C:\Users\traecit2\Desktop\excelsoft_logo.png

**AWS - Image Pipeline Documentation**

**Description of the Document -**

This document provides a brief description to create an Image Pipeline in AWS using multiple resources and services by the AWS provider.

**Note:** This document is the sole property of EXCELSOFT, the contents of this document shall not be reproduced either partly or wholly without appropriate approvals from EXCELSOFT.

|  |  |
| --- | --- |
| **Document Details** | |
| **Type Of Information** | **Document Data** |
| **Document Title** | AWS - Image Pipeline |
| **Document Reference Number** | ES/RE/1 |
| **Document Owner** | ES/RE Team |
| **Security Classification** | Internal Use |
| **Review Frequency** | Daily |

|  |  |  |  |
| --- | --- | --- | --- |
| **Document Approval** | | | |
| **Name** | **Role** | **Signature** | **Date** |
| Pavan Prasad R | Lead Engineer | Pavan Prasad | 20 MAY 2025 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Version Control Details** | | | |
| **Version**  **Number** | **Revision Date** | **Prepared by** | **Nature of Change** |
| **1.0** | 20 MAY 2025 | Nikhil MG | Initial release |

Table of Contents

1. **OBJECTIVE ……………………………………………………………………**
2. **PREREQUISITE ………………..……………………………………………**
3. **Step-by-step guide ……………………………………..………………….…**

# OBJECTIVE

The purpose of this document is to provide a comprehensive and practical guide on integrating **EC2 Image Builder** and **AWS CodeDeploy** to automate the creation, validation, and deployment of secure, up-to-date, and consistent Amazon Machine Images (AMIs) across environments. This includes:

* Capturing every step in configuring EC2 Image Builder pipelines to generate golden images.
* Documenting IAM roles, components, and lifecycle hooks used by CodeDeploy for application delivery.
* Demonstrating how these services support CI/CD workflows and enable scalable, repeatable, and secure deployments.
* Providing implementation examples, best practices, and troubleshooting tips to empower DevOps engineers and cloud administrators.

# Prerequisites

* 1. **AWS Account :** A valid AWS account with administrative privileges or specific permissions to use EC2, Image Builder, CodeDeploy, IAM, and S3.
  2. **IAM Roles and Policies :** 
     1. **For Ec2 Image builder:**  An **Image Builder Service Role** (ec2imagebuilder.amazonaws.com) with permissions to create EC2 instances, build and test images, access S3 logs, and publish AMIs. An **Instance Profile Role** for build and test instances (attached to EC2 instances launched during image building).
     2. ****For CodeDeploy**:** A **CodeDeploy Service Role** (codedeploy.amazonaws.com) with permission to deploy applications.An **EC2 Instance Profile Role** with codedeploy-agent permissions, AmazonEC2RoleforAWSCodeDeploy, and access to CodeDeploy, S3, CloudWatch, and EC2.
  3. **VPC and Networking :** A configured VPC with, Public/private subnets and route tables.Internet Gateway (for downloading packages during image building).Security Groups allowing necessary ports (e.g., SSH, HTTP/S, CodeDeploy agent communication).
  4. **Amazon S3 :** A versioned S3 bucket to store the artifact/packages/executables.
  5. **Code Deploy Agent :** Pre-installed in base AMI or installed via a script during the build process (in EC2 Image Builder).

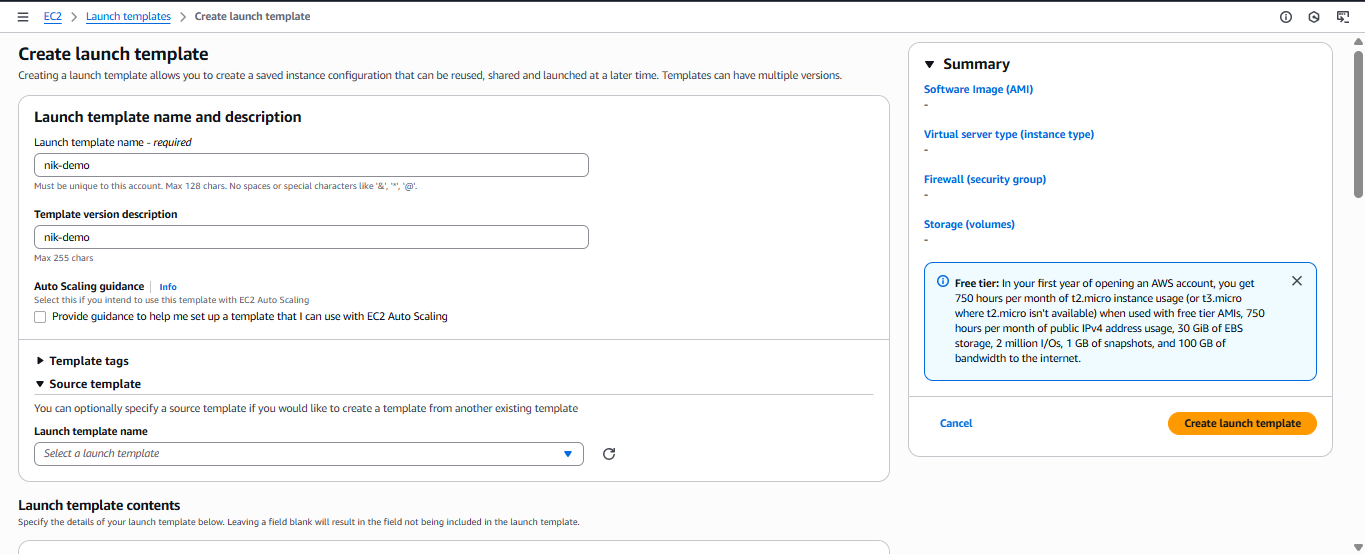
# Step-by-step guide

Once the required roles are created and validated, we’ll create a Image pipeline as mentioned below.

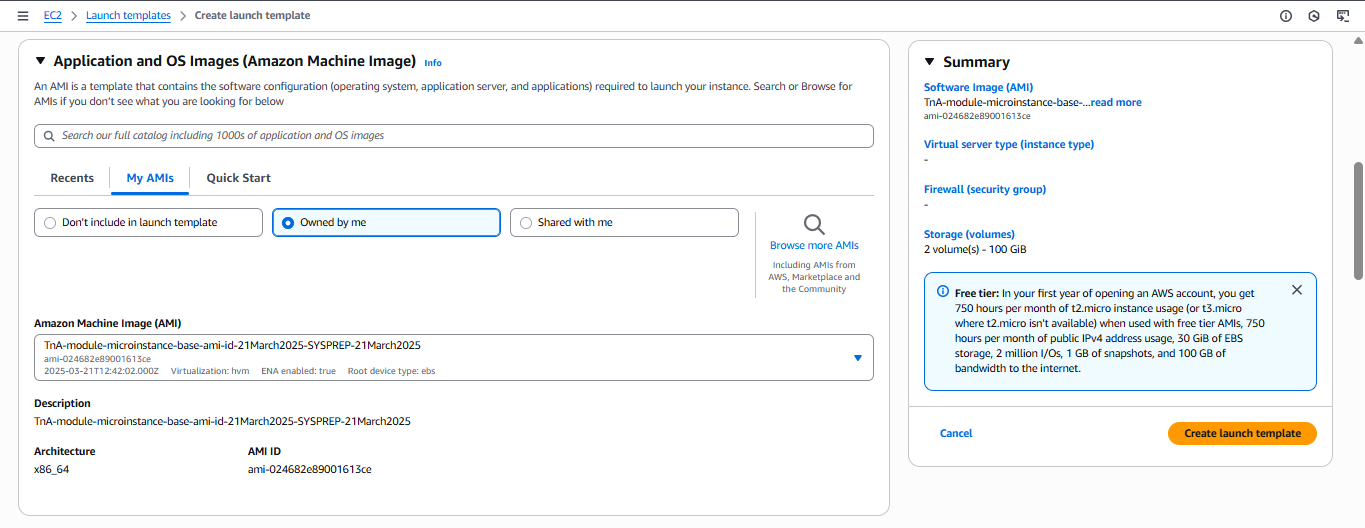
To create an image pipeline we need to create few things prior to the pipeline setup and those are Launch Template, Build component and Image recipe, Infrastructure Configuration, Distribution Settings.

**Launch Template** : - A **Launch Template** is a **reusable blueprint** that defines how your EC2 instances should be launched. It has several configuration settings like AMI ID, Instance Type , Key Pair, Security Groups, User data script, Storage/Volumes, tags, IAM roles, Network settings.

Go to AWS search bar -> Launch Template -> Create a Launch Template



Scroll down and in the AMI section choose the base AMI prepared earlier,

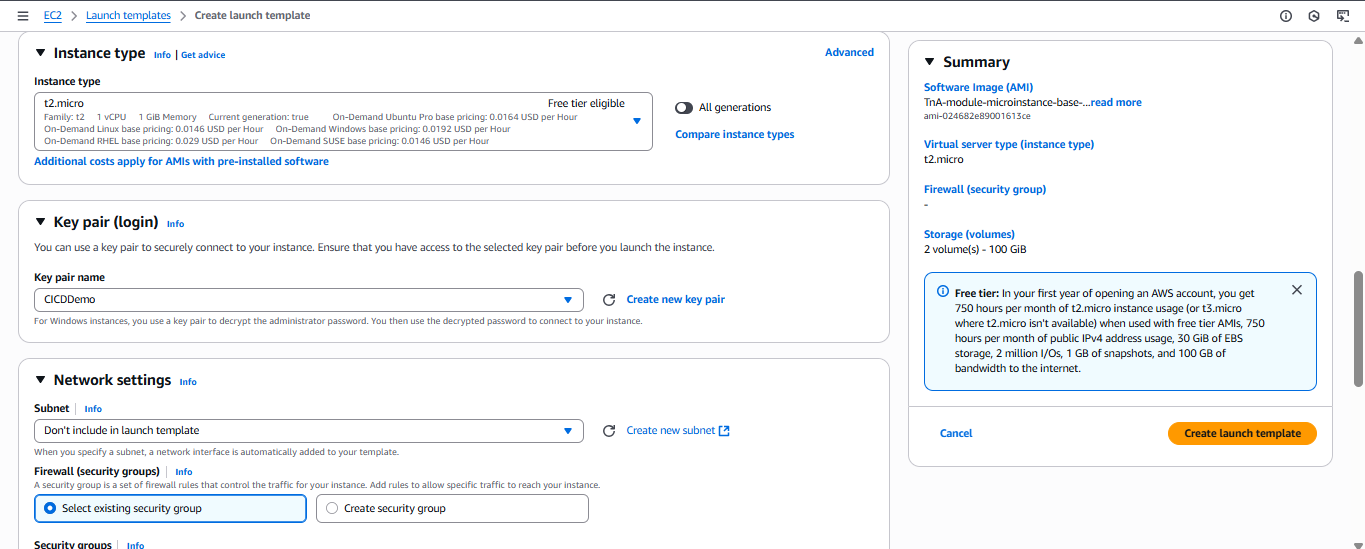


Scroll down and in the

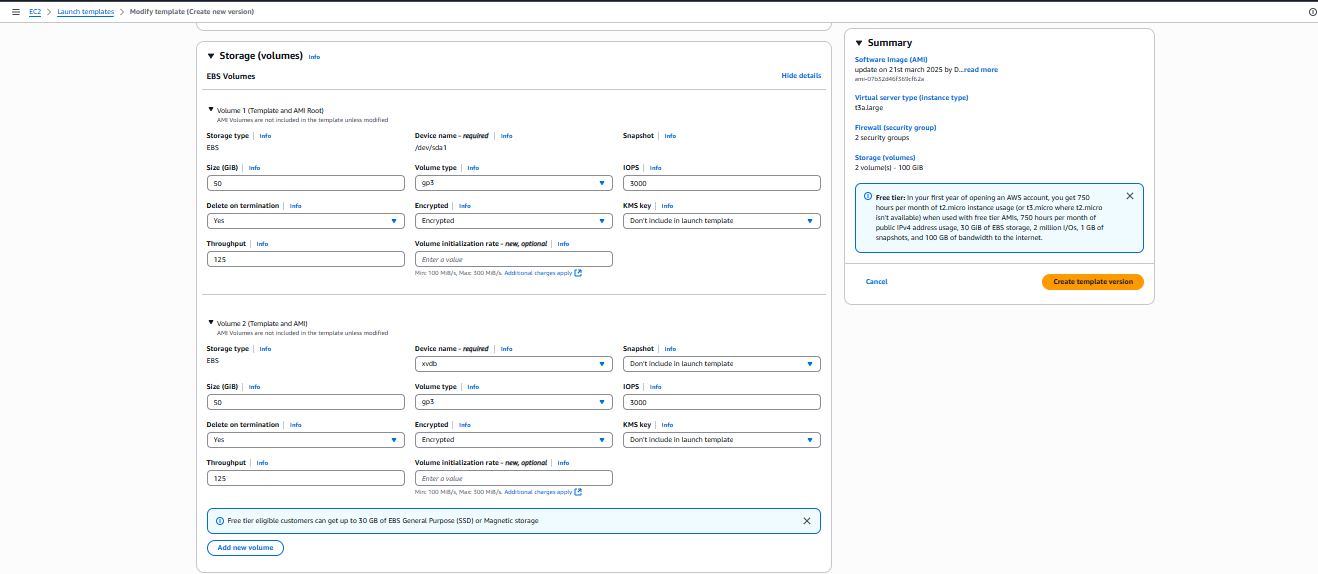
Instance type section choose the type as per requirement,

Key-pair section choose the appropriate key-pair

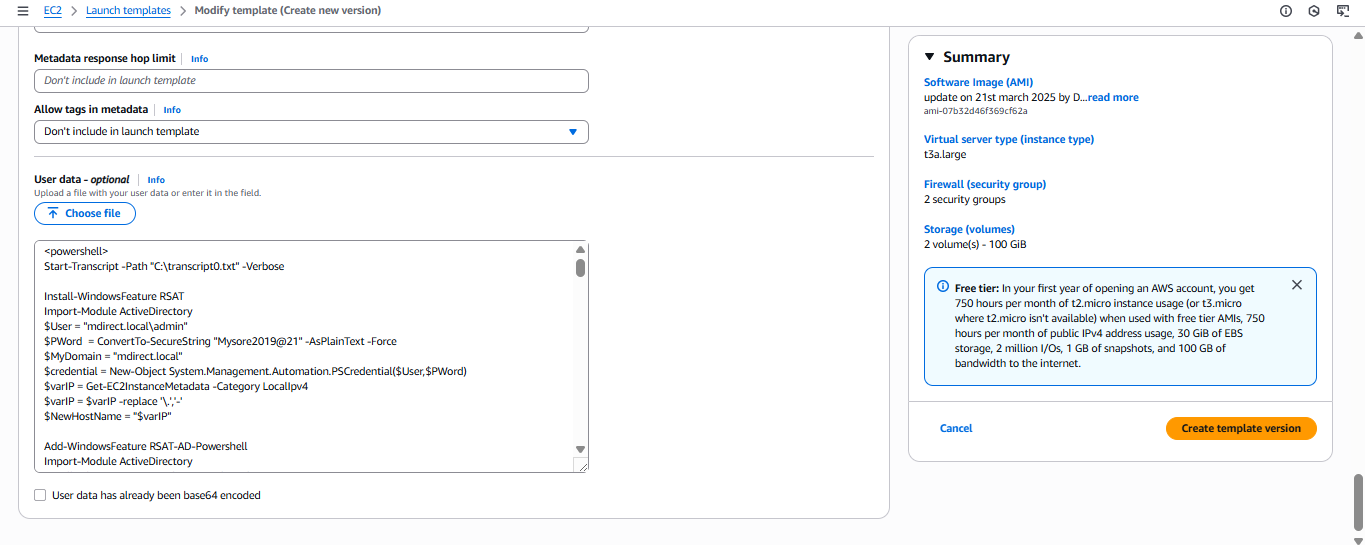
Network settings choose the security-group created earlier as per the requirement



Scroll down and in the volume section select the Root volume and additional volume as required and

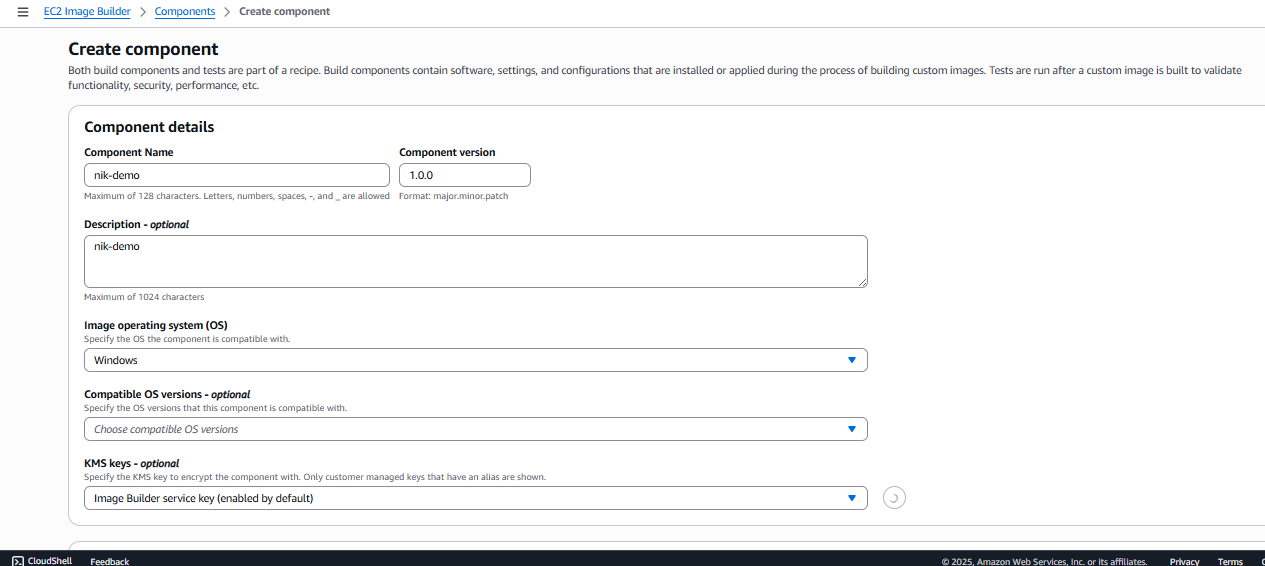


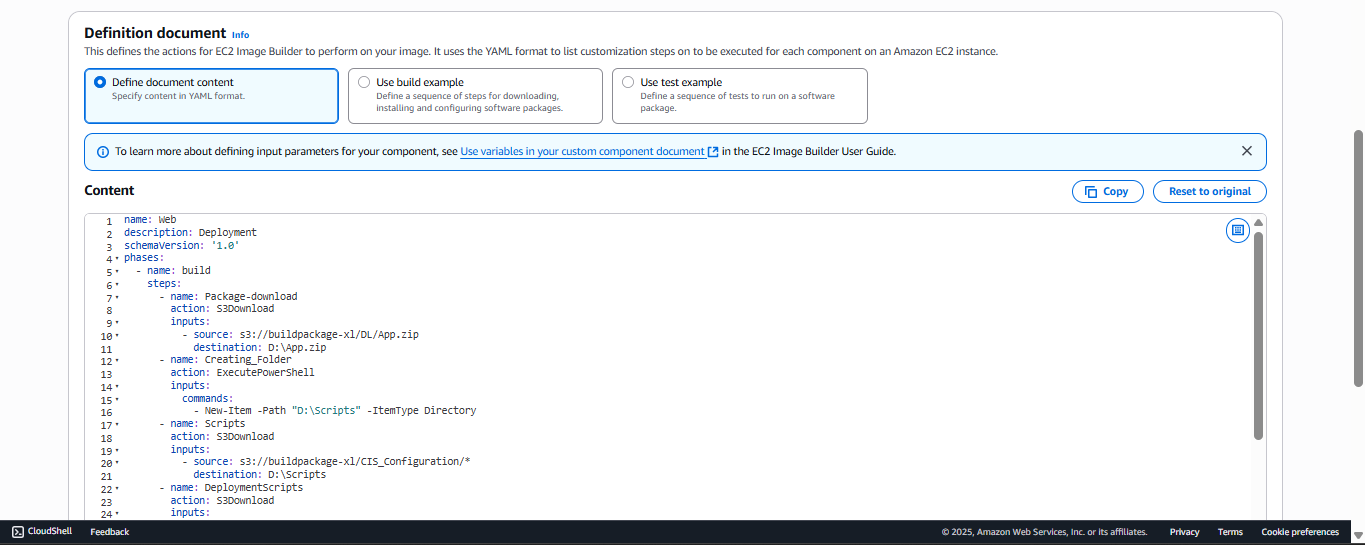
In Addvanced Details section scroll down to end and you’ll find user data section where you can place the scripts/commands that exectes when the EC2 intializes. Add user data as below and click on create template.



**Build Component** : - A **Build Component** is like a **script or set of instructions** used to customize your EC2 AMI (Amazon Machine Image). A component contains a YAML definition file with Steps: name, action (ExecuteBash, ExecutePowerShell), inputs, etc.

Go to AWS search bar -> Build components -> Create Component



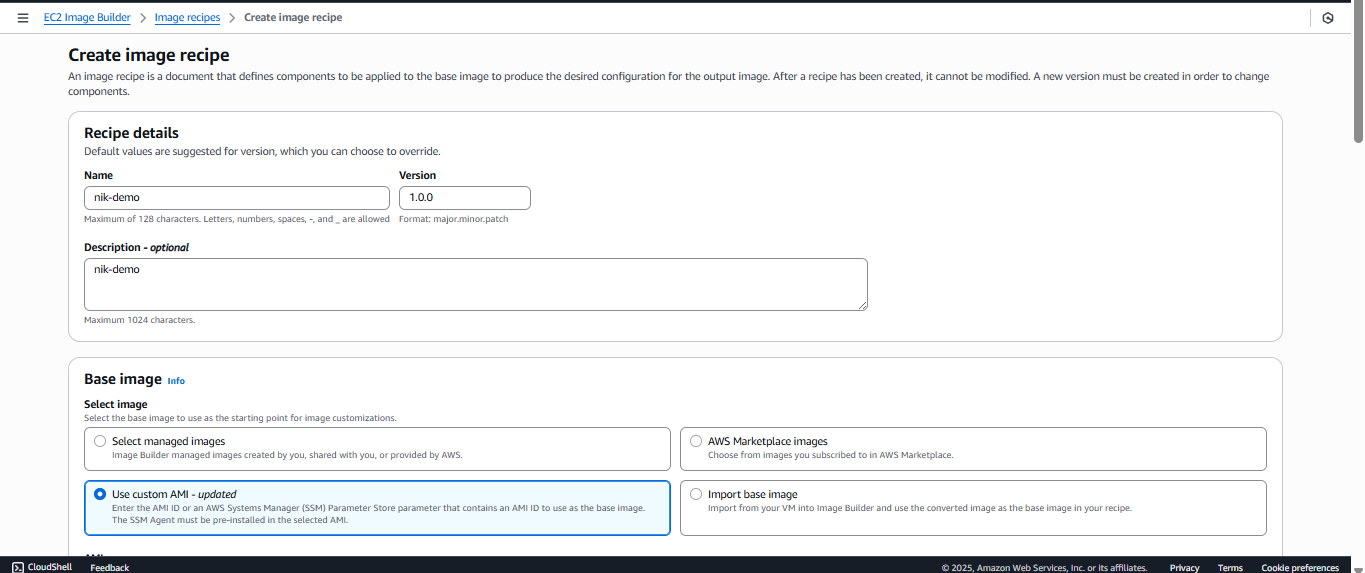


As listed above we need to define a YAML which is the crux of Build component. In the definition document place the YAML contents.   
***Note: -*** Indentation and action defined must be validated and syntactically correct.

Click on createcomponent and the component will be created.

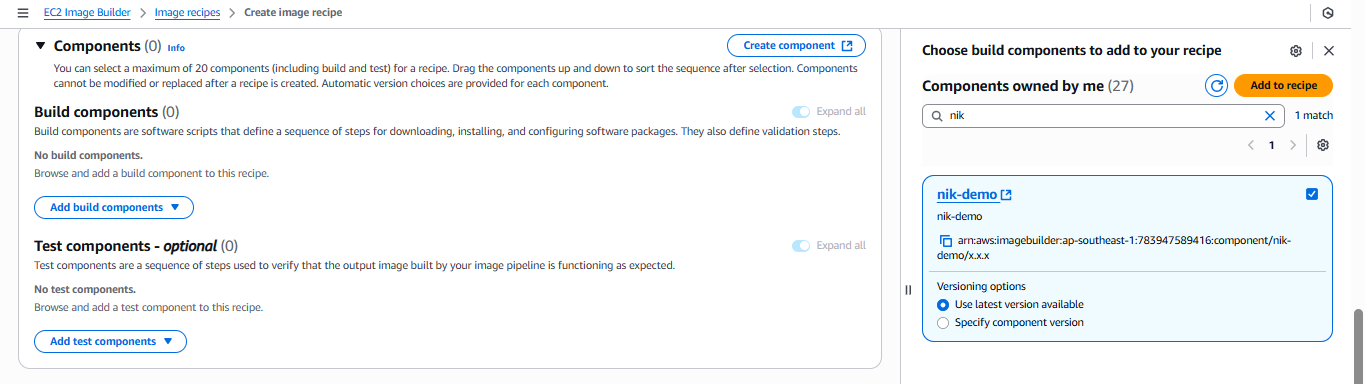
**Image Recipe : -** An Image Recipe is the full recipe for building your AMI. It combines base image, build components, Output settings (AMI name, tags, etc.) and settings into one blueprint.

Go to AWS search bar -> Image Recipe -> Create Image Recipe

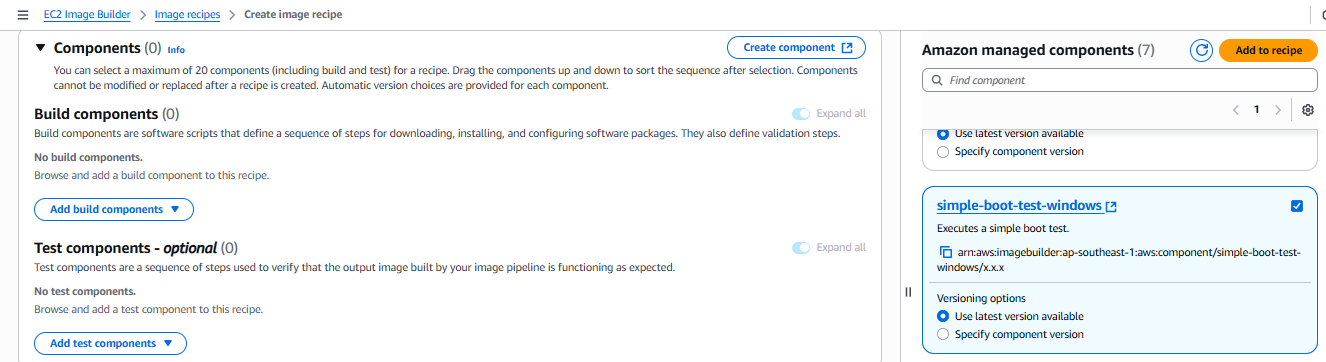


In the base image definition , which is the main part of image recipe use custom AMI as to select the AMI prepared before hand with required best practices and softwares installed.

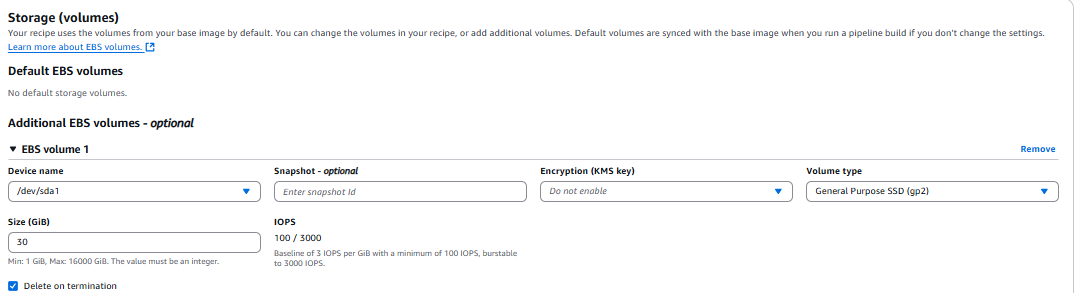
Once this is done, in the components section select the component created as above and add.



After buildcomponent select test components and add it to recipe.Use AWS managed testcomponents.

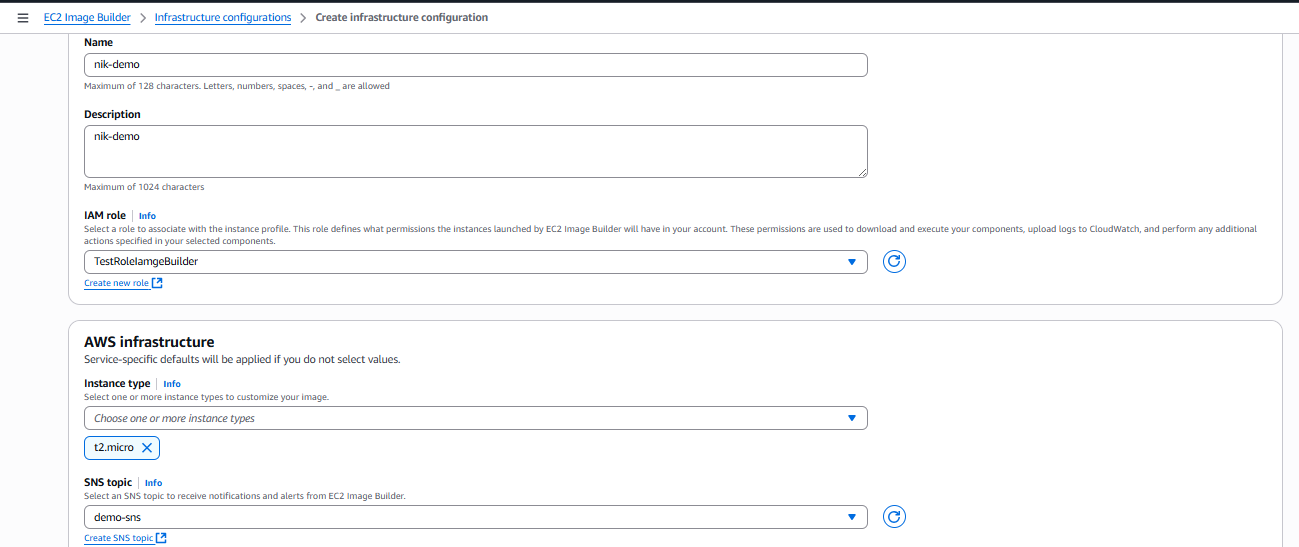


In the storage volume sections , configure EBS as below



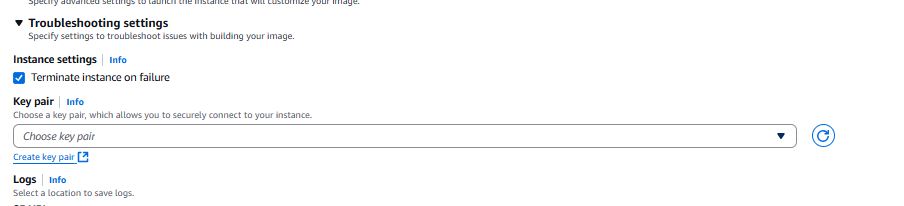
Click on CreateRecipe.  
***Note:-*** Atleast one build and test component needs to be specified to create a recipe.

**Infrastructure configuration** :- The Infrastructure Configuration defines where and how the image build will run. Also serves as the environment settings for the temporary EC2 instance that Image Builder launches to create your custom AMI which will be fed to Image Builder.

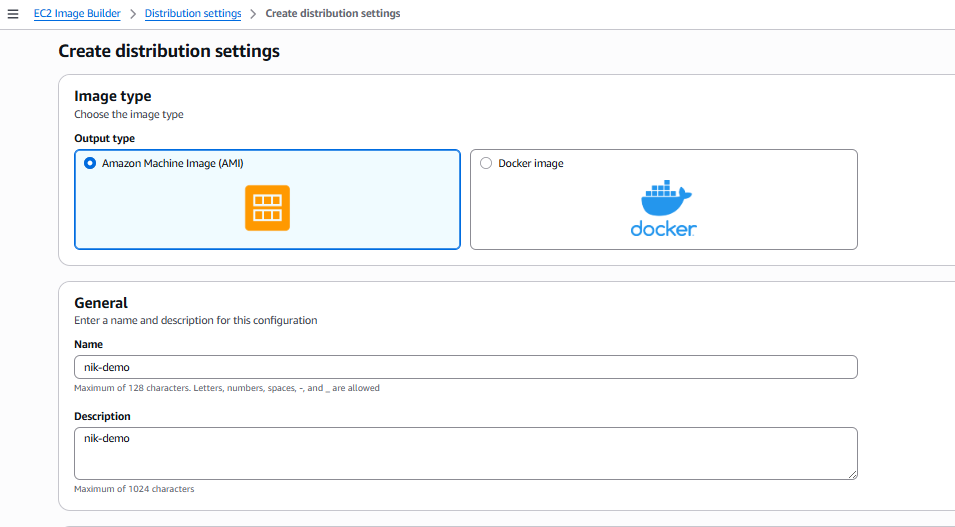


Select the IAM role, Instance type , SNS topic as above

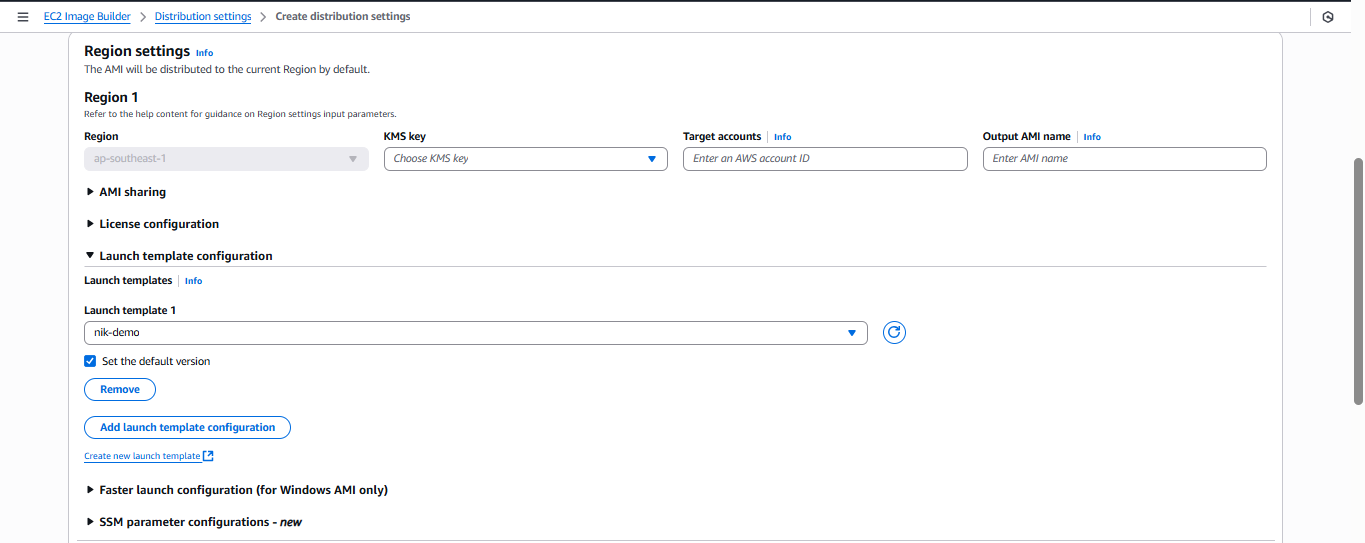
Under Troubleshooting settings enable the checkbox as shown , this is to clean up the resource and a best practice. Only disable this if any troubleshoot or debugging has to be done for the pipeline failure in initial stages.



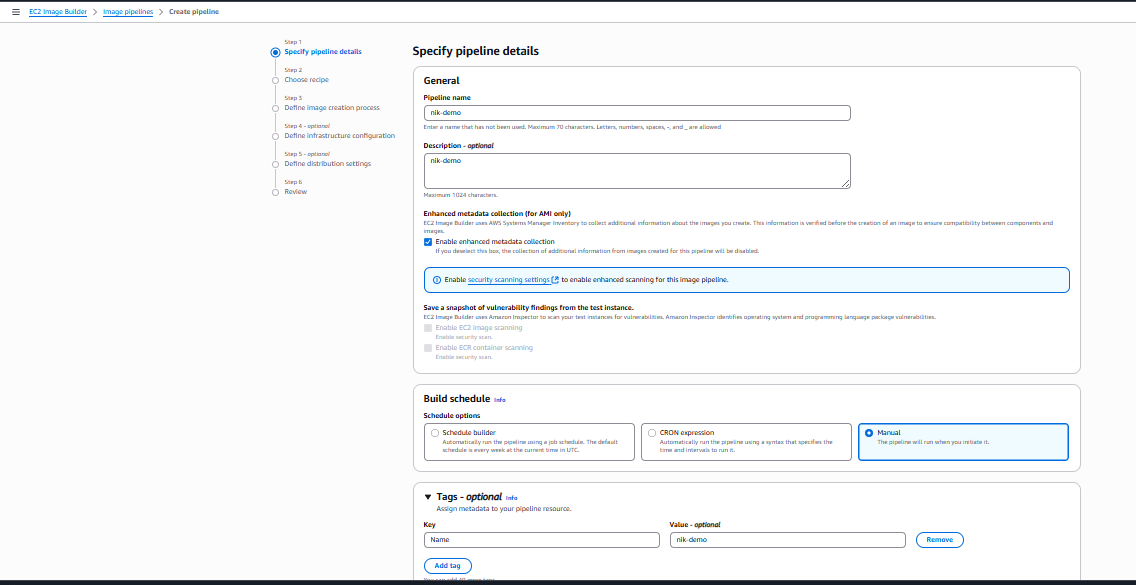
**Distribution Settings : -** A Distribution Settings tell EC2 Image Builder **where and how to share the AMI** after it’s built.



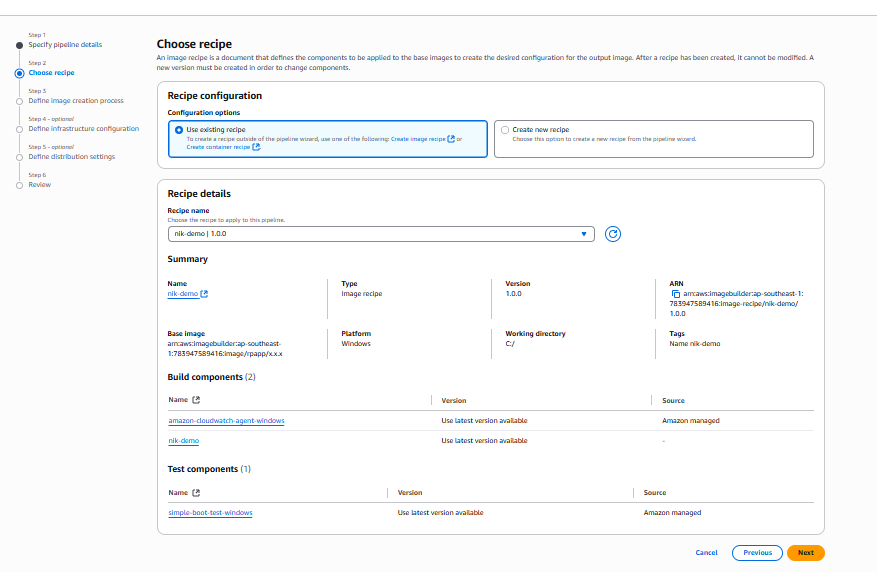
Select the region and Launch Template as shown



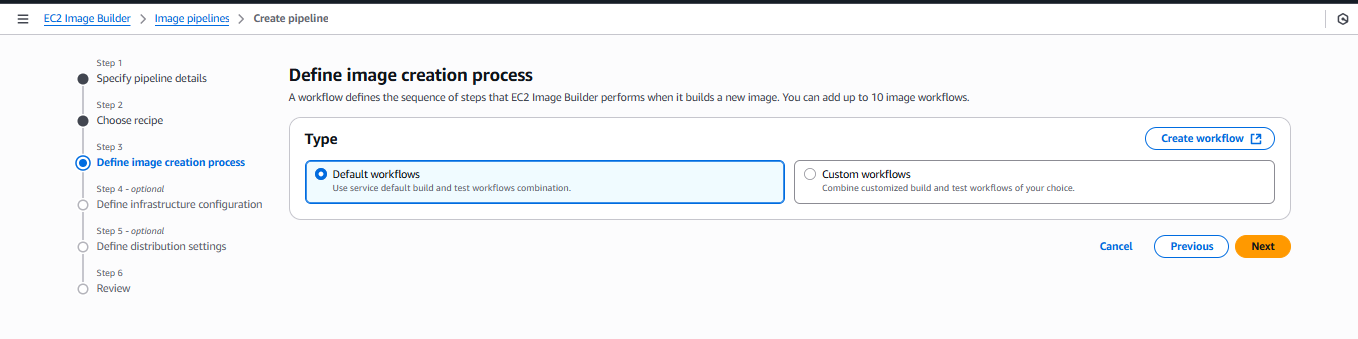
**Once all the above things are ready we can proceed to create Image-Pipeline as below,**



Give the required name and description, Build schedule keep it Manual and add Tags as required and click next

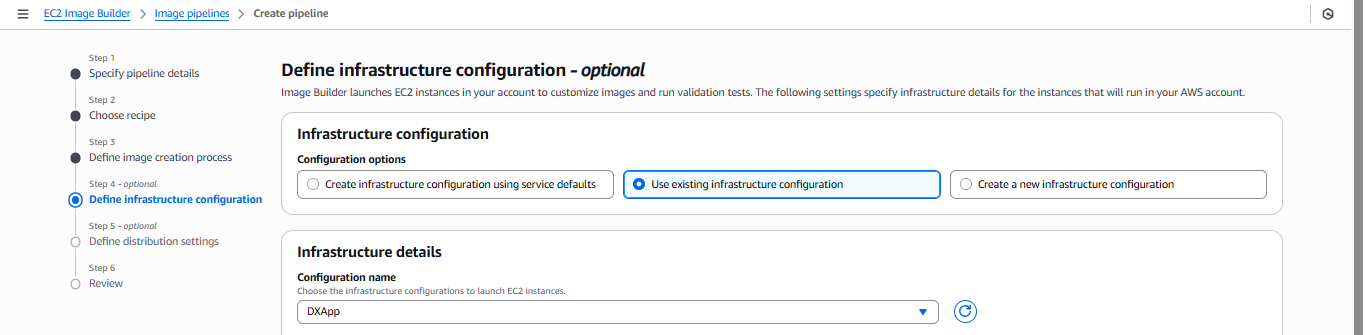


Choose the recipe created earlier and click next

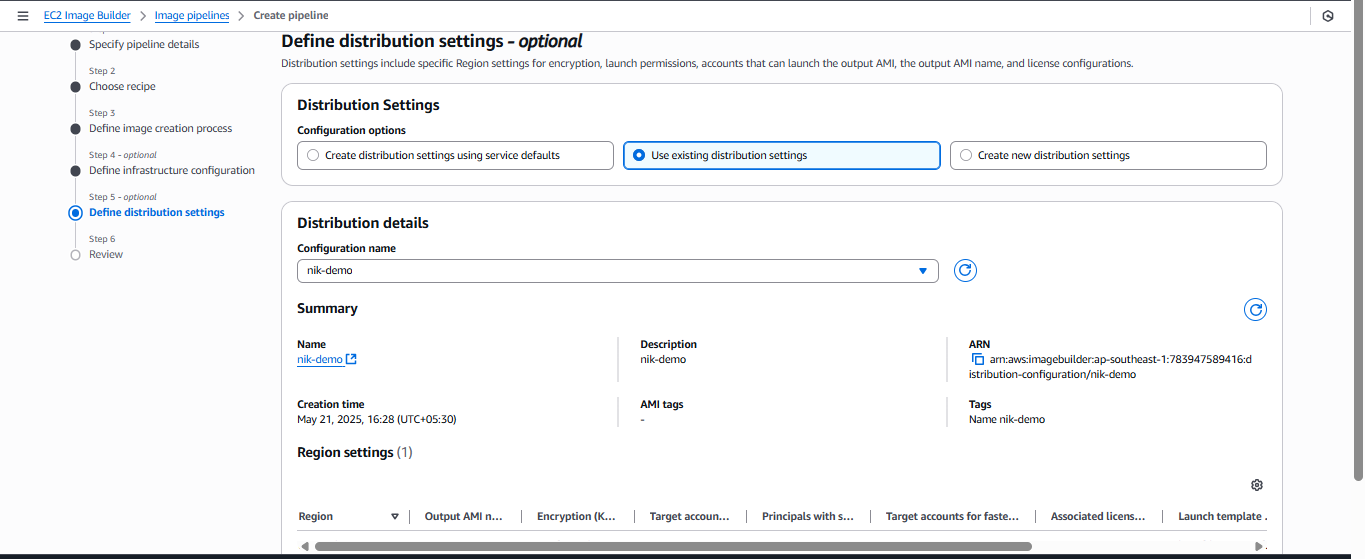


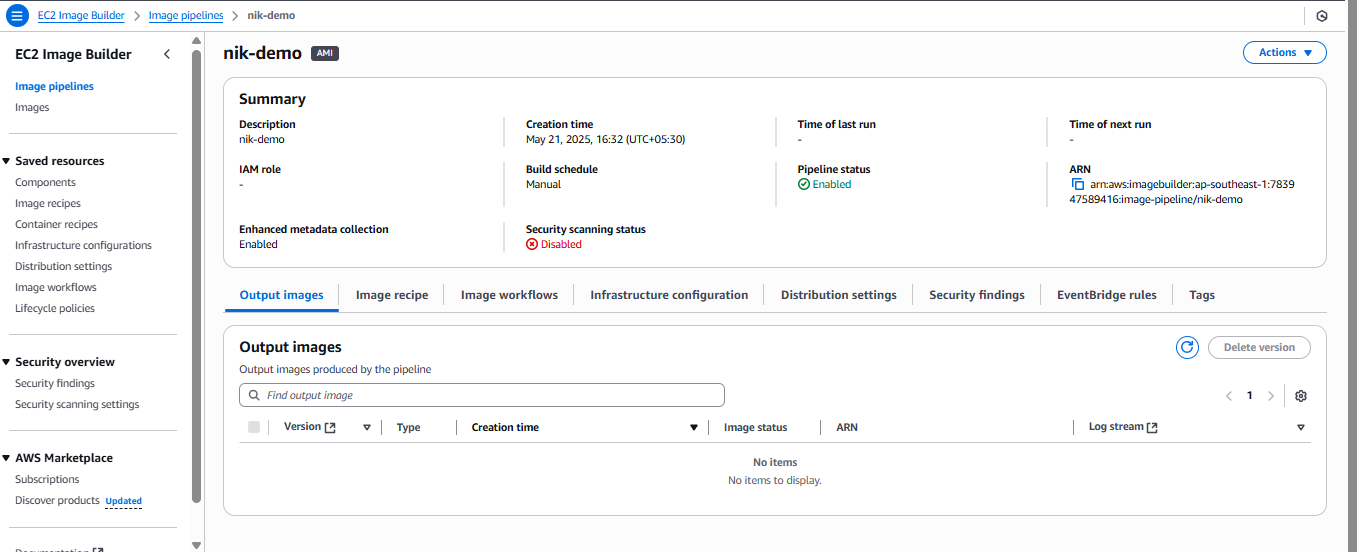
Keep the image creation process as default and click next,

Use existing Infrastructure Configuration created earlier, next



Use existing Distribution settings created earlier, next



Click on Create Pipeline, you have the pipeline ready to run on demand.   


**Congratulations…. now with this document you can create an Image pipeline of your own using multiple AWS resources.**

Once the Image Pipeline is completed the configured sns and lambda will trigger the CodePipeline which again triggers the CodeDeploy.

The output AMI from image pipeline will be automatically fetched by launch template and the configured ASG and Target group will be updated with instances created by latest AMI after CodeDeploy execution  
  
Kindly refer :

CICD\_Pipeline\_document\_V1.2.docx for more details on Codepipeline and Codedeploy.