Hello World… Welcome to the ultimate hands-on guide for launching an AWS EC2 instance and deploying a simple application using Azure pipelines. So, let’s get started…

**Overview:**

* What is Azure DevOps Pipeline?
* What is YAML?
* What AWS EC2 Server is?
* What is a User Data script?
* Simple hands-on guide: using the PEM key
* Simple hands-on guide: using User Data script
* Architectural Diagram
* Source Code

**What is Azure DevOps Pipeline?**

Azure DevOps Pipeline is a cloud-based service that enables developers to automate the entire software delivery process. It provides a platform for building, testing, and deploying code across various platforms and environments. It supports integration with multiple version control systems, including Git, Subversion, TFVC, etc. It offers a flexible and customizable YAML-based configuration for defining build and release pipelines.

**What is YAML?**

“YAML Ain’t Markup Language” is a human-readable data serialization format commonly used for configuration files. It is often used in various programming languages and applications where data needs to be stored or transmitted in a structured format. YAML is commonly used for configuration files because of its simplicity and readability. It uses indentation to define structure and relies on whitespace indentation rather than explicit symbols or characters like braces or semicolons. YAML is often used for configuration files in software projects, such as defining build configurations, specifying application settings, or creating deployment pipelines.

**What AWS EC2 Server is?**

EC2, Amazon Elastic Compute Cloud, is a web service that is provided by Amazon Web Services (AWS) that allows users to rent virtual servers, also known as instances, in the cloud. These instances provide scalable computing capacity and enable users to run applications and workloads without investing in and managing physical hardware.

**What is a User Data script?**

User data script refers to a script or a set of commands we provide when launching an instance in a cloud computing environment, such as AWS EC2. The script is executed automatically when the instance starts up and allows to perform various initialization tasks or configurations on the instance.

**Simple hands-on guide: Using the PEM key**

**Task Overview:** The objective is to use an Azure DevOps pipeline to automate launching an EC2 instance and deploying a simple web app onto the server, utilizing a PEM key for secure access.

**Step 1:**Store the AWS access keys.

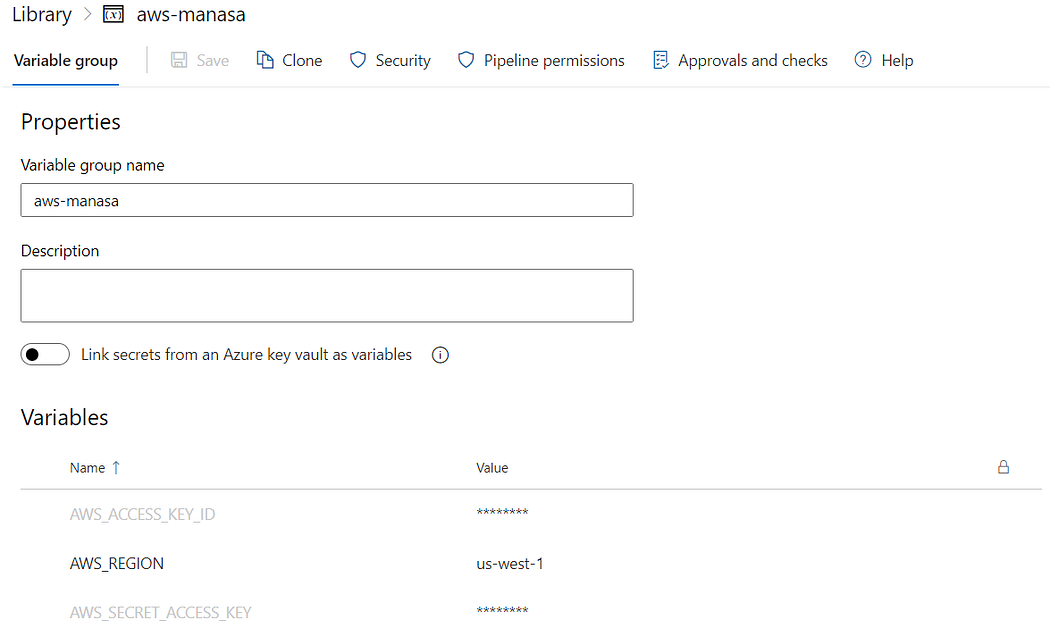
* The AWS access keys must be stored as ‘Pipeline Variables’ or ‘Environment Variables’ for securely managing and accessing AWS resources during pipeline execution.
* These variables can be used at various scopes within Azure Pipelines.
* Below snap is the process of how to store them:

Zoom image will be displayed



Flow: How to store the variables in Azure Pipeline

Zoom image will be displayed



Snapshot of stored variables

**Step 2:** Store the PEM key.

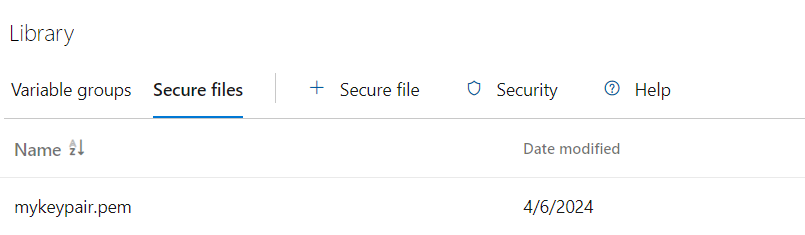
* A PEM (Privacy Enhanced Mail) key, often referred to as a PEM file or PEM certificate, is a file format used to store cryptographic keys and certificates.
* In the context of AWS EC2 instances, a PEM key is typically used for securely accessing the instance over SSH (Secure Shell) protocol.
* Below snap is the process of how to store it:

Zoom image will be displayed



Flow: How to store the PEM key in Azure Pipeline

Zoom image will be displayed



Snapshot of stored PEM key

**Step 3:** Write a pipeline definition in YAML.

* The above pipeline definition launches an EC2 instance in AWS, logs in to the server using the PEM key, and then the app is deployed into the server.

**Step 4:** Create a simple web app.

**Step 5:**Create a pipeline and run.

* I’ve created the pipeline definition file in the Azure Repos itself.
* Below snap is the process of how to create a pipeline and run it:

Zoom image will be displayed

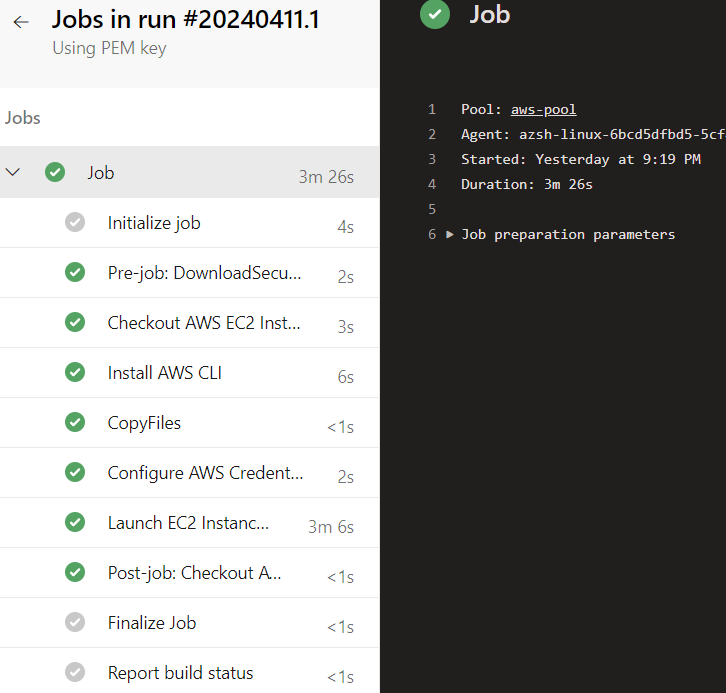


Flow: How to create a pipeline and run it

**Step 6:**Results.

* Upon the successful completion of the pipeline run…

Zoom image will be displayed



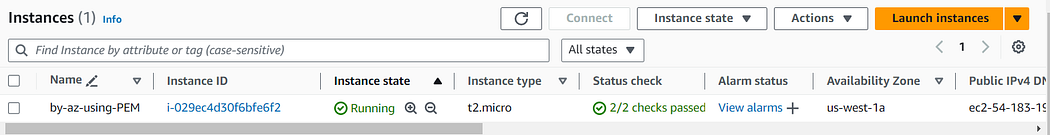
* Check the status EC2 server and access the web app.
* Steps to check the EC2 server:

Zoom image will be displayed



Flow: How to check the state of EC2 Instance

Zoom image will be displayed



* Steps to access the web app:

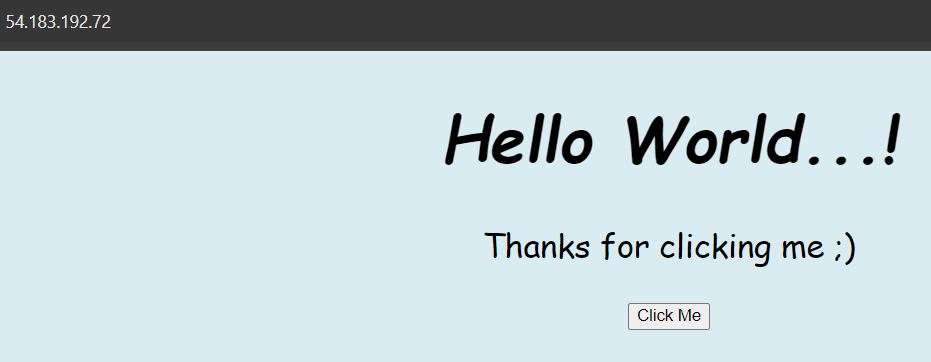


Flow: How to access the web page

Zoom image will be displayed



Zoom image will be displayed



* And it’s done successfully using the PEM key… 🙌

**Simple hands-on guide: using User Data script**

**Task Overview:** The objective is to use an Azure DevOps pipeline to automate launching an EC2 instance and deploying a simple web app onto the server, using a User Data script.

**Step 1:**Store the AWS access keys.

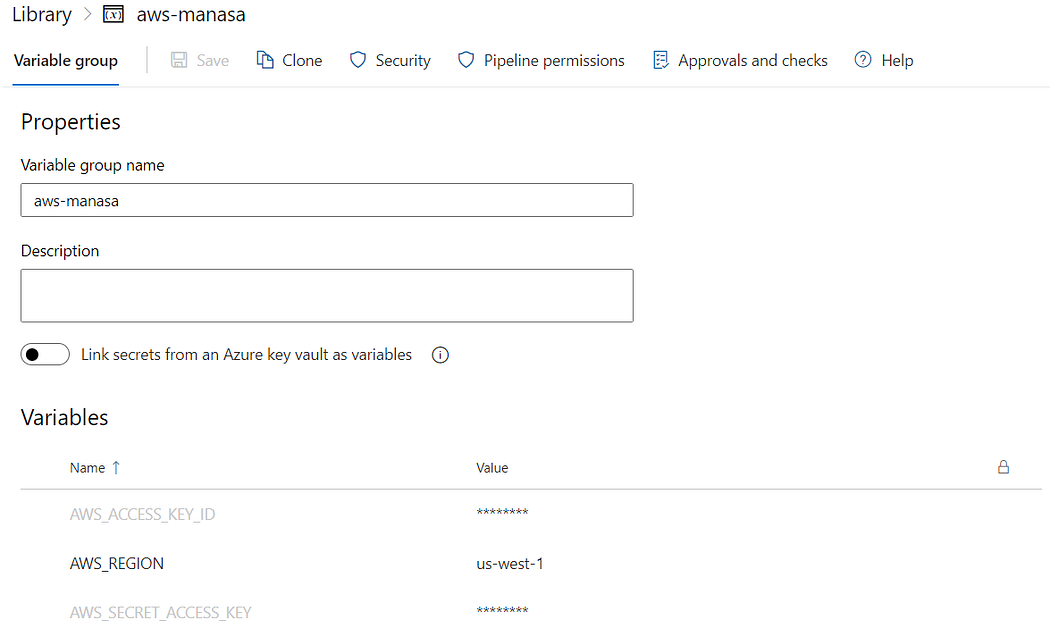
* The AWS access keys must be stored as ‘Pipeline Variables’ or ‘Environment Variables’ for securely managing and accessing AWS resources during pipeline execution.
* These variables can be used at various scopes within Azure Pipelines.
* Below snap is the process of how to store them:

Zoom image will be displayed



Flow: How to store the variables in Azure Pipeline

Zoom image will be displayed



Snapshot of stored variables

**Step 2:**Write a User Data script.

* The script above installs, starts, and enables the Apache service, while also writing the application code to the server.

**Step 3:**Write a pipeline definition in YAML.

* The provided pipeline definition launches an EC2 instance in AWS, then accesses the server and deploys the application using a user data script.

**Step 4:**Create a pipeline and run.

* I’ve created the pipeline definition file in the Azure Repos itself.
* Below snap is the process of how to create a pipeline and run it:

Zoom image will be displayed

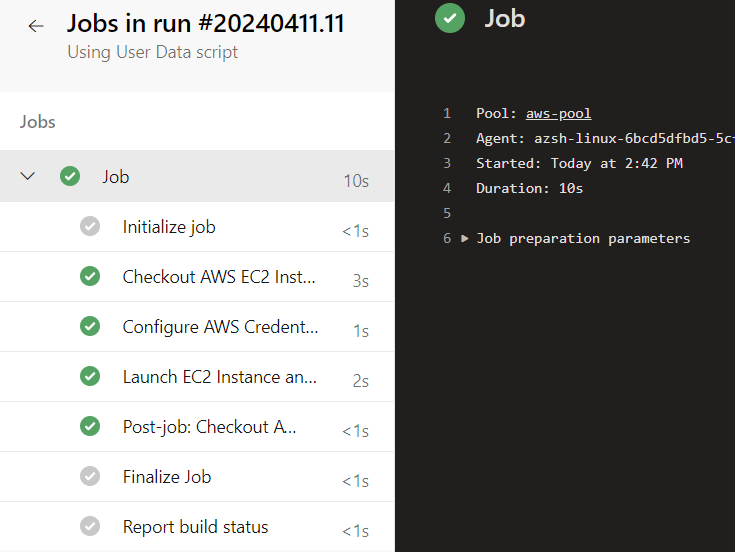


Flow: How to create a pipeline and run it

**Step 5:** Results.

* Upon the successful completion of the pipeline run…

Zoom image will be displayed



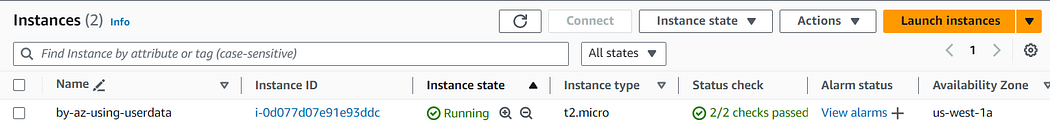
* Check the status EC2 server and access the web app.
* Steps to check the EC2 server:

Zoom image will be displayed



Flow: How to check the state of EC2 instance

Zoom image will be displayed



* Steps to access the web app:



Flow: How to access the web page