**Project : EC2 Instance Launch with Enforced Tagging Policy Using IAM**

**Obejctive :**

The objective of this project is to enforce mandatory tagging during EC2 instance creation without using AWS Organizations. By applying a custom IAM policy, the EC2 launch process is restricted unless specific tags—Name, emailID, phoneNo, and Place—are provided. This ensures proper resource identification, cost tracking, and management consistency. The project also demonstrates both successful and failed EC2 launches based on tag compliance, using only identity-based policies within a single AWS account.

**Introduction :**

In cloud environments, resource tagging is a critical practice used for cost tracking, automation, security, and resource organization. To ensure consistency and governance, organizations often enforce mandatory tagging standards.

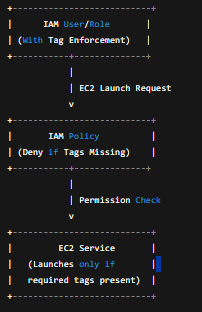
This project focuses on implementing tag enforcement for Amazon EC2 instance launches using IAM policies—without relying on AWS Organizations or service control policies (SCPs). The goal is to deny the creation of any EC2 instance that does not include a predefined set of tags: Name, emailID, phoneNo, and Place.

By attaching a custom IAM policy to a user or role, the project simulates real-world compliance enforcement, ensuring that only properly tagged EC2 resources are allowed to be launched. It includes practical validation of successful and failed EC2 launch scenarios based on tag compliance, offering a secure and cost-efficient approach to resource management within a standalone AWS account.

**Techonology Stack :**

|  |  |
| --- | --- |
| AWS EC2 | Used to launch and manage virtual server instances in the cloud. |
| AWS IAM | Identity and Access Management service used to create users, roles, and apply tag enforcement policies. |
| IAM Policy (Custom | A JSON-based identity policy created to deny EC2 instance launches if required tags are missing. |

**System architecture diagram :**



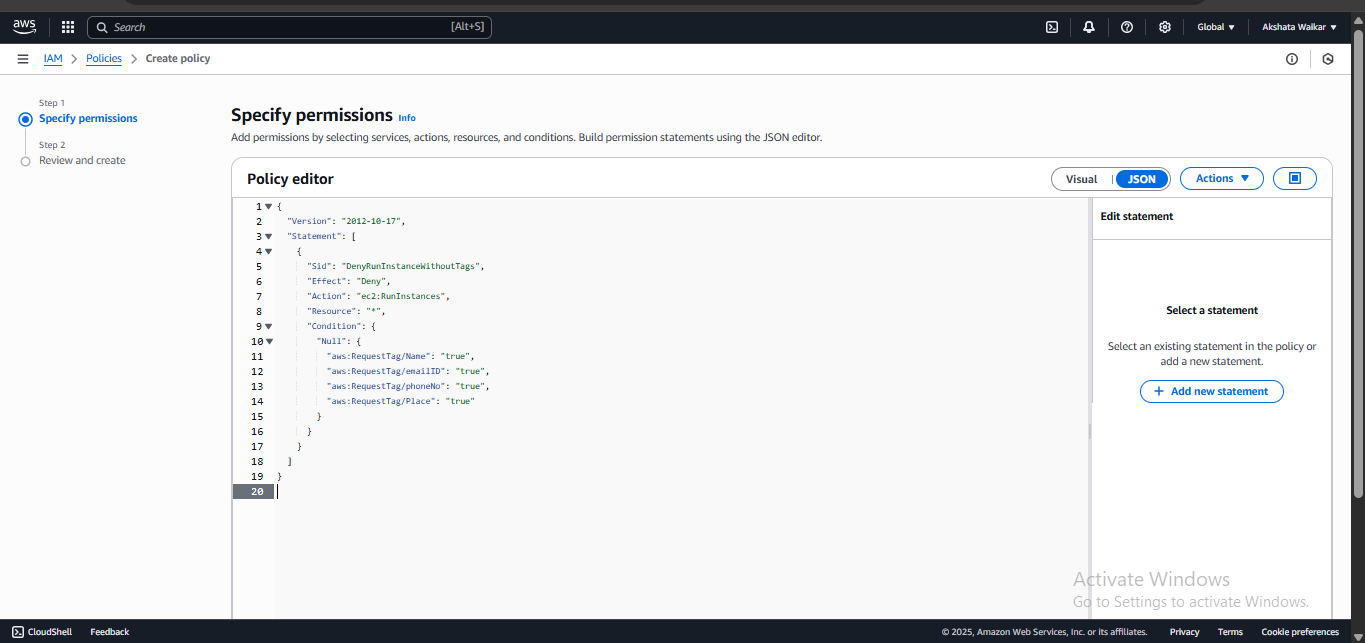
**Implementation Steps :**

**✅ Step 1: Log in to AWS Console**

* Log in with an admin/root user to access IAM and EC2 services.

**✅ Step 2: Create a Custom IAM Policy to Enforce Tags**

1. Navigate to **IAM → Policies → Create policy**.



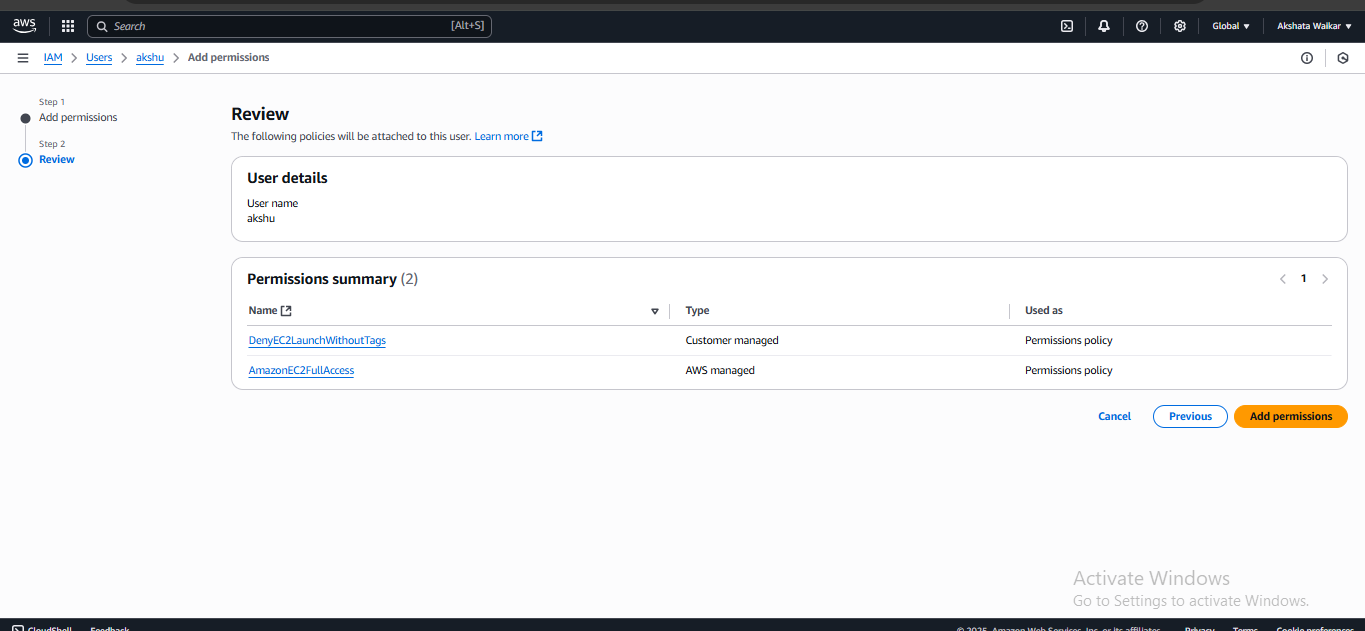
1. Click **Next**, give the policy a name like DenyEC2WithoutTags, and create it.

#### ✅ ****Step 3: Create or Use an IAM User****

1. Navigate to **IAM → Users → Add user**.
2. Create a user

#### ✅ ****Step 4: Attach Policies to the IAM User****

1. Go to the user you just created.
2. Click **Add permissions → Attach policies directly**.
3. Select:
   * AmazonEC2FullAccess (grants EC2 launch permissions)
   * DenyEC2WithoutTags (your custom policy to enforce tags)

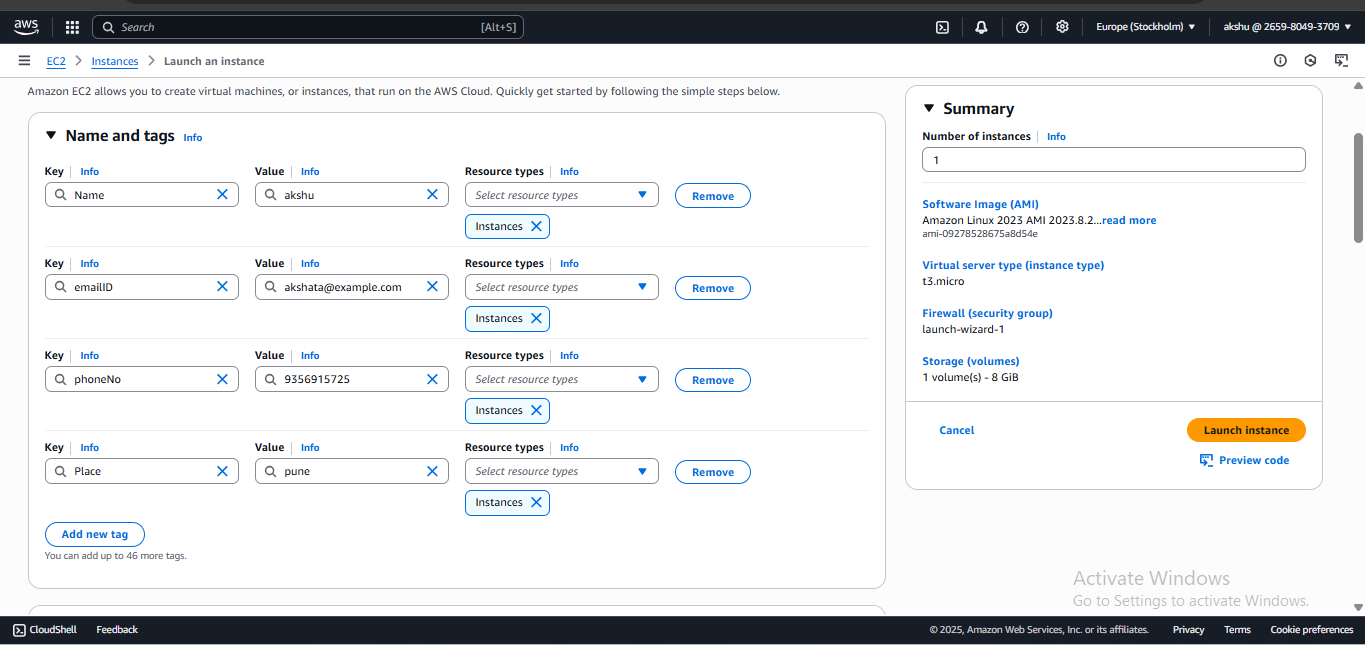


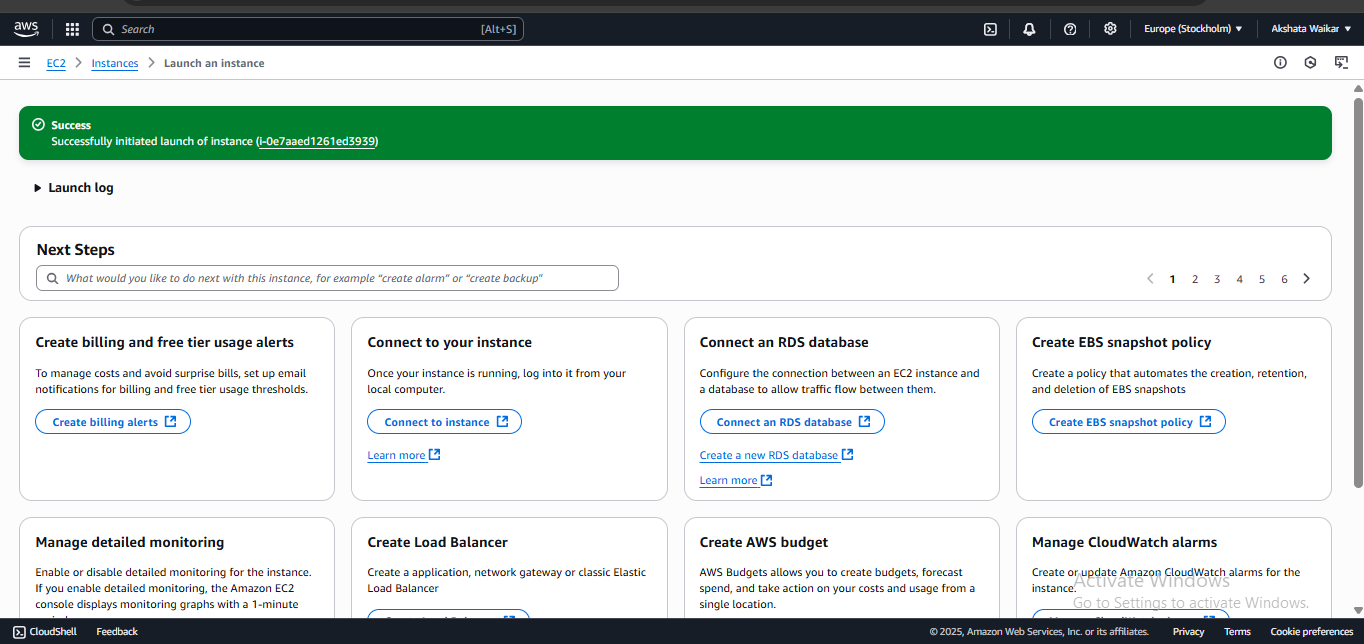
#### ✅ ****Step 5: Launch EC2 Instance with Required Tags (Success Test)****

1. Go to **EC2 → Launch instance**.
2. Choose:
   1. **AMI**: Amazon Linux 2
   2. **Instance type**: t2.micro
   3. **Key pair**: Create or use existing
3. In the **Add Tags** section, add:

| **Key** | **Value** |
| --- | --- |
| Name | Akshata |
| emailID | akshata@example.com |
| phoneNo | 9876543210 |
| Place | Pune |

1. Launch the instance.  
    The instance should launch successfully.

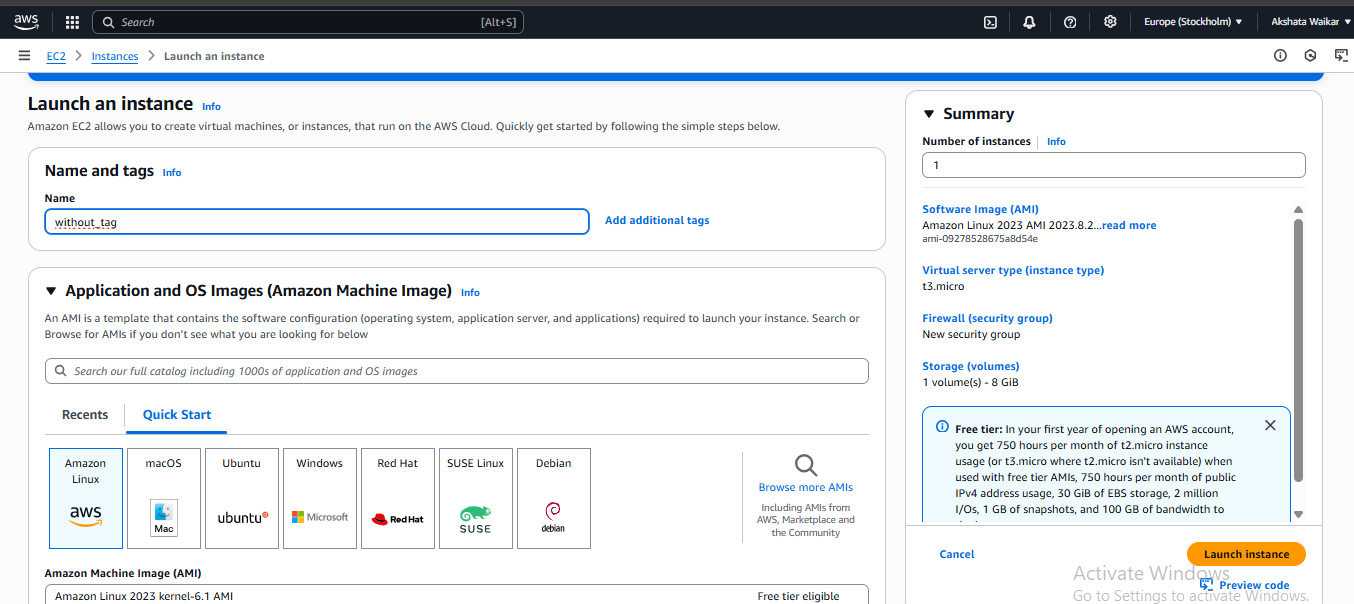


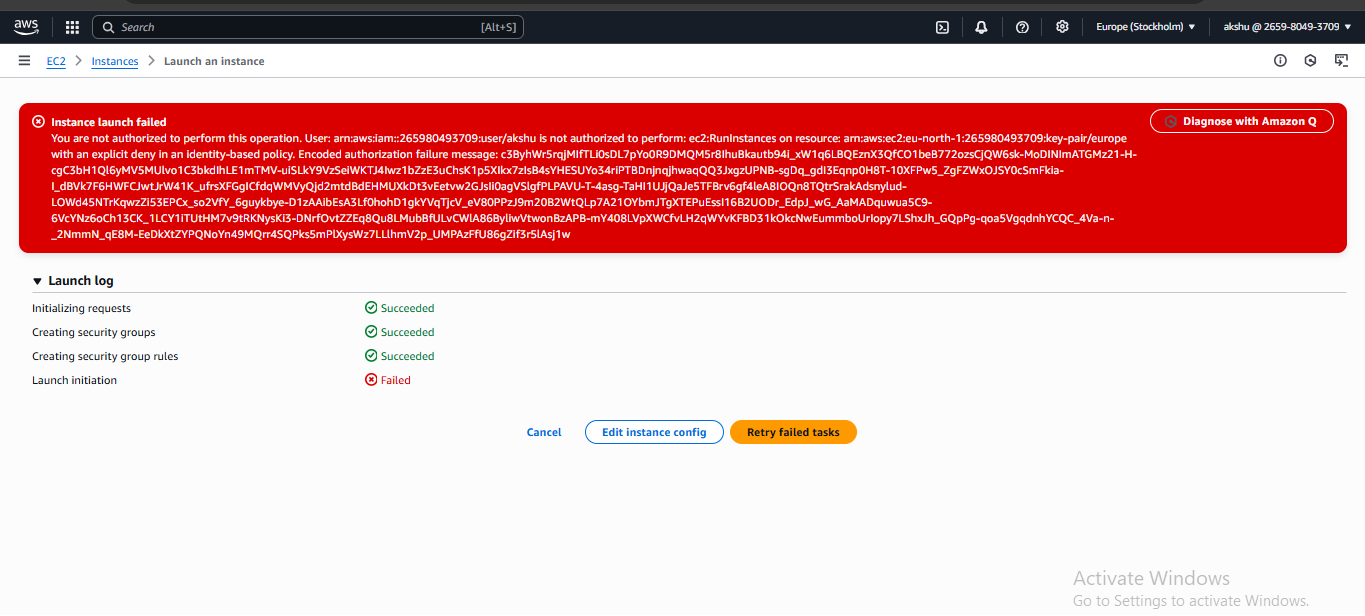


#### ✅ ****Step 6: Launch EC2 Without Tags (Failure Test)****

1. Try launching a new EC2 instance.
2. Skip the "Add Tags" section or miss one tag.
3. Click **Launch**

You should see an error





**Conclusion :**

In this project, we successfully implemented a tag enforcement mechanism for EC2 instance launches using AWS IAM policies, without relying on AWS Organizations or service control policies (SCPs). By creating a custom IAM policy, we ensured that EC2 instances cannot be launched unless they include all required tags: Name, emailID, phoneNo, and Place. This method is simple, scalable, and ideal for individual accounts or smaller teams that want to maintain tagging discipline without the complexity of AWS Organizations.

**GitHub Repository link** : https://github.com/Akshata-Waikar/IAM-Tag\_Policies

**Prepared By:**  
**Akshata Waikar**  
Cloud & DevOps Intern  
Cravita Technologies