Azure Pipelines

Azure DevOps Service

Chakravarthy B

What is Azure Pipelines?

- Automatically builds and test code projects to make them available
- Supports any programming language - including Python, Java, JavaScript, PHP, Ruby, C#, C++, node.js and Go.
- Combines Continuous Integration(CI) and Continuous deliver(CD) to test and build your code and ship to any target - include virtual machines, environments, containers, on-premises and cloud platforms, or PaaS services
- We define Azure Pipelines using YAML Syntax

Why Should I use Azure Pipelines?

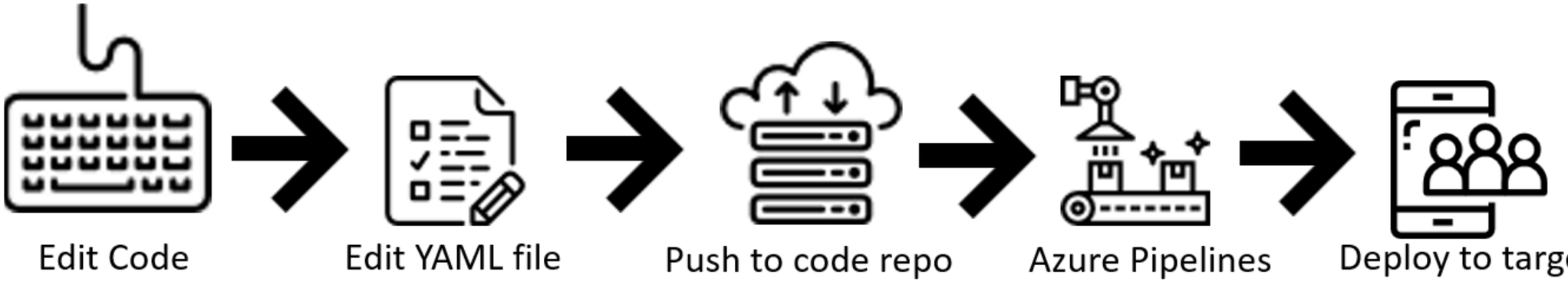
Use Azure Pipelines because it supports the following scenarios:

- 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

Pipeline Concepts

- **Continuous Delivery**: Continuous delivery (CD) is a process by which code is built, tested, and deployed to one or more test and production stages
- **Continuous Integration:** Continuous integration (CI) is the practice used by development teams to simplify the testing and building of code
- **Deployment:** A deployment typically refers to a **deployment job**. A deployment job is a collection of steps that are run sequentially against an environment
- **Environment**: An environment is a collection of resources, where you deploy your application. It can contain one or more virtual machines, containers, web apps, or any service

Azure Pipelines



Define your pipeline in a YAML file called **azure-pipelines.yml** with the rest of your app

YAML Syntax

- YAML is data serialization language
- YAML Stands for yet another markup language or YAML ain’t markup language (a recursive acronym)
- YAML is a popular programming language because it is human-readable and easy to understand
- YAML has features that come from Perl, C, XML, HTML, and other programming languages. YAML is also a superset of JSON, so JSON files are valid in YAML.
- YAML files use a .yml or .yaml extension.

YAML Syntax

- The structure of a YAML file is a map or a list
- Maps allow you to associate key-value pairs. Each key must be unique, and the order doesn't matter
- A list includes values listed in a specific order and may contain any number of items needed
- A list sequence starts with a dash (-) and a space, while indentation separates it from the parent

YAML Syntax

```
---
# An employee record
name: Martin D'vloper
job: Developer
skill: Elite
employed: True
foods:
  - Apple
  - Orange
  - Strawberry
  - Mango
languages:
  perl: Elite
  python: Elite
  pascal: Lame
education: |
  4 GCSEs
  3 A-Levels
  BSc in the Internet of Things
```

YAML vs JSON

YAML

```yaml
simple-property: a simple value

object-property:
  a-property: a value
  another-property: another value


array-property:
  - item-1-property-1: one
    item-1-property-2: 2
  - item-2-property-1: three
    item-2-property-2: 4

# no comment in JSON
```

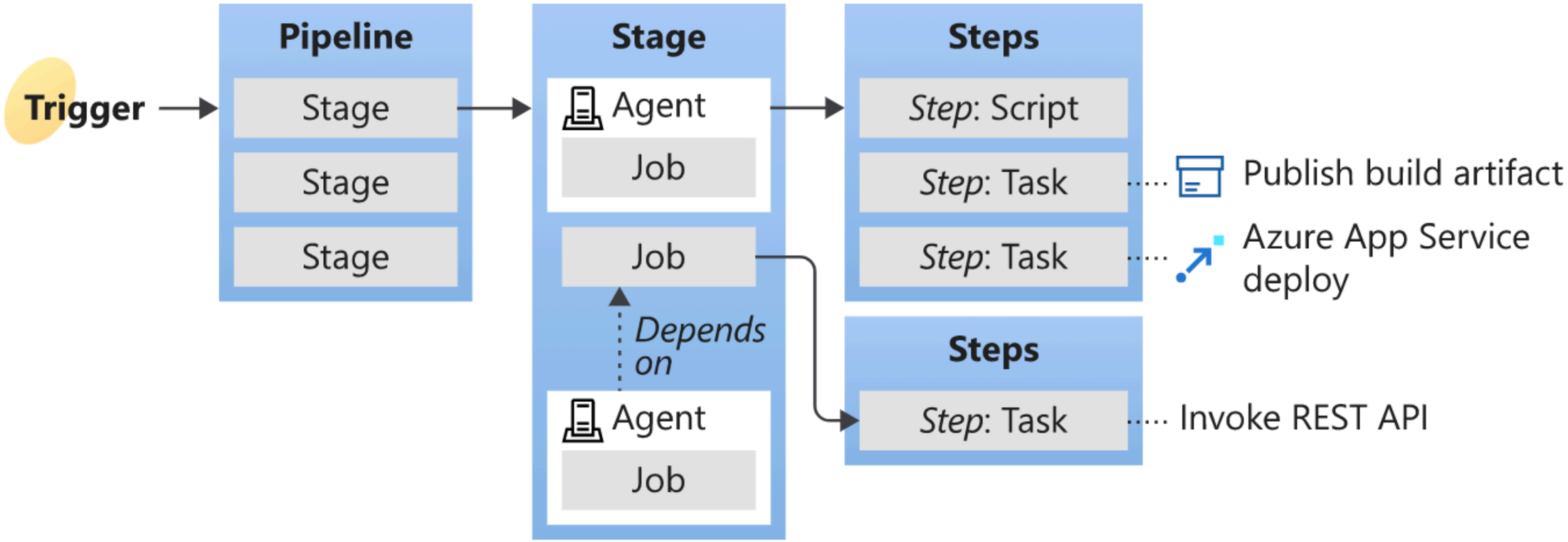
JSON

```json
{
  "simple-property": "a simple value",

  "object-property": {
    "a-property": "a value",
    "another-property": "another value"
  },

  "array-of-objects": [
    { "item-1-property-1": "one",
      "item-1-property-2": 2 },
    { "item-2-property-1": "three",
      "item-2-property-2": 4 }
  ]
}
```

Azure Pipelines Key Concepts



Pipeline Concepts

- **Trigger**: A trigger tells a Pipeline to run
- A **pipeline** is made up of one or more stages. A pipeline can deploy to one or more environments.
- A **stage** is a way of organizing jobs in a pipeline and each stage can have one or more jobs
- Each **job** runs on one agent. A job can also be agentless.
- Each **agent** runs a job that contains one or more steps.
- A **step** can be a task or script and is the smallest building block of a pipeline
- A **task** is a pre-packaged script that performs an action, such as invoking a REST API or publishing a build artifact
- An **artifact** is a collection of files or packages published by a run

Pipeline Concepts

- **Pipeline:** A pipeline defines the continuous integration and deployment process for your app. It's made up of one or more stages.
- **Release**: A release is a versioned set of artifacts specified in a pipeline. The release includes a snapshot of all the information required to carry out all the tasks and actions in the release pipeline
- **Trigger**: A trigger is something that's set up to tell the pipeline when to run. You can configure a pipeline to run upon a push to a repository, at scheduled times, or upon the completion of another build.
- **Run:** A run represents one execution of a pipeline. It collects the logs associated with running the steps and the results of running tests.

Azure Pipeline sample code

```yaml
trigger:
- main

pool:
  vmImage: ubuntu-latest

steps:
- script: echo Hello, world!
  displayName: 'Run a one-line script'

- task: AzureResourceManagerTemplateDeployment@3
  inputs:
    deploymentScope: 'Resource Group'
    azureResourceManagerConnection: 'Azure subscription 1(9cb6023b-c3d7-48ae-a774-b672029e02b6)'
    subscriptionId: '9cb6023b-c3d7-48ae-a774-b672029e02b6'
    action: 'Create Or Update Resource Group'
    resourceGroupName: 'NetworkWatcherRG'
    location: 'South India'
    templateLocation: 'URL of the file'
    csmFileLink: 'https://github.com/kbattiprolu88/AzureRmPipeline-repo/blob/main/CreateWebApp/azuredeploy.json'
    overrideParameters: '-projectName AzureRmPipeline-repo/'
    deploymentMode: 'Incremental'
```