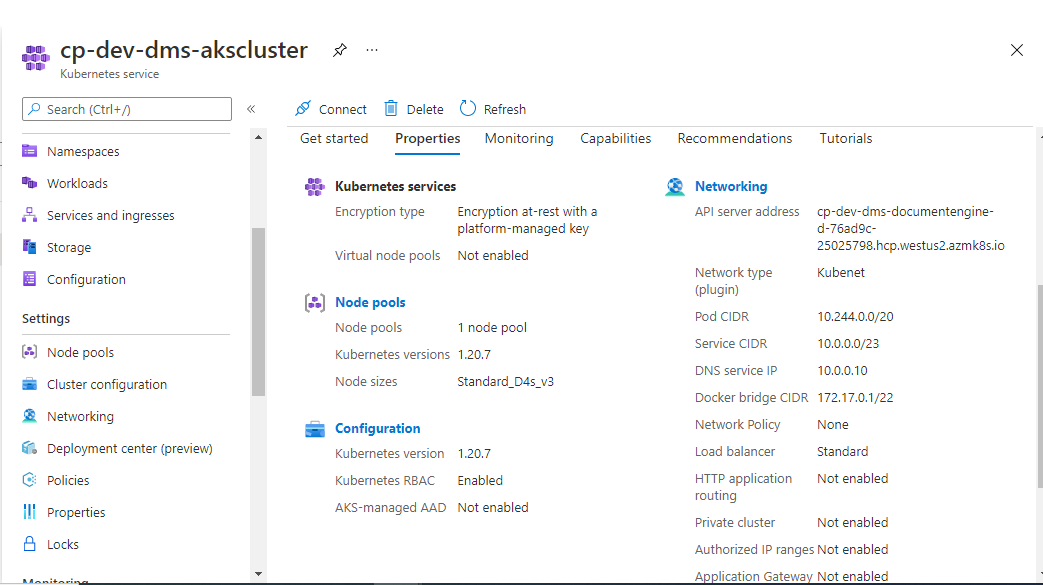


**4.1 go to Azure Services and Create AKS Service.**

**Steps and commands to create AKS is mentioned in attached notepad.**

**4.2 Below are the AKS cluster details after creation.**

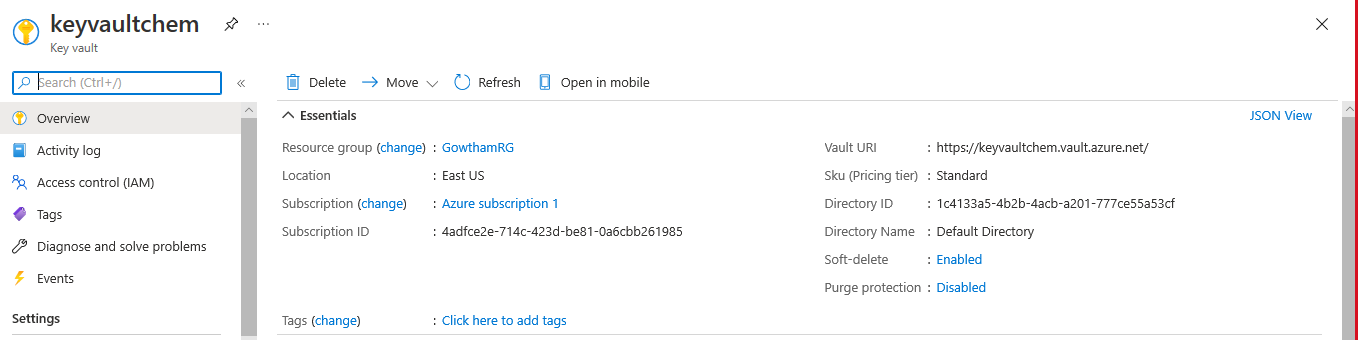




**5. Key Vault**

Azure Key Vault is a cloud service for securely storing and accessing secrets. A secret is anything that you want to tightly control access to, such as API keys, passwords, certificates, or cryptographic.

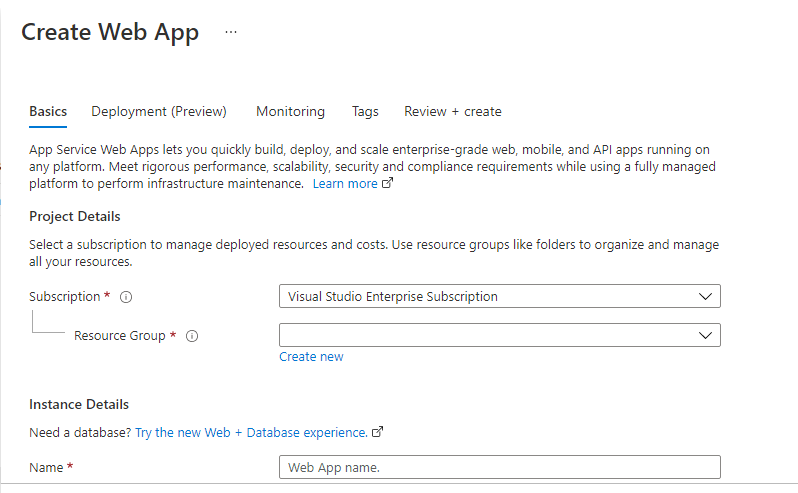
5.1 go to Azure Services and create Key Vault

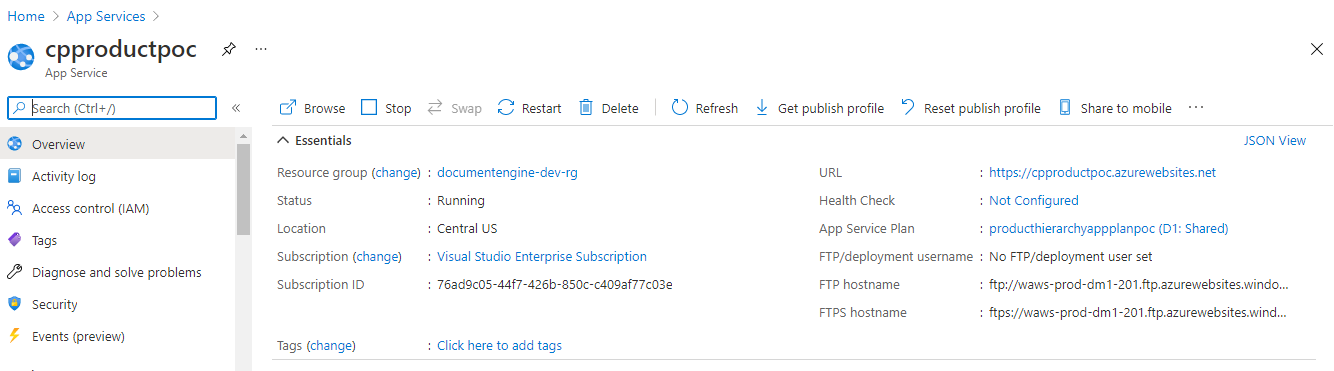


**6. App Service**

Azure App Service is an HTTP-based service for hosting web applications, REST APIs. We can develop in language’s .NET, .NET Core, Java, Ruby, Node.js, PHP, or Python. Applications which are developed will run and scale with ease on both Windows and [Linux](https://docs.microsoft.com/en-us/azure/app-service/overview#app-service-on-linux)-based environments. We also take advantage of its DevOps capabilities, such as continuous deployment from Azure DevOps, GitHub, Docker Hub, and other sources.

* 1. **go to Azure Services to create App Services**



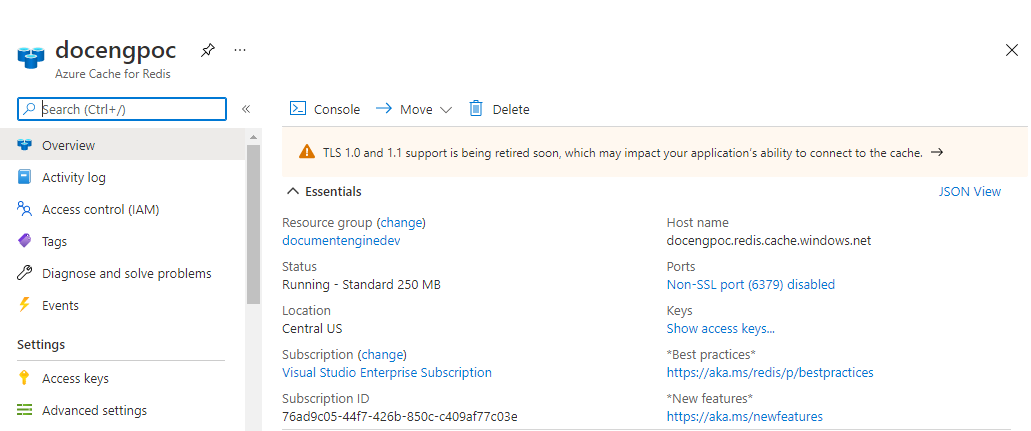


**7.Redis**

Azure Cache for Redis is a fully managed, in-memory cache that enables high-performance and scalable architectures. use it to create cloud or hybrid deployments that handle millions of requests per second at sub-millisecond latency all with the configuration, security and availability benefits of a managed service.

We need to add a quick caching layer to the application architecture to handle thousands of simultaneous users with near-instant speed with the benefits of a fully managed service.

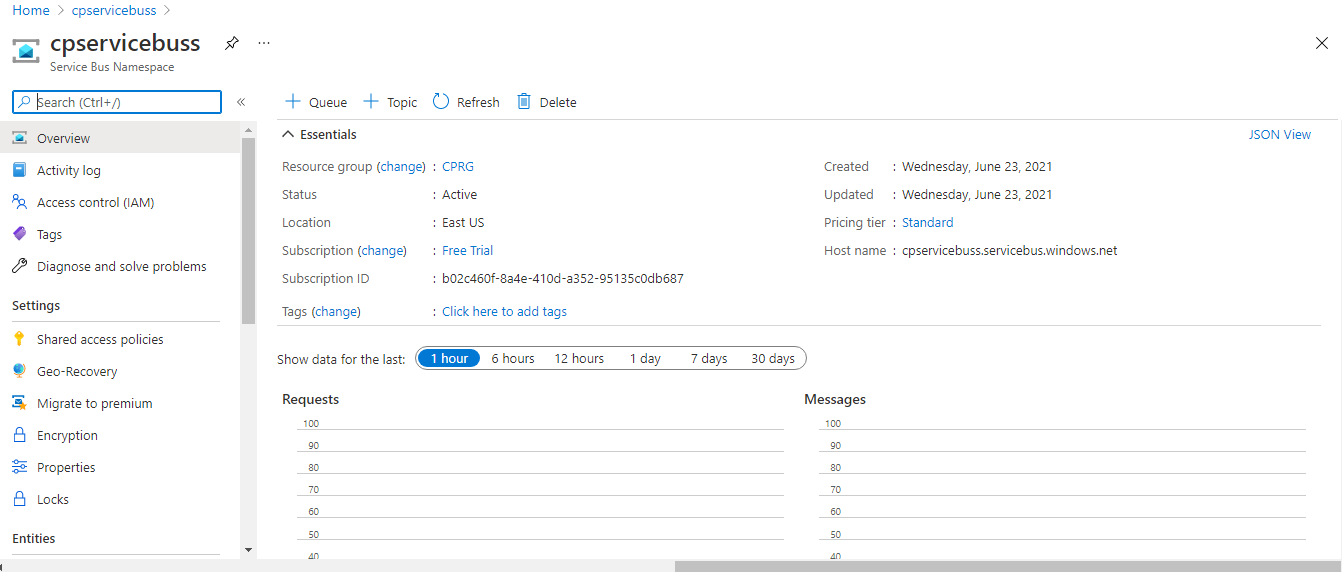
**go to Azure Services to create Redis**

****

**8. Azure Service Bus**

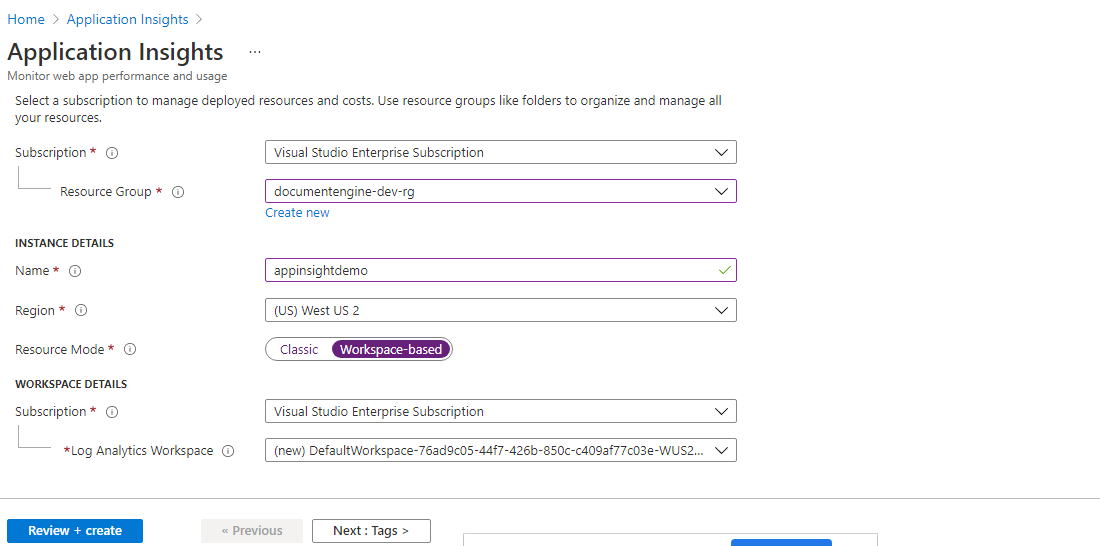
Microsoft Azure Service Bus is a fully managed enterprise message broker with message queues and publish-subscribe topics.

**go to Azure Services to create Azure Service bus**

****

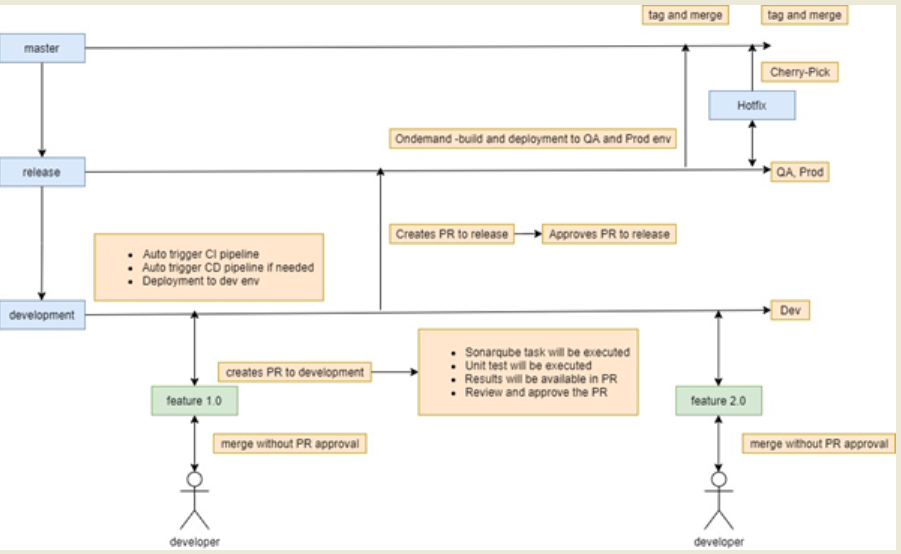
**9. Azure Monitor**

**Go to Application Insights and create as shown below.**

****

**10.Branching Strategy**

Master branch is a stable one which is of production environment. The develop branch is an intermediate one which gets created from master. When the development is over, release branch will be created from this branch. Hotfix branch is created out of master when patch release needs to be pushed.



### 10.1 Build Process

The DNS Front end components are be compiled and build artifacts are pushed to centralized Azure Artifacts repository.

### 10.2 Quality Assurance

There are plenty of test cases are executed in each stage. The testing team starts writing the test cases in parallel with development activity. Therefore, the test steps are aligned with the requirement most of the time and test cases are approved internally before executing in any environment.

### 10.3 Promotion Pipeline

In promotion Pipeline will get Approval’s from Approvers to deploy to various Environments like QA/ Stage & Prod.

### 10.4 Release Process

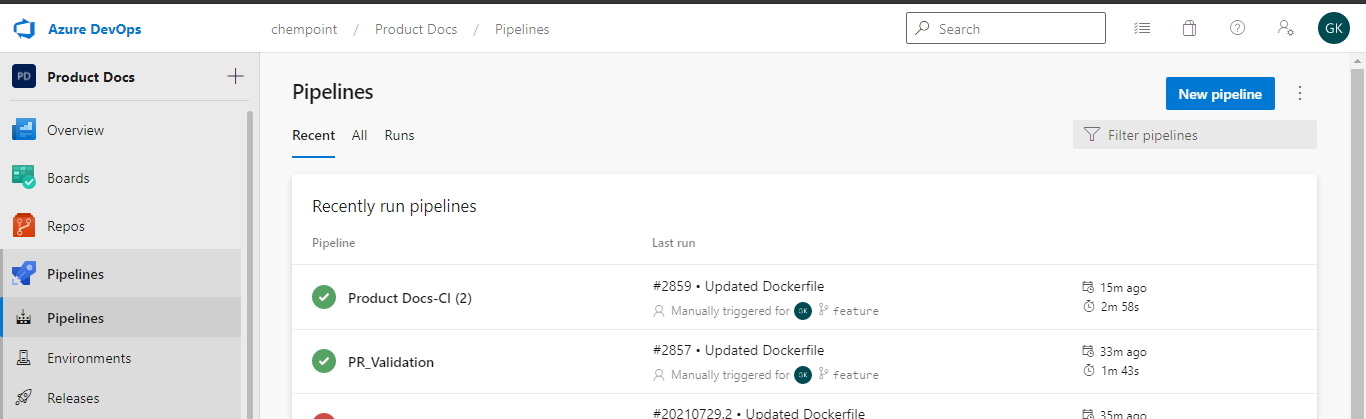
A developer creates new branch in the components specific repository. Ideally, the new branch will be created from master; otherwise, the latest release branch. Once the new branch is created, the developer will clone the repository into their local development environment and start pushing custom changes. When the development is complete, the Pull Request will be created from current working branch to destination branch (e.g. master). Eventually, all the members who are working in CI/CD will be added as a reviewer and based on no conflicts; the PR will be merged to the destination branch.

**11. Azure CI Pipeline**

**11.1 Azure CI Pipeline Steps -DMS Download Service:**

**Step1:**

Goto Azure Devops 🡪 Create New Pipeline 🡪

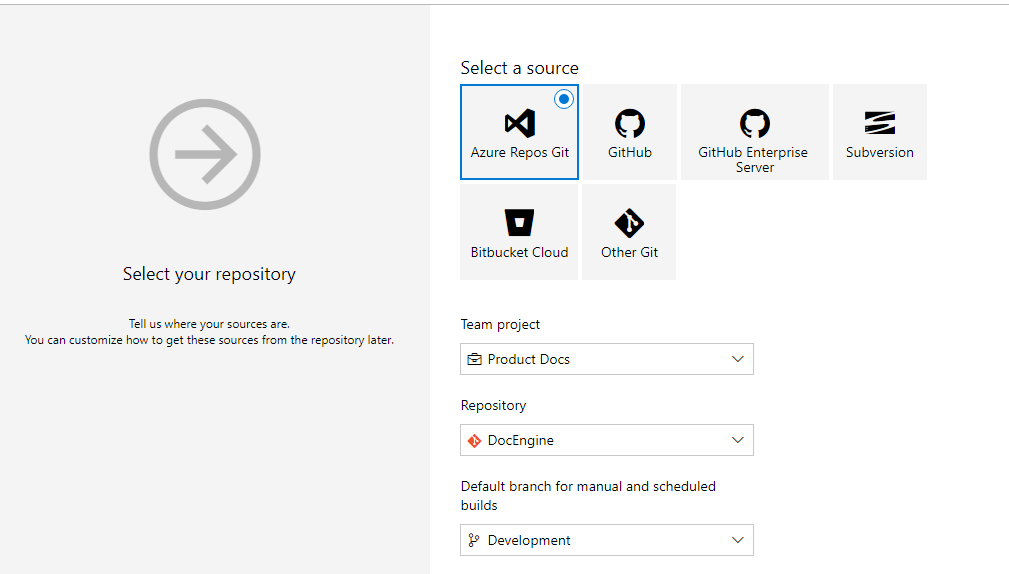


**Step2:**

Select and click on [Use the classic editor](https://dev.azure.com/chempoint/Product%20Docs/_apps/hub/ms.vss-ciworkflow.build-ci-hub?_a=build-definition-getting-started&id=0) to create a pipeline

**Step3:**

Select a Source 🡪 Azure Repo Git

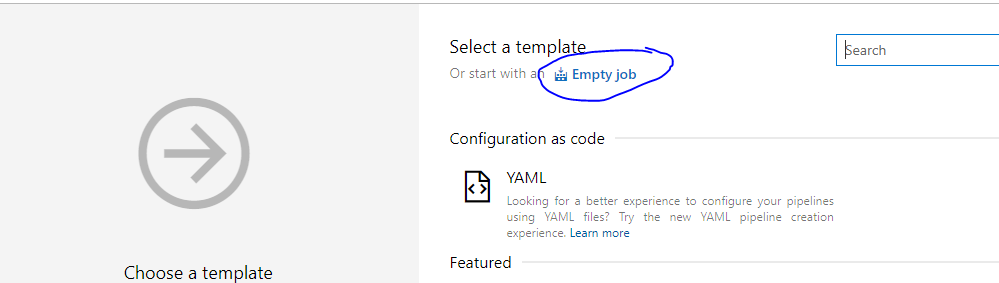


Select The Repository from the drop down.

Click On the Button Continue.

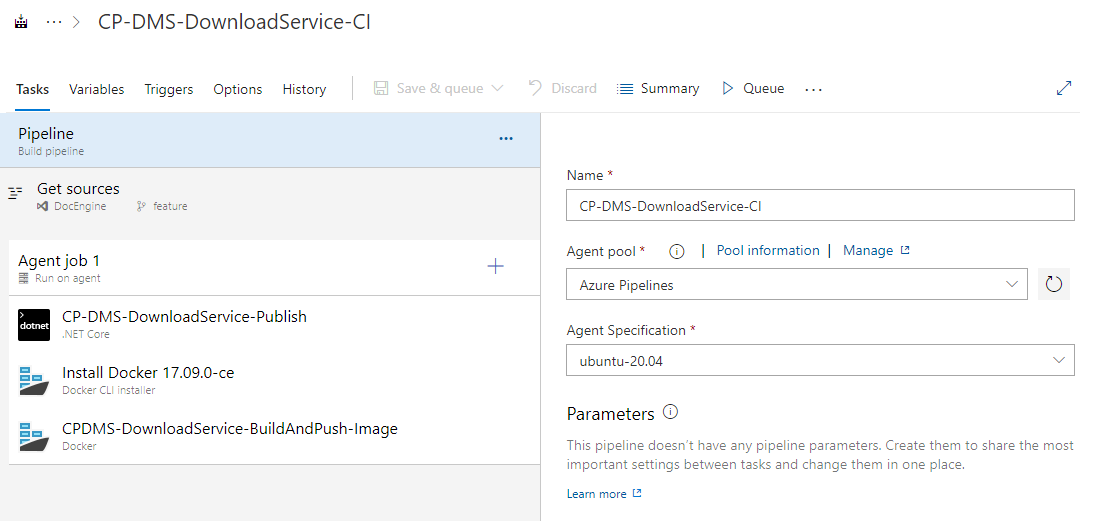
**Step4:**

Select Empty job



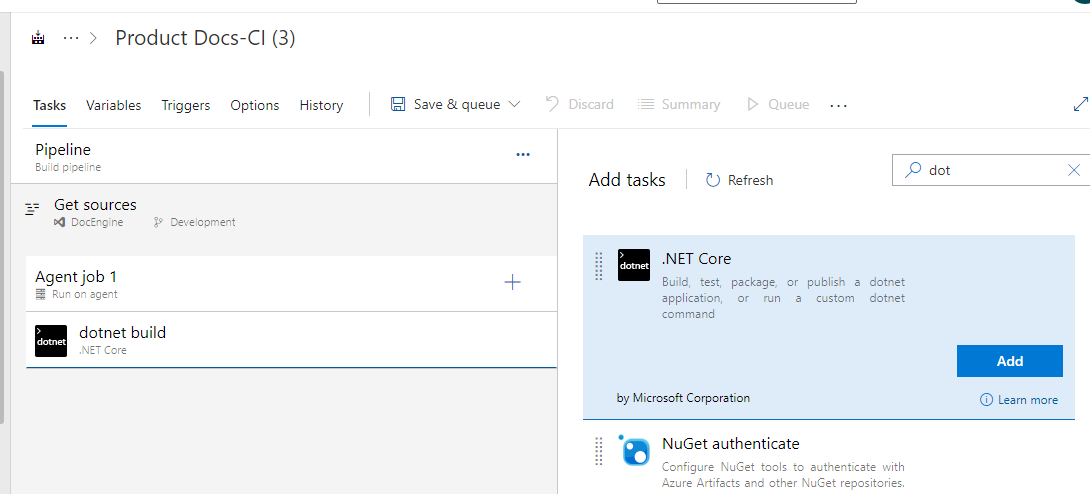
**Step5:**

Create Pipeline for DMS Download Service with Ubuntu Agent pool



**Step6:**

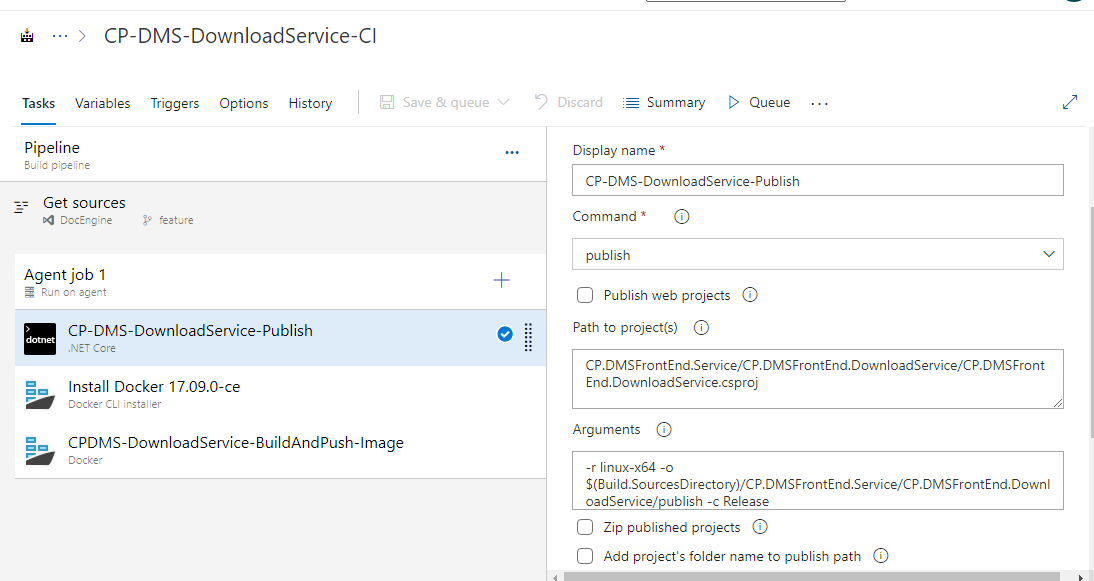
Select The AgentJob 🡪 Click On + Icon (add a task to agentjob) 🡪 Add Dotnet Core



Select from dropdown Command name as publish

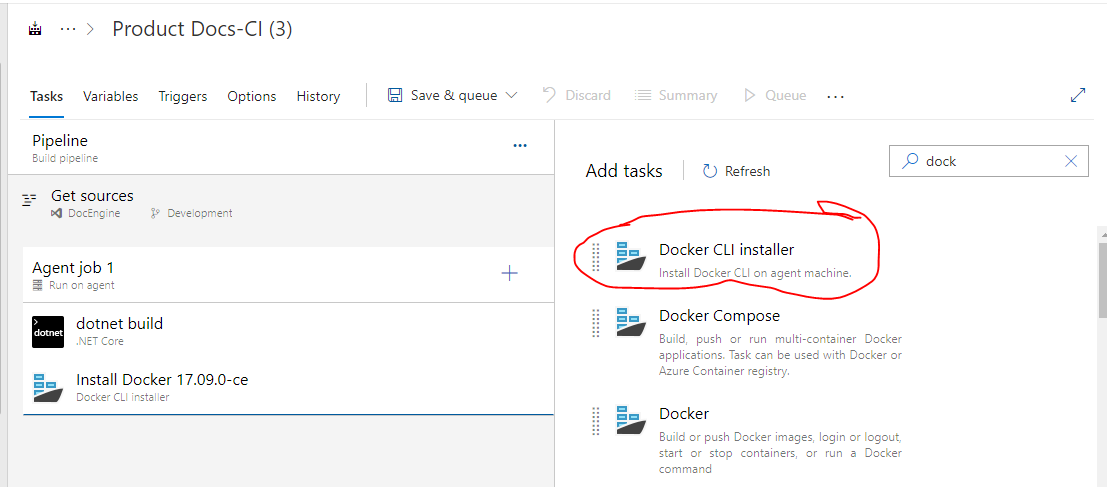
Provide Path to project details as show in below

Provide Arguments details as show in below



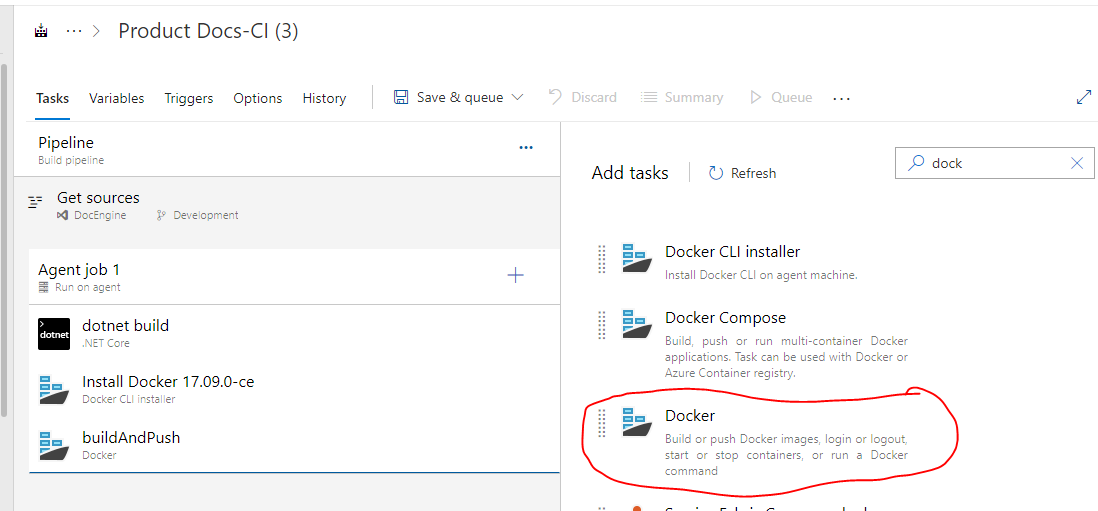
**Step7:**

Select The AgentJob 🡪 Click On + Icon (add a task to agentjob) 🡪 Add Docker CLI Installer

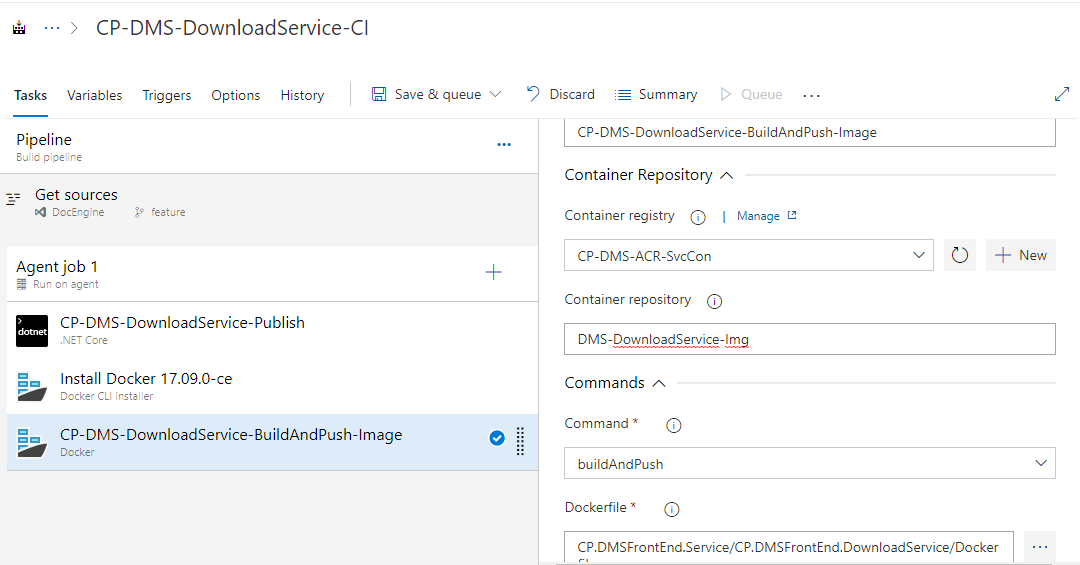


**Step8:**

Select The AgentJob 🡪 Click On + Icon (add a task to agentjob) 🡪 Add Docker



Please provide Container registry And Container Repository And Docker File Path as shown below.



**Step9: Create a Docker File for download service.**

FROM mcr.microsoft.com/dotnet/sdk:5.0 AS build

WORKDIR /src

COPY CP.DMSFrontEnd.DownloadService.csproj .

RUN dotnet restore

COPY . .

RUN dotnet publish -c release -o /app

FROM mcr.microsoft.com/dotnet/aspnet:5.0

WORKDIR /app

COPY --from=build /app .

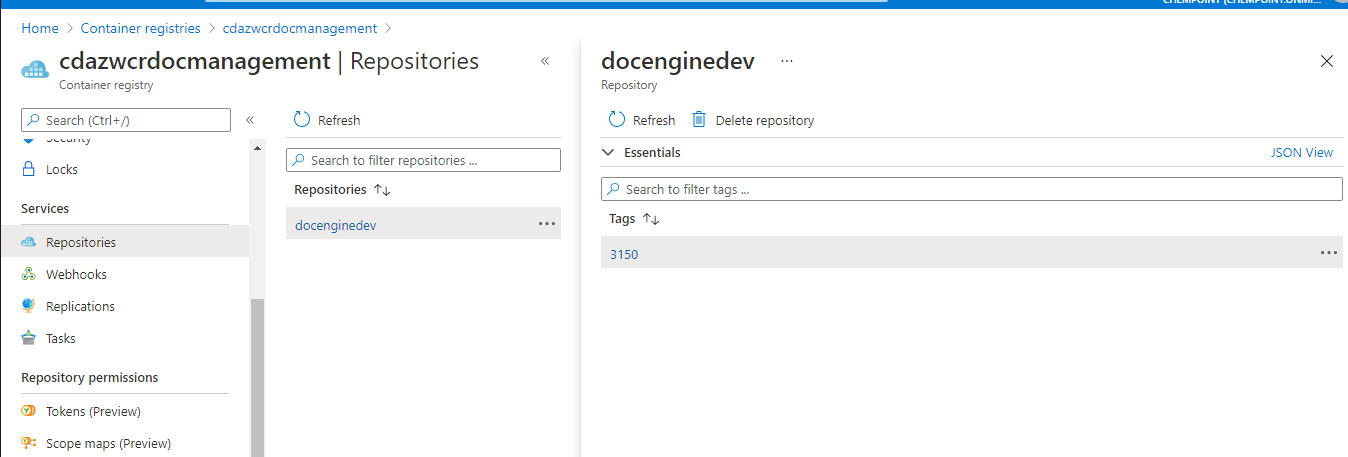
ENTRYPOINT ["dotnet", "CP.DMSFrontEnd.DownloadService.dll"]

**Step 10:**

Select Save&Queue Button to run the job.

**Step 11:**

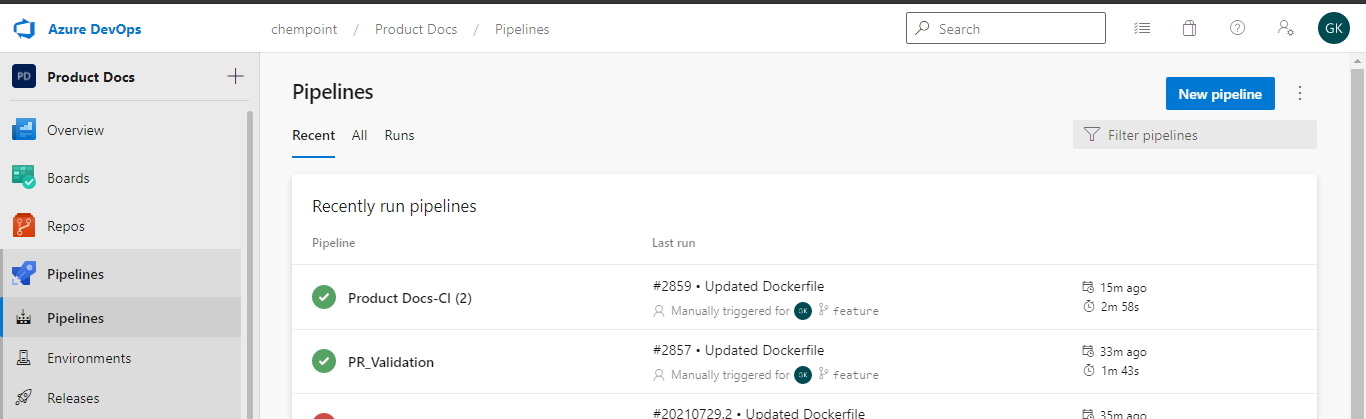
Verify the docker image is pushed to ACR as shown below.



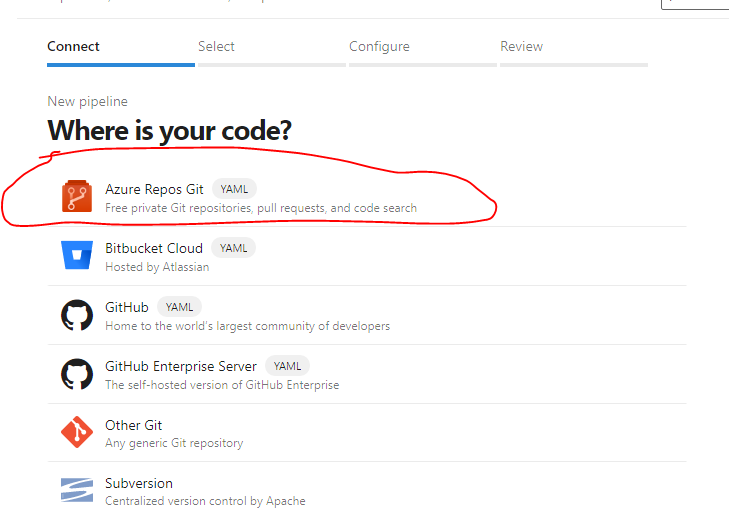
**11.2 Azure CI Pipeline Steps –Ai&ML Service:**

**Step1:**

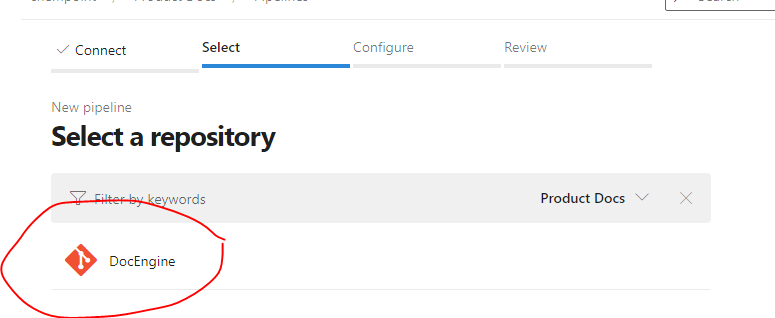
Goto Azure Devops --> Create New Pipeline -->



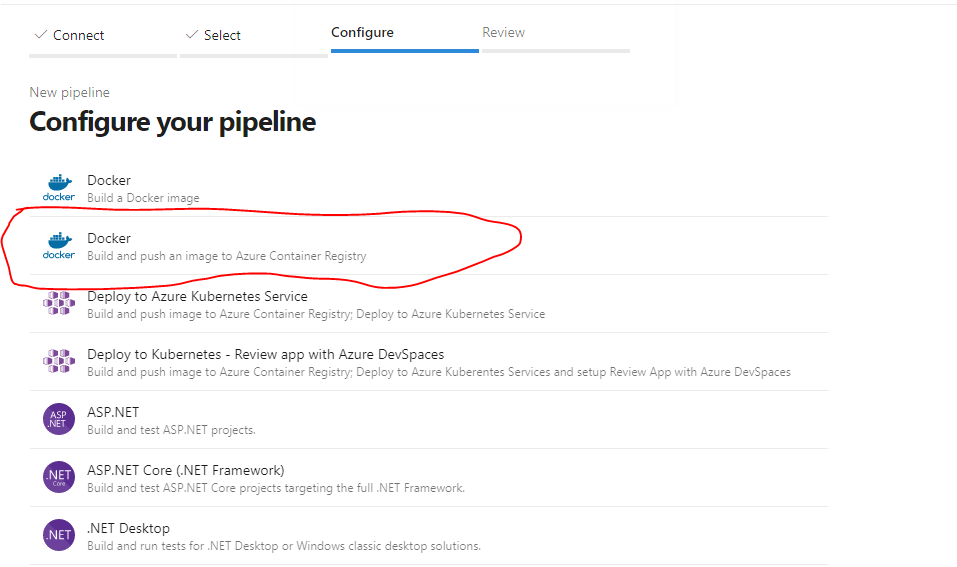
**Step2**: Select the Azure Repos Git



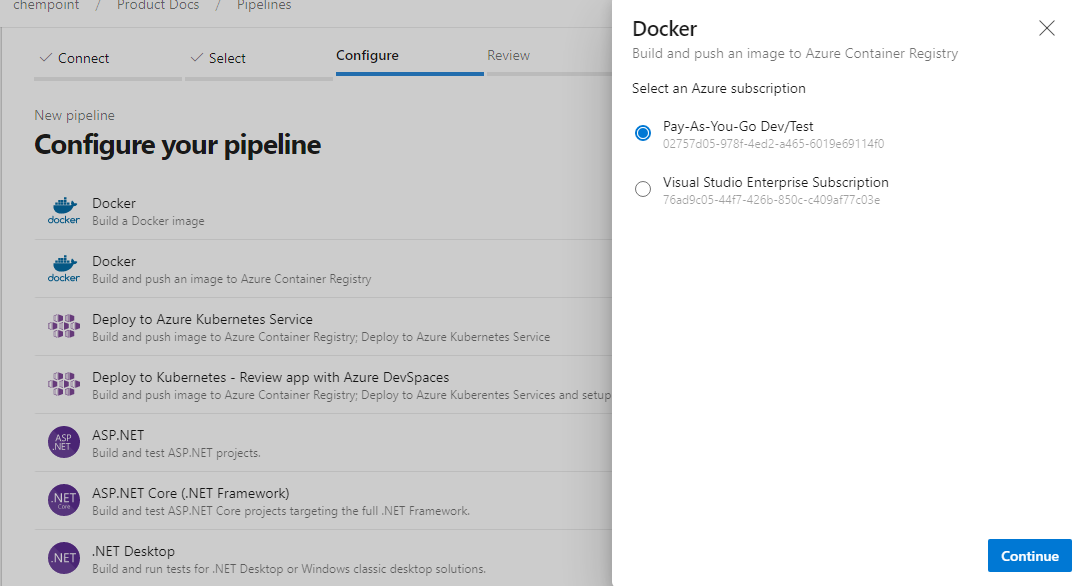
**Step3:** Select a repository



**Step4:** Configure your pipeline

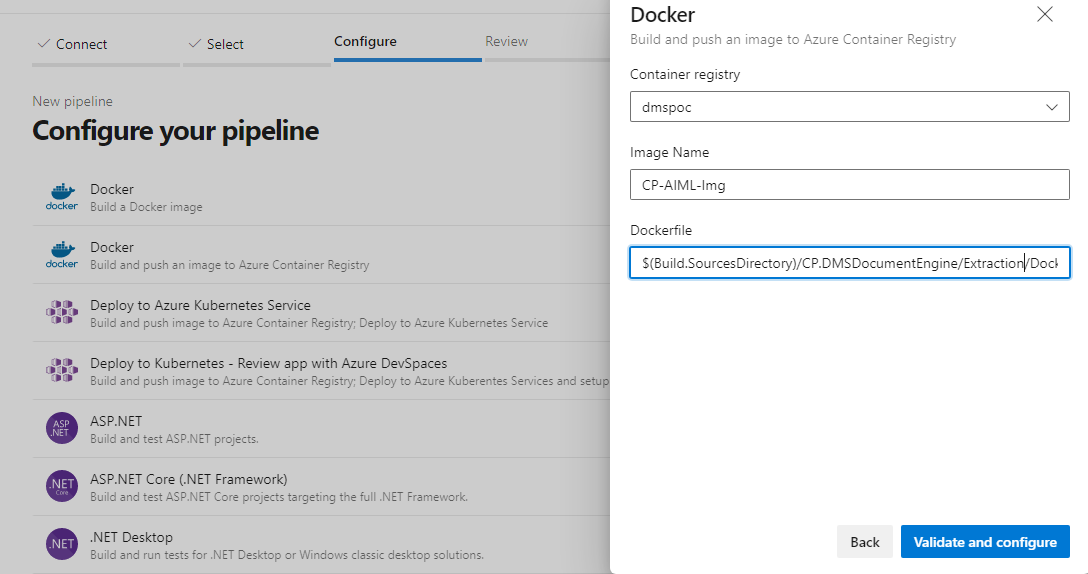


**Step5**: Click on Docker Docker and select Pay As You Go Dev/Test Subscription



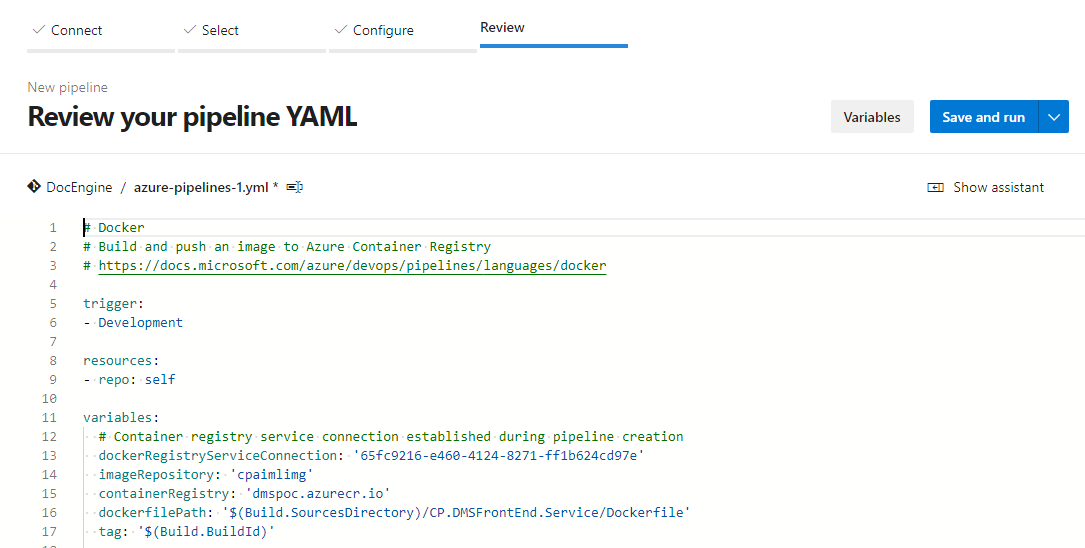
Click on Continue

**Step6**: Enter the details Container Registry Name, Image name Dockerfile

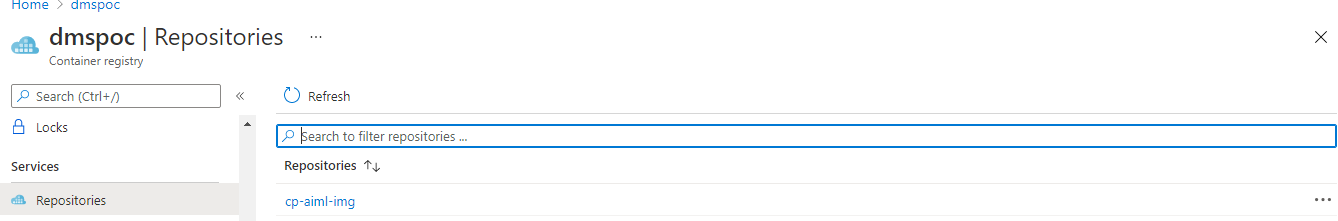


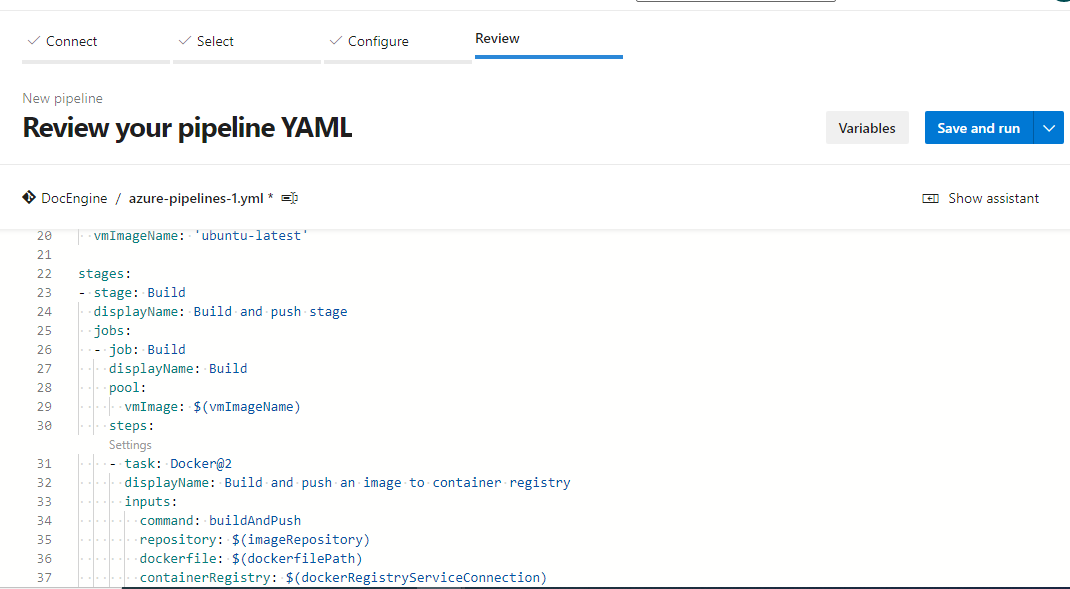
Click on Validate and Configure

**Step7**:It will configure resources and generating yaml file as shown below

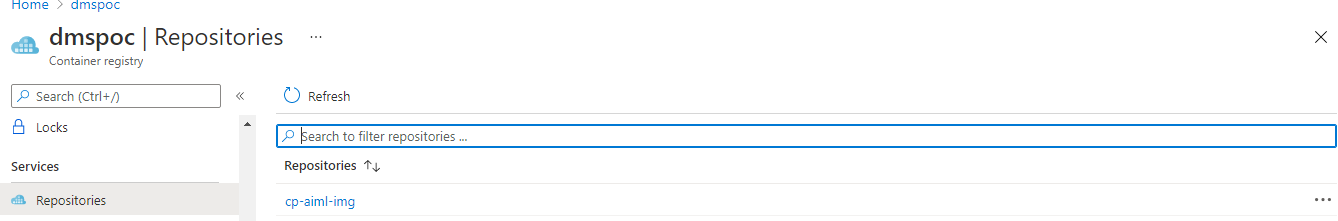
**Step 8:**





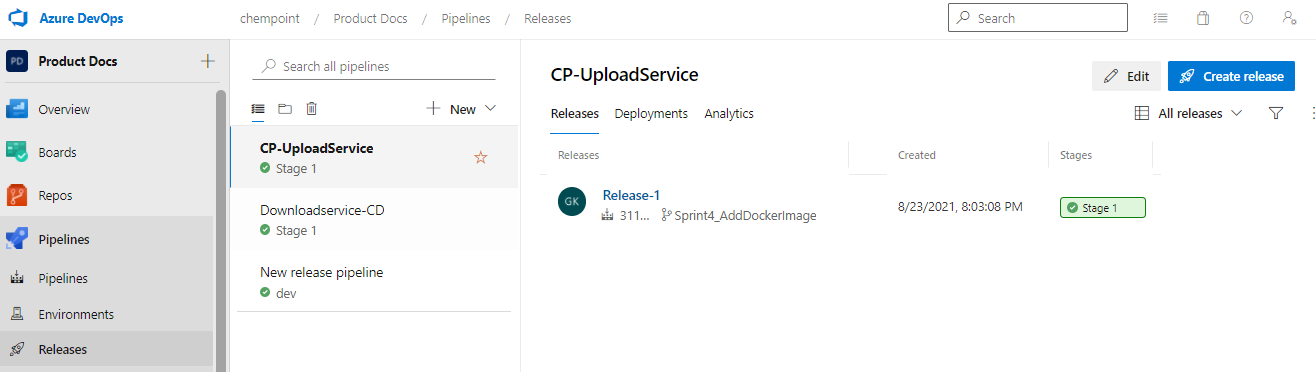
**Step 9**: Click on save and run

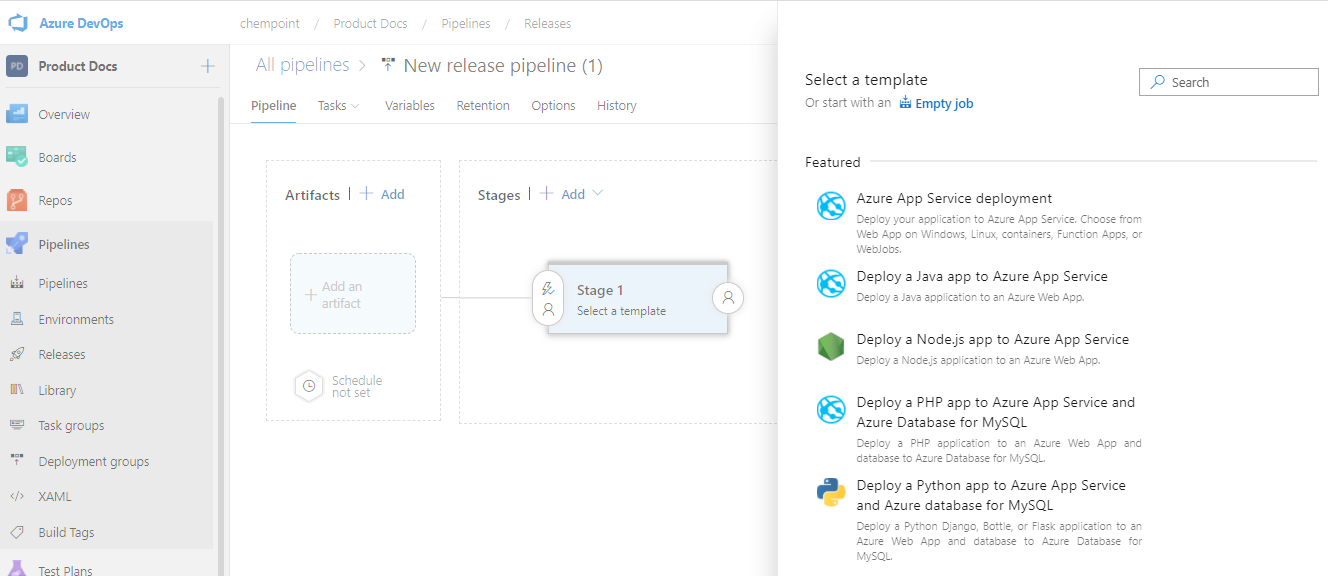
**Step 10**: Verify the docker image is pushed to ACR as shown below.



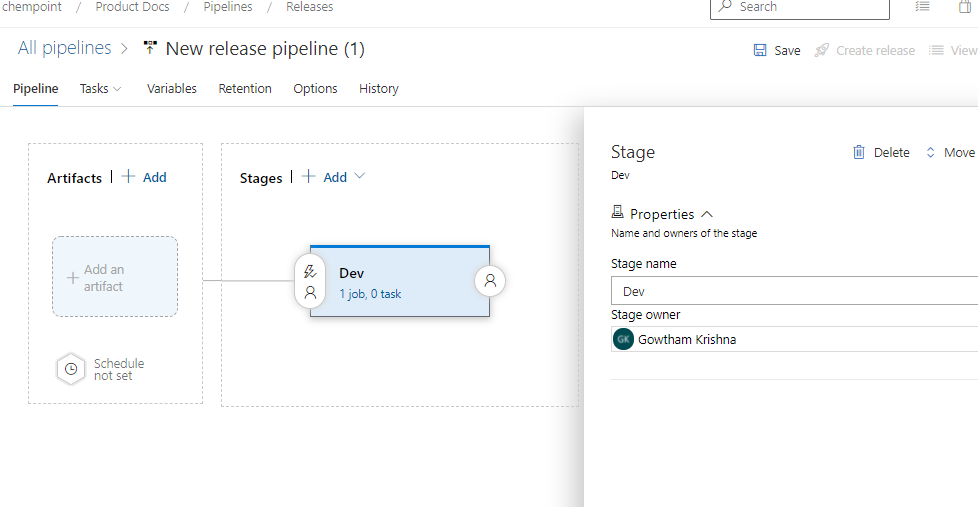
**11.3 Azure CD Pipeline Steps –Ai&ML Service:**

Create a Service Principal before create the CD Pipleline.

1. Go to Azure DevOps Click on The Releases
2. Click New and select New release Pipeline from drop down box

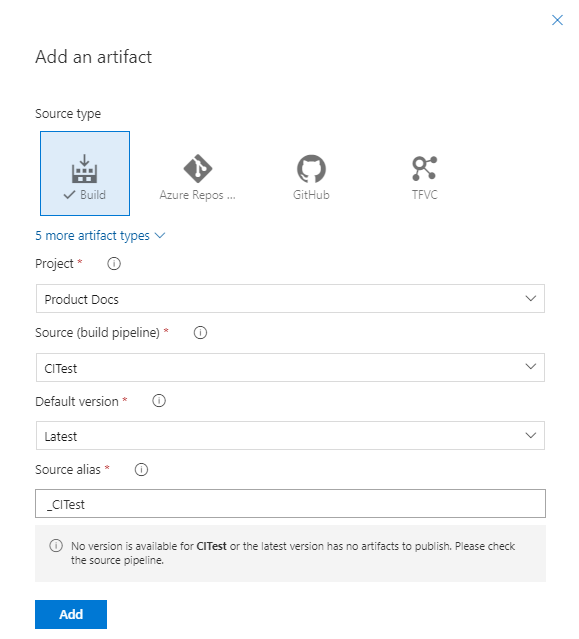


3) Click on Empty Job



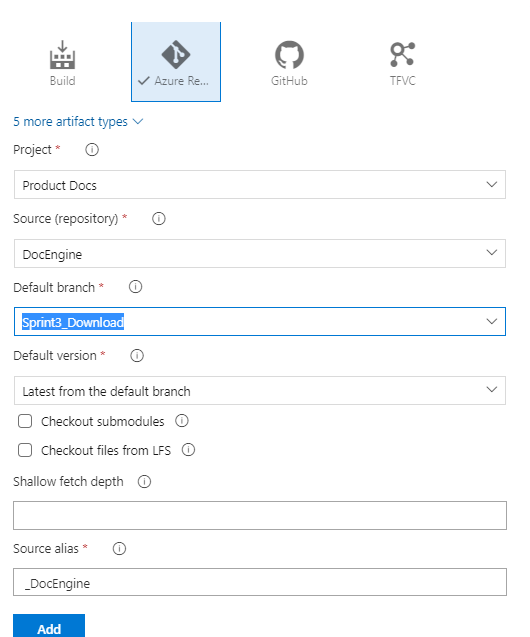
Click on Add on artifact

4)



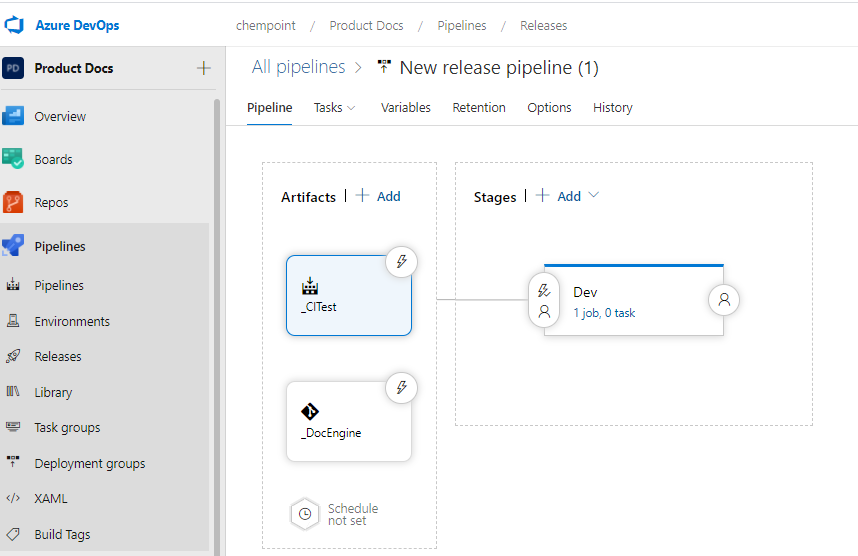
Click on Add button

5) Click on add Artifacts| 🡪 Add and selct Azure Repos



Click on Add

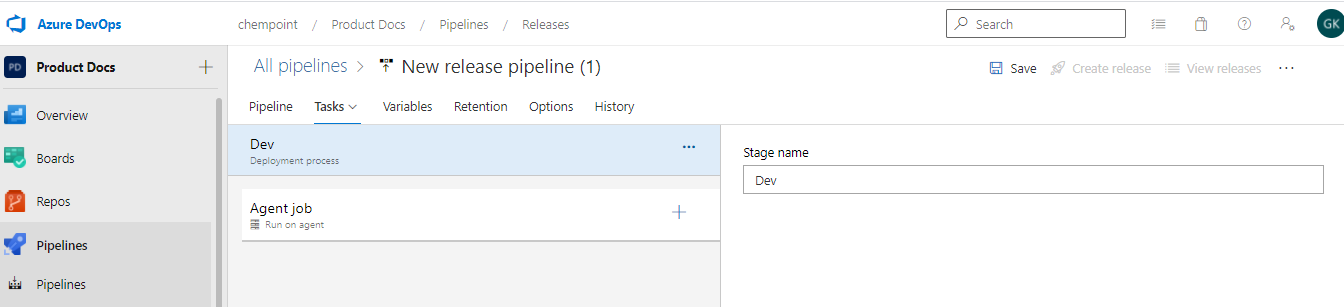
6



6) Go to stages|+ Add

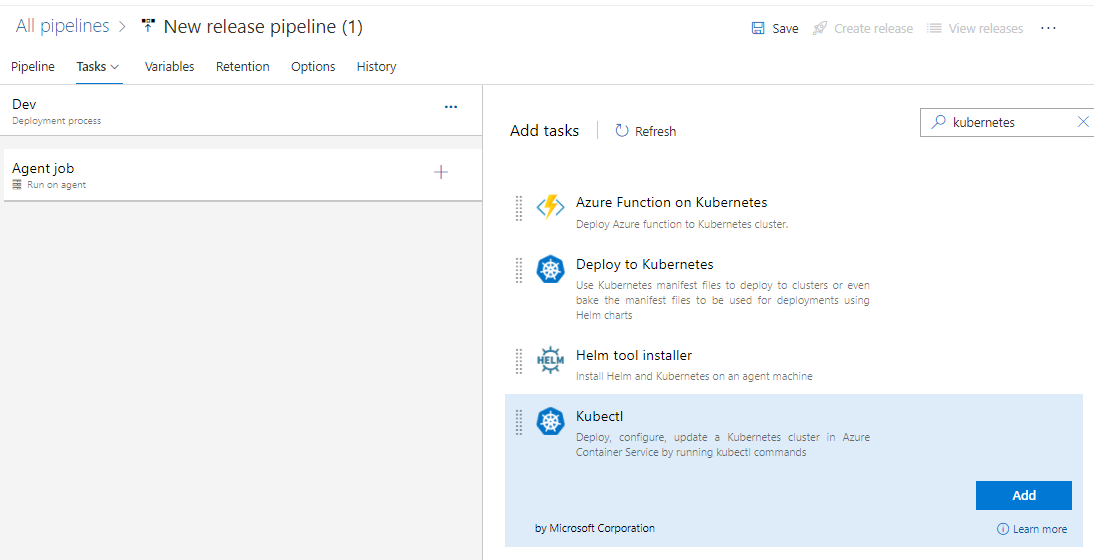
Click on Dev Stages(Job)

7)

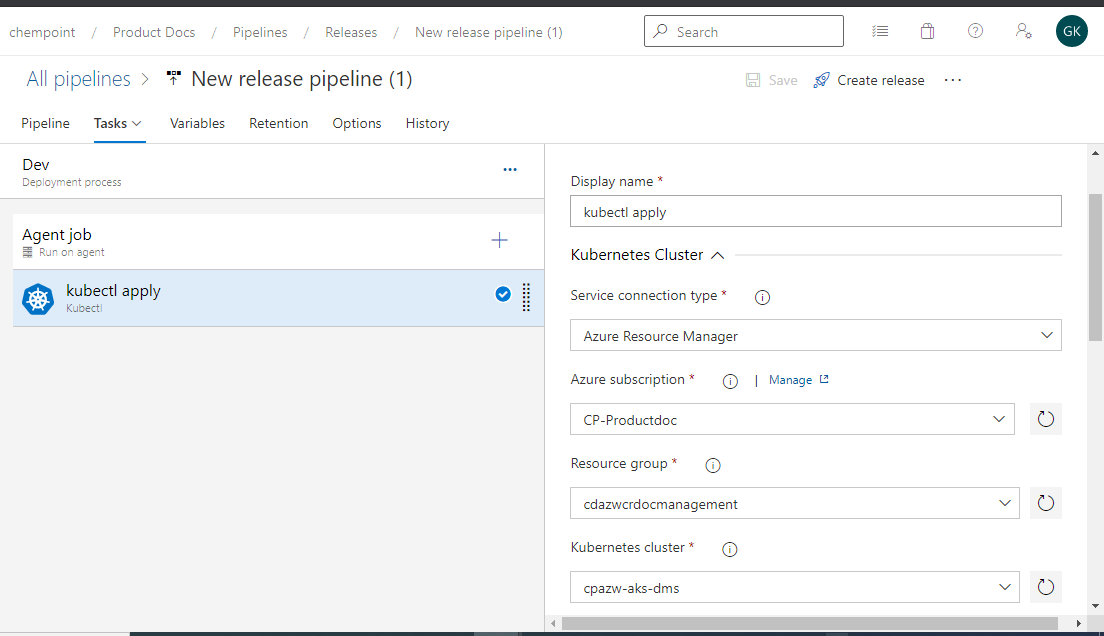


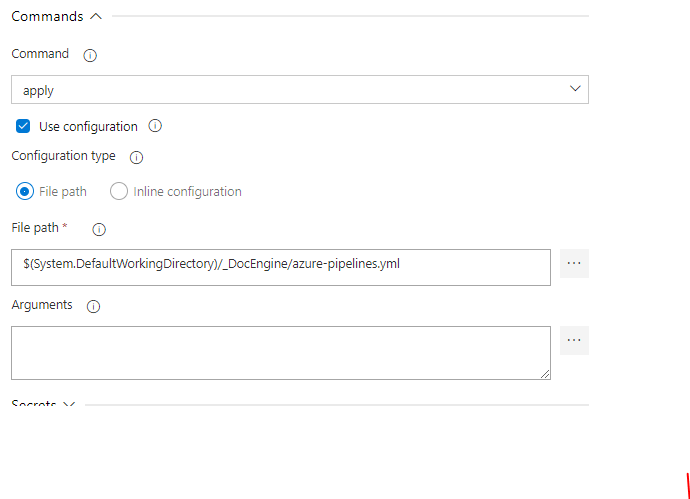
Click on Agent job on +symbol

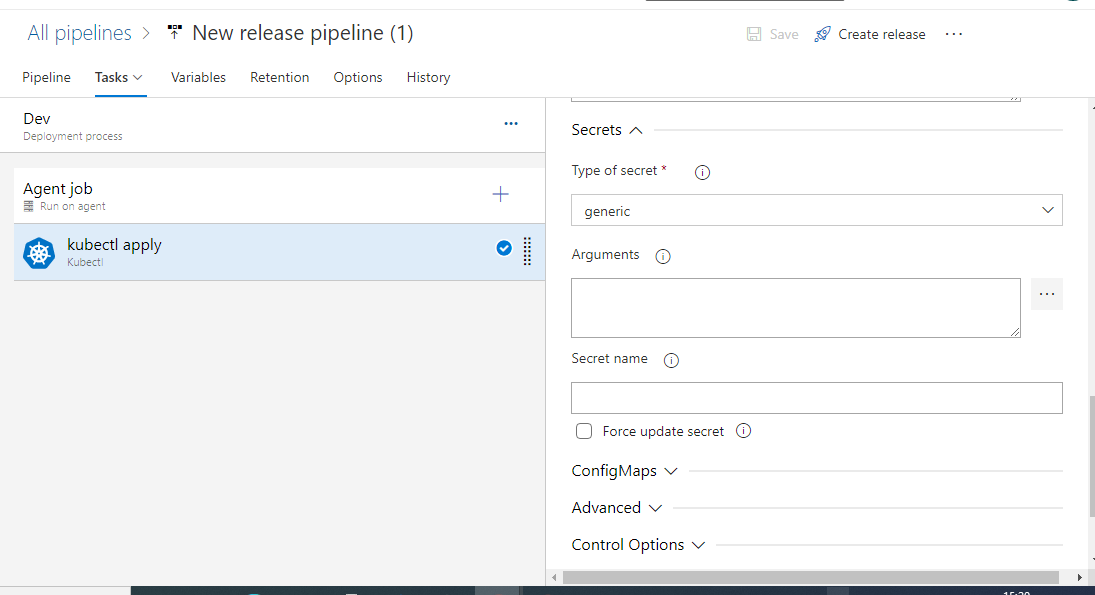
8)



Click on Add button







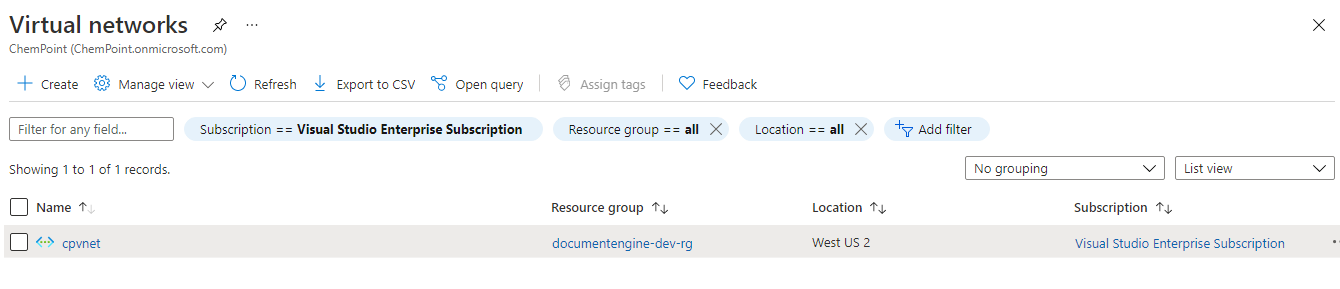
Click save &) Create release

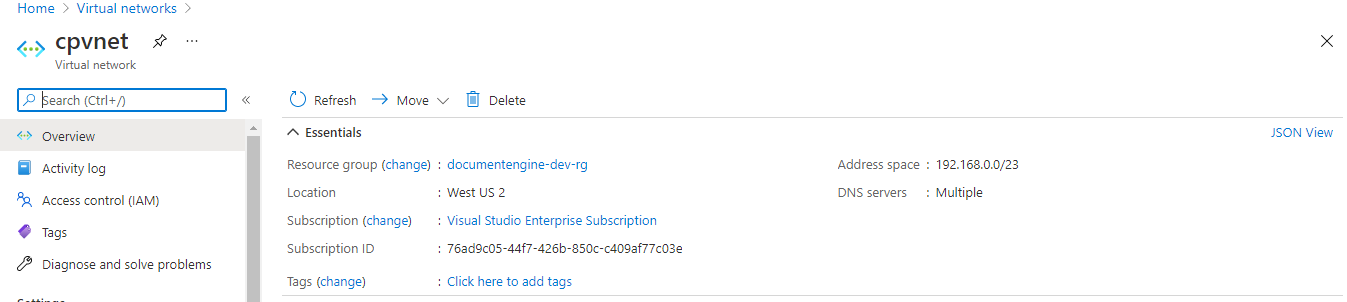
**Appendix**

VNET

An Azure Virtual Network (VNet) is a representation of your own network in the cloud. It is a logical isolation of the Azure cloud dedicated to your subscription. We can use VNets to provision and manage virtual private networks (VPNs) in Azure and, optionally, link the VNets with other VNets in Azure, or with your on-premises IT infrastructure to create hybrid or cross-premises solutions. Azure virtual network enables Azure resources to securely communicate with each other, the internet, and on-premises networks.

5.1 Go to Azure Services and create VNET



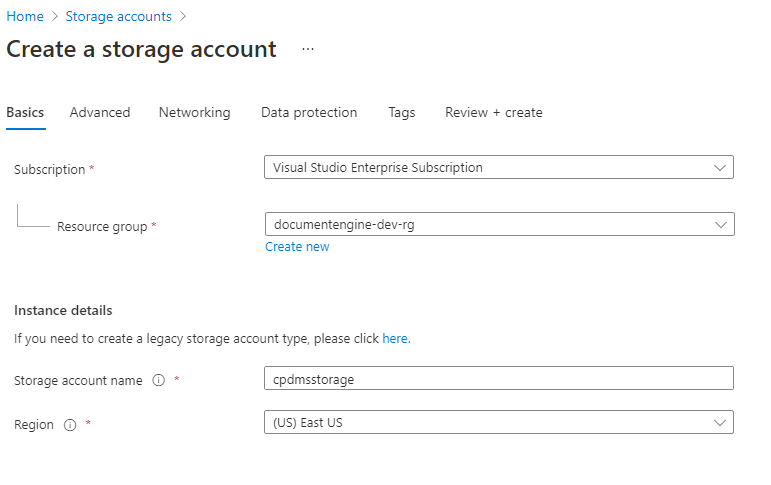


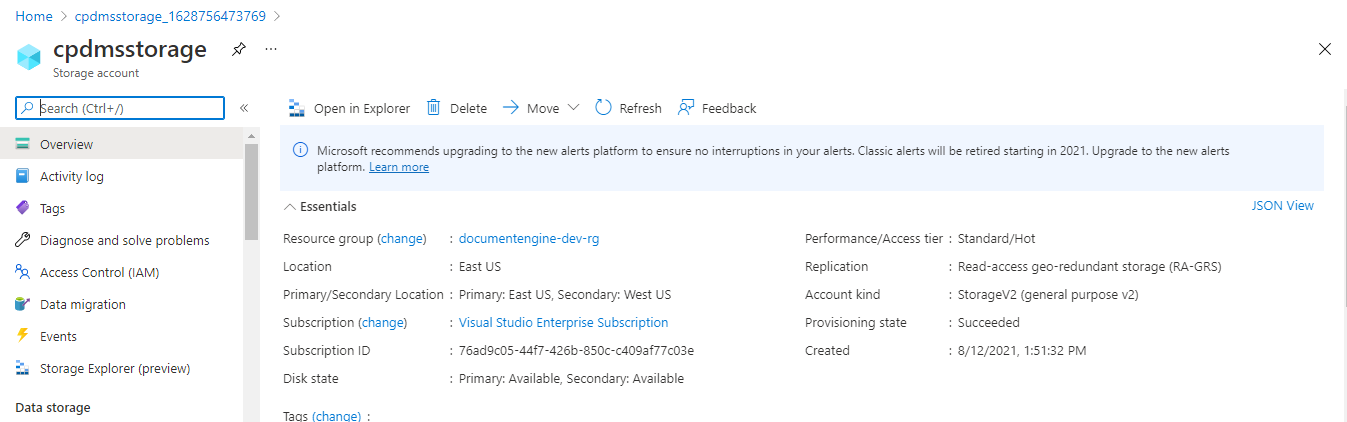
**Storage Account(cpstdev)**

An Azure storage account contains all Azure Storage data objects: blobs, file shares, queues, tables, and disks. The storage account provides a unique namespace for Azure Storage data that is accessible from anywhere over HTTP or HTTPS.

Steps to create Storage Account

4.1 Go to Azure Services and Create the storage Account by clicking on New



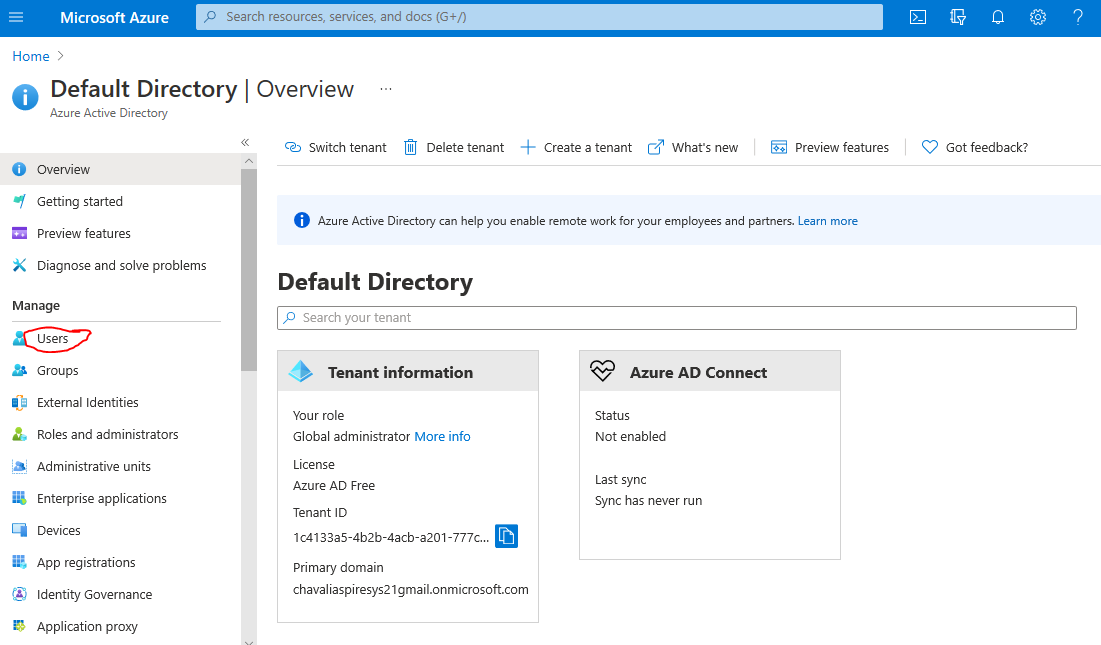


**9. Azure Active Directory**

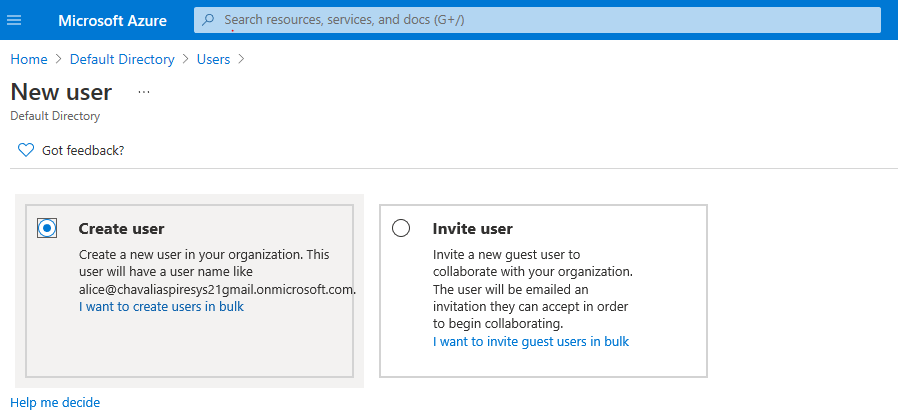
Azure Active Directory (Azure AD) is Microsoft's enterprise cloud-based identity and access management (IAM) solution.

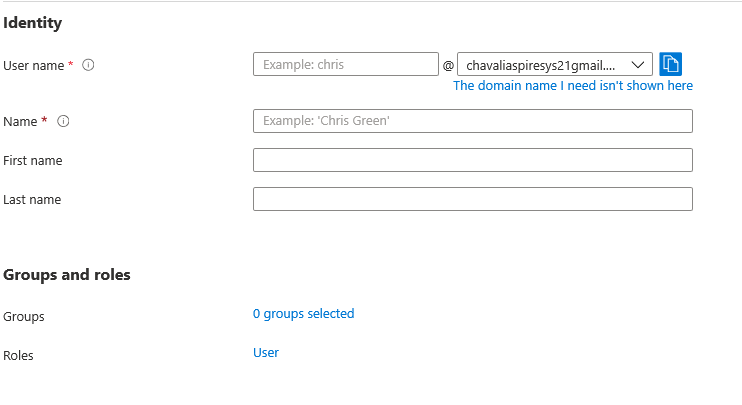
Steps to create AAD

**10.1 Create users in Azure Active Directory(AAD)**



**10.2 Click on the users in left side navigation bar**

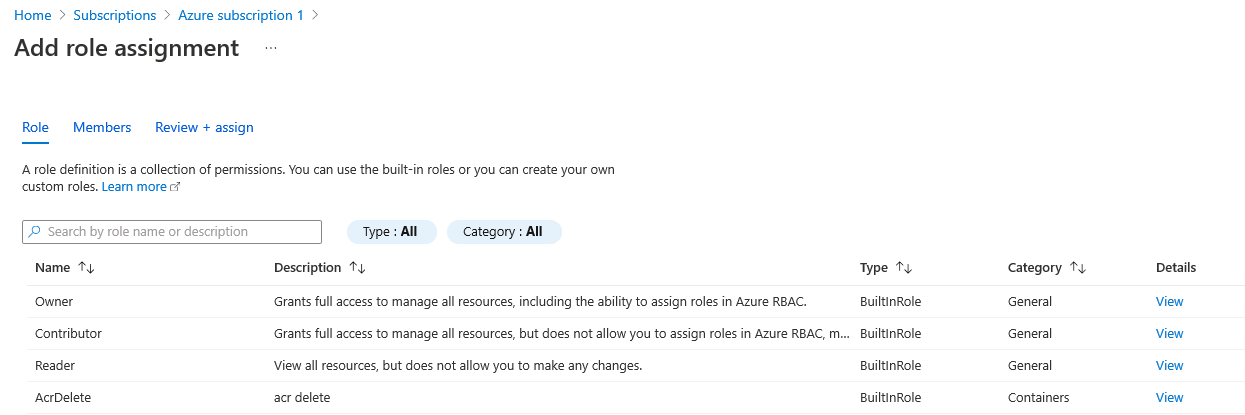




Click on create button after giving user details.

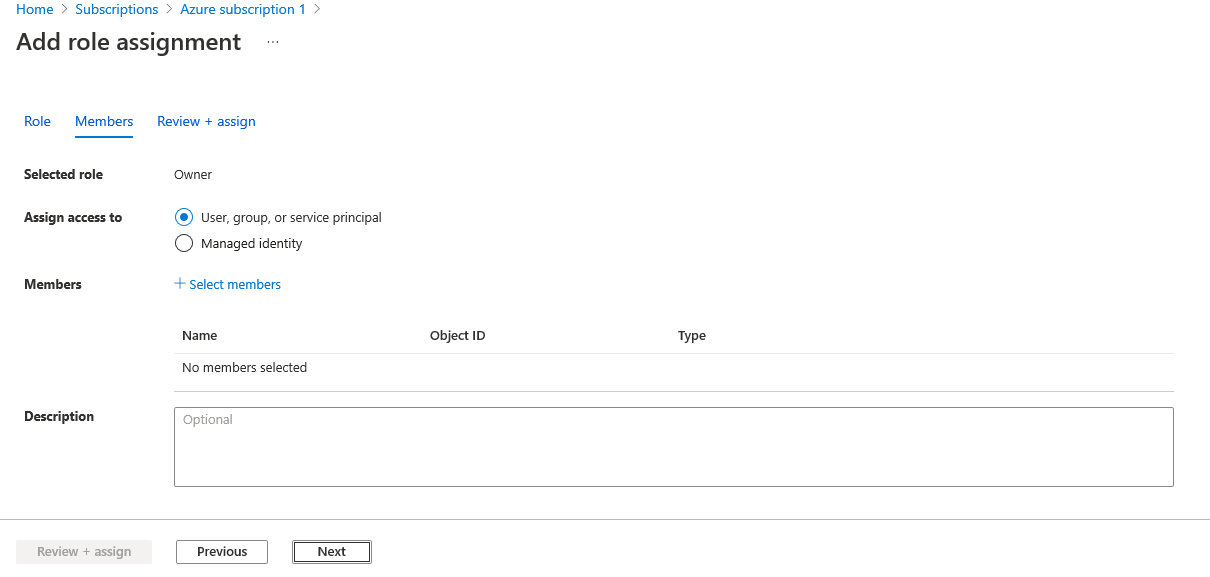
**10.3 Assign role to users**

Home🡪 Subsriptions🡪IAM🡪 Click on Add role assignment

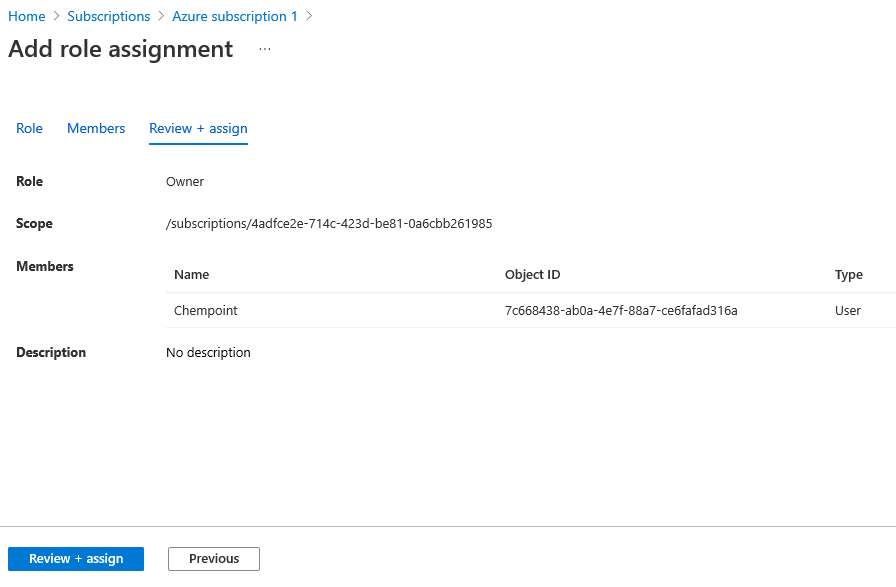


**10.4 Select the role for example owner and click on next button**

**10.5 click on select members**



**10.6 select user/member to assign owner role**



**10.7 Click on review+assign button.**