Azure DevOps Assignment week 8

---

1. Configure Dashboard and Queries for Work Items

Objective: To track and visualize work items effectively using Azure Boards, dashboards, and query configurations.

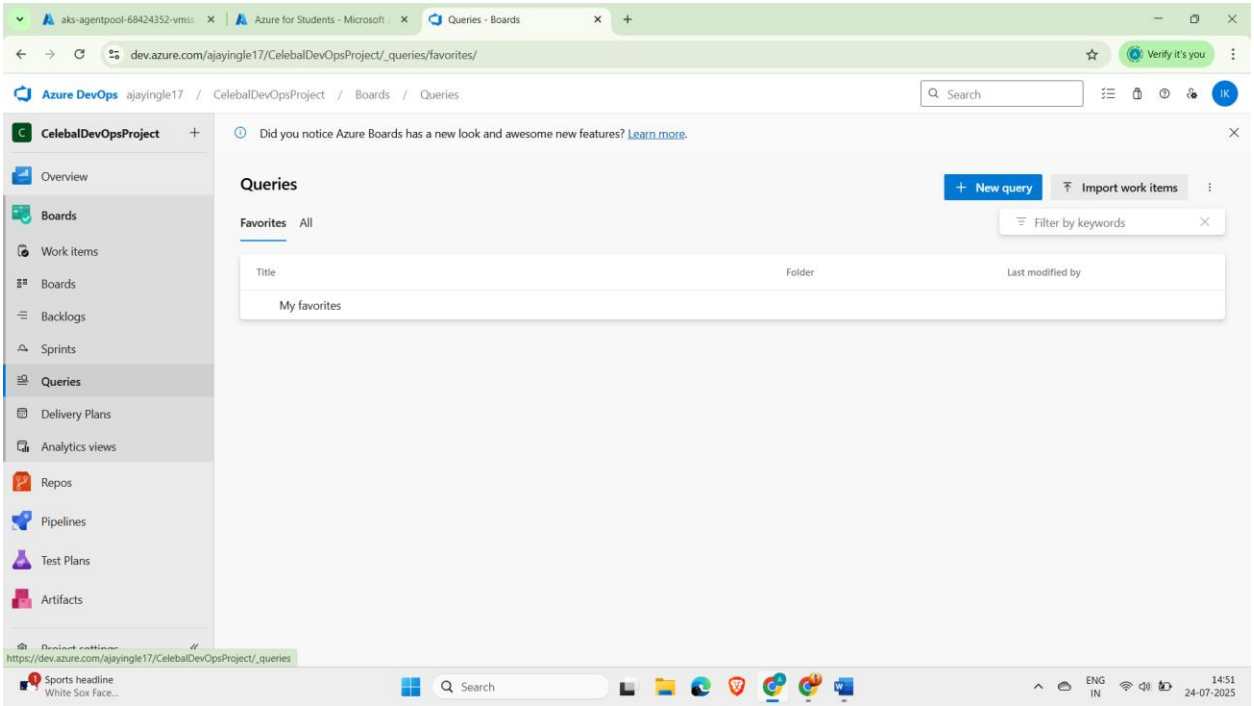
Theoretical Description: Azure DevOps allows you to manage work using Boards. You can create queries to filter work items and then visualize them using Dashboards with charts and widgets.

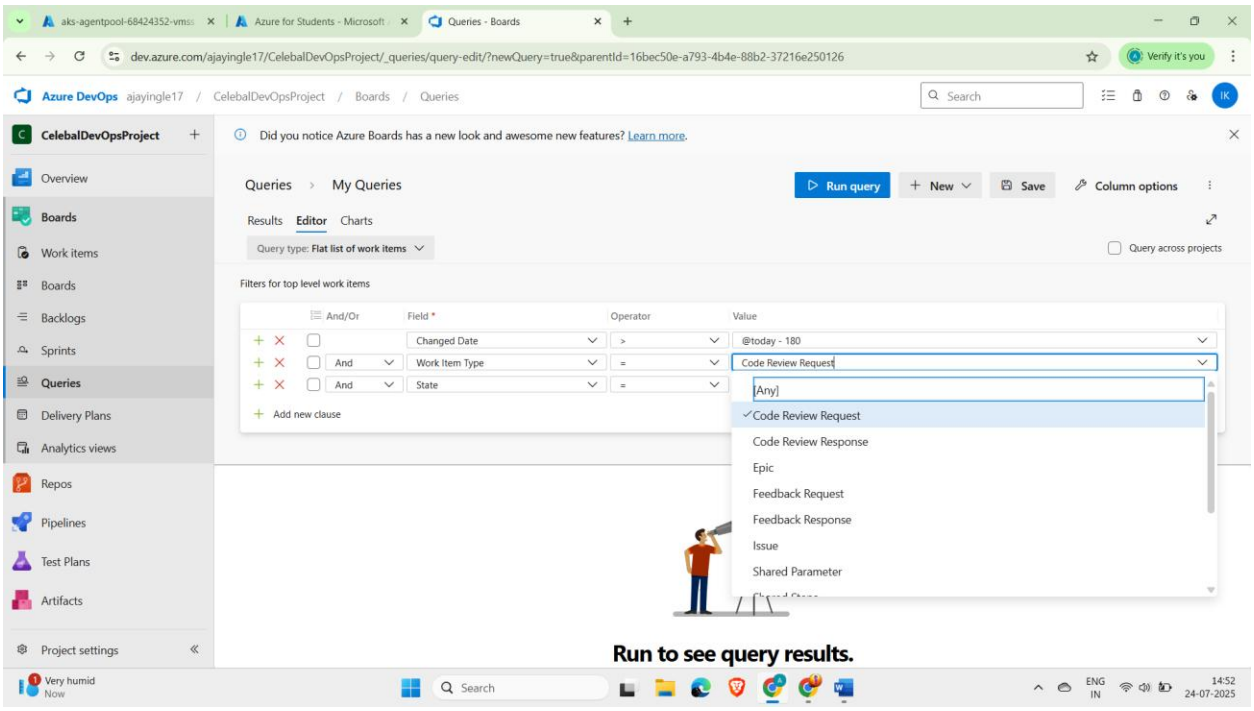
Steps:

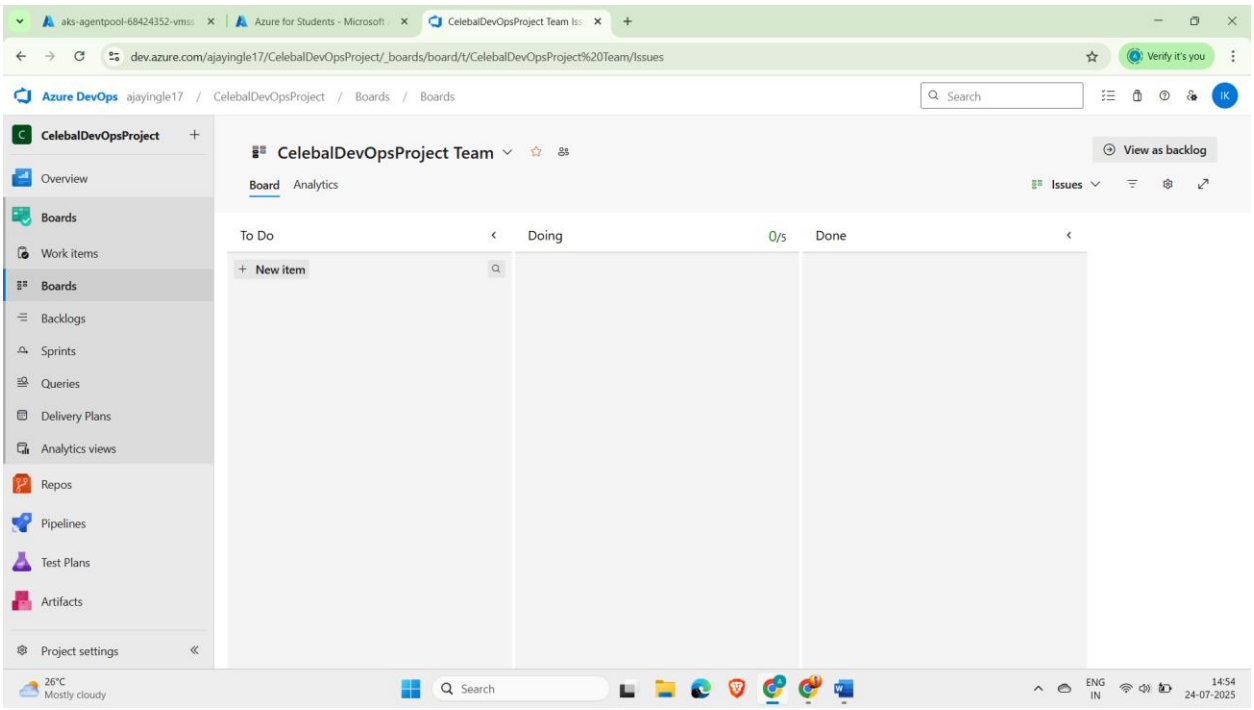
1. Go to Azure DevOps Project > Boards > Queries.
2. Click “New Query” and define conditions (e.g., Assigned To, State).
3. Save the query in Shared Queries.
4. Navigate to Dashboards > Click on “New Dashboard”.
5. Add widgets like Charts, Query Tile, Sprint Overview, etc.
6. Configure each widget to point to your saved queries.

Resources:

- Using Queries in Azure Boards
- YouTube Video: Watch Here




Azure DevOps ajayingle17 / CelebalDevOpsProject / Boards / Queries
Search
CelebalDevOpsProject
Overview
Boards
Work items
Boards
Backlogs
Sprints
Queries
Delivery Plans
Analytics views
Repos
Pipelines
Test Plans
Artifacts
Project settings
Did you notice Azure Boards has a new look and awesome new features? Learn more.
Queries › My Queries
Run query
New
Save
Column options
Results Editor Charts
Query type: Flat list of work items
Query across projects
Filters for top level work items
And/Or
Field *
Operator
Value
Changed Date
@today - 180
And
Work Item Type
Code Review Request
And
State
Add new clause
[Any]
Code Review Request
Code Review Response
Epic
Feedback Request
Feedback Response
Issue
Shared Parameter
Run to see query results.


Azure DevOps ajayingle17 / CelebalDevOpsProject / Boards / Boards
Search
CelebalDevOpsProject
Overview
Boards
Work items
Boards
Backlogs
Sprints
Queries
Delivery Plans
Analytics views
Repos
Pipelines
Test Plans
Artifacts
Project settings
CelebalDevOpsProject Team
View as backlog
Board Analytics
Issues
To Do
Doing 0/5
Done
New item

2. Use Pipeline Variables while Configuring Pipelines

Objective: To make pipelines dynamic and reusable using variables.

Theoretical Description: Pipeline variables store values used across pipelines. They allow dynamic configuration of values like environment name, version, etc.

Steps:

1. Go to Pipelines > Select/Edit your YAML pipeline.
2. Declare variables:

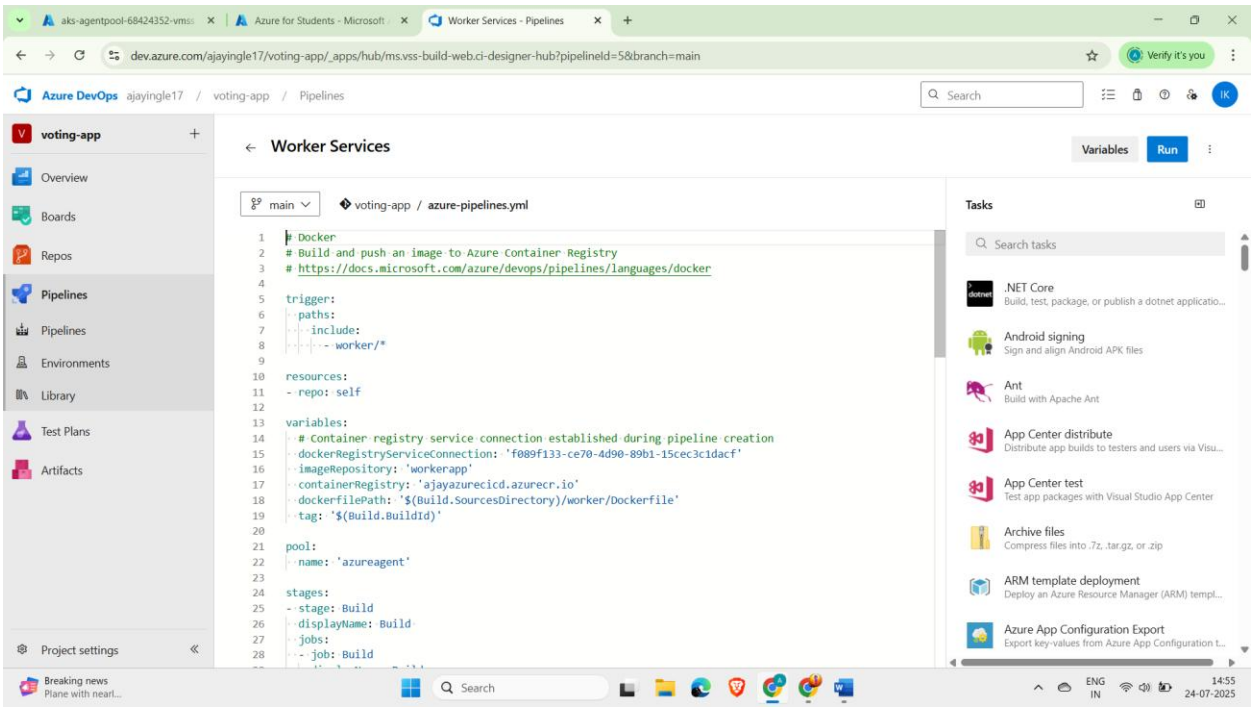
```yaml
variables:
  environment: 'dev'
  version: '1.0.0'
```

3. Use them in tasks:

```yaml
script: echo Deploying version $(version) to $(environment)
```

Resources:

- Pipeline Variables Documentation
- YouTube Video: Watch Here



3. Use Variable and Task Groups in Pipelines and Set Scopes

Objective: To manage common variables/tasks across pipelines and apply scoped values to different stages.

Theoretical Description: Variable groups store shared variables across pipelines. Task groups are reusable sets of tasks. Scopes can control when and where variables apply.

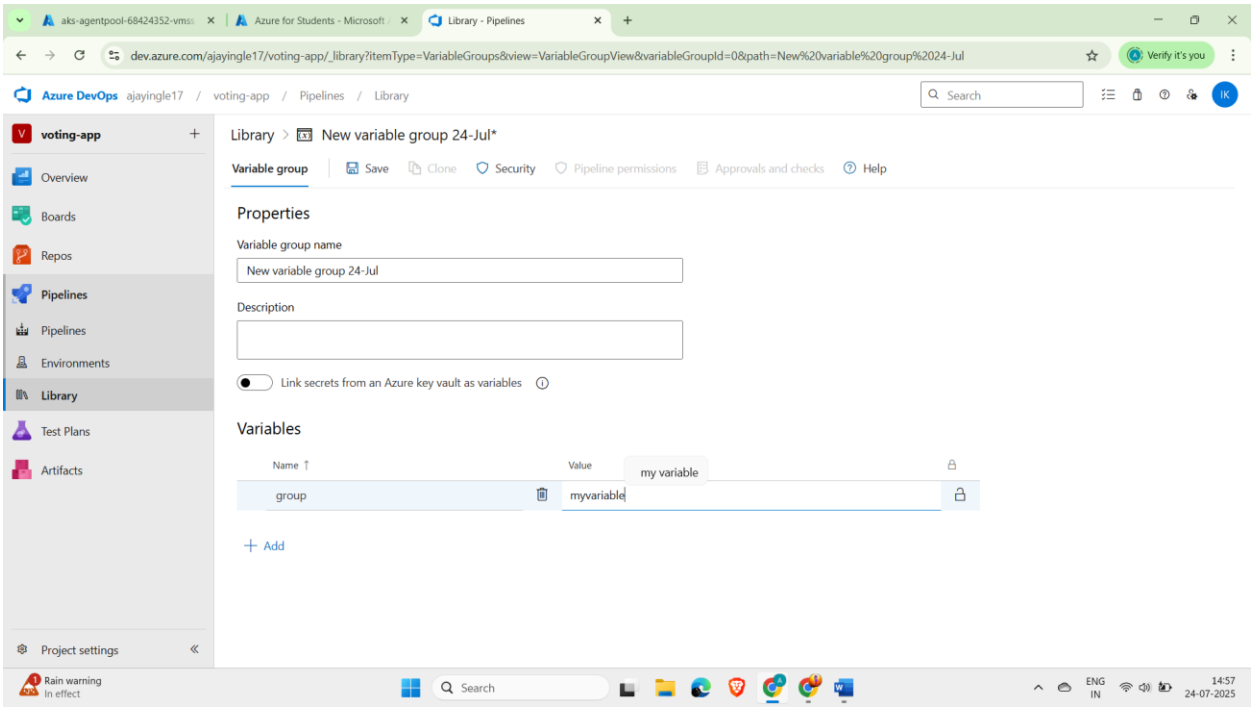
Steps:

1. Navigate to Pipelines > Library > Variable groups.
2. Click “Add” and define variables.
3. Link to pipeline YAML using `variableGroup`.
4. To scope:

```yaml
variables:
  - group: MyVariables
stages:
  - stage: Build
    variables:
      - name: env
        value: 'dev'
```

Resources:

- Variable Groups
- YouTube Video: Watch Here



4. Create a Service Connection

Objective: To connect Azure DevOps to external services like Azure, Docker, GitHub, etc.

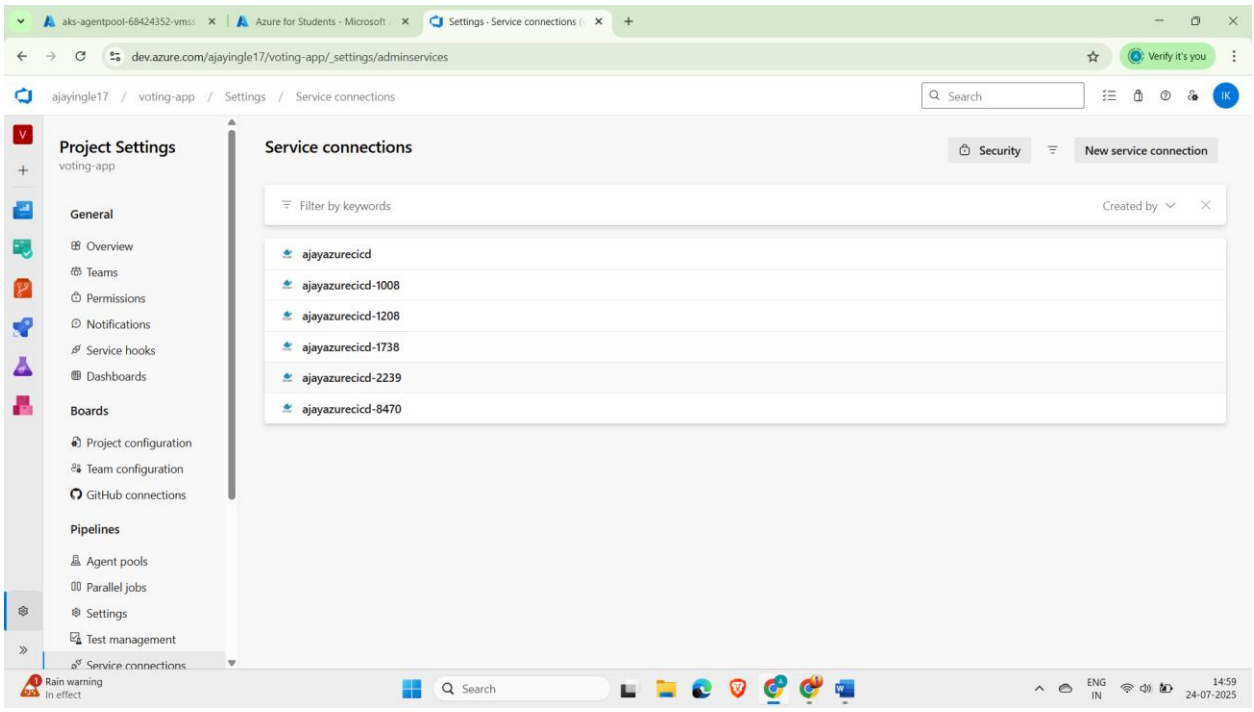
Theoretical Description: Service connections provide credentials for external systems, allowing tasks like deployment and artifact push.

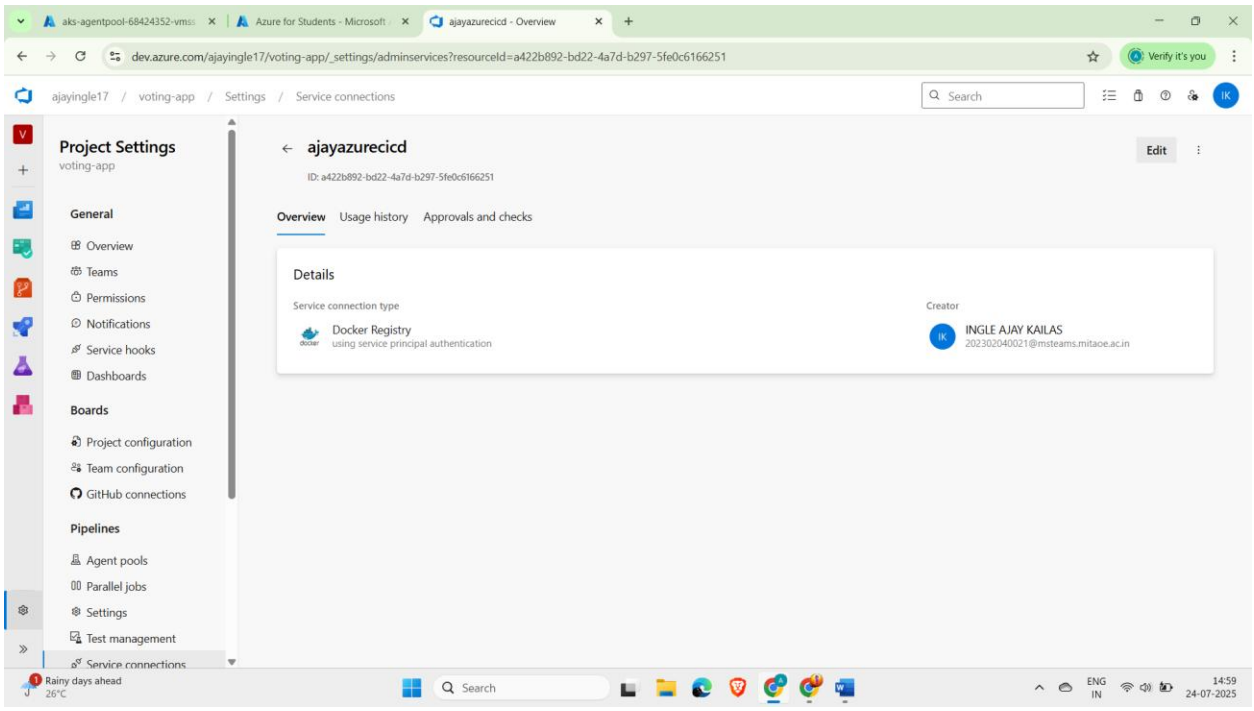
Steps:

1. Navigate to Project Settings > Service Connections.
2. Click “New Service Connection” > Select Azure Resource Manager.
3. Authenticate via Service Principal or Azure CLI.
4. Give connection a name and verify.

Resources:

- Service Connections
- YouTube Video: Watch Here





---

5. Create a Linux/Windows Self-Hosted Agent

Objective: To create custom agent machines to execute pipelines.

Theoretical Description: Self-hosted agents allow more control over the environment and installed tools.

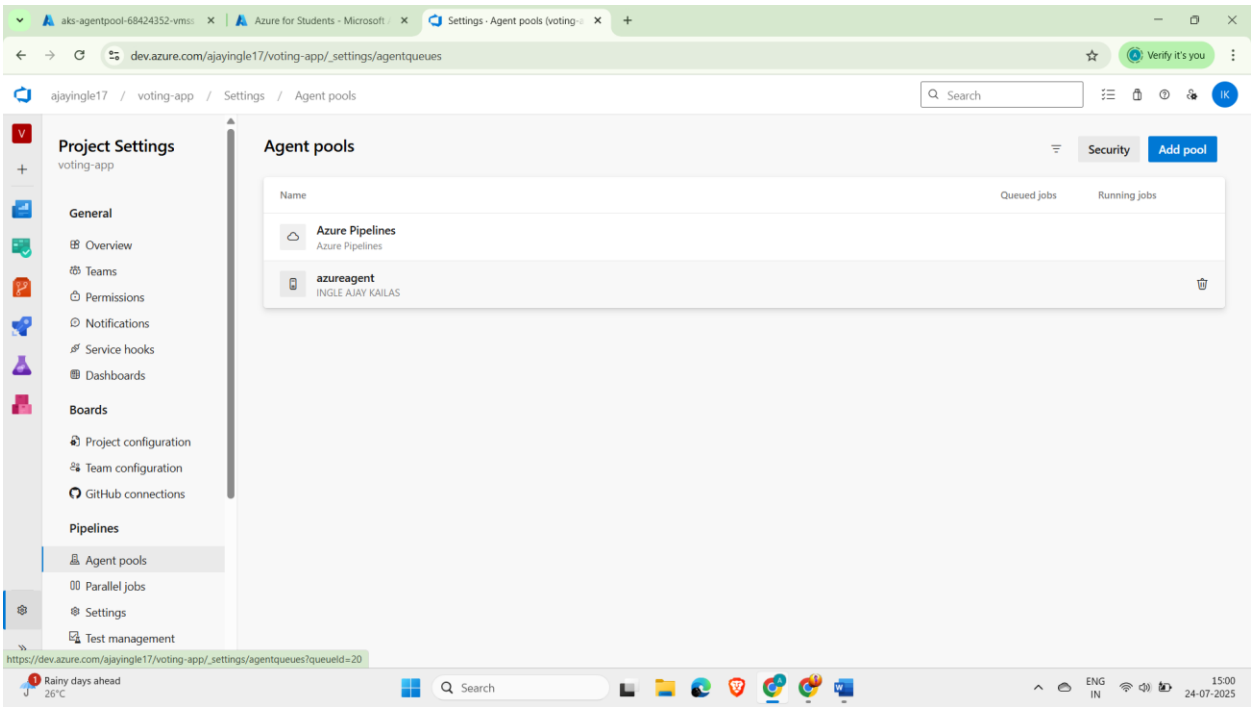
Steps:

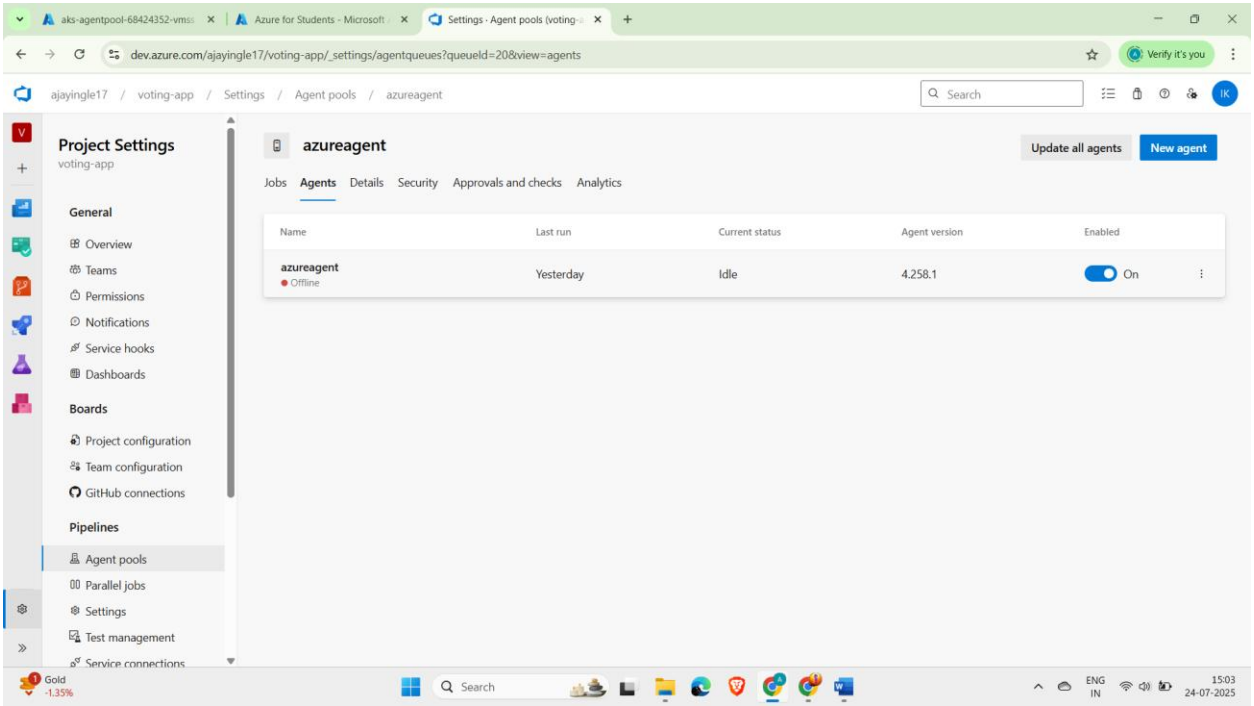
1. Navigate to Project Settings > Agent Pools.
2. Add a new pool or use default.
3. Click “New agent” and choose OS.
4. Download agent package and configure using command-line:

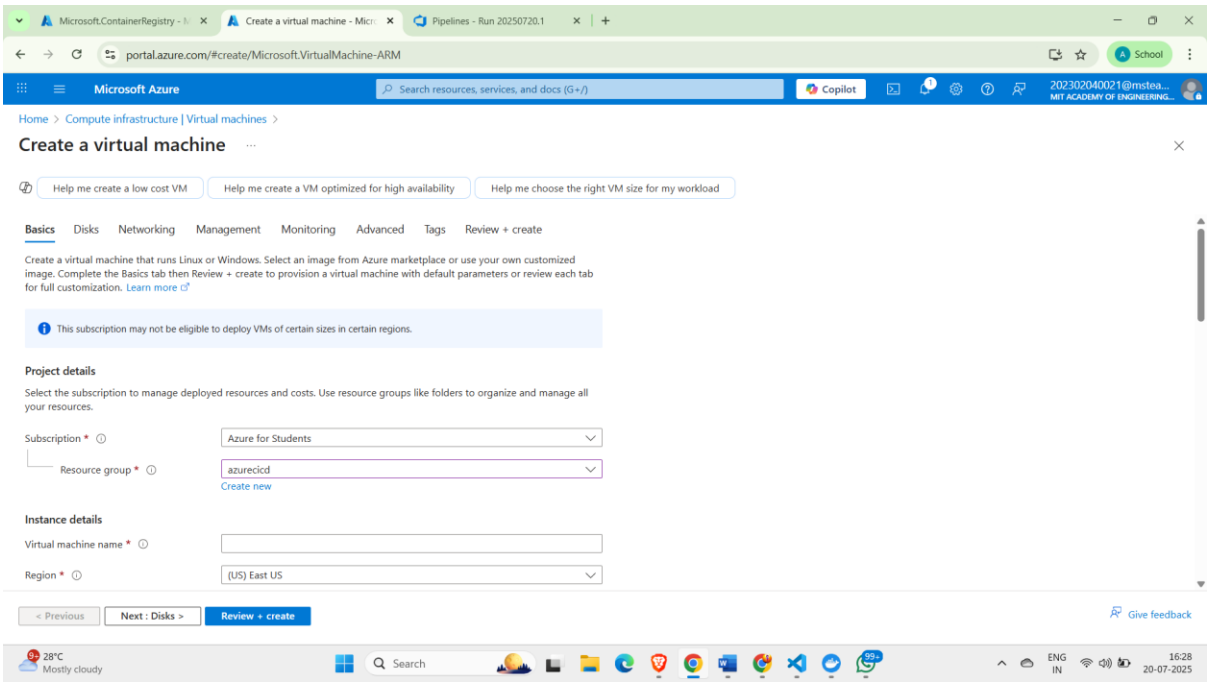
```
./config.sh --unattended --url https://dev.azure.com/ORG --auth PAT
./svc.sh install
./svc.sh start
```

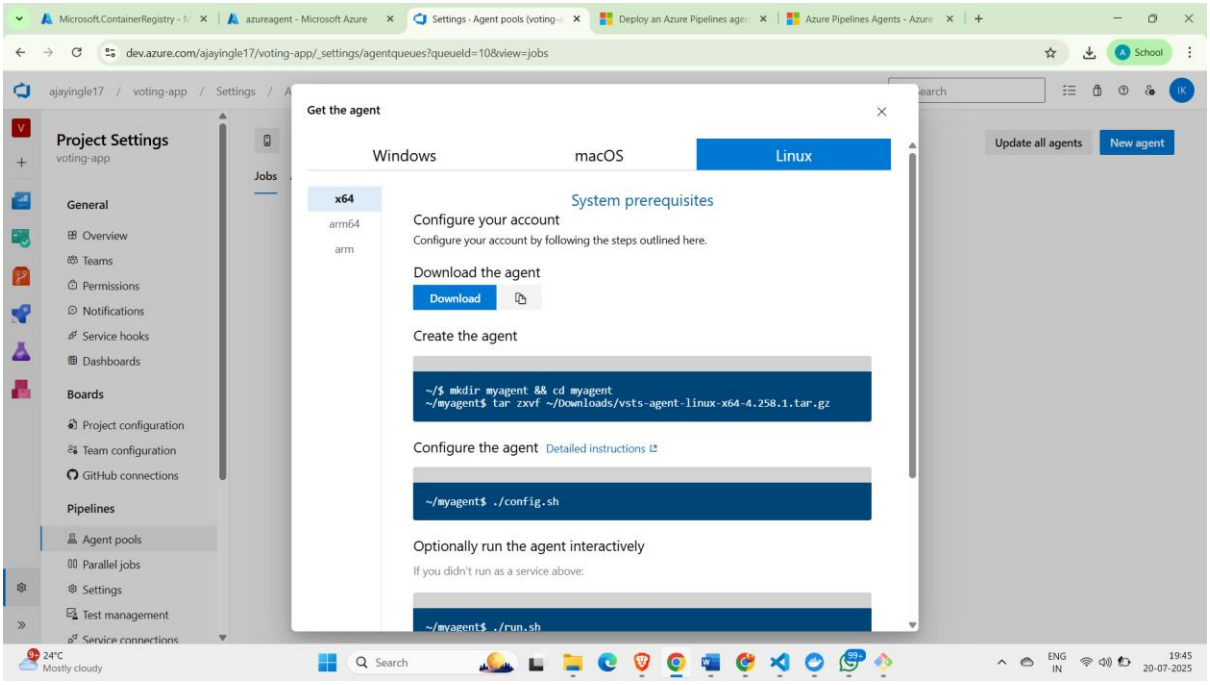
Resources:

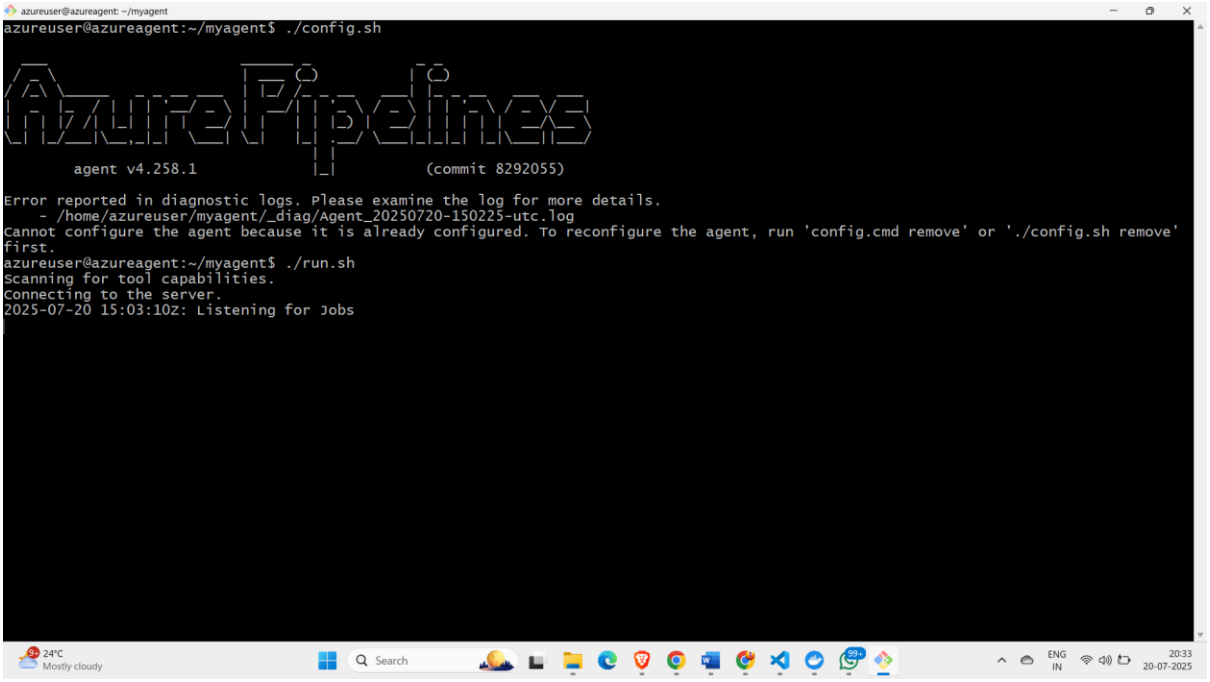
- Linux Agent Guide
- YouTube Video: Watch Here

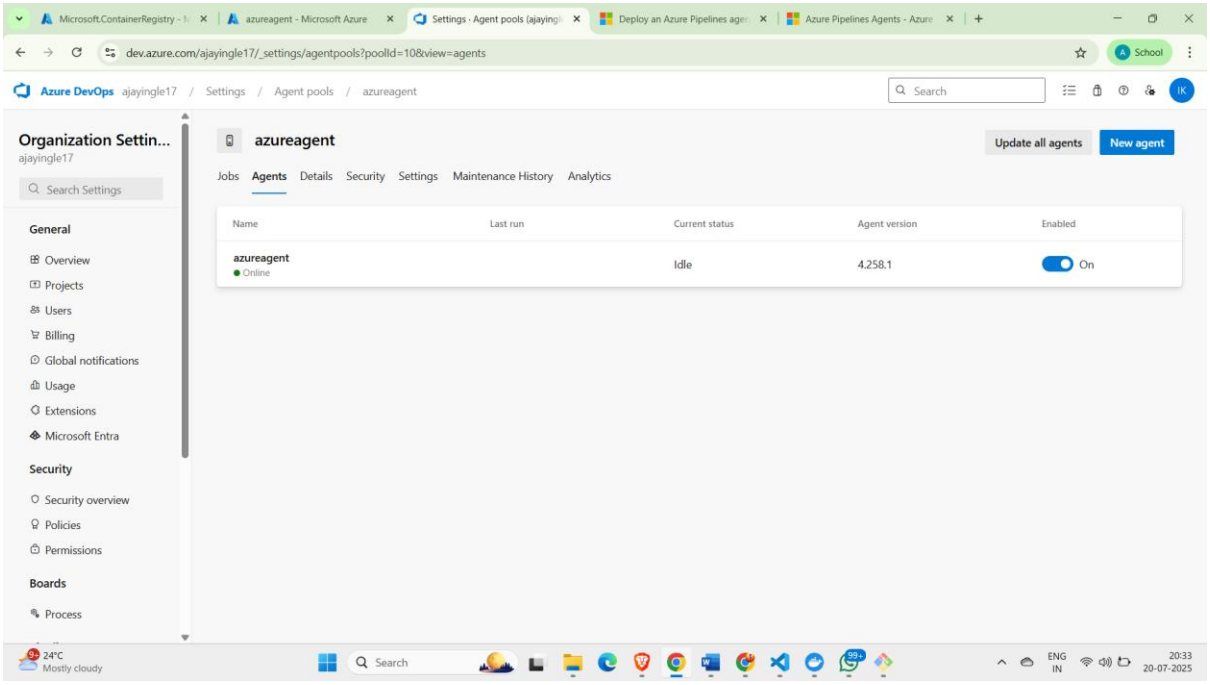


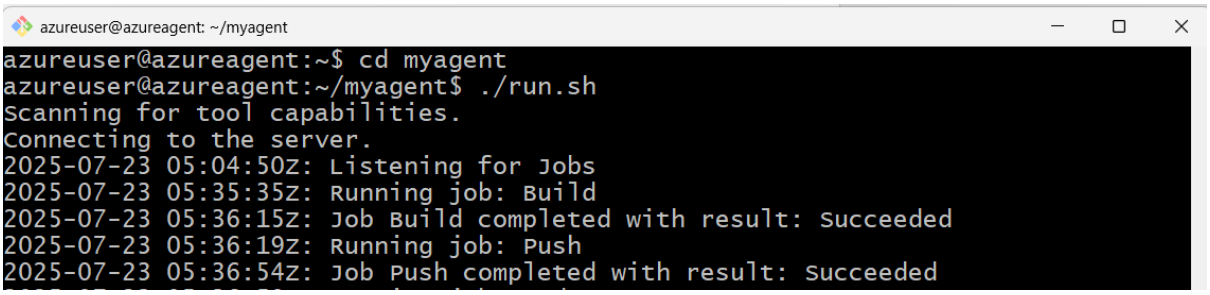



Microsoft.ContainerRegistry - M
Create a virtual machine - Micr
Pipelines - Run 20250720.1
portal.azure.com/#create/Microsoft.VirtualMachine-ARM
School
Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
202302040021@mstea...
MIT ACADEMY OF ENGINEERING...
Home > Compute infrastructure | Virtual machines >
Create a virtual machine
Help me create a low cost VM
Help me create a VM optimized for high availability
Help me choose the right VM size for my workload
Basics Disks Networking Management Monitoring Advanced Tags Review + create
Create a virtual machine that runs Linux or Windows. Select an image from Azure marketplace or use your own customized image. Complete the Basics tab then Review + create to provision a virtual machine with default parameters or review each tab for full customization. Learn more
This subscription may not be eligible to deploy VMs of certain sizes in certain regions.
Project details
Select the 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 *
azurecicd
Create new
Instance details
Virtual machine name *
Region *
(US) East US
< Previous
Next : Disks >
Review + create
Give feedback
28°C
Mostly cloudy
Search
ENG
IN
16:28
20-07-2025


Microsoft.ContainerRegistry - M
azureagent - Microsoft Azure
Settings · Agent pools (voting-
Deploy an Azure Pipelines age
Azure Pipelines Agents - Azure
dev.azure.com/ajayingle17/voting-app/_settings/agentqueues?queueId=10&view=jobs
School
ajayingle17 / voting-app / Settings
Project Settings
voting-app
General
Overview
Teams
Permissions
Notifications
Service hooks
Dashboards
Boards
Project configuration
Team configuration
GitHub connections
Pipelines
Agent pools
Parallel jobs
Settings
Test management
Service connections
Jobs
Update all agents
New agent
Get the agent
Windows
macOS
Linux
x64
arm64
arm
System prerequisites
Configure your account
Configure your account by following the steps outlined here.
Download the agent
Download
Create the agent
~/$ mkdir myagent && cd myagent
~/myagent$ tar zxvf ~/Downloads/vsts-agent-linux-x64-4.258.1.tar.gz
Configure the agent
Detailed instructions
~/myagent$ ./config.sh
Optionally run the agent interactively
If you didn't run as a service above:
~/myagent$ ./run.sh
24°C
Mostly cloudy
Search
ENG
IN
19:45
20-07-2025


azureuser@azureagent: ~/myagent
azureuser@azureagent:~/myagent$ ./config.sh
agent v4.258.1
(commit 8292055)
Error reported in diagnostic logs. Please examine the log for more details.
- /home/azureuser/myagent/_diag/Agent_20250720-150225-utc.log
Cannot configure the agent because it is already configured. To reconfigure the agent, run 'config.cmd remove' or './config.sh remove' first.
azureuser@azureagent:~/myagent$ ./run.sh
Scanning for tool capabilities.
Connecting to the server.
2025-07-20 15:03:10Z: Listening for Jobs


dev.azure.com/ajayingle17/_settings/agentpools?poolId=10&view=agents
Azure DevOps
ajayingle17 / Settings / Agent pools / azureagent
Organization Settin...
ajayingle17
Search Settings
General
Overview
Projects
Users
Billing
Global notifications
Usage
Extensions
Microsoft Entra
Security
Security overview
Policies
Permissions
Boards
Process
azureagent
Update all agents
New agent
Jobs
Agents
Details
Security
Settings
Maintenance History
Analytics
Name
Last run
Current status
Agent version
Enabled
azureagent
Online
Idle
4.258.1
On


azureuser@azureagent: ~/myagent
azureuser@azureagent:~$ cd myagent
azureuser@azureagent:~/myagent$ ./run.sh
Scanning for tool capabilities.
Connecting to the server.
2025-07-23 05:04:50Z: Listening for Jobs
2025-07-23 05:35:35Z: Running job: Build
2025-07-23 05:36:15Z: Job Build completed with result: Succeeded
2025-07-23 05:36:19Z: Running job: Push
2025-07-23 05:36:54Z: Job Push completed with result: Succeeded

6. Apply Pre and Post-Deployment Approvers in Release Pipeline

Objective: To control deployment flow and enforce approvals.

Theoretical Description: Approvers can review and approve deployments before or after stages.

Steps:

1. Go to Pipelines > Releases > Edit pipeline.
2. Select stage > Click on pre/post-deployment conditions.
3. Add users/groups as approvers.
4. Enable timeout, rejections, and comments.

Resources:

- Approvals Documentation
- YouTube Video: Watch Here

---

7. CI/CD Pipeline to Build and Push Docker Image to ACR and Deploy to AKS

Objective: Automate containerization and deployment to Kubernetes (AKS).

Steps:

1. Configure ACR and AKS service connections.
2. YAML Example:

```yaml
trigger:
  branches:
    include: [main]

variables:
  imageName: 'myapp'

pool:
  vmImage: 'ubuntu-latest'

steps:
  - task: Docker@2
    inputs:
      command: buildAndPush
      repository: $(imageName)
      dockerfile: '**/Dockerfile'
      containerRegistry: 'ACR-SC'
```

```
    - task: Kubernetes@1
      inputs:
        connectionType: 'Azure Resource Manager'
        kubernetesServiceEndpoint: 'AKS-SC'
        command: apply
        useConfigurationFile: true
        configuration: 'manifests/deployment.yaml'
```

Resources:

- CI/CD to AKS
- YouTube Video: Watch Here