



## Azure Pipelines

### 1. What is Azure Pipelines?

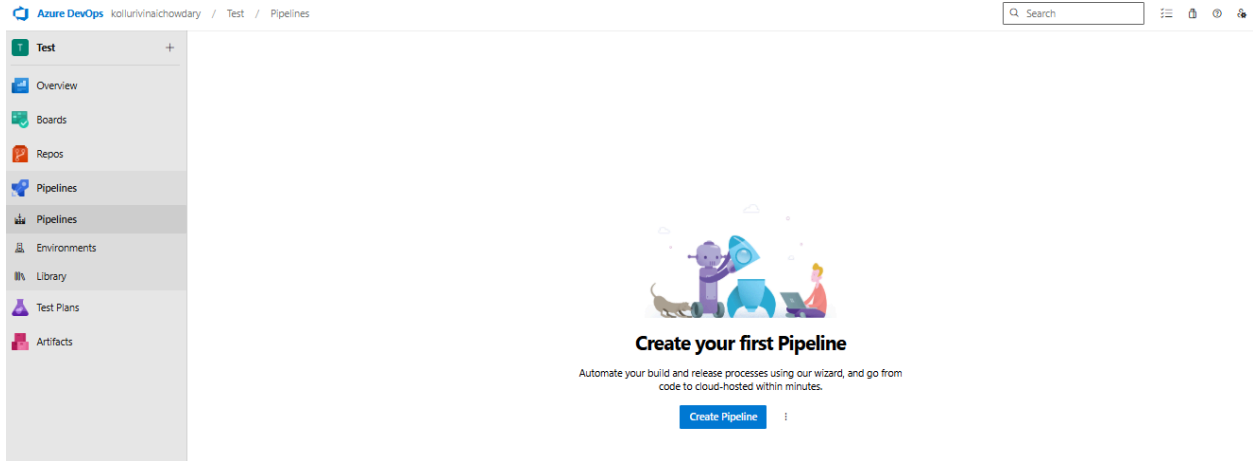
#### Definition

Azure Pipelines is a **CI/CD service** in Azure DevOps used to:

- Build code
- Run tests
- Package applications
- Deploy to environments (VMs, App Service, AKS, etc.)

#### Why Enterprises Use Azure Pipelines

- Cloud & on-prem support
- YAML + Classic pipelines
- Deep integration with Azure Repos, GitHub, Artifacts
- Enterprise security, approvals, auditing



## 2. Azure Pipelines Architecture

### Core Components

- Pipeline
- Agent
- Stages
- Jobs
- Steps
- Tasks

### Flow

Code Commit → Pipeline Trigger → Agent → Tasks → Artifacts → Deployment



## 3. Types of Pipelines

### 3.1 Build Pipeline (CI)

- Compiles code
- Runs unit tests
- Publishes artifacts

### 3.2 Release Pipeline (CD – Classic)

- Deploys build artifacts
- Uses environments & approvals

### 3.3 YAML Pipeline (Modern – Recommended)

- CI + CD in one file
- Version controlled



## 4: Pipeline Infrastructure & Execution Control

*(Azure DevOps Enterprise Execution Layer)*

This explains how Azure Pipelines actually run, where jobs execute, how access is secured, and how enterprises control speed, scale, and permissions.

### 4.1 Azure Pipeline Agents – Execution Engines

#### What is an Agent?

An **Azure Pipeline Agent** is a **machine that executes pipeline jobs** such as:

- Building code
- Running tests
- Packaging artifacts
- Deploying applications

Pipelines do not run inside Azure DevOps.

They always run **on agents**.

Examples:



## AZURE

- `ubuntu-latest`
- `windows-latest`

## 4.2 Agent Pools

### What is an Agent Pool?

An **Agent Pool** is a **collection of agents** that pipelines can use to run jobs.

Agent pools help enterprises:

- Reuse agents
- Control access
- Isolate workloads
- Scale execution

### Types of Agent Pools

#### 1. Microsoft-Hosted Agent Pools

- Managed by Microsoft
- No maintenance required
- New VM for every run
- Recommended for learning and cloud-native workloads



Examples:

- `ubuntu-latest`
- `windows-latest`
- `macos-latest`

## 2. Self-Hosted Agent Pools (Enterprise Usage)

- Installed on customer-managed servers or VMs
- Used when:
  - Internal network access is required
  - Custom tools are needed
  - Compliance policies exist

## Why Enterprises Use Multiple Agent Pools

- Separate **Build** and **Deployment** agents
- Restrict **Production deployments**
- Assign agents to specific teams

Example:

- `Build-Agent-Pool`
- `Deploy-Agent-Pool`
- `Prod-Restricted-Pool`



## 4.3 Agent Pool Settings & Management

### Key Agent Pool Settings

Setting	Purpose
Pool Name	Logical identification
Agent Type	Hosted or self-hosted
Permissions	Who can use/manage
Auto-Provision	Managed by Microsoft
Capabilities	OS, tools, software

### Agent Capabilities

Capabilities define **what an agent can do**:

- OS type
- Installed software
- Custom labels

Used for:

- Matching jobs to correct agents
- Avoiding failures due to missing tools



## 4.4 Parallel Jobs (Speed & Scale Control)

### What are Parallel Jobs?

Parallel jobs define **how many pipeline jobs can run at the same time.**

Parallel jobs = **Pipeline execution capacity**

### Types of Parallel Jobs

#### Microsoft-Hosted Parallel Jobs

- Limited by Azure DevOps license
- Shared across organization
- Additional capacity requires paid plans

#### Self-Hosted Parallel Jobs

- Limited only by:
  - Number of agents
  - Server resources





## Why Parallel Jobs Matter in Enterprises

- Faster CI/CD
- Multiple teams running pipelines
- Reduced waiting time
- High productivity

### Example:

- 1 parallel job → pipelines queue
- 10 parallel jobs → faster feedback

### Enterprise Best Practice

- Use hosted agents for CI
- Use self-hosted agents for CD
- Allocate parallel jobs per team

## 4.5 Service Connections (Secure Access Layer)

### What is a Service Connection?

A **Service Connection** is a **secure authentication method** that allows Azure Pipelines to connect to external systems.

Pipelines never use user credentials directly.



## Why Service Connections Are Mandatory

- Secure authentication
- Centralized access control
- Audit and compliance
- No hard-coded secrets

### Common Service Connection Types

Service	Purpose
Azure Resource Manager	Deploy to Azure
GitHub	Access GitHub repos
Docker Registry	Push/pull images
Kubernetes	Deploy to AKS
Generic	External APIs

## Service Connection Security (Enterprise Level)

- RBAC controlled
- Scoped to:
  - Subscription
  - Resource group
- Approval-based usage
- Auditable access



## AZURE

### Execution Flow





## 4.6 Real Enterprise Design Example

### Scenario

- 50 developers
- 10 pipelines
- Production compliance required

### Setup

- Microsoft-hosted agents → Build
- Self-hosted agents → Deployment
- Separate prod agent pool
- Limited service connection access
- Parallel jobs allocated per team



# AZURE

Project Settings

Overview

Teams

Permissions

Notifications

Service hooks

Dashboards

Boards

Project configuration

Team configuration

GitHub connections

Pipelines

Agent pools

Parallel jobs

Settings

Test management

Service connections

XAML build services

Project details

Name

Test

Description

Process

Basic

Save

Project administrators

VK

vinal kolluri

kollurivinaichowdary@gmail.com

Add administrator

Project Settings

Test

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

XAML build services

Agent pools

Name

Queued jobs

Running jobs

Azure Pipelines

Azure Pipelines

Default

Azure Pipelines

Azure DevOps

kollurivinaichowdary

Test / Settings / Agent pools / Default

Search

Test

Overview

Boards

Repos

Pipelines

Test Plans

Artifacts

Project Settings

Permissions

Notifications

Service hooks

Dashboards

Boards

Project configuration

Team configuration

GitHub connections

Pipelines

Agent pools

Parallel jobs

Settings

Test management

Service connections

XAML build services

Default

Jobs Agents Details Security Approvals and checks Analytics

Update all agents

New agent

Add your first agent

Manage agents and run pipeline jobs on this pool.

New agent

Contact : +91 9966107782  
info@sdlctechacademy.com

**Azure DevOps** kollurvinachowdary / Test / Settings / Agent

**Test** +

- Overview
- Boards
- Repos
- Pipelines
- Test Plans
- Artifacts

**Project Settings**

- Permissions
- Notifications
- Service hooks
- Dashboards
- Boards
  - Project configuration
  - Team configuration
  - GitHub connections
- Pipelines
  - Agent pools
  - Parallel jobs
  - Settings
  - Test management
  - Service connections
  - XAML build services
- Repos
  - Repositories
- Artifacts
  - Storage
- Test

**Get the agent**

Windows macOS Linux

**x64**

**System prerequisites**

Configure your account  
Configure your account by following the steps outlined here.

Download the agent  
[Download](#)

Create the agent

```
PS C:\> mkdir agent ; cd agent
PS C:\agent> Add-Type -AssemblyName System.IO.Compression.FileSystem ;
[System.IO.Compression.ZipFile]::ExtractToDirectory("$HOME\Downloads\vs-agent-win-x64-4.266.2.zip", "$PWD")
```

Configure the agent [Detailed instructions](#)

```
PS C:\agent> .\config.cmd
```

Optionally run the agent interactively  
If you didn't run as a service above:

```
PS C:\agent> .\run.cmd
```

That's it!

Update all agents [New agent](#)



## AZURE

kollurivinalchowdary / Test / Settings / Agent pools / Azure Pipelines

Search

### Project Settings

Test

- General
  - Overview
  - Teams
  - Permissions
  - Notifications
  - Service hooks
  - Dashboards
- Boards
  - Project configuration
  - Team configuration
  - GitHub connections
- Pipelines
  - Agent pools

### Azure Pipelines

Jobs Agents Details Security Approvals and checks Analytics

Name	Last run	Current status	Agent version	Enabled
Hosted Agent		Idle	4.266.2	<input checked="" type="checkbox"/> On

kollurivinalchowdary / Test / Settings / Settings

Search

### Settings

Retention policy

The artifacts, symbols and attachments retention setting is being ignored because the runs retention setting is evaluated first.

Days to keep artifacts, symbols and attachments: 30

Days to keep runs: 30

Days to keep pull request runs: 10

Number of recent runs to retain per pipeline: 3

[Learn more about run retention](#)

General

☒ On **Disable anonymous access to badges**  
Anonymous users can access the status badge API for all pipelines unless this option is enabled.

☒ On **Limit variables that can be set at queue time**  
You can set any variables at queue time unless this option is enabled. With this option enabled, only those variables that are explicitly marked as "Settable at queue time" can be set. [Learn more](#)

☒ On **Limit job authorization scope to current project for non-release pipelines**  
Non-Release Pipelines can run with collection scoped access tokens unless this option is enabled. With this option enabled, you can reduce the scope of access for all non-release pipelines to the current project.

koilurvinaichowdary / Test / Settings / Parallel jobs

Search

### Project Settings

Test

- 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
  - XAML build services

#### Private projects

Microsoft-hosted	0
<a href="#">View in progress jobs</a>	Parallel jobs
Monthly purchases	0 <a href="#">Change</a>
Self-hosted	1
<a href="#">View in progress jobs</a>	Parallel jobs
Free parallel jobs	1
Visual Studio Enterprise subscribers	0
Monthly purchases	0 <a href="#">Change</a>

#### Public projects

Microsoft-hosted	0
<a href="#">View in progress jobs</a>	Parallel jobs
Self-hosted	Unlimited
<a href="#">View in progress jobs</a>	Parallel jobs


koilurvinaichowdary / Test / Settings / Service connections

Search

### Project Settings

Test

- 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
  - XAML build services



### Create your first service connection

Service connections help you manage, protect, and reuse authentications to external services.

[Create service connection](#)



kolur/vina Chowdary / Test / Settings / Service connections

**Project Settings**  
Test

**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
- XAML build services

**Repos**

**Create your first service connection**

Service connections help you manage, protect, and reuse authentications to external services.

[Create service connection](#)

**New service connection**

Choose a service or connection type

Search connection types

- ☒ Azure Repos/Team Foundation Server
- ☐ Azure Resource Manager
- ☐ Azure Service Bus
- ☐ Bitbucket Cloud
- ☐ Cargo
- ☐ Chef
- ☐ Docker Host
- ☐ Docker Registry
- ☐ Generic
- ☐ GitHub
- ☐ GitHub Enterprise Server
- ☐ Incoming WebHook
- ☐ Jenkins

[Learn more](#) [Next](#)



## AZURE

### New GitHub service connection ×

**Authentication method**

☒ Grant authorization  
☐ Personal Access Token

**Authentication**

**OAuth Configuration**

Authorize

**Service connection details**

**Service Connection Name**

**Description (optional)**

**Security**

☐ Grant access permission to all pipelines

[Learn more](#)  
[Troubleshoot](#)

Back Save



## 5. Pipeline Creation – Step by Step (YAML)

### Step 1: Go to Pipelines

- Azure DevOps → Pipelines → New Pipeline

### Step 2: Select Code Source

- Azure Repos Git
- GitHub
- Bitbucket



## AZURE

Azure DevOps kollurivinaichowdary / Test / Pipelines

Connect Select Configure Review

New pipeline

### Where is your code?

**Azure Repos Git** YAML  
Free private Git repositories, pull requests, and code search

**GitHub** YAML  
Home to the world's largest community of developers More options

**Bitbucket Cloud** YAML  
Hosted by Atlassian

Connect Select Configure Review

New pipeline

### Select a repository

Filter by keywords Test ×

practice

testing

### Step 3: Select Pipeline Template

- Starter pipeline
- ASP.NET
- Java
- Node.js

Azure DevOps kollurivinaichowdary / Test / Pipelines

Test +

- Overview
- Boards
- Repos
- Pipelines
- Pipelines
- Environments
- Library
- Test Plans
- Artifacts

Project settings <<

✓ Connect ✓ Select **Configure** Review

New pipeline

### Configure your pipeline

- Starter pipeline**  
Start with a minimal pipeline that you can customize to build and deploy your code.
- Existing Azure Pipelines YAML file**  
Select an Azure Pipelines YAML file in any branch of the repository.
- .NET Core Function App to Windows on Azure**  
Build a .NET Core function app and deploy it to Azure as a Windows function App.
- .NET Desktop**  
Build and run tests for .NET Desktop or Windows classic desktop solutions.
- Android**  
Build your Android project with Gradle.
- Ant**  
Build your Java projects and run tests with Apache Ant.
- ASP.NET**  
Build and test ASP.NET projects.
- ASP.NET Core**  
Build and test ASP.NET Core projects targeting .NET Core.
- ASP.NET Core (.NET Framework)**  
Build and test ASP.NET Core projects targeting the full .NET Framework.
- C/C++ with GCC**  
Build your C/C++ project with GCC using make.
- Deploy to Azure Kubernetes Service**  
Build and push image to Azure Container Registry; Deploy to Azure Kubernetes Service

Select an existing YAML file

Select an Azure Pipelines YAML file in any branch of the repository.

Branch  
dev

Path  
/Azurepipeline.yml

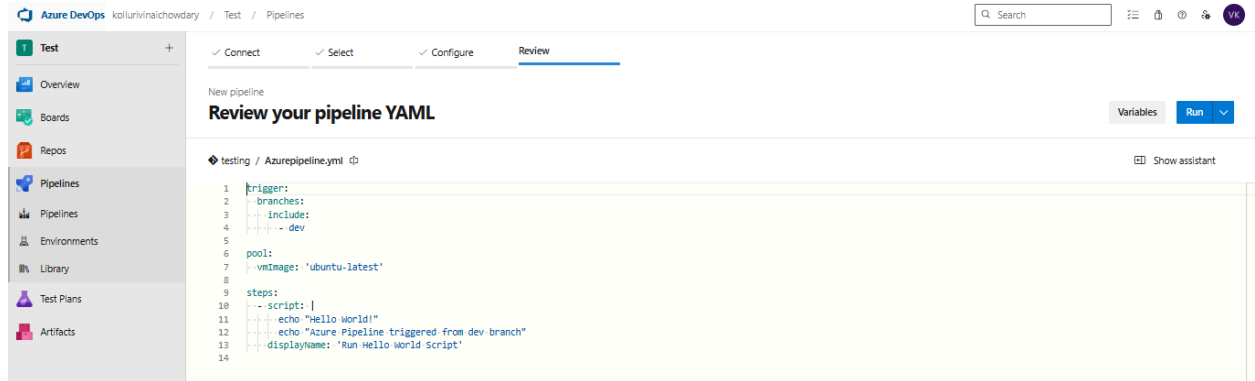
Select a file from the dropdown or type in the path to your file

testing

Cancel Continue



# AZURE



## Step 4: YAML File Created

- `azure-pipelines.yml`
- Stored inside repo



## 6. YAML Pipeline Structure (Core Learning)

```
trigger:
```

```
- main
```

```
pool:
```

```
  vmImage: ubuntu-latest
```

```
stages:
```

```
- stage: Build
```

```
  jobs:
```

```
    - job: BuildJob
```

```
      steps:
```

```
        - task: ExampleTask
```



## Key Sections Explained

### Key Sections Explained

Section	Purpose
trigger	When pipeline runs
pool	Agent selection
stages	High-level phases
jobs	Logical execution units
steps	Actual tasks

## 7. Triggers in Azure Pipelines

### 7.1 CI Trigger

- Runs on code commit

### 7.2 Branch Filters

- main
- develop
- feature/\*

### 7.3 Path Filters

- Trigger only if specific folders change





## 7.4 Scheduled Triggers

- Nightly builds
- Weekly security scans

## 8. Variables & Variable Groups (Enterprise Usage)

### 8.1 Pipeline Variables

- Key-value pairs
- Used across pipeline

### 8.2 Variable Groups

- Centralized variables
- Shared across pipelines

### 8.3 Secret Variables

- Passwords
- Tokens
- Hidden in logs
- Secure Files

The screenshot displays the Azure DevOps interface across three panels, illustrating the process of reviewing a pipeline and creating a variable group.

**Top Panel: Review your pipeline YAML**

The left sidebar shows the navigation menu with 'Test' selected. The main area displays the 'Review' tab for a new pipeline named 'testing / Azurepipeline.yml'. The pipeline YAML content is as follows:

```

1 trigger:
2   branches:
3     - dev
4   pool:
5     vmImage: 'ubuntu-latest'
6   steps:
7     - script: |
8       echo "Hello World!"
9       echo "Azure Pipeline triggered from dev branch"
10      displayName: 'Run Hello World script'

```

**Right Panel: Variables**

This panel provides information about variables, stating: "Use variables to store values or encrypted secrets separately from your repository." It includes a 'New variable' button and a link to 'Learn about variables'.

**Bottom Panel: Library - New variable group**

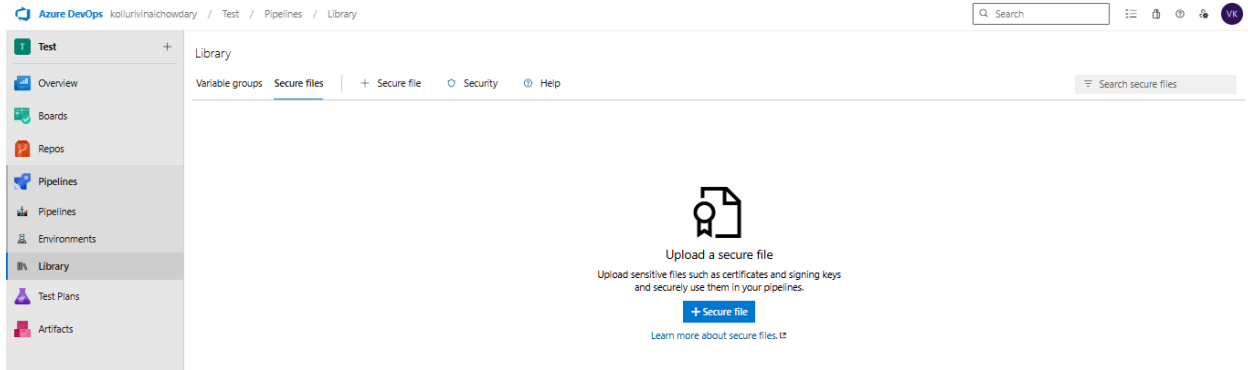
The bottom panel shows the 'Library' view with the 'Variable groups' tab selected. It displays a 'New variable group' card with the instruction: "Create groups of variables that you can share across multiple pipelines." A '+ Variable group' button is present, along with a link to 'Learn more about variable groups.'.

**Bottom Panel: Variable group configuration**

Below the 'New variable group' card, the configuration page for a variable group named 'Test' is shown. The 'Properties' section includes a 'Variable group name' field with the value 'Test' and a 'Description' field. A checkbox for 'Link secrets from an Azure key vault as variables' is checked. The 'Variables' section contains a table with the following data:

Name	Value
username	vinay
password	Vinay@7782

A '+ Add' button is located at the bottom of the variables table.



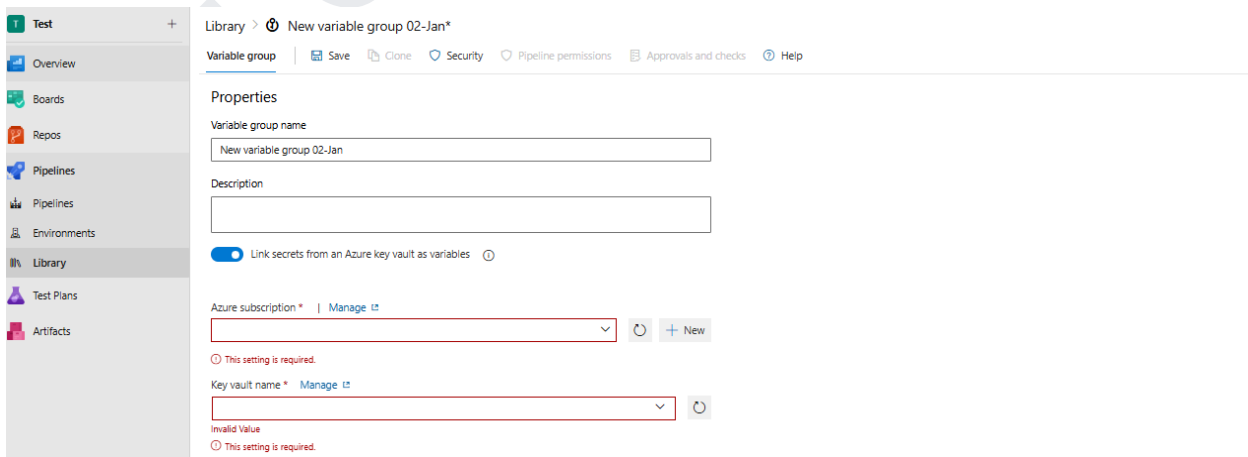
## 9. Secure Secrets with Azure Key Vault

### Why?

- Enterprise security
- No hard-coded secrets

### Flow

Pipeline → Variable Group → Key Vault → Secret





## 10. Tasks in Azure Pipelines

### What is a Task?

- Prebuilt action
- Example:
  - Build
  - Copy files
  - Publish artifacts
  - Deploy

### Types

- Built-in tasks
- Marketplace extensions
- Custom scripts

## 11. Artifacts (Build Outputs)

### Why Artifacts?

- Store build output
- Used by deployment pipelines



## Artifact Types

- Pipeline artifact
- Universal package

## 12. Multi-Stage Pipelines

### Example Stages

- Build
- Test
- Deploy-Dev
- Deploy-QA
- Deploy-Prod

### Benefits

- Single pipeline
- Full visibility
- Controlled promotion



## 13. Environments in Azure Pipelines

### What is an Environment?

- Logical deployment target
- Example:
  - Dev
  - QA
  - Prod

### Features

- Deployment history
- Approvals
- Resource tracking



## AZURE

Azure DevOps kolluvinsachowdary / Test / Pipelines / Environments

Search

New environment

Environment	Status	Last activity
dev	Never deployed	Just now
qa	Never deployed	Just now
uat	Never deployed	Just now

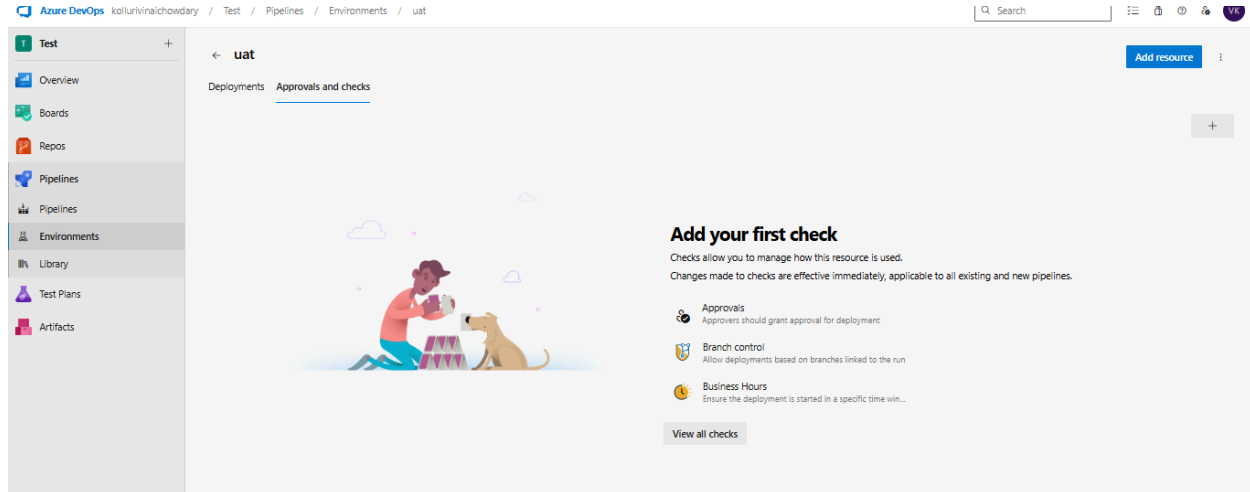
## 14. Approvals & Gates

### Manual Approvals

- Required before deployment

### Automated Gates

- Health checks
- Monitoring validation



## 15. Deployment Strategies

### Types

- RunOnce
- Rolling
- Canary
- Blue-Green (via YAML logic)





## 16. Service Connections

### What is Service Connection?

- Secure connection to external systems

Examples:

- Azure subscription
- Docker registry
- Kubernetes cluster

## 17. Permissions & Security

### Pipeline Security

- Who can edit pipeline
- Who can run pipeline

### Environment Security

- Deployment approvals
- Role-based access



## AZURE

The screenshot shows the Azure DevOps interface with the 'Permissions for testing' dialog box open. The dialog lists various permissions for the 'Project Collection Administrators' group. The permissions are as follows:

Permission	Value
Administer build permissions	Allow (inherited)
Delete build pipeline	Allow (inherited)
Delete builds	Allow (inherited)
Destroy builds	Allow (inherited)
Edit build pipeline	Allow (inherited)
Edit build quality	Allow (inherited)
Edit queue build configuration	Allow (inherited)
Manage build qualities	Allow (inherited)
Manage build queue	Allow (inherited)
Override check-in validation by build	Allow (inherited)
Queue builds	Allow (inherited)
Retain indefinitely	Allow (inherited)
Stop builds	Allow (inherited)
Update build information	Not set
View build pipeline	Allow (inherited)
View builds	Allow (inherited)

## 18. Monitoring & Troubleshooting

### Features

- Pipeline logs
- Task-level logs
- Timeline view

### Common Failures

- Agent issues



## AZURE

- Permission issues
- Variable not found

## 19. Pipeline Best Practices

- Use YAML pipelines
- Separate build & deploy stages
- Use Key Vault for secrets
- Enable approvals for prod
- Use reusable templates
- Enforce branch policies

## 20. Azure Pipelines vs Jenkins

Azure Pipelines	Jenkins
Fully managed	Self-managed
Azure native	Plugin dependent
Secure by default	Needs hardening



## 21. Real Enterprise Pipeline Flow

