**Placement Empowerment Program**

***Cloud Computing and DevOps Centre***

**Implement Auto-scaling in the Cloud**Set up an auto-scaling group for your cloud VMs to handle variable workloads.

Name: Kowshica N Department: Biotech



**Introduction**

Auto-scaling in the cloud refers to the ability to automatically adjust the number of virtual machines (VMs) in a system based on workload demands. This ensures that resources are allocated dynamically, optimizing performance and cost-efficiency. Cloud providers like AWS, Azure, and Google Cloud offer native auto-scaling capabilities, such as AWS Auto Scaling, Azure Scale Sets, and Google Compute Engine Autoscaler.

**Overview**

Auto-scaling uses predefined metrics (e.g., CPU utilization, memory usage, or custom application metrics) to determine when to scale in (reduce resources) or scale out (add resources). Common components of an auto-scaling system include:

1. **Launch Configuration**: Specifies the type of instances to launch (e.g., instance type, Amazon Machine Image (AMI), key pairs).
2. **Auto-scaling Group**: A collection of VMs that are managed collectively to meet demand.
3. **Scaling Policies**: Define when and how scaling actions should occur (e.g., scale up when CPU > 70%).

**Objectives**

The objective of auto-scaling is to:

1. Dynamically manage the number of instances based on traffic patterns or resource utilization.
2. Improve system reliability by responding to workload spikes and reducing resource waste during low-demand periods.
3. Enhance cost efficiency by only running the resources needed for a specific workload.

**Importance**

 **Enhanced Performance**: Ensures application availability and reliability during high-demand periods.

 **Cost Optimization**: Automatically shuts down unused instances to reduce costs.

 **Scalability**: Easily adapts to changing business requirements.

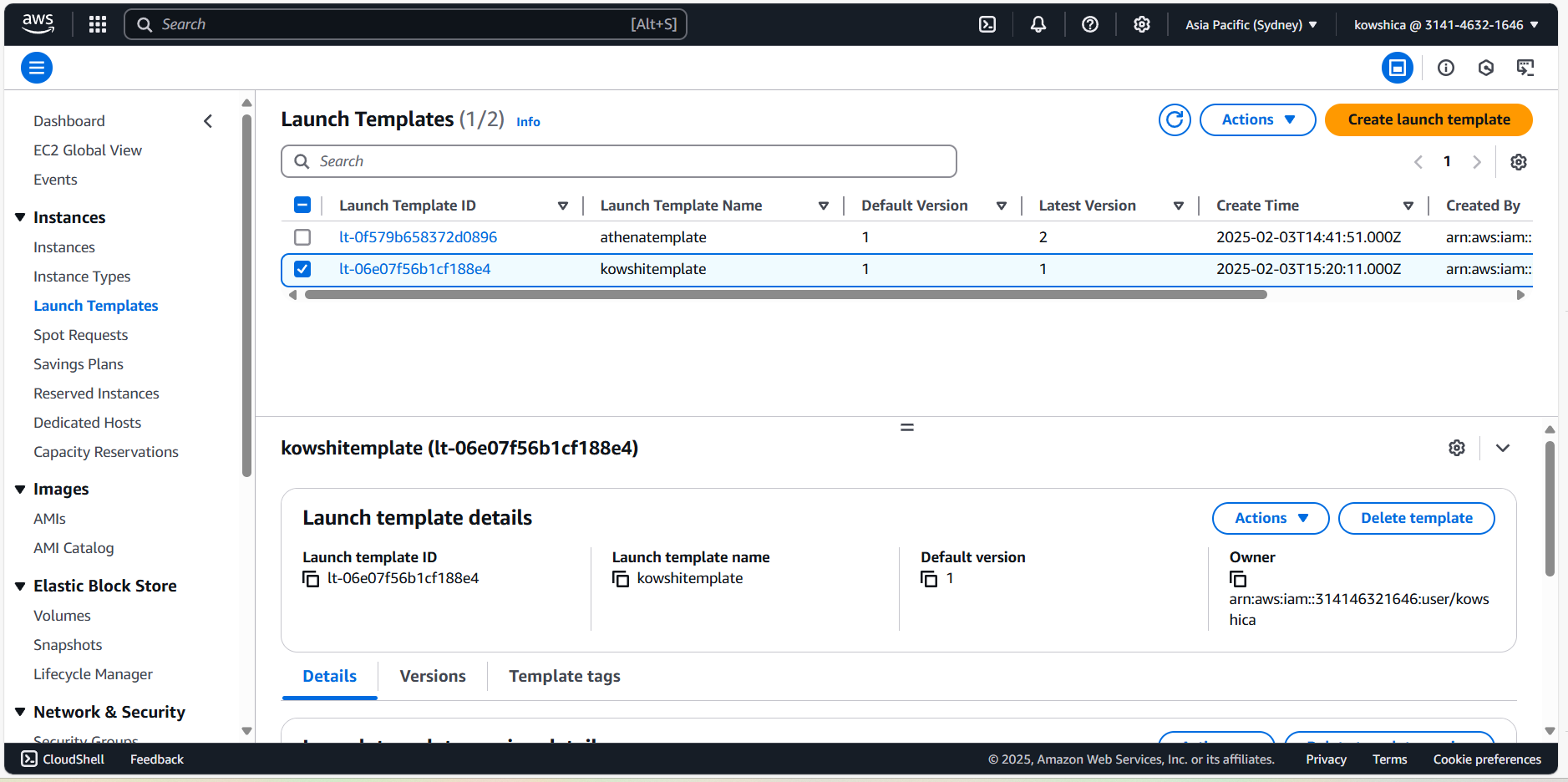
 **Operational Efficiency**: Reduces the need for manual intervention in resource allocation.

**Step-by-Step Overview**

Step 1:

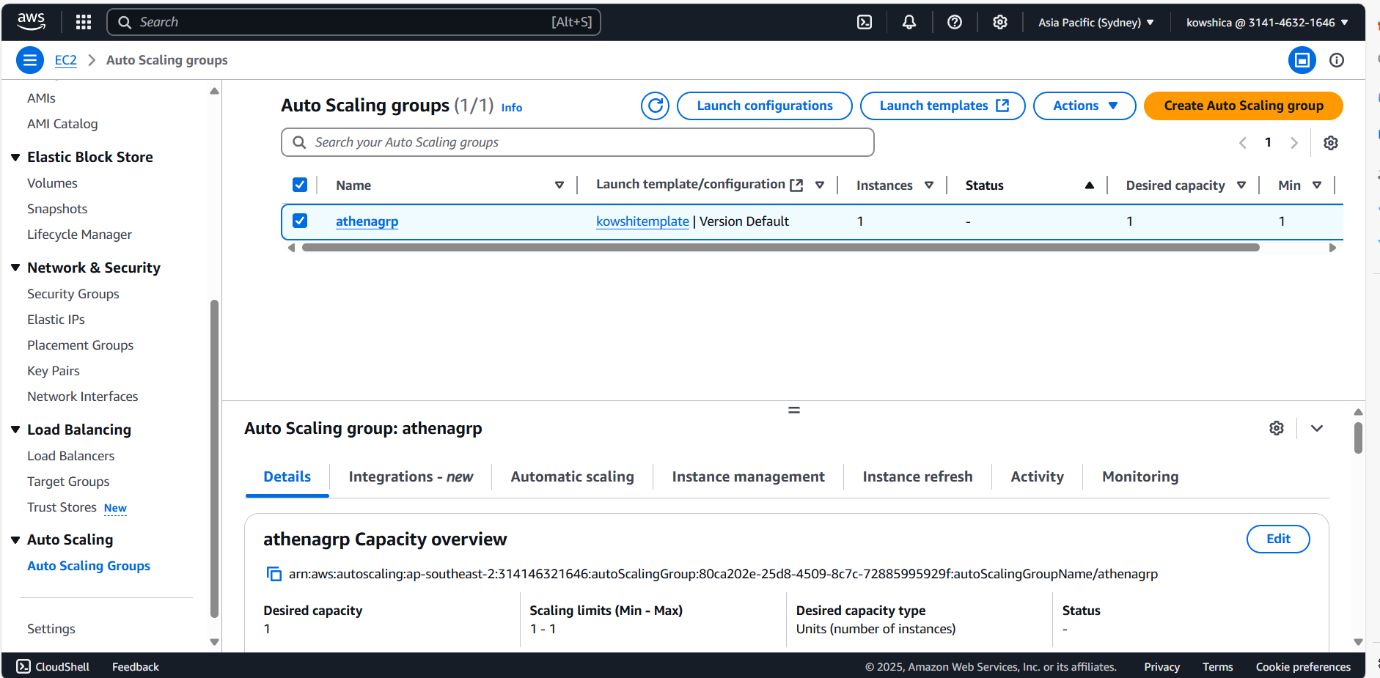
**Set Up Auto-Scaling in AWS**

1. **Create a Launch Template (or Configuration):**
   * Navigate to the **EC2** service in the AWS Management Console.
   * Select **Launch Templates**.
   * Create a new launch template with:
     + AMI ID (choose the image for your VMs).
     + Instance type (e.g., t2.micro).
     + Key pair and security group settings.



Step 2 **Create an Auto-Scaling Group:**

* Navigate to **Auto Scaling Groups** under the EC2 Dashboard.
* Click on **Create Auto Scaling group**.
* Configure:
  + **Launch Template**: Select the template created earlier.
  + **Instance Count**: Set the desired, minimum, and maximum number of instances.
  + **Network**: Choose the VPC and subnets.



Step 3

**Set Scaling Policies:**

* Select **Dynamic Scaling Policy**:
  + Example: Scale out when CPU utilization exceeds 70%.
  + Example: Scale in when CPU utilization is below 30%.
* Configure alarm actions using **CloudWatch**.

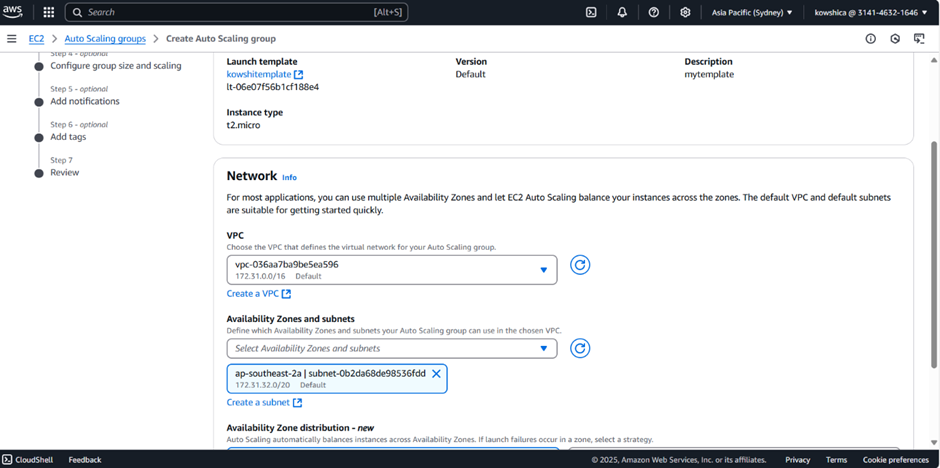
Step 4

**Configure Notifications:**

* Set up Amazon SNS (Simple Notification Service) for email or SMS alerts on scaling events.

Step 5

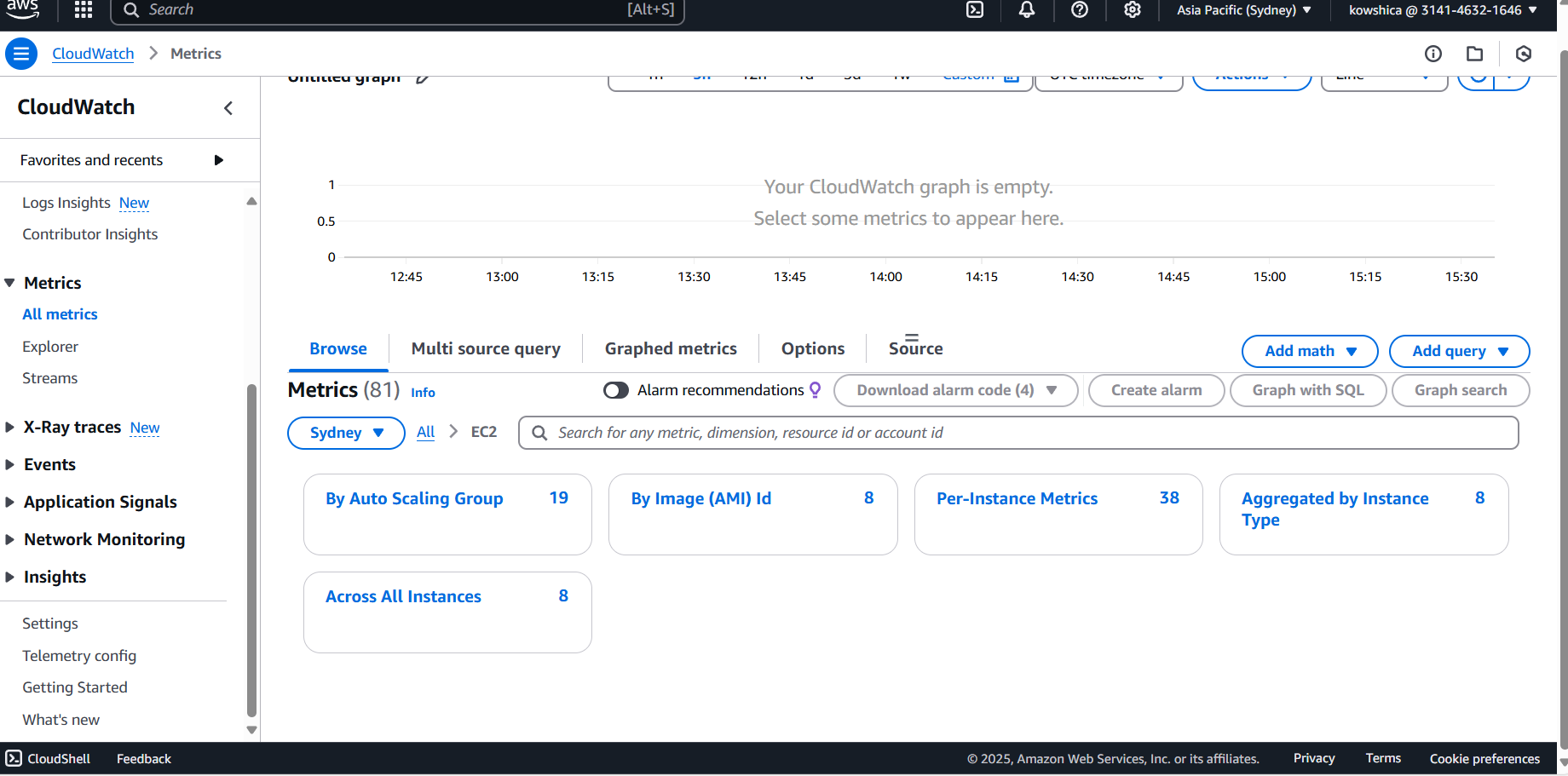
**Test the Auto-Scaling Setup:**

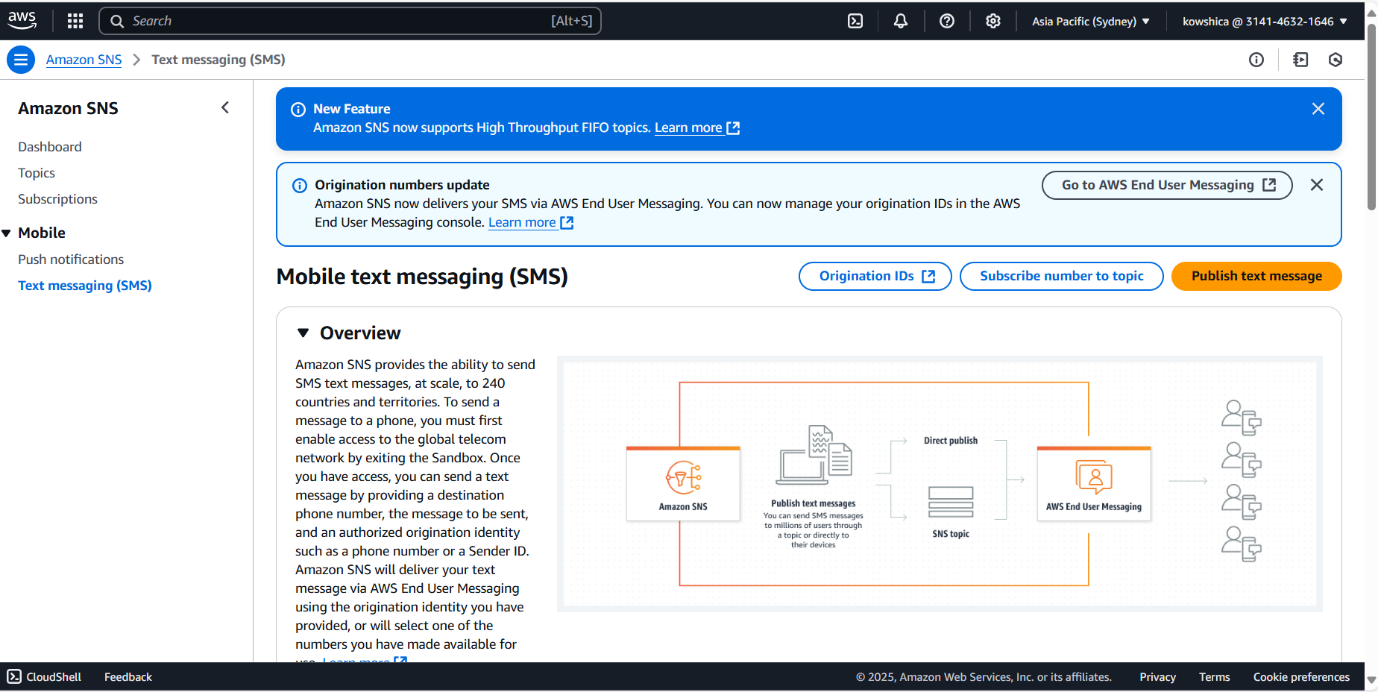
* Increase the workload (e.g., by generating traffic to a web application).
* Monitor the scaling activity in the AWS Management Console.

Step 6

**Set Up Auto-Scaling in Other Cloud Providers**

1. **Azure Auto-Scaling:**
   * Use **Azure Virtual Machine Scale Sets**.
   * Configure scaling rules in the Azure portal (e.g., scale out when CPU > 60%).
   * Enable notifications using Azure Monitor.



. 

**Expected Outcome**

 **Dynamic Resource Management**: The system automatically adjusts resources to match demand.

 **Improved System Resilience**: Ensures high availability and reliability for end users.

 **Cost Savings**: Reduces over-provisioning and associated expenses.

 **Simplified Operations**: Minimizes manual intervention and management overhead.