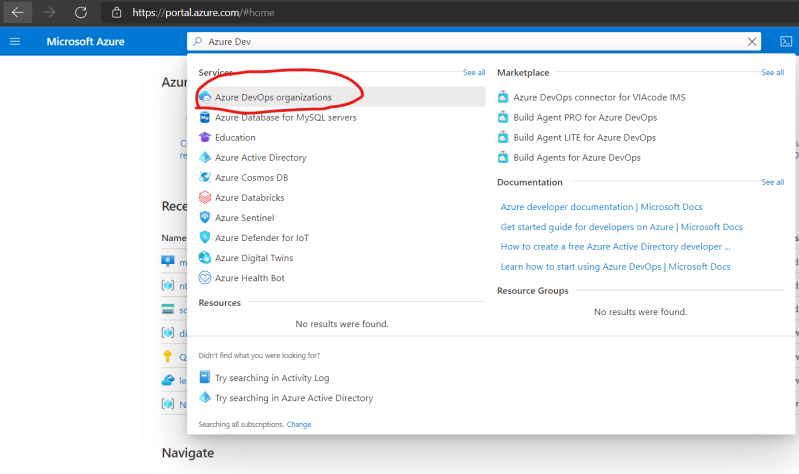
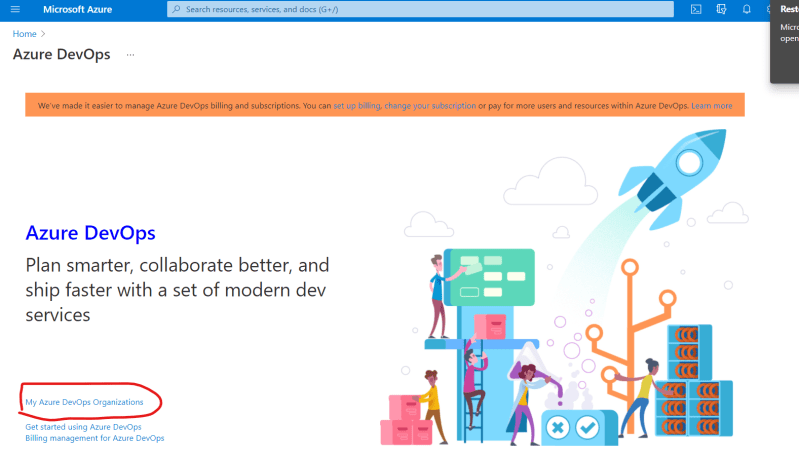
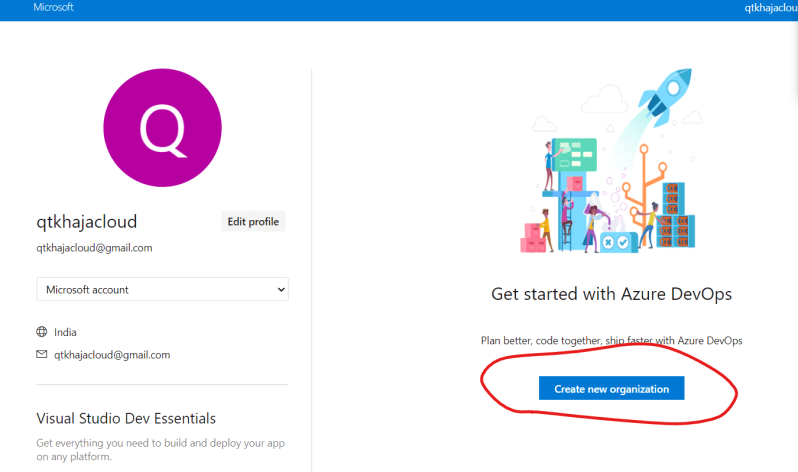
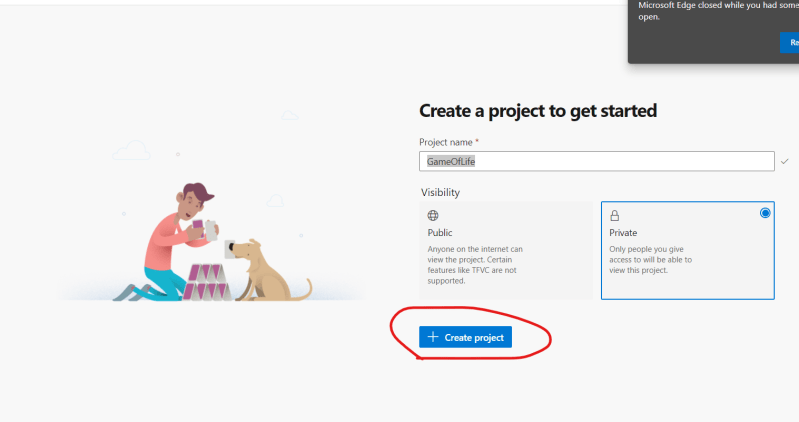
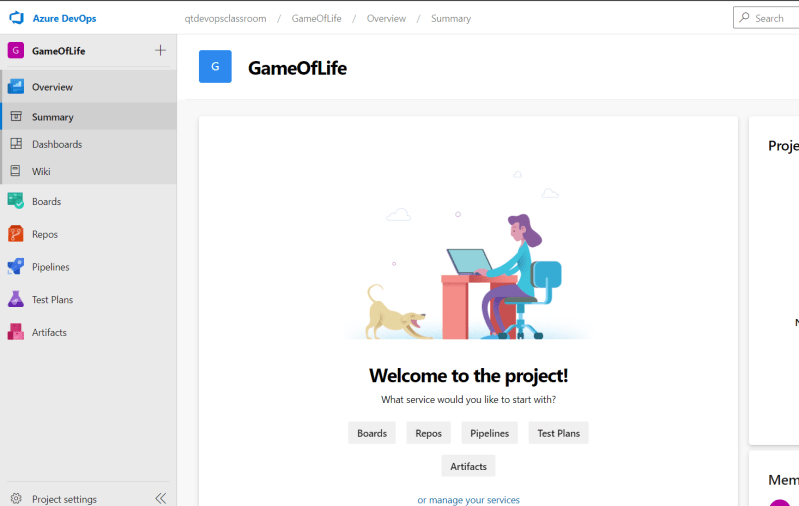
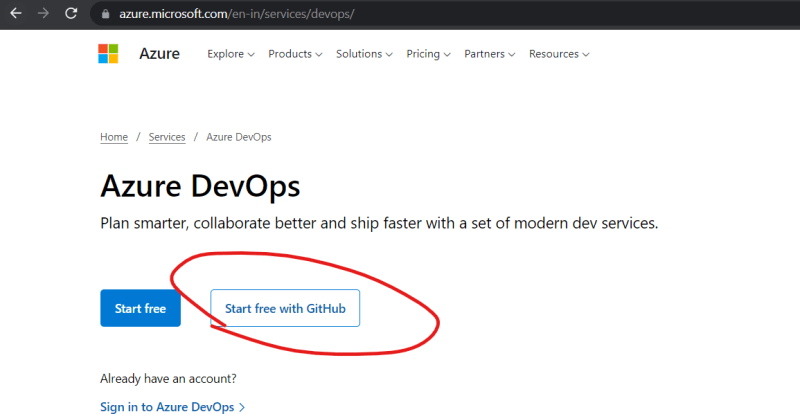
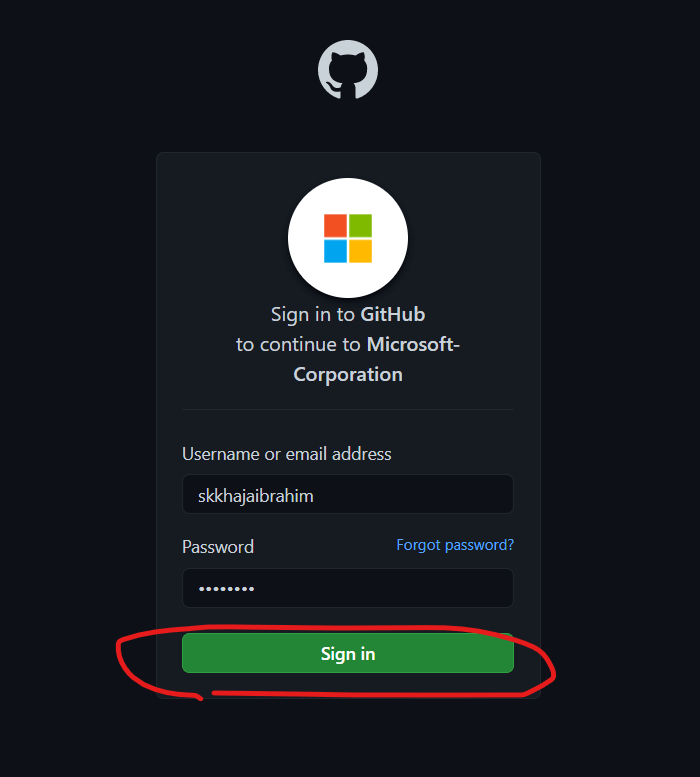
Day-1:

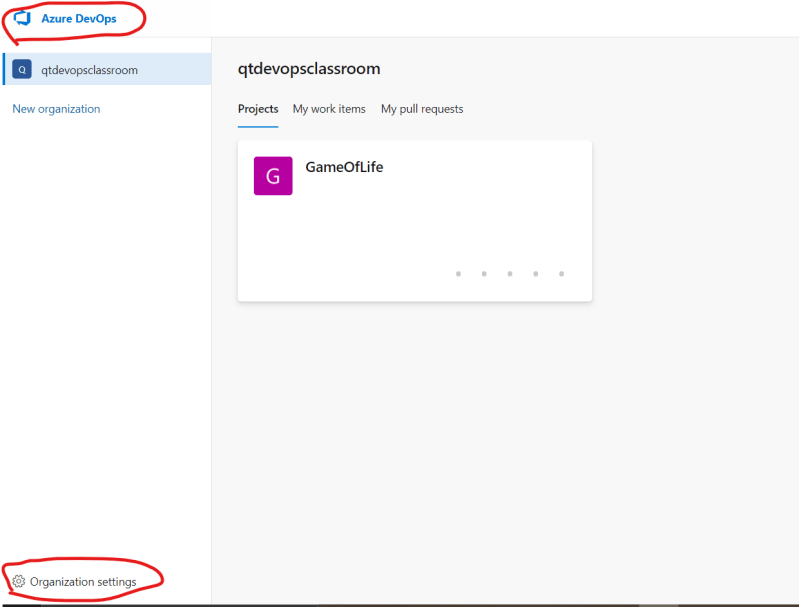
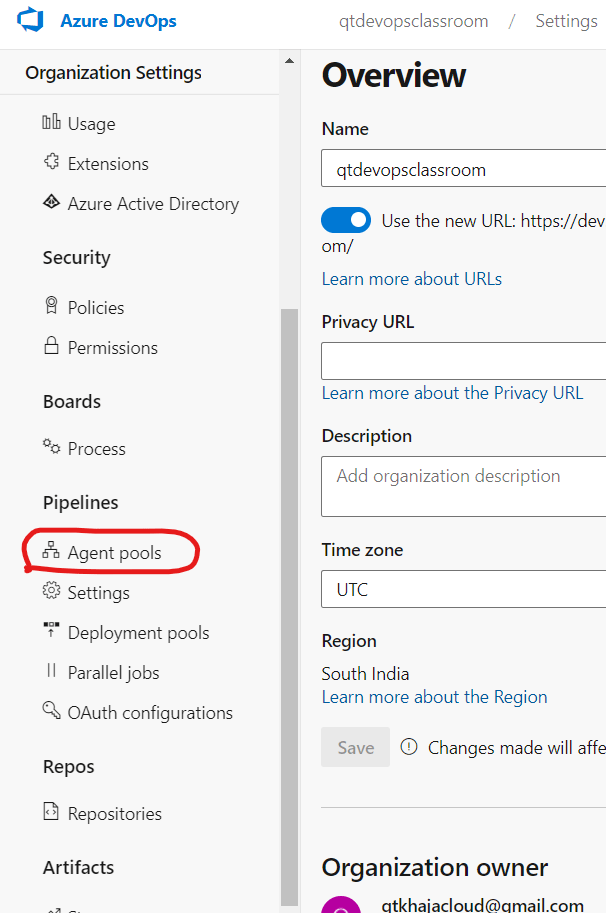
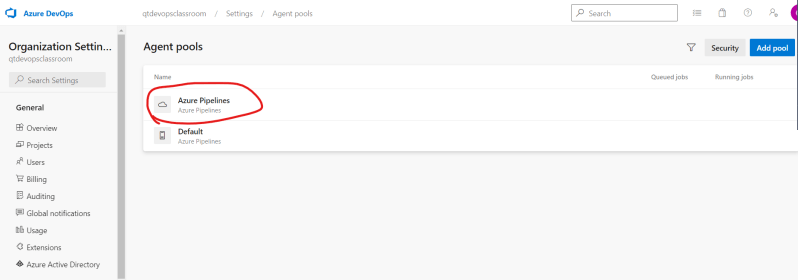
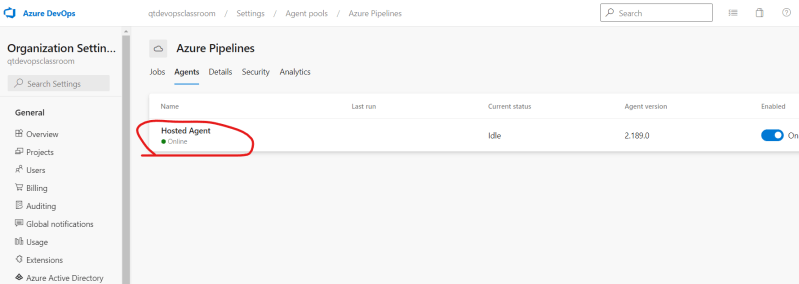
AZD-1

31/Jul/2021

Azure DevOps

* Accessing Azure DevOps
  + For existing Azure Users     
  + If you are not an Azure user we can use github account Naviagate to [Refer Here](https://azure.microsoft.com/en-in/services/devops/)  
* Azure DevOps offers two ways of using Azure Devops
  + Hosted Azure DevOps: Azure DevOps will be hosted and you need to create an account and manage users
    - pricing: [Refer Here](https://azure.microsoft.com/en-in/pricing/details/devops/azure-devops-services/)
  + Installing Azure DevOps on your on-premises server: [Refer Here](https://docs.microsoft.com/en-us/azure/devops/server/install/single-server?view=azure-devops-2020)
* Once the git repo is imported/created in Azure DevOps Source Repos, All we need to do is to create an azure-pipelines.yml file with the necessary build steps configured.
* Exercise: Create an Azure DevOps Organization and create a project called as gameoflife and push the master branch of gameoflife from github into AzureSourceRepo

Writing Azure Pipelines in YAML

* YAML Tutorial [Refer Here](https://www.youtube.com/watch?v=ggOmHlnhPaM&list=PLuVH8Jaq3mLud3sVDvJ-gJ__0zd15wGDd&index=15)
* To write the Pipeline in YAML we need to understand the schema from Azure DevOps [Refer Here](https://docs.microsoft.com/en-us/azure/devops/pipelines/yaml-schema?view=azure-devops&tabs=schema%2Cparameter-schema)
* Pipeline Schema [Refer Here](https://docs.microsoft.com/en-us/azure/devops/pipelines/yaml-schema?view=azure-devops&tabs=schema%2Cparameter-schema#pipeline)
* Triggers [Refer Here](https://docs.microsoft.com/en-us/azure/devops/pipelines/yaml-schema?view=azure-devops&tabs=schema%2Cparameter-schema#triggers)
  + For periodic builds equivalent => Scheduled Trigger [Refer Here](https://docs.microsoft.com/en-us/azure/devops/pipelines/yaml-schema?view=azure-devops&tabs=example%2Cparameter-schema#scheduled-trigger)
  + For poll scm equivalent => Push Trigger [Refer Here](https://docs.microsoft.com/en-us/azure/devops/pipelines/yaml-schema?view=azure-devops&tabs=example%2Cparameter-schema#push-trigger)
* Azure Pipleline job needs to run on some node. Azure DevOps uses the term agent to refer it. In Azure DevOps we have two types of agent
  + Microsoft Hosted Agents:
    - These are hosted by Azure/Microsoft
    - Free plan => 1800 minutes per month is free
  + Self Hosted Agents
    - We can add our agent to the Azure DevOps
    - Adding one agent is part of free plan
* For configuring agents    
* To configure a pool [Refer Here](https://docs.microsoft.com/en-us/azure/devops/pipelines/yaml-schema?view=azure-devops&tabs=example%2Cparameter-schema#pool) and use a microsoft hosted agent [Refer Here](https://docs.microsoft.com/en-us/azure/devops/pipelines/agents/hosted?view=azure-devops&tabs=yaml#use-a-microsoft-hosted-agent) (<https://docs.microsoft.com/en-us/azure/devops/pipelines/agents/hosted?view=azure-devops&tabs=yaml#use-a-microsoft-hosted-agent>)
* In Azure Devops

stages

----> jobs

-----> steps

------> tasks

* For all the tasks [Refer Here](https://docs.microsoft.com/en-us/azure/devops/pipelines/tasks/?view=azure-devops) (<https://docs.microsoft.com/en-us/azure/devops/pipelines/tasks/?view=azure-devops>)
* The Azure pipeline which is created in the class

---

trigger:

- master

variables:

MVN\_GOAL: 'package'

pool:

name: Azure Pipelines

vmImage: 'ubuntu-latest'

steps:

- task: Maven@3

inputs:

mavenPomFile: 'pom.xml'

goals: $(MVN\_GOAL)

jdkVersionOption: 1.8

testResultsFiles: '\*\*/surefire-reports/TEST-\*.xml'

* Equivalent in Jenkins

pipeline {

agent { label 'ubuntu' }

triggers {

pollSCM('\* \* \* \* \*')

}

stages {

stage('SCM') {

steps {

git 'htps:/lksdfskl'

}

}

stage('build') {

steps {

sh 'mvn package'

junit '\*\*/surefire-reports/TEST-\*.xml'

}

}

}

}

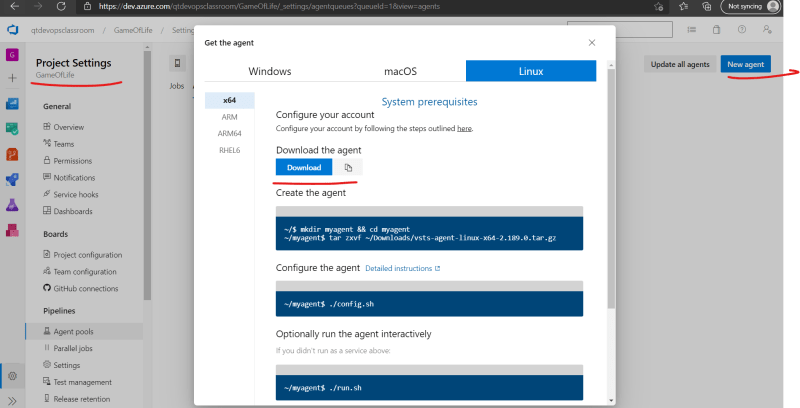
==============================================================================

31/Jul/2021

AZD-2

Adding Agents to Azure DevOps

* For configuring self host agents go through the classroom video
* Create a linux agent (ubuntu 20)
  + Install java 8
  + install maven



* With the azure-pipelines.yaml as shown below and with the self hosted agent

---

trigger:

- master

variables:

MVN\_GOAL: 'package'

pool:

name: Default

steps:

- task: Maven@3

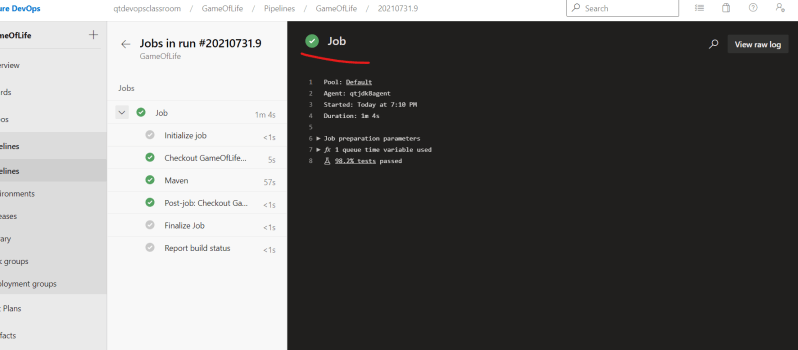
inputs:

mavenPomFile: 'pom.xml'

goals: $(MVN\_GOAL)

jdkVersionOption: 1.8

testResultsFiles: '\*\*/surefire-reports/TEST-\*.xml'



* For working with different technologies in Azure DevOps [Refer Here](https://docs.microsoft.com/en-us/azure/devops/pipelines/ecosystems/ecosystems?view=azure-devops) (<https://docs.microsoft.com/en-us/azure/devops/pipelines/ecosystems/ecosystems?view=azure-devops>)
* For building .net application on Windows
  + Install Visual Studio Build Tools
  + Restore the nuget package C:\hostedtoolcache\windows\NuGet\5.10.0\x64\nuget.exe restore D:\a\1\s\WebApplication1\WebApplication1.sln
  + Run the msbuild command C:\Program Files (x86)\Microsoft Visual Studio\2019\Enterprise\MSBuild\Current\Bin\msbuild.exe" "D:\a\1\s\WebApplication1\WebApplication1.sln"
  + Run the testcases C:\Program Files (x86)\Microsoft Visual Studio\2019\Enterprise\Common7\IDE\Extensions\TestPlatform\vstest.console.exe "@D:\a\\_temp\nml1iy2lu5z.tmp"
* Exercise:
  + Ansible:
    - Write an ansible playbook to deploy game-of-life on tomcat
  + Terraform:
    - Write a terraform template to create a linux instance (in any cloud) and deploy game of life application
  + Docker and K8s:
    - Write a Dockerfile for game-of-life
    - Create a K8s Manifest for game-of-life deployment

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***