edureka!

Microsoft Azure DevOps Solution Certification (AZ-400)

COURSE OUTLINE

Azure AZ-400

MODULE 1: Introduction to Azure DevOps

MODULE 2: Implementing Continuous Integration

MODULE 3: Build Containers with Azure DevOps

MODULE 4: Designing a Dependency Management Strategy and Managing

Artifact Versioning

MODULE 5: Setting up Release Management Workflow

MODULE 6: Implementing Deployment Models and Services

MODULE 7: Implement and Optimize Continuous Feedback Mechanism

MODULE 8: Azure Tools: Infrastructure and Configuration, and Third-Party Tools

MODULE 9: Implementing Compliance and Security

MODULE 10: Azure Case Studies

edureka!

Implementing Continuous Integration

Topics

Following are the topics covered in this module:

- DevOps Pipeline
- Azure Pipeline
- Hosted and Private Agents
- Pipeline and Concurrency
- Azure Pipeline YAML and Visual Designer
- Continuous Integration
- Build Strategy Implementation
- Integrating Azure Pipelines
- Setting up Private Agents
- Analyze and Integrate Docker Multi-Stage Builds

Objectives

After completing this module, you should be able to:

- Understand DevOps Pipeline
- Explain Azure Pipeline
- Describe Hosted and Private Agents
- Explain Pipeline and Concurrency
- Understand Azure Pipeline YAML and Visual Designer
- Explain Continuous Integration
- Understand Build Strategy and its Implementation
- Integrate Azure Pipelines
- Set up Private Agents
- Analyze and Integrate Docker Multi-Stage Builds

edureka!

Zomato’s Release Concerns

edureka!

Zomato Service

Zomato is an Indian restaurant aggregator and food delivery company. Zomato provides information, menus, and user-reviews of restaurants as well as food delivery options from partner restaurants in select cities by using mobile and web apps

Zomato App Management

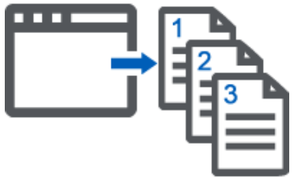
It receives customer feedback based on the service and the quality of food. So, the application needs regular changes based on the requirement. It hired Mr. Cook as a team lead to make the necessary changes and release the new version of the app to the consumers

The Issue

Mr. Cook must find out and add new features to the app. But he is facing the following issues:

FEATURE MANAGEMENT

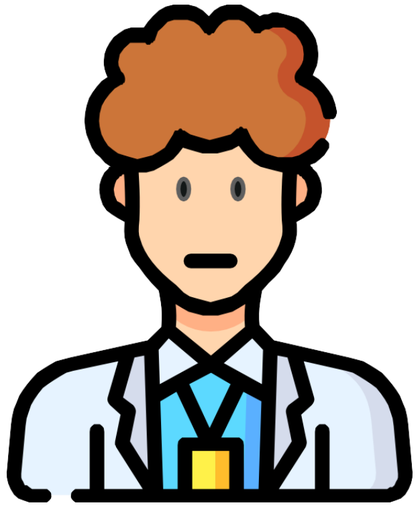
- Understanding the customer requirements
- Faster integration of new features to the application



CODE MANAGEMENT

- Tracking and managing changes made in code by developers
- Manage the deployment and the updates

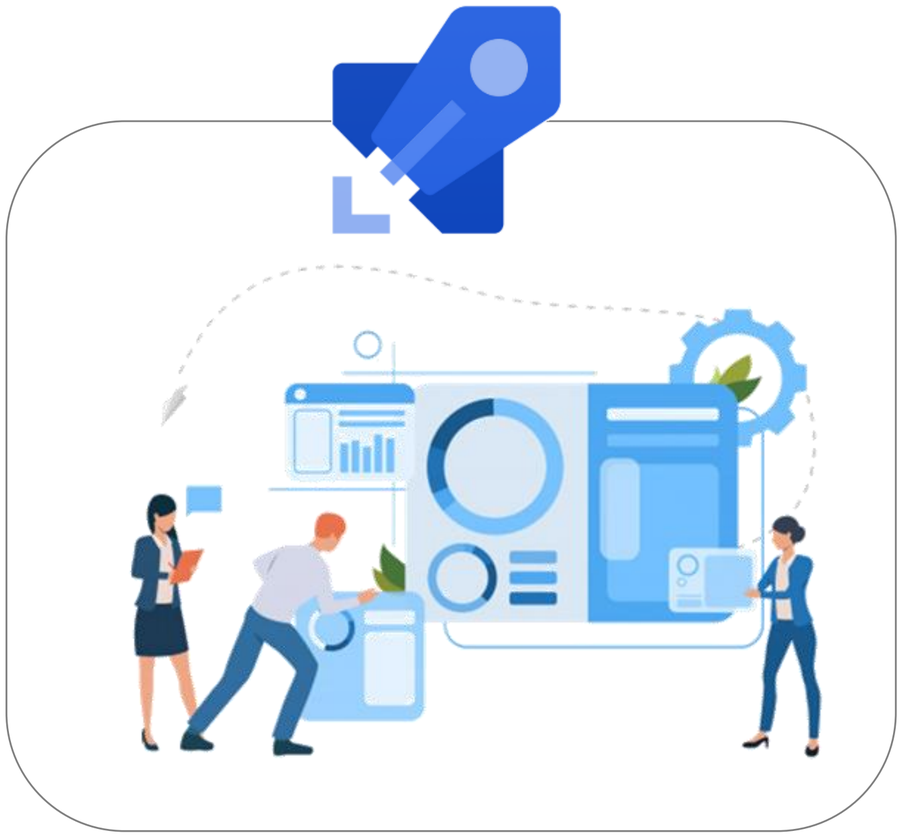
The Research



Mr. Cook
The Lead

From my research, it is clear that I have to use Azure pipeline to support the code updates and perform continuous integration and integrate external source controls

The Solution



AZURE PIPELINE

- Azure pipelines allow you to automatically run builds, perform tests and deploy code (release) to various development and production environments

- To continuously and consistently test and create the code and ship it to any target, Azure Pipelines combines continuous integration (CI) and continuous delivery (CD)

Azure DevOps Pipeline

edureka!

What is DevOps Pipeline?

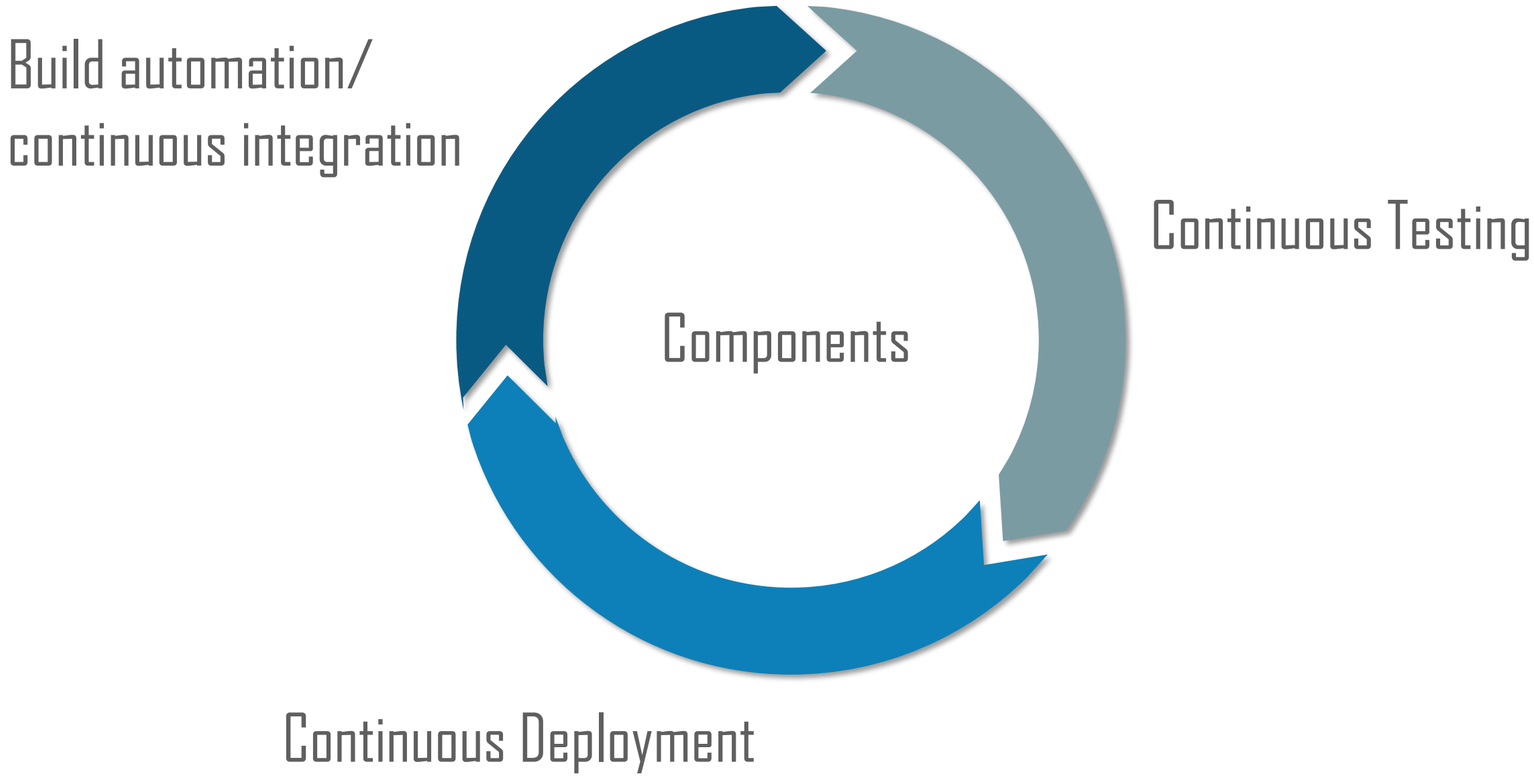
DevOps pipeline is a set of automated processes

Allows developers and DevOps professionals to efficiently compile the solution, generate the installer, run the tests, and deploy the installer to the production system.

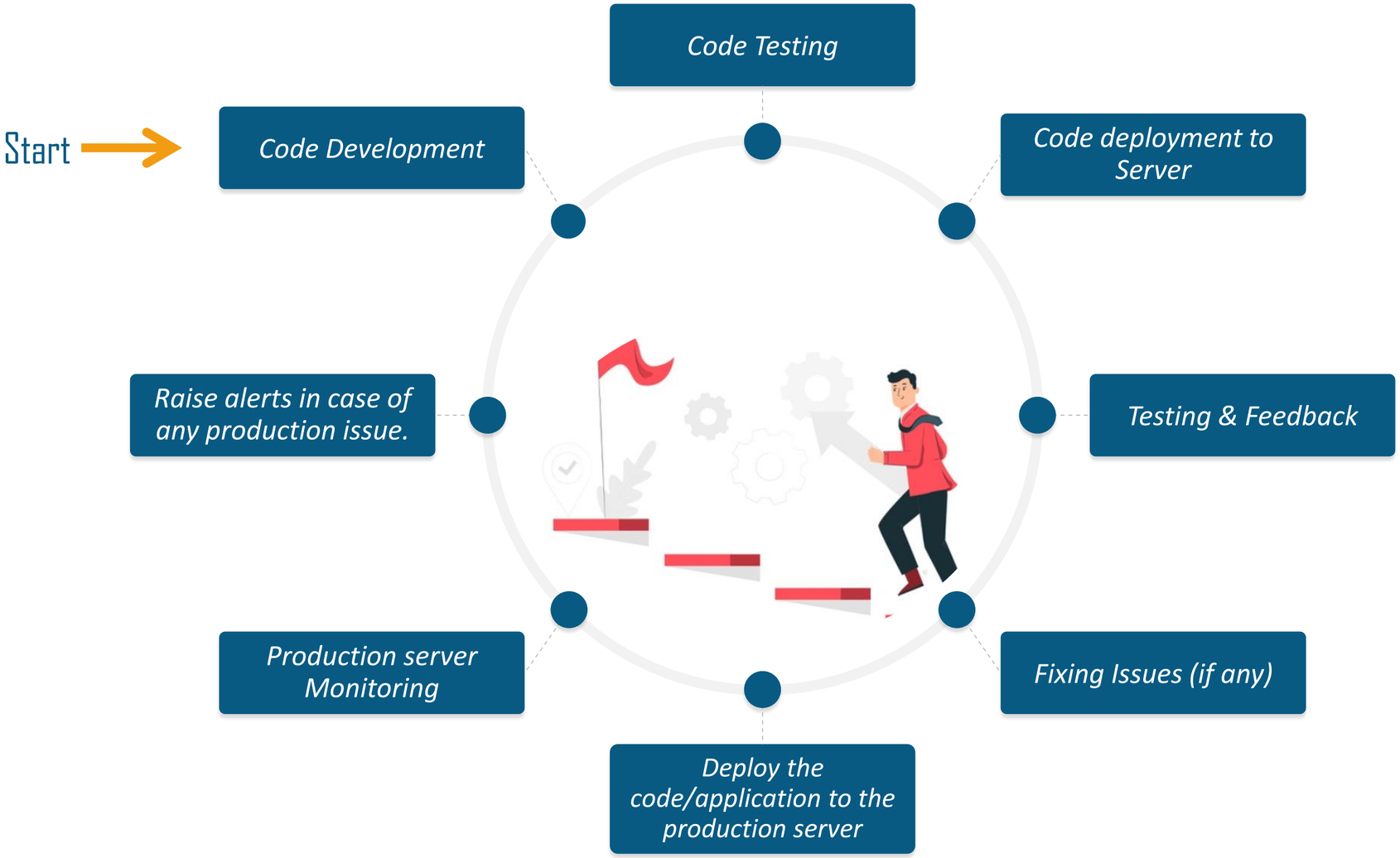
Results in efficient and quicker deployment

edureka!

DevOps Pipeline Components



Pipeline Steps



Introduction to Azure Pipelines

edureka!

Azure Pipeline

Azure Pipeline is an Azure-based cloud service that is used to build, test, and make the projects available to other users.

Works with any language

Python, Java, PHP, Ruby, JavaScript, C#, C++, Go, etc.

Integration

Integrates with:

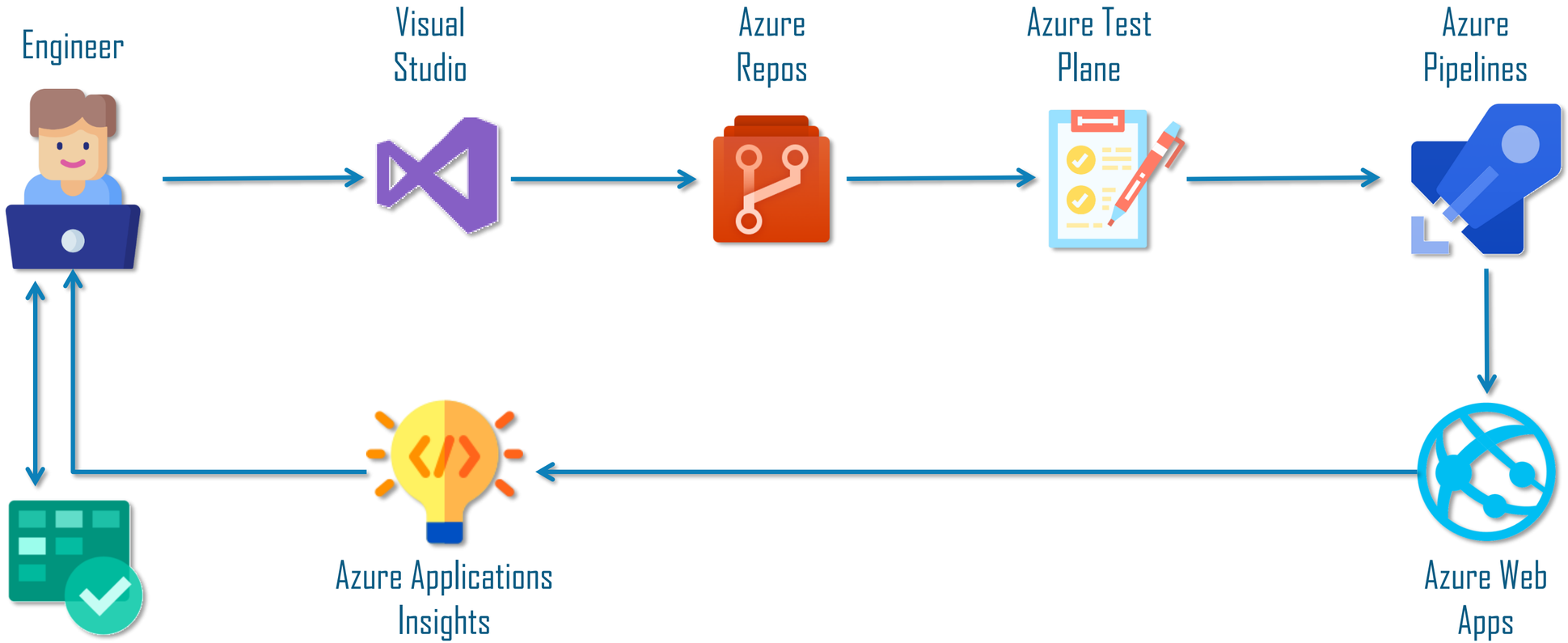
- GitHub
- GitHub Enterprise
- Azure Repos Git
- TFVC
- Bitbucket Cloud
- Subversion

Can be used with many application types

Java, Node.js, Python, .NET, C++, PHP, JavaScript XCode, Go etc.

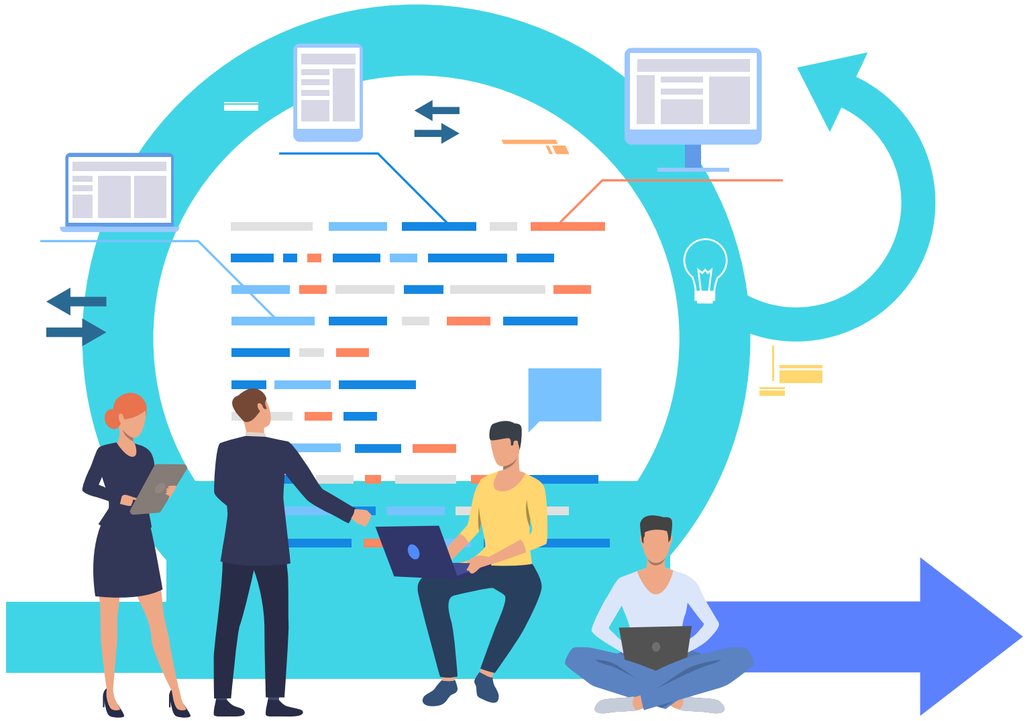
Azure Pipeline Flow

The objective of Pipeline is to automate the process as much as possible and thus reducing the human intervention.



edureka!

Azure Pipeline Advantages



- Helps constantly and consistently test and build the code and ship it to any target
- Provides a quick and safe way to automate building the projects and make them available to users

Azure Pipeline Benefits

Works with any language or platform

Deploys to different types of targets at the same time

Integrates with Azure deployments

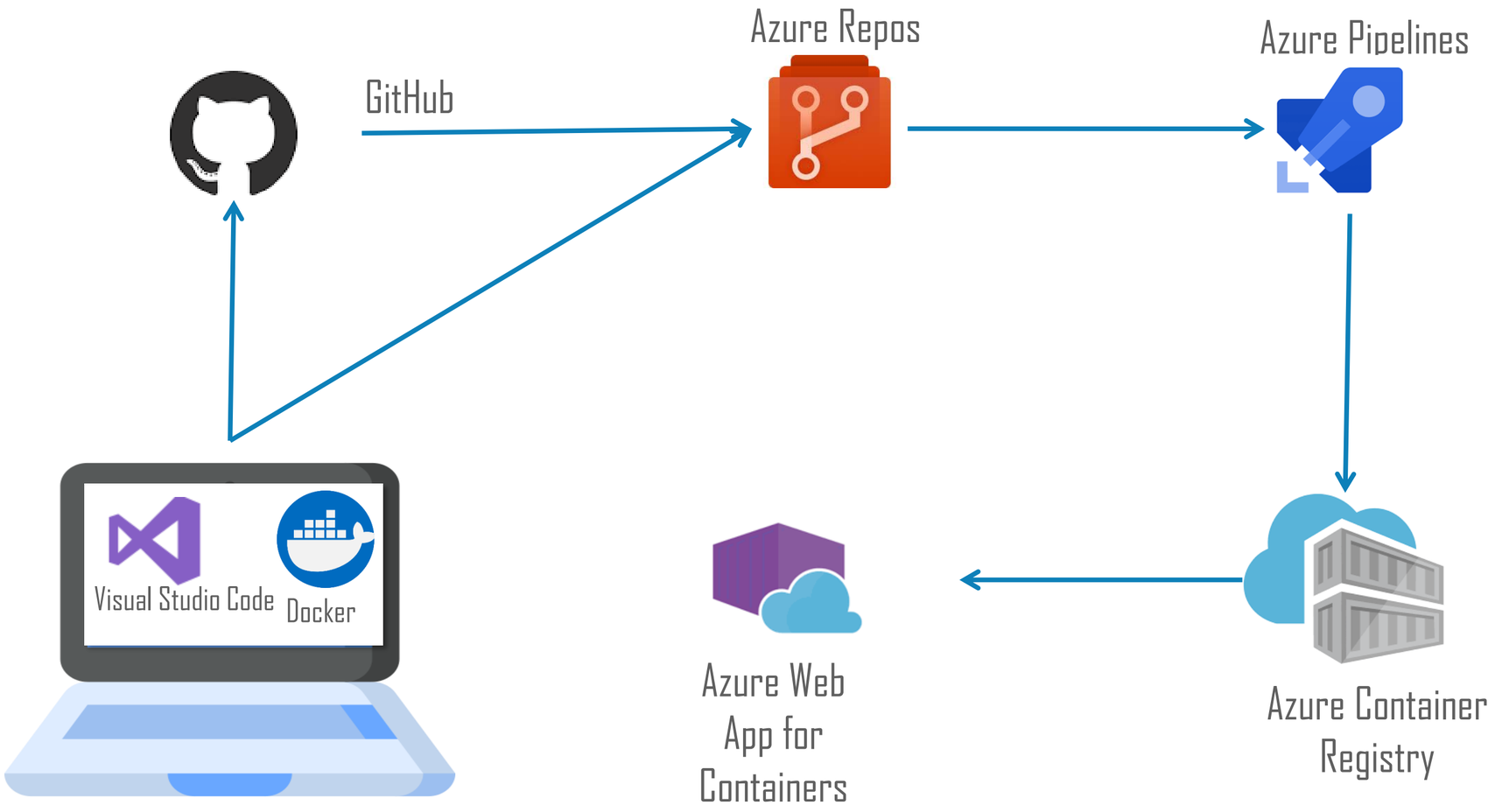
Builds on Windows, Linux, or Mac machines

Integrates with GitHub

Works with open-source projects

edureka!

Azure Pipeline Benefits



Hosted vs. Private Agents

edureka!

Build Agent

A Build Agent is a computing infrastructure with installed agent software to build and deploy the code in Azure pipelines.

Runs one job at a time

Runs build process

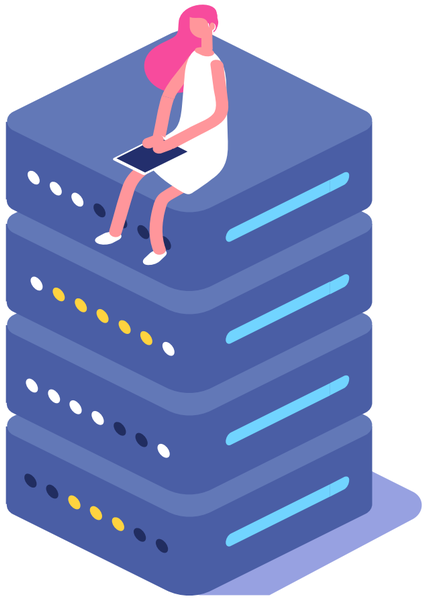
Build Agent Types

Build Agents can run job either on the host machine or in container.

Microsoft Hosted Agents

Self Hosted Agents

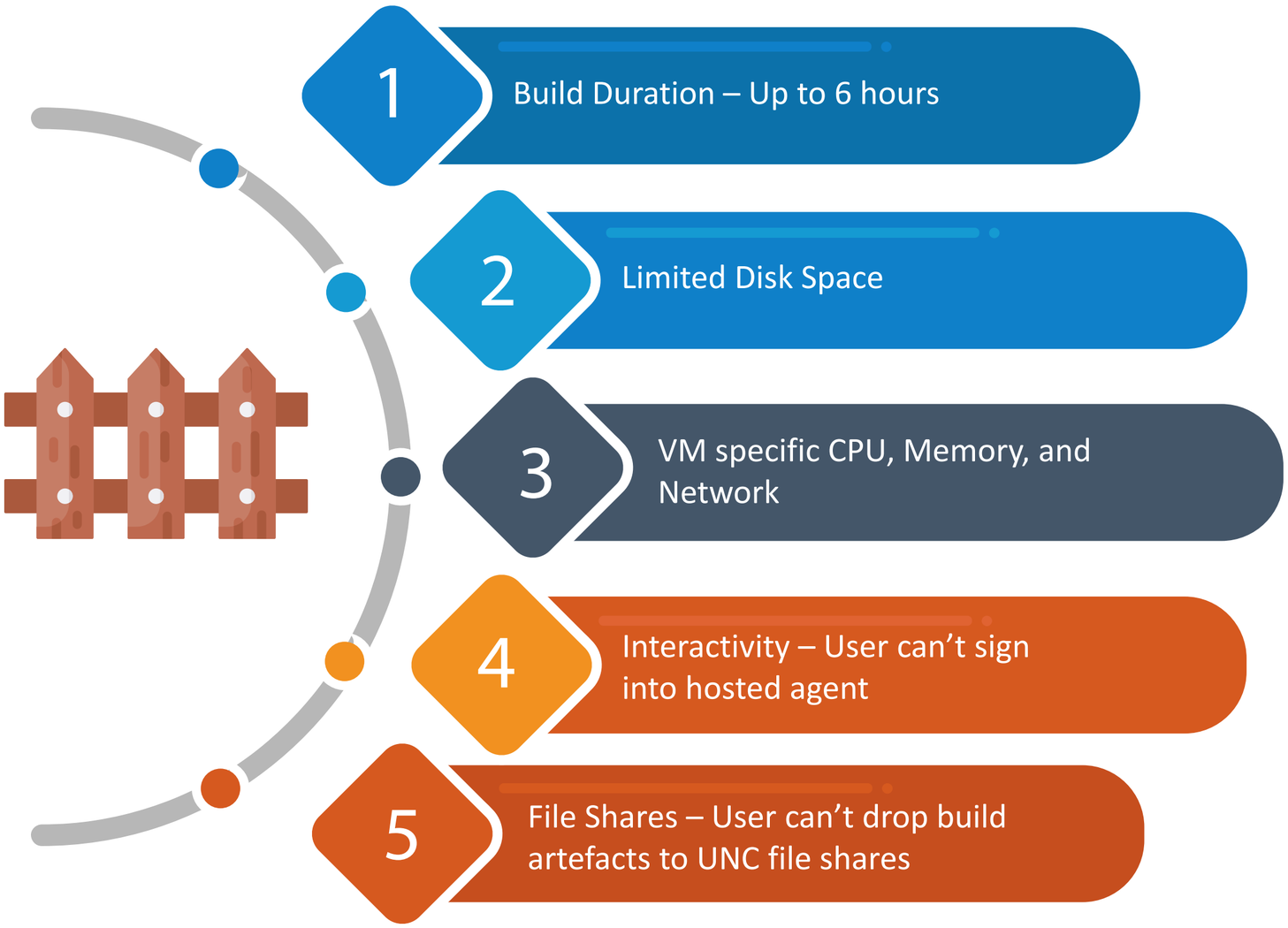
Microsoft Hosted Agents



- With Microsoft-hosted agents, maintenance and upgrades are taken care of for you
- Users have to define the build configuration
- Each time you run a pipeline, users get a fresh virtual machine
- The virtual machine is discarded after one use to free up the resources

👉 This is the simplest way of running the pipeline job.

Limitations of Microsoft Hosted Agent



edureka!

Self-Hosted/Private Agents

Private Agent

A private agent is an agent that users set up and manage on their own.

- Gives more control to the user to install dependent software needed for builds and deployments
- Cache and configuration persist from run to run, thus improving the performance
- Users can specify the software required to build the application
- Users can customize the environment as per application requirements
- Users can use the image or snapshot to create a configured environment
- Image can be used to create as many identical environments as required

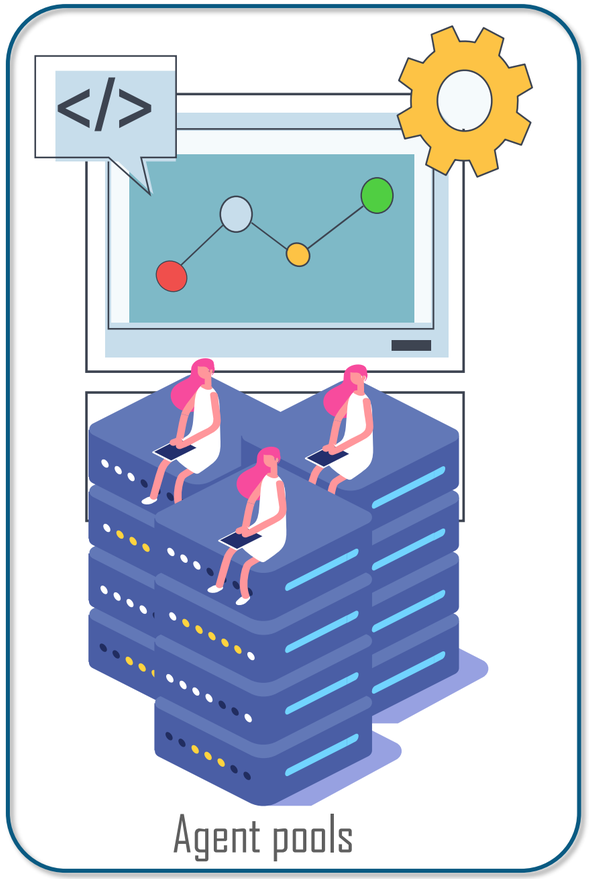
edureka!

Agent Pools

edureka!

Agent Pools

Agent pools contain various agents so that agents can be served on the fly to multiple requests.



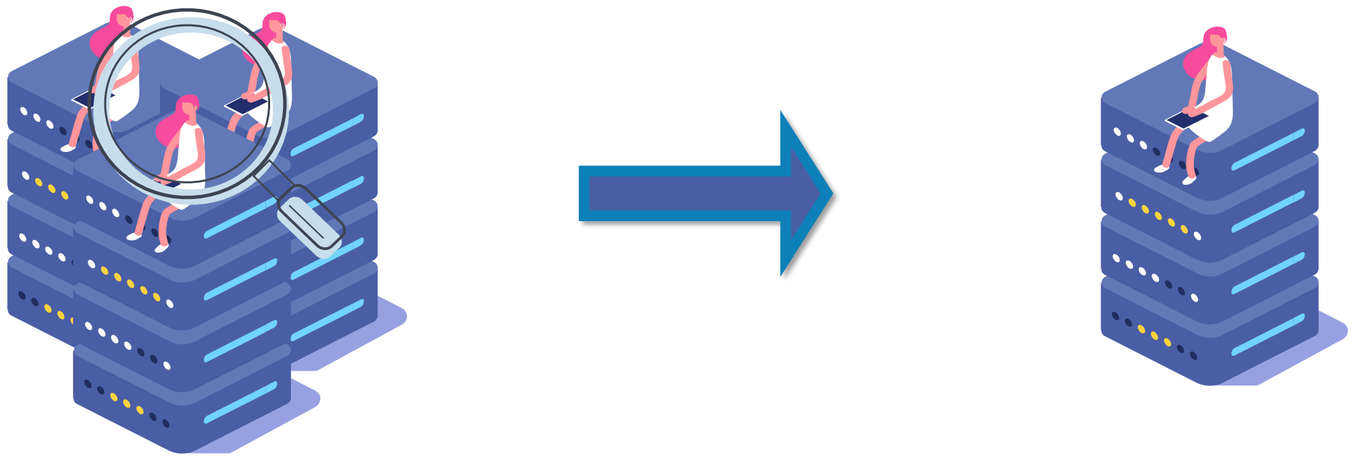
- The Agent pools are required for building the code and deploying using Azure Pipeline
- When the code increases and more people get involved, then we need more agent pools
- When the pipeline runs, then jobs defined in the pipeline executes
- The agent pool's objective is to ensure that the agent is ready to process each build request

edureka!

Agent Selection

When a build is triggered, then the Azure pipeline selects an available agent from the Agent pool

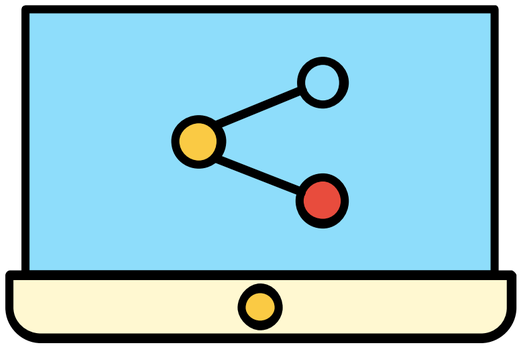
If all agents are busy, the Azure pipeline waits for an agent to process it



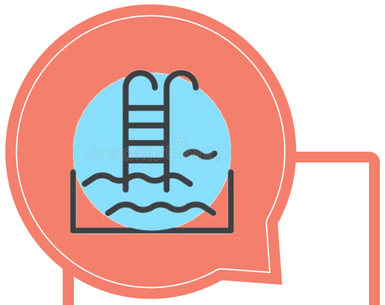
👉 Agent pools can be selected and mentioned in the YAML file for pipeline

Agent Pools Scope

- In Azure Pipelines, the pools are scoped to the entire organization. The user can share the agent machines across all projects
- In Azure DevOps Server, agent pools are scoped to the entire server
- Users can share the agent machines across projects and collections

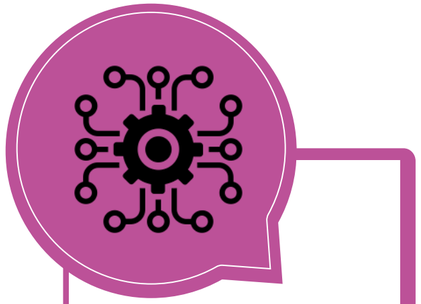


Agent Pools Configuration



When you configure an agent, it is registered with a single pool

When the user creates a pipeline, the user specifies which pool the pipeline will use



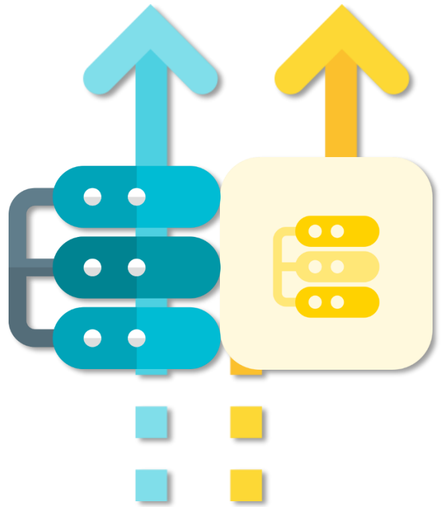
When the user runs the pipeline, it runs on an agent from the pool that meets the pipeline's demands

edureka!

Concurrent Jobs Execution

In Azure Pipelines, users can run parallel jobs on Microsoft-hosted infrastructure or in self-hosted infrastructure.

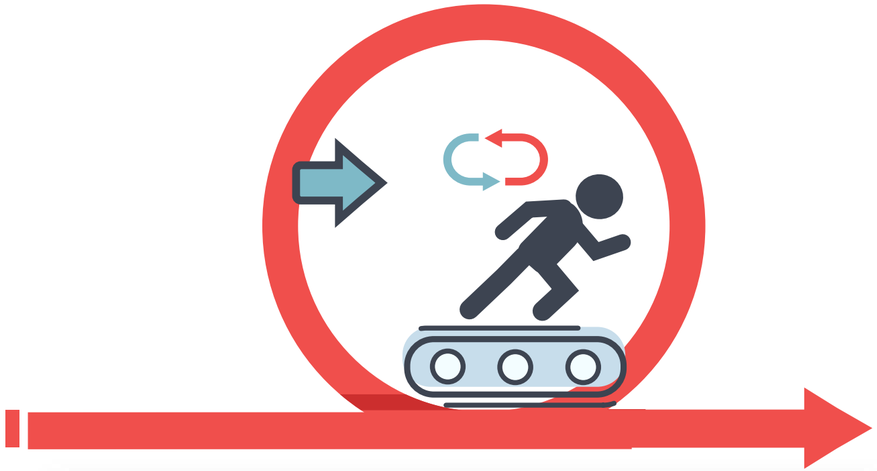
- With both Microsoft hosted agents or self-hosted agents, users can have parallel pipelines executing
- Due to the multiple requests, pipelines can be invoked in parallel
- To achieve this, users or organization need to purchase parallel jobs



Microsoft-Hosted CI/CD and Self-Hosted CI/CD

edureka!

Microsoft-hosted CI/CD



Microsoft-hosted CI/CD

If users want to run jobs on machines that Microsoft manages, they need to use Microsoft-hosted parallel jobs

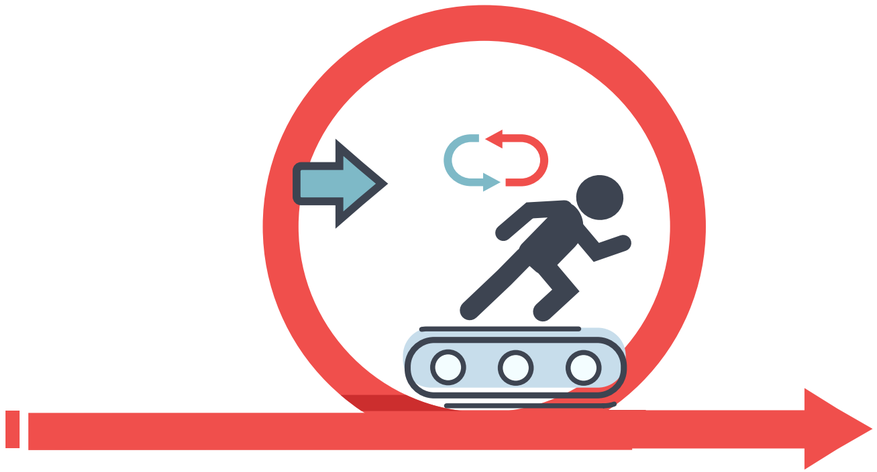
Jobs are run in the pool of Microsoft-hosted agents

When the user creates an Azure Pipeline, the user gets a convenient option to run the jobs using a Microsoft-hosted agent

With Microsoft-hosted agents, you don’t need to take care of the maintenance and upgrades

edureka!

Microsoft-hosted CI/CD



- Each time a user runs a pipeline, the user gets a new VM
- The VM is discarded after one use
- Microsoft hosted agents can run jobs in VM or a container
- Azure Pipelines provide a pre-defined agent pool named Azure Pipelines with Microsoft-hosted agents

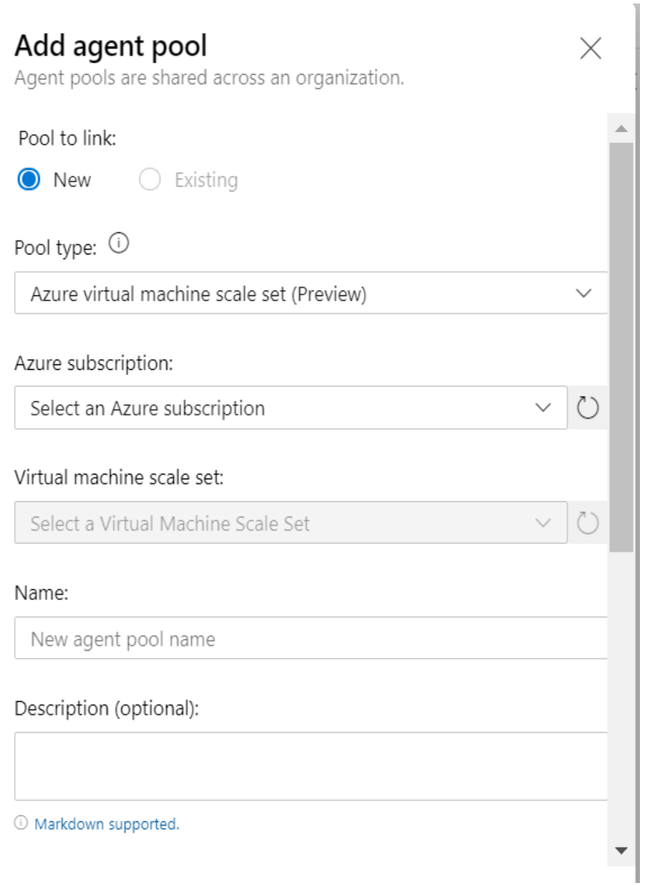
Azure Pipelines Agent Pool: VM Images

The Azure Pipelines agent pool offers several virtual machine images to choose from, including a broad range of tools and software.

Image	Classic Editor Agent Specification	YAML VM Image Label
Windows Server 2019 with Visual Studio 2019	Windows-2019	Windows-latest OT windows-2019
Windows Server 2016 with Visual Studio 2017	Vs2017-win2016	Vs2017-win2016
Ubuntu 20.04	Ubuntu-2014	Ubuntu-20.04
Ubuntu 18.04	Ubuntu-18.04	Ubuntu-latest OR ubuntu-18.04
Ubuntu 16.04	Ubuntu-16.04	Ubuntu-16.04
macOS X Mojave 10.14	macOS-10.14	macOS-10.14
macOS X Cataline 10.15	macOS-10.15	macOS-latest OR macOS-10.15

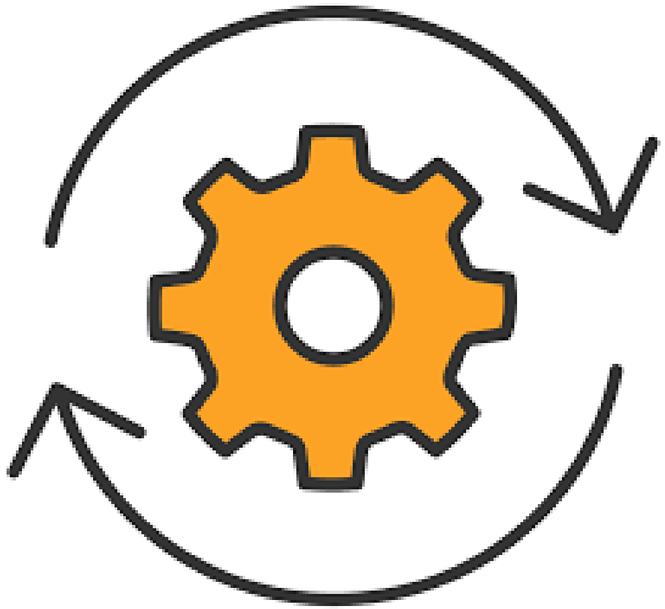
New Agent Creation

- User can create new agents using Microsoft VMs using a virtual machine scale set
- User must have an Azure subscription
- Create a virtual machine scale set in Azure
  - In the snapshot, we use a virtual machine scale set created in Azure as an agent. This will provide the infrastructure that is required for the agent job
  - Use this agent in Pipeline YAML for pool objects



edureka!

Self-hosted CI/CD



Self-hosted parallel jobs enable users to use Azure Pipelines with their own infrastructure to orchestrate builds and releases. This helps in customization of machines to suit requirement

You can assume more control from self-hosted agents for installing dependent software required for your builds and deployments. Also, machine-level caches and configuration persist from run to run, which can boost speed

Azure virtual machine scale set agents can be used as a self-hosted agent to select the virtual machine configuration. This is done to suit the requirement for enhanced speed

Azure DevOps and Open-Source Projects

Azure DevOps supports open-source projects.

The sources can be hosted in different version controls like:

- Azure Repos Git
- Azure Repos TFVC
- GitHub
- GitHub Enterprise Server
- Bitbucket Cloud
- Bitbucket Server
- Subversion

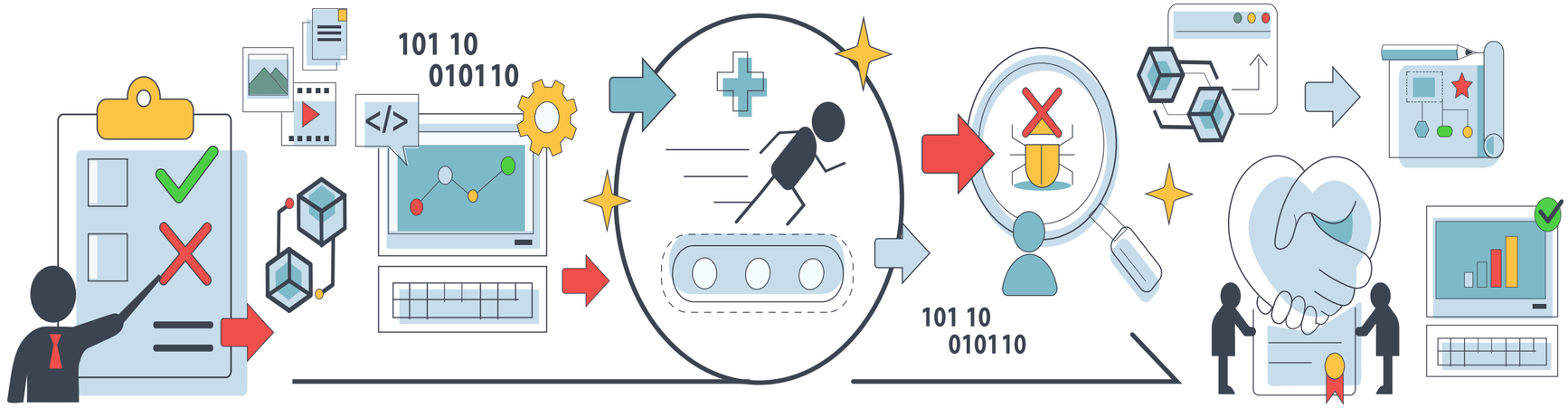
Features and functionalities of Azure DevOps can be integrated and used in the version controls mentioned above.

Azure Pipelines YAML and Visual Designer

edureka!

Azure Pipelines YAML

- Azure pipeline consists of one or more stages that describe the CI/CD pipeline process
- There are various stages of the pipeline:
  - Build
  - Run tests
  - Deploy to production



edureka!

Azure Pipelines YAML

Stage

A stage consists of one or more jobs assigned to the same machine

Job

Job is a series of steps

Built Artifact

The output of Azure pipeline is Built Artifact

Stage and Job hierarchy

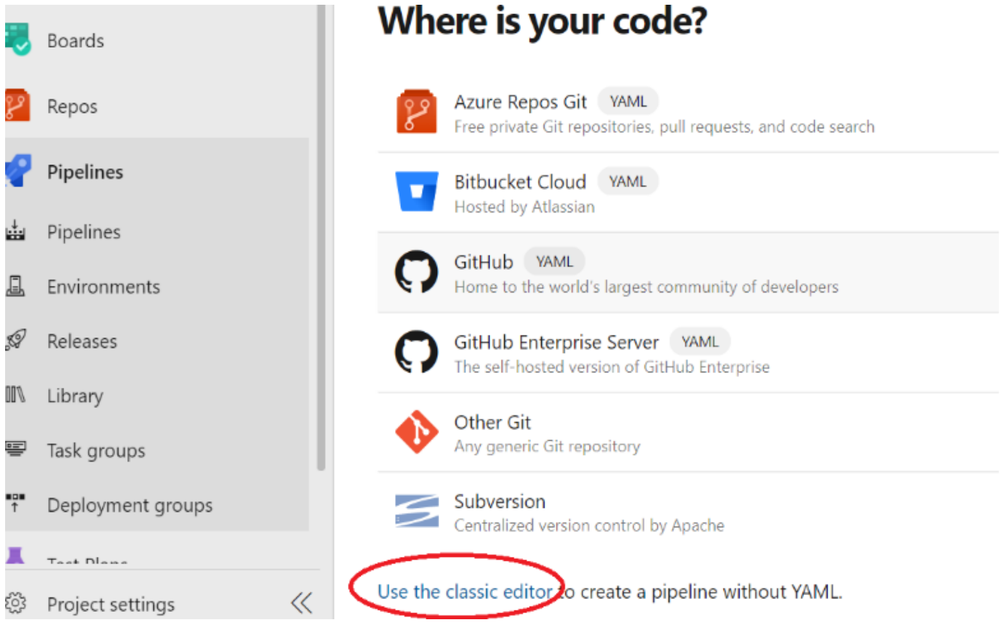
- Pipeline
  - Stage A
    - Job 1
      - Step 1.1
      - Step 1.2
      - Step 1.3
      - Step 1.4
      - …
    - Job 2
      - Step 2.1
      - Step 2.2
      - ..
  - Stage B
    - …

A simple pipeline may not require all these steps. The steps, as shown above, are defined in the YAML file.

edureka!

Visual Designer

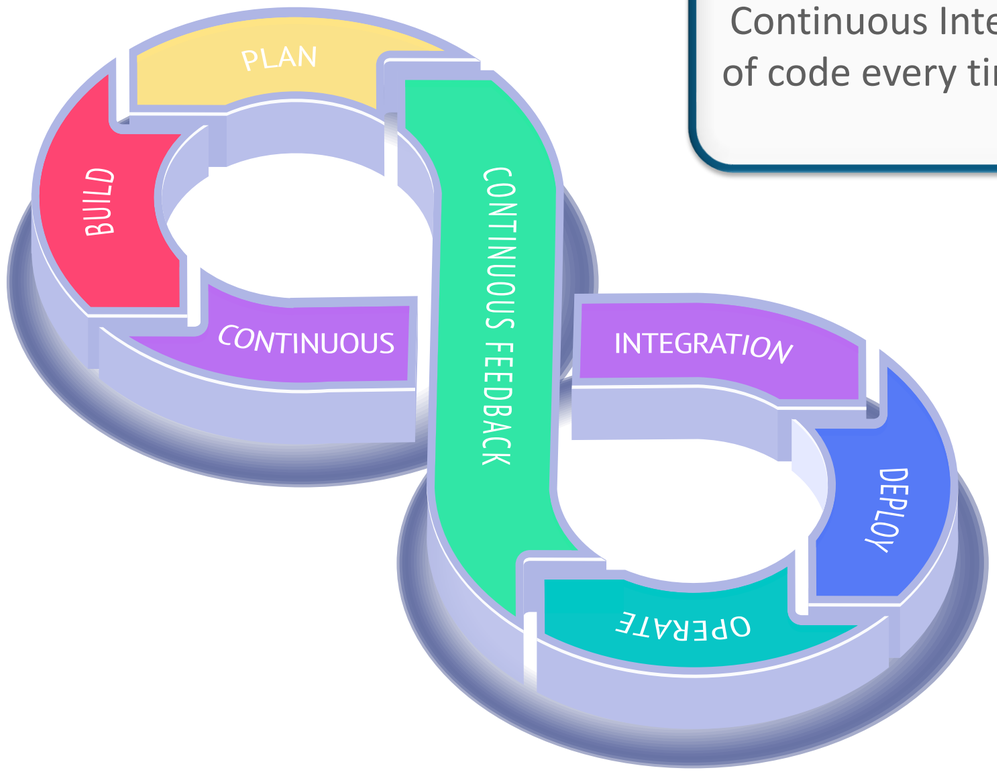
The Azure pipeline can be created using Visual designer as well as shown:



Continuous Integration

edureka!

Introduction to Continuous Integration

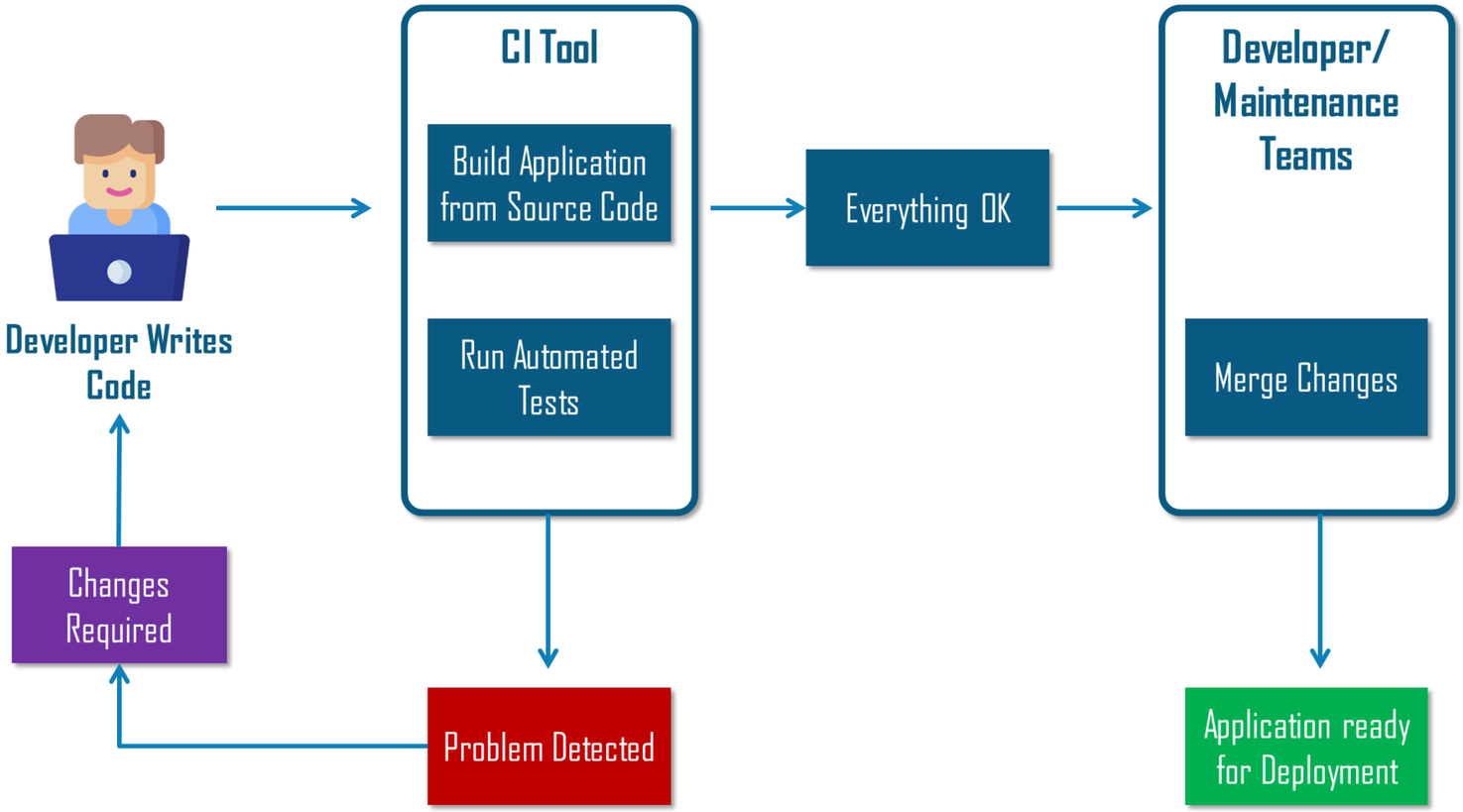


Continuous Integration (CI) is the mechanism to automate the building and testing of code every time a development team member commits version control changes.

Introduction to Continuous Integration

CI encourages the developers to share their code and unit tests by merging the changes into a shared version control repository after completing a small task.

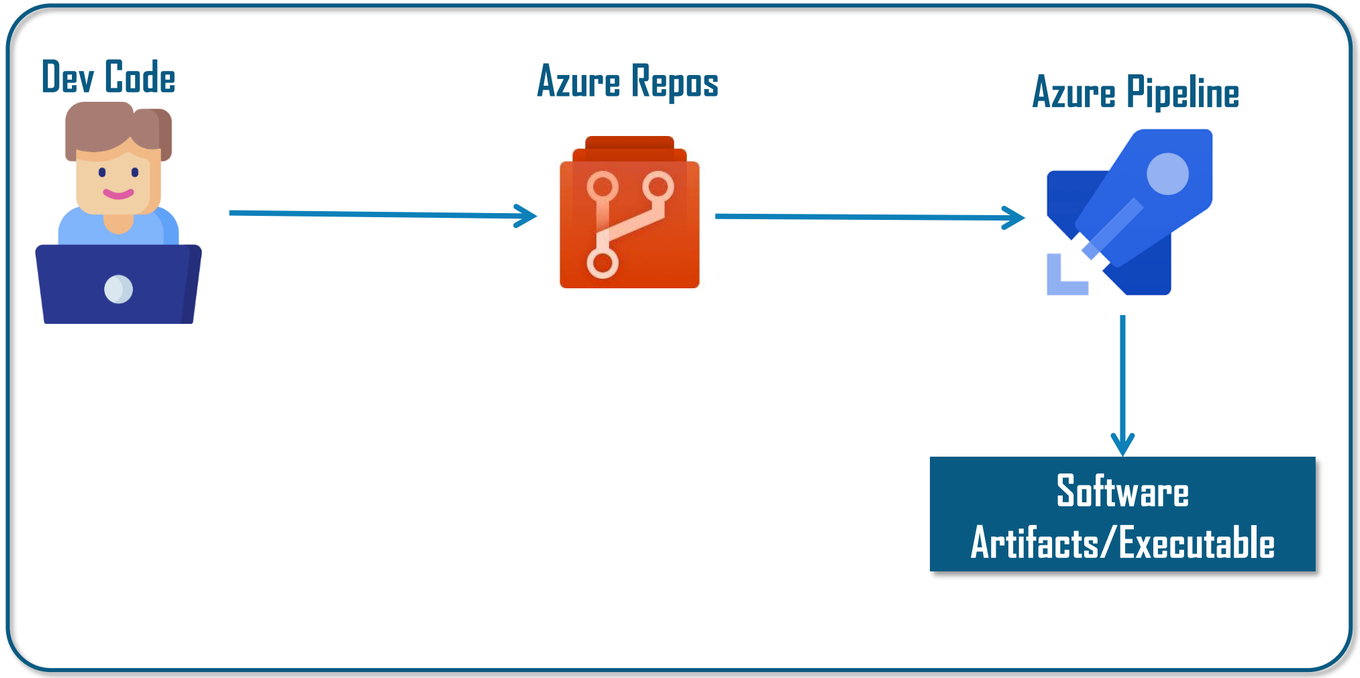
Committing the code triggers to build an automated system. This helps get the latest code from the shared repository to build, test, and validate the full master branch (also known as the trunk or main)



edureka!

Continuous Integration Flow

- Developer checks-in the code in Azure Repos
- Due to Continuous Integration enabled in the Azure Repos Pipeline, the build process is triggered, and subsequent artifact is created
- Subsequent deployment can be configured based on the artifacts or executable generated

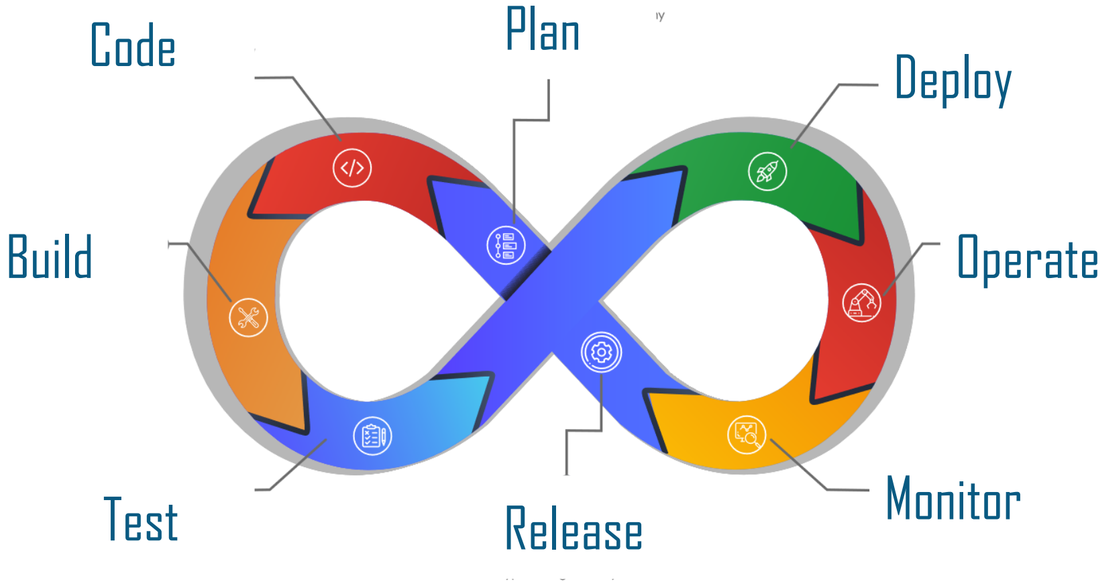


Implementing a Build Strategy

- Build and release strategy is implemented through Azure pipelines.
- Build strategy includes:
  - Build
  - Unit test
  - Code Coverage
  - Including build dependencies as part of the pipeline
  - Deciding about Microsoft agent or self-hosted agent
  - Deployment of the application to deployment servers
  - Multistage build for Development, Testing, Staging, and Production environments
  - Execution of functional and non-functional tests as part of the build process of Azure Pipeline

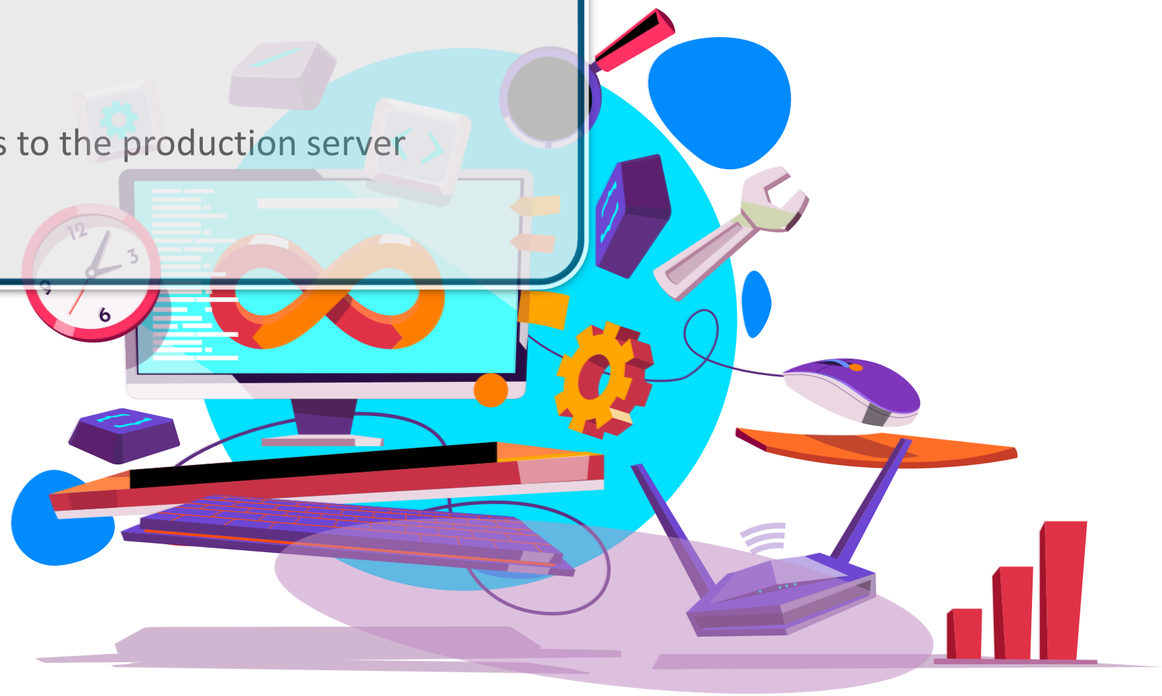
Integrating Azure Pipelines

- The pipeline automates the process from code check-in to build, test, and deployment
- This removes manual errors, provides feedback loops to developers, and enables fast product iterations



Integrating Azure Pipelines

- Azure pipeline is used for the Continuous Integration process covering:
  - Code Check-in
  - Deployment of code
  - Test server
  - Promoting the deployment to the staging server
- If all tests are executed successfully, then the pipeline further deploys to the production server

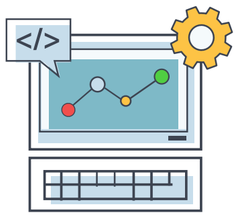


Integrating Azure Pipelines

This process includes

Continuous integration (CI)

- This process involves generation of software artifact/output/executable on code build



Continuous Deployment (CD)

- Azure Pipeline is also used in the continuous deployment process used for application deployment to various servers such as Test Server/Staging Server/ Production Server
- CD sets up a working environment

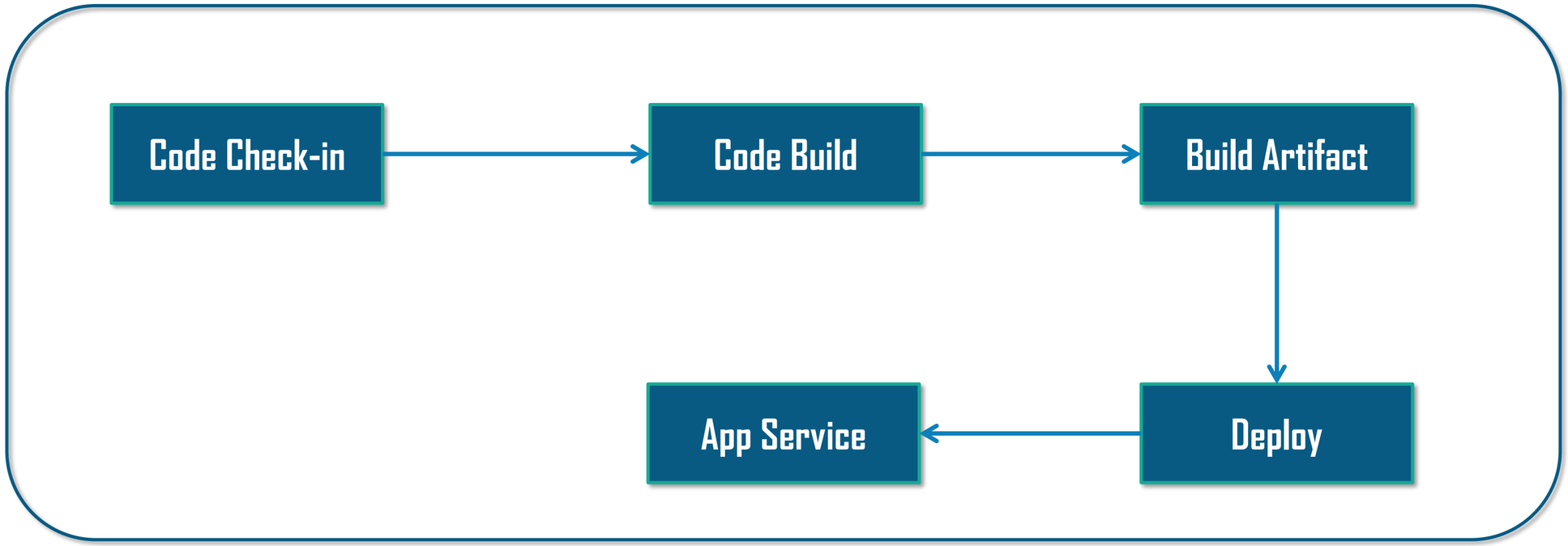
CI Steps

1. Code development by developer
2. Code Check-in to Azure Repos
3. Automatic invocation of Azure Pipeline on code check-in
4. Code build initiation
5. Generation of software Artifact/output/executable

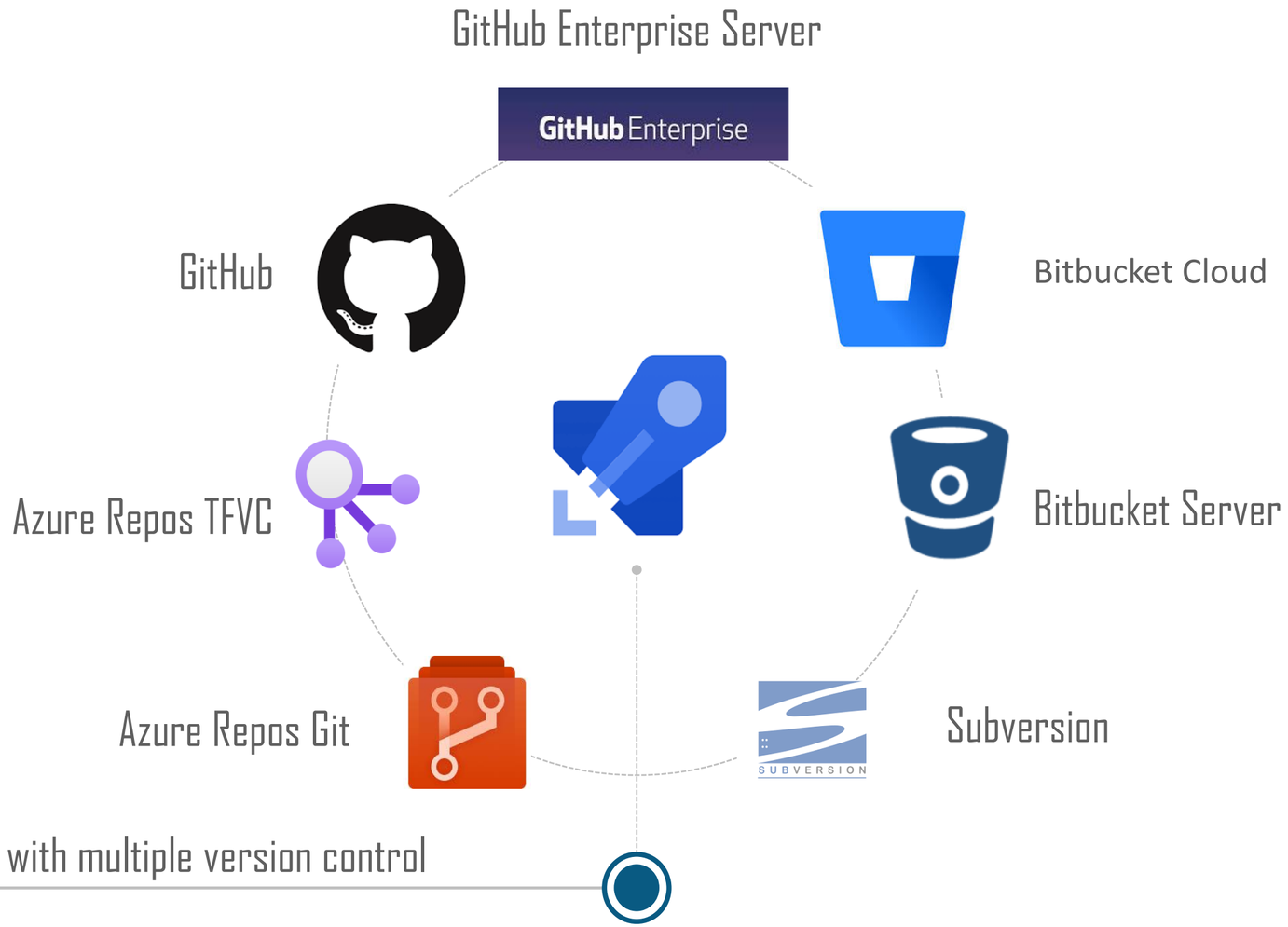
Continuous Integration Process

The Continuous Integration process involves the process from Code Check-in to Build Artifact, and the Continuous Deployment involves Build Artifact to deploy to the app service.

This entire process can be implemented using Azure Pipeline



Integrating External Source Control with Azure Pipelines



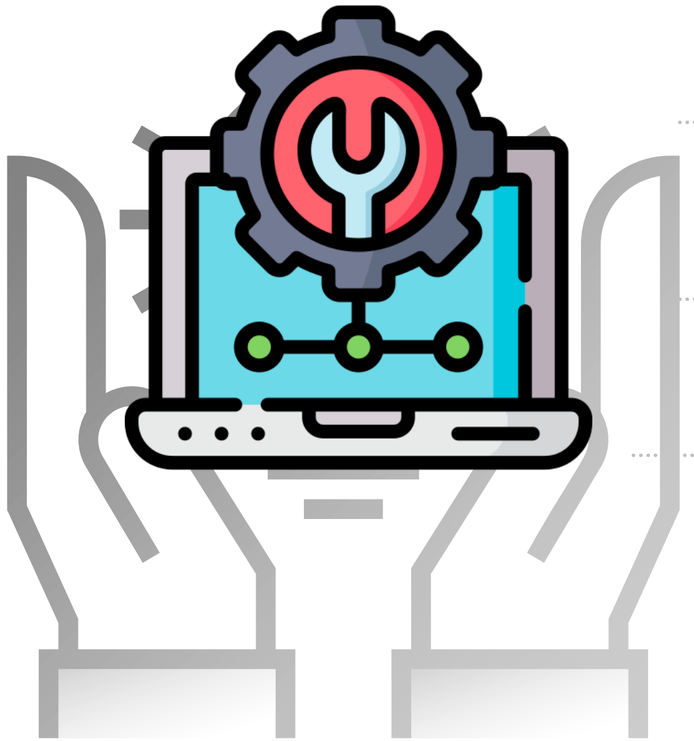
edureka!

Integrating External Source Control with Azure Pipelines

Supported version control systems:

Repository Type	Azure Pipelines (YAML)	Azure Pipelines (Classic Editor)	Azure DevOps Server 2019, TFS 2018, TFS 2017, TFS 2015.4	TFS 2015 RTM
Azure Repos Git	Yes	Yes	Yes	Yes
Azure Repos TFVC	No	Yes	Yes	Yes
GitHub	Yes	Yes	No	No
GitHub Enterprise Server	Yes	Yes	TFS 2018.2 and higher	No
Bitbucket Cloud	Yes	Yes	No	No
Bitbucket Server	No	Yes	Yes	Yes
Subversion	Yes	Yes	Yes	No

Setting Up Private Agents

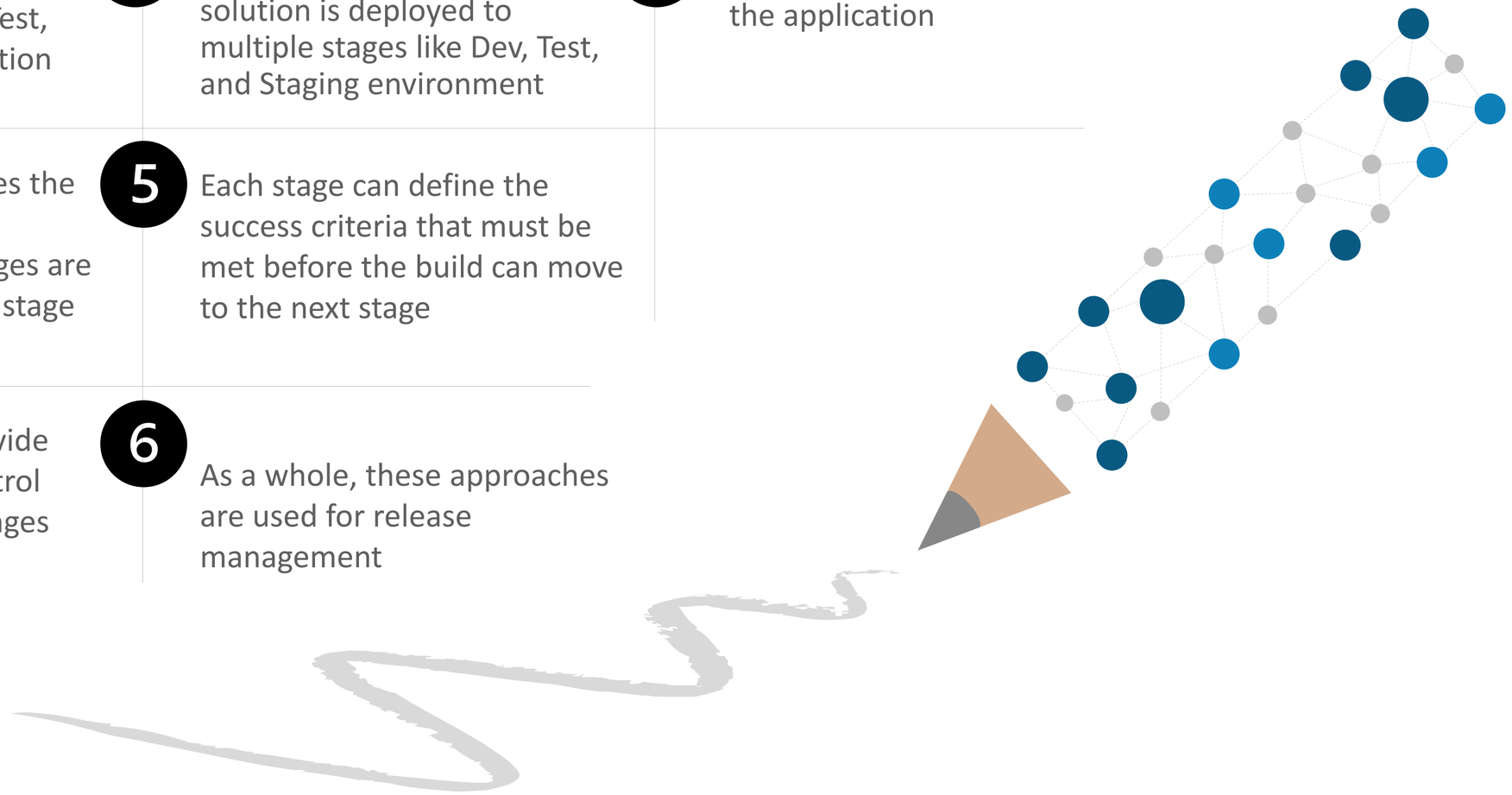


- A private agent can be used in place of a Microsoft hosted agent
- Users can create a private agent for enhanced storage and performance. This agent can be used in the YAML file for the Azure pipeline
- Microsoft hosted agents provide the capabilities for all the job execution for the Azure pipeline

Sometimes the capacity required for the Azure pipeline process may be more than what is offered by Microsoft hosted agents. Due to this reason, the users may need a dedicated agent each, which is higher in capability, processing speed compared to that of Microsoft agent

Analyze and Integrate Docker Multi-Stage Builds

1. The multi-stage build is required to promote the changes from Dev, Test, Staging, and Production
2. After the user defines the stages required, it considers how changes are promoted from one stage to the next
3. Azure Pipelines provide several ways to control how and when changes move through the pipeline
4. As part of a good release management process, the solution is deployed to multiple stages like Dev, Test, and Staging environment
5. Each stage can define the success criteria that must be met before the build can move to the next stage
6. As a whole, these approaches are used for release management
7. Each stage has its instance of the application



edureka!

Analyze and Integrate Docker Multi-Stage Builds

In Azure Pipelines, a condition enables us to run a task or job based on the pipeline's state.

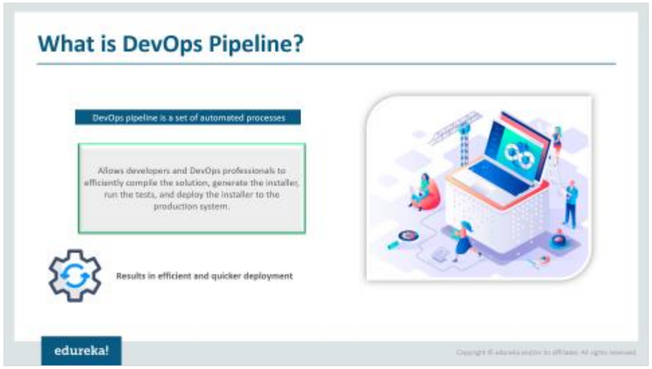
Some of the conditions include:

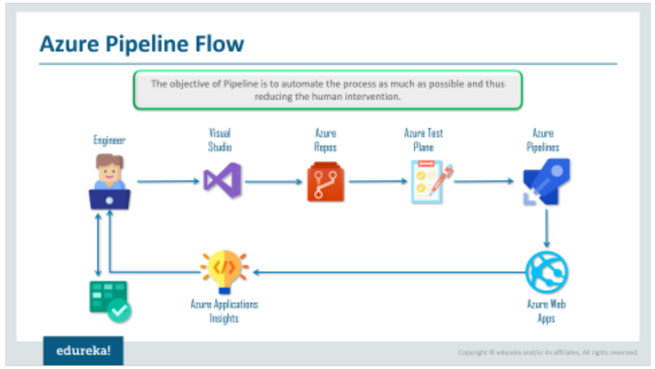
- Only when all previous dependent tasks have succeeded
- Even if a previous dependency has failed unless the run was canceled
- Even if a previous dependency has failed, even if the run was canceled
- Only when a previous dependency has failed
- Custom conditions

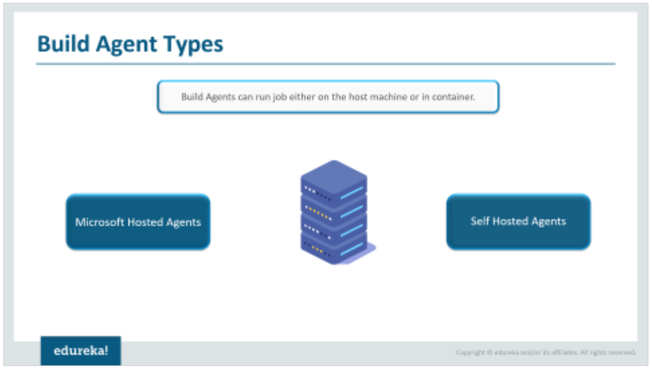
Demonstration

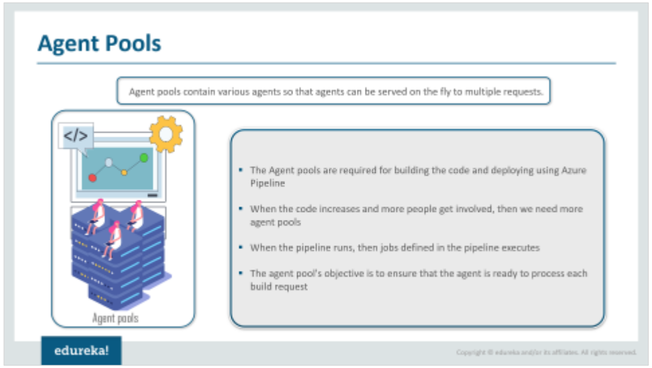
- Demo 1: Implement Continuous Integration with Azure Pipelines
- Demo 2: Integrate External Source Control with Azure Pipelines

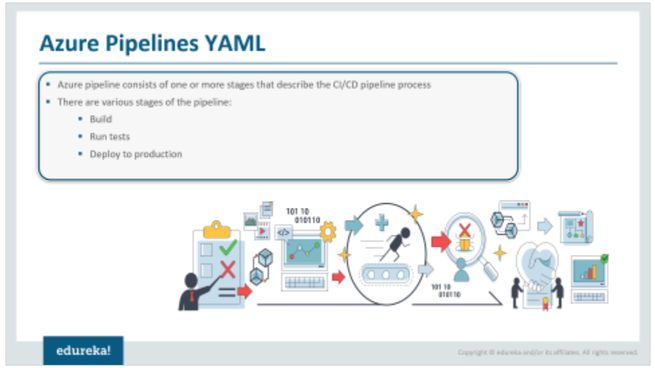
Summary

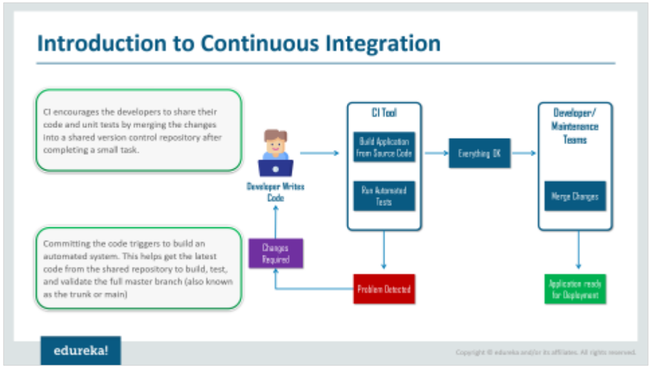












edureka!

edureka!

Thank You

For more information please visit our website
www.edureka.co