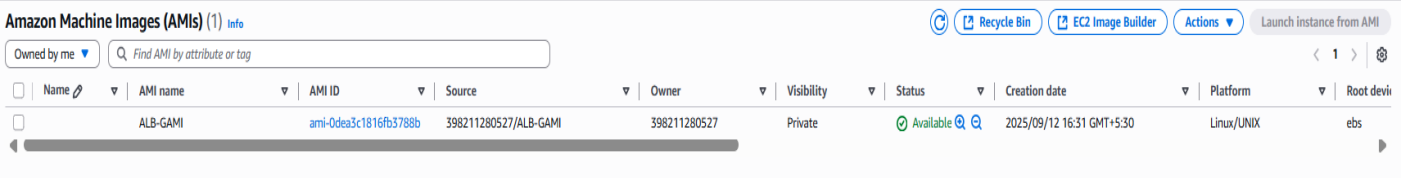
Practical Lab on EC2 Auto Scaling using AWS

Linkedin : https://www.linkedin.com/in/uppala-venkata-sai/
Github : https://github.com/UppalavenkataSai

Step 1: Create an Amazon Machine Image (AMI)

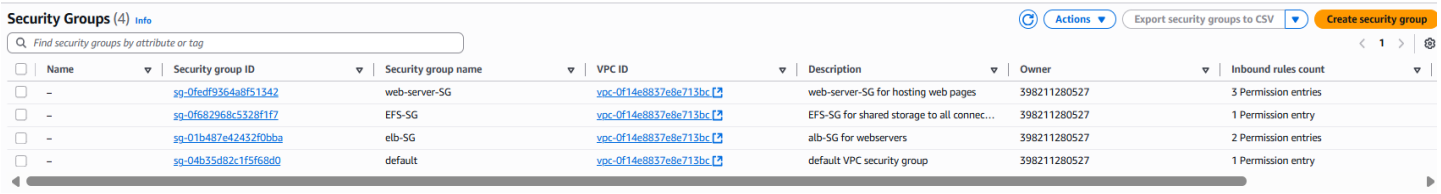
1. Log in to your AWS Management Console.
2. Navigate to the EC2 Dashboard.
3. Launch an EC2 instance to serve as your base image. Install all necessary software on this instance, such as a web server (e.g., Apache or Nginx) and your web application code.
4. Once the instance is configured, select it from the EC2 Instances list.
5. Go to `Actions` -> `Image and templates` -> `Create image`.
6. Give the image a name (e.g., `ALB-GA-AMI`) and a description.
7. Click `Create image`. This will create a snapshot of your instance's root volume and create a new AMI.



Step 2: Create Security Groups

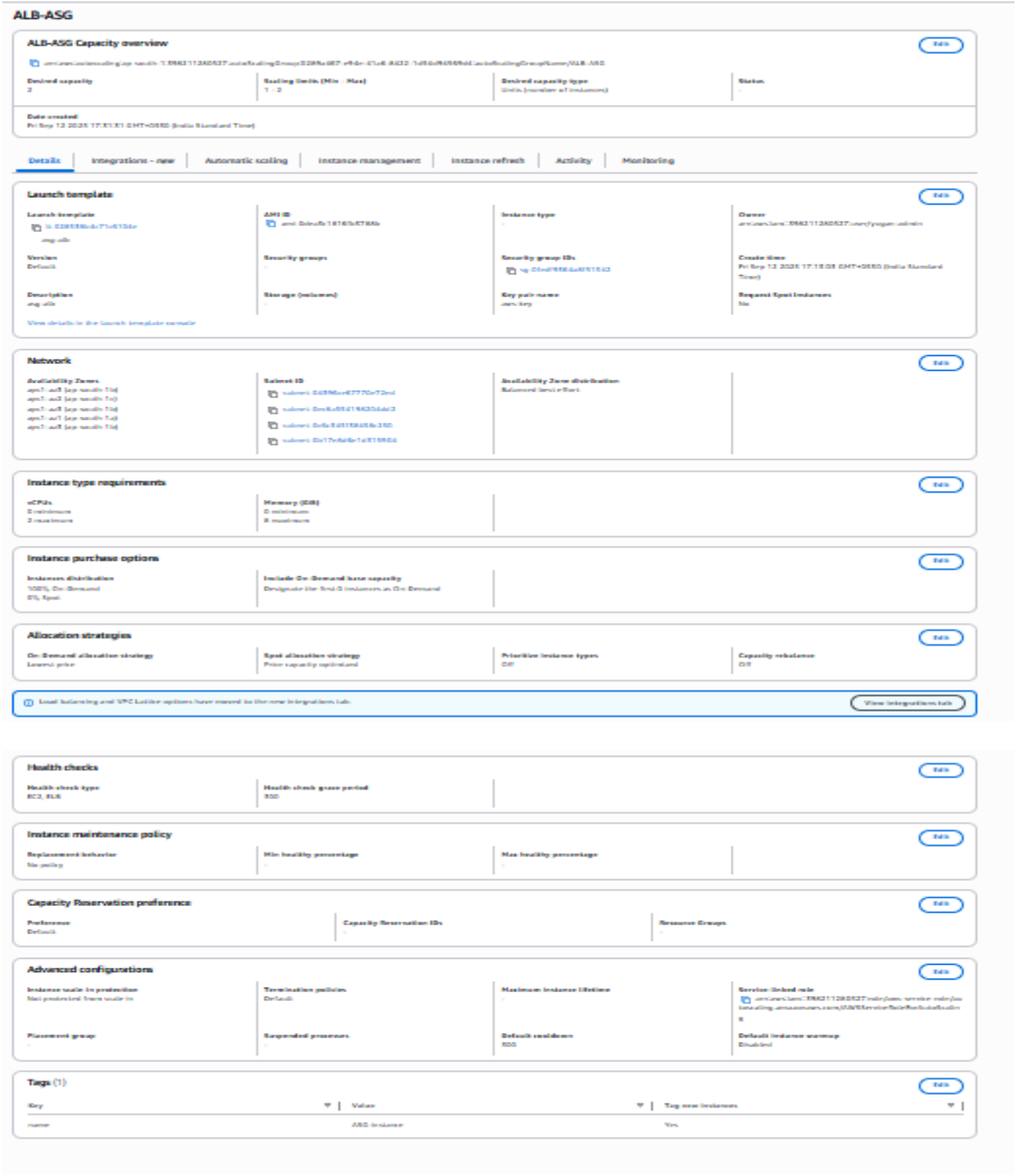
1. From the EC2 Dashboard, navigate to `Security Groups`.
2. Click `Create security group` to create the following three security groups:
    - **ALB Security Group:**
        - Give it a name (e.g., `elb-SG`).
        - Add an **Inbound rule** for HTTP traffic:
            - **Type:** HTTP
            - **Protocol:** TCP
            - **Port range:** 80
            - **Source:** Anywhere (`0.0.0.0/0`)
        - This security group allows all incoming HTTP traffic to the load balancer.
    - **Web Server Security Group:**
        - Give it a name (e.g., `web-server-SG`).
        - Add an **Inbound rule** for HTTP traffic:
            - **Type:** HTTP
            - **Protocol:** TCP
            - **Port range:** 80
            - **Source:** `Custom`, and enter the security group ID for your `elb-SG` to allow traffic only from the ALB.
        - Add an inbound rule for SSH access for management.
        - This security group allows incoming HTTP traffic only from the load balancer.

- **EFS Security Group:**
  - Give it a name (e.g., `EFS-SG`).
  - Add an **Inbound rule** for EFS traffic:
    - **Type:** NFS
    - **Protocol:** TCP
    - **Port range:** 2049
    - **Source:** `Custom`, and enter the security group ID for your `web-server-SG` to allow traffic only from the web servers.
  - This security group allows the EC2 instances to communicate with the EFS file system.



Step 3: Create an Application Load Balancer (ALB)

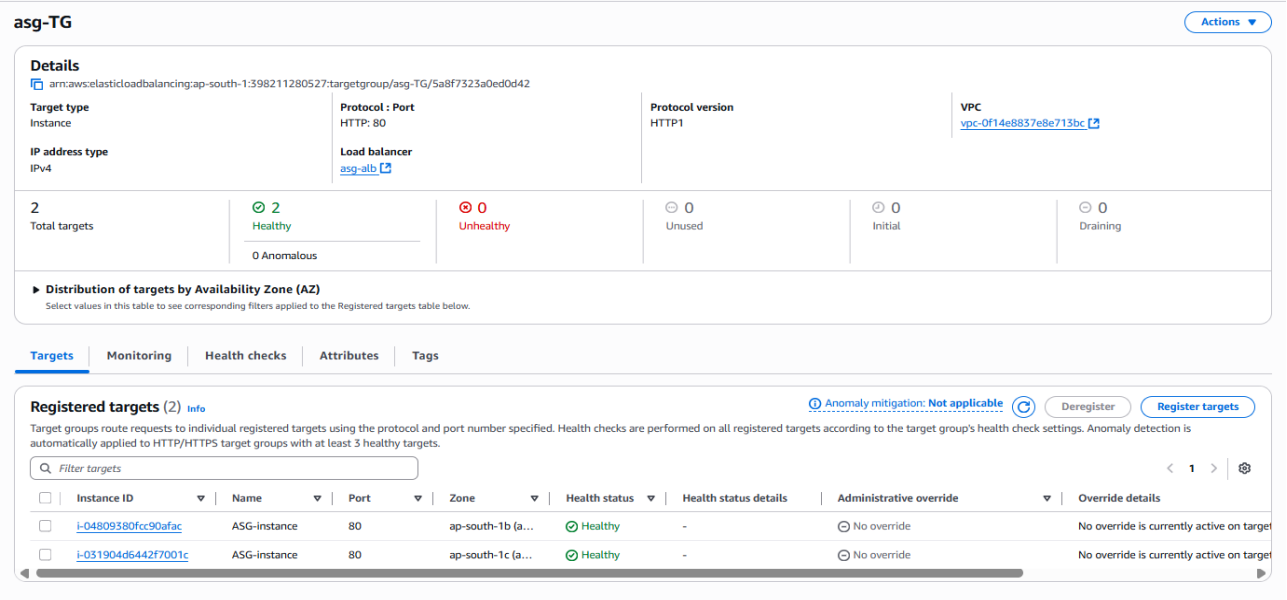
1. From the EC2 Dashboard, navigate to `Load Balancers`.
2. Click `Create Load Balancer`.
3. Choose `Application Load Balancer`.
4. Configure the load balancer:
   - **Name:** `asg-alb`
   - **Scheme:** Internet-facing
   - **VPC:** Select your default VPC.
   - **Availability Zones:** Select at least two Availability Zones (e.g., `ap-south-1b` and `ap-south-1c`).
   - **Security Groups:** Select the `elb-SG` security group



Linkedin : https://www.linkedin.com/in/uppala-venkata-sai/
Github : https://github.com/UppalavenkataSai

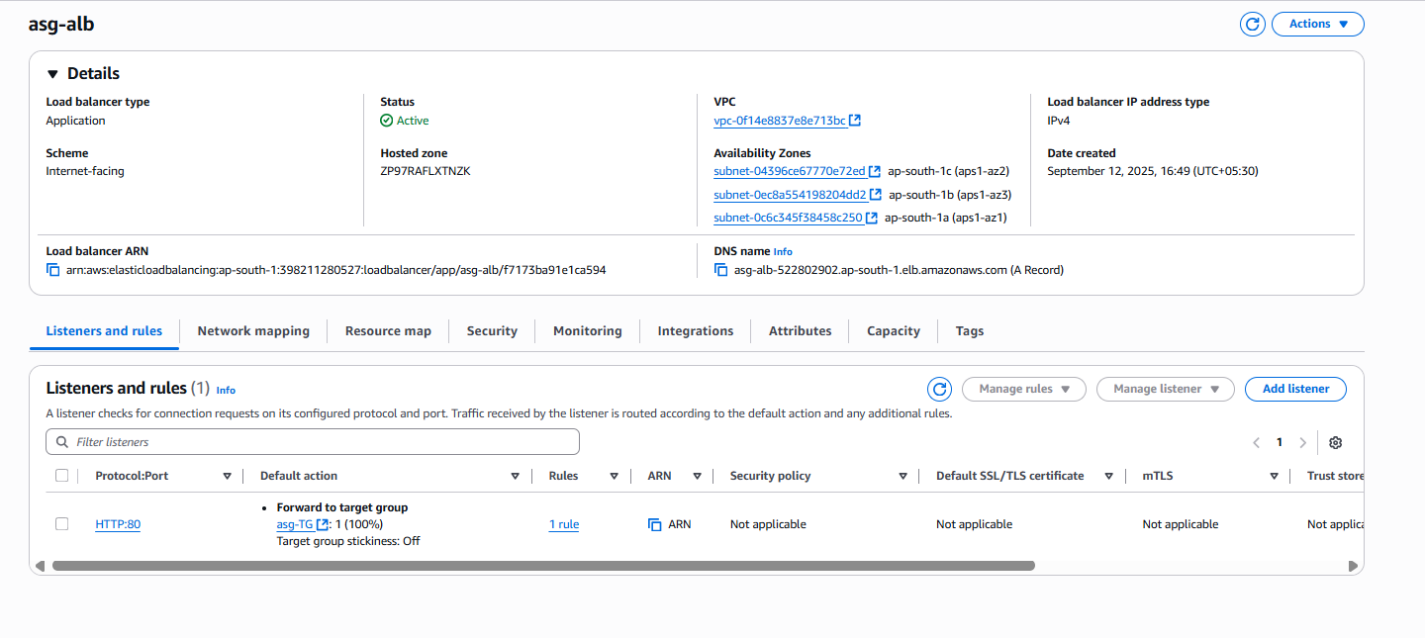
Step 4: Create a Target Group

1. From the EC2 Dashboard, navigate to `Target Groups`.
2. Click `Create target group`.
3. Choose `Instances`.
4. **Target group name:** `asg-tg`
5. **Protocol:** HTTP, **Port:** 80.
6. **VPC:** Select your default VPC.



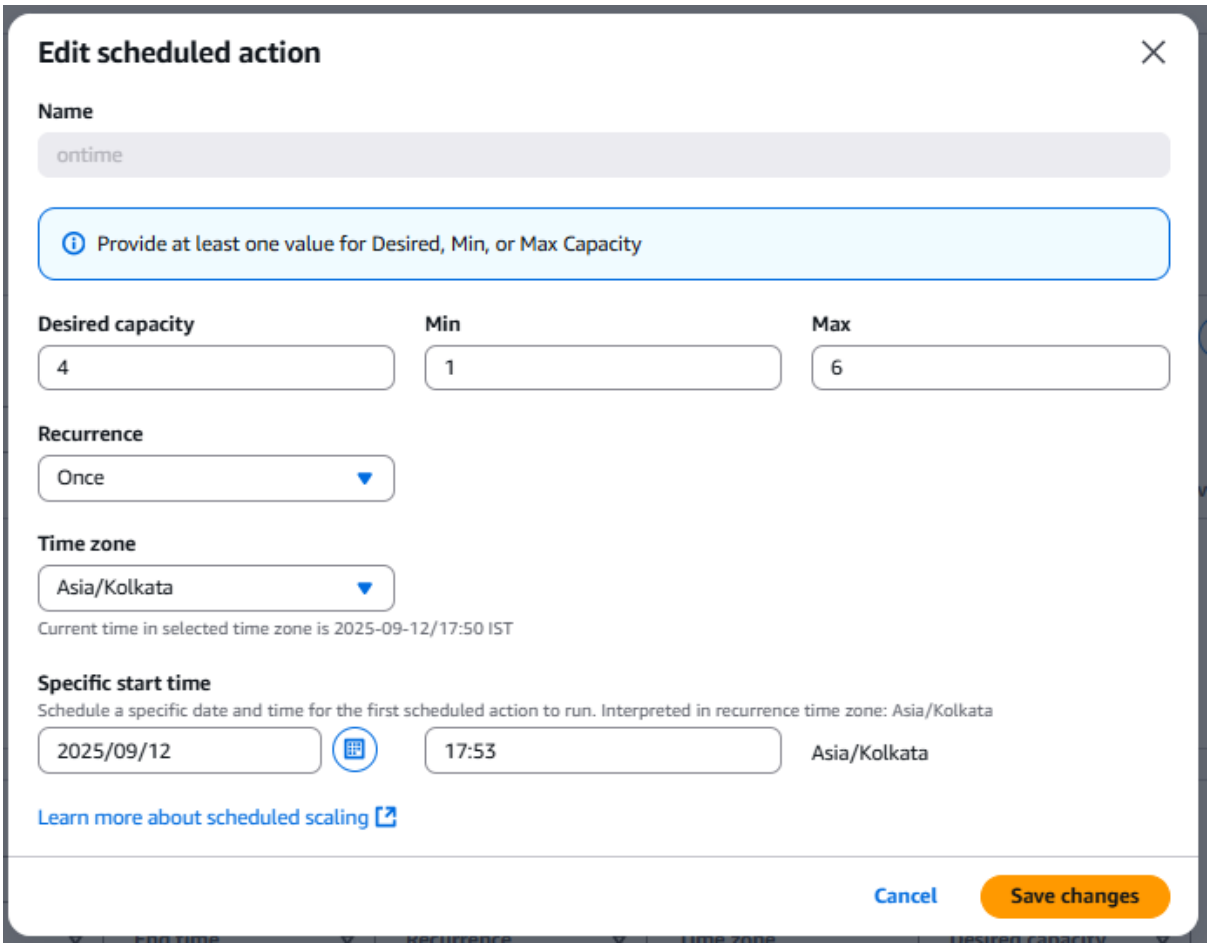
Step 5: Configure the ALB Listener

1. Go back to your `asg-alb` in the `Load Balancers` section.
2. Click on the `Listeners` tab.
3. Click `Add listener`.
4. **Protocol:** HTTP, **Port:** 80.
5. **Default action:** `Forward to`, and select the `asg-tg` target group you just created.
6. Click `Add`.



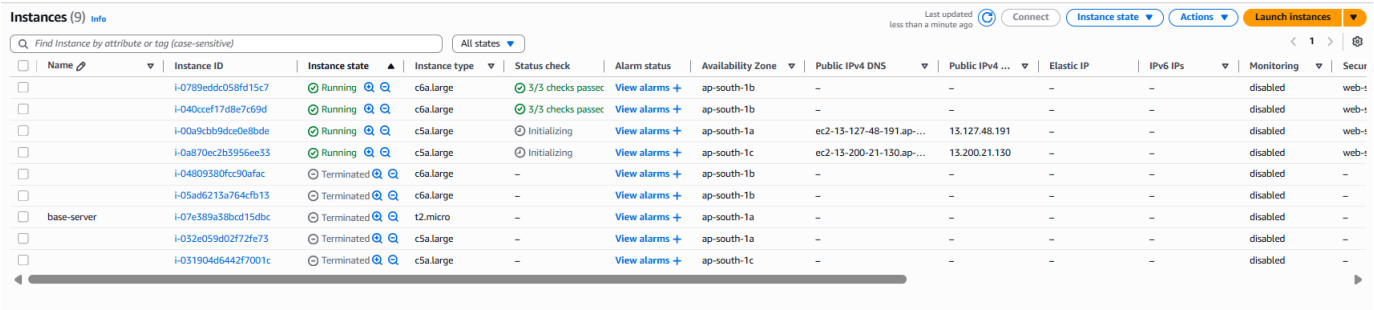
Step 6: Create an Auto Scaling Group (ASG)

1. From the EC2 Dashboard, navigate to `Auto Scaling Groups`.
2. Click `Create Auto Scaling group`.
3. **Name:** Give it a name, such as `my-asg`.
4. **Launch Template:** Create a new launch template or select an existing one. In the template, specify:
    - The AMI you created in Step 1 (`ALB-GA-AMI`).
    - The instance type (`c5a.large`).
    - The security group you created for the web server (`web-server-SG`).
5. **Network:**
    - **VPC:** Select your default VPC.
    - **Subnets:** Select the same subnets that you used for your ALB.
6. **Load balancing:**
    - **Attach to an existing load balancer:** Choose your `asg-tg` target group.
7. **Group size:** Set your desired, minimum, and maximum capacity (e.g., **Desired:** 4, **Minimum:** 1, **Maximum:** 6).
8. **Scaling policies:** Configure scaling policies based on metrics like CPU utilization or network traffic.
9. Review and click `Create Auto Scaling group`.



Step 7: Configure Scheduled Scaling

1. Select your newly created Auto Scaling Group.
2. Go to the `Automatic scaling` tab.
3. Click `Create scheduled action`.
4. **Name:** `ontime`
5. **Desired capacity:** 4, **Min capacity:** 1, **Max capacity:** 6.
6. **Recurrence:** `Once`.
7. **Start Time:** Set the specific date and time for the scheduled action to occur (e.g., `2025/09/12` at `17:53`).
8. Click `Create`.



Instances (9) Info

Last updated less than a minute ago | Connect | Instance state | Actions | Launch instances

Find Instance by attribute or tag (case-sensitive) | All states

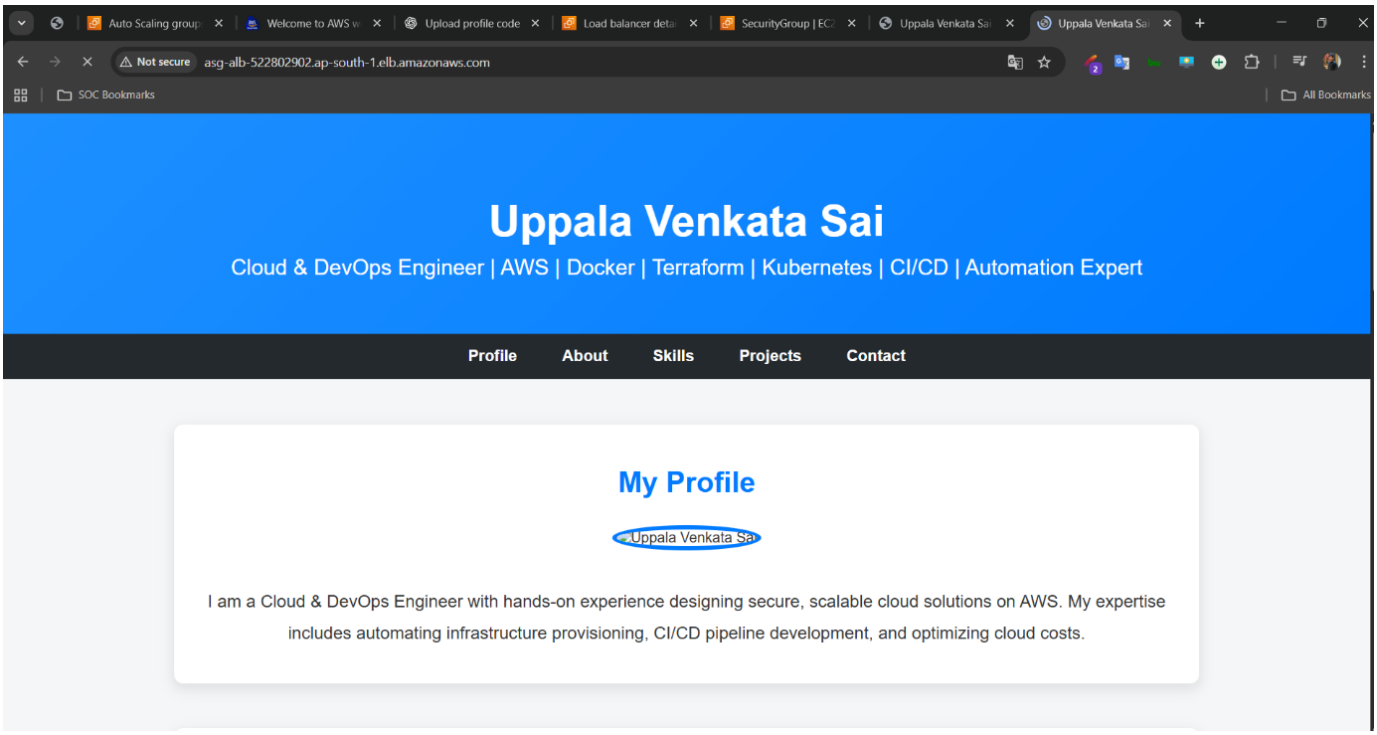
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6 IPs	Monitoring	Secur
	i-0789eddc058fd15c7	Running	c6a.large	3/3 checks passed	View alarms +	ap-south-1b	–	–	–	–	disabled	web-s
	i-040ccef17d8e7c69d	Running	c6a.large	3/3 checks passed	View alarms +	ap-south-1b	–	–	–	–	disabled	web-s
	i-00a9cbb9dce0e8bde	Running	c5a.large	Initializing	View alarms +	ap-south-1a	ec2-13-127-48-191.ap-...	13.127.48.191	–	–	disabled	web-s
	i-0a870ec2b3956ee33	Running	c5a.large	Initializing	View alarms +	ap-south-1c	ec2-13-200-21-130.ap-...	13.200.21.130	–	–	disabled	web-s
	i-04809380fcc90afac	Terminated	c6a.large	–	View alarms +	ap-south-1b	–	–	–	–	disabled	–
	i-05ad6213a764cfb13	Terminated	c6a.large	–	View alarms +	ap-south-1b	–	–	–	–	disabled	–
base-server	i-07e389a38bcd15dbc	Terminated	t2.micro	–	View alarms +	ap-south-1a	–	–	–	–	disabled	–
	i-032e059d02f72fe73	Terminated	c5a.large	–	View alarms +	ap-south-1a	–	–	–	–	disabled	–
	i-031904d6442f7001c	Terminated	c5a.large	–	View alarms +	ap-south-1c	–	–	–	–	disabled	–

In this lab, Auto Scaling was successfully demonstrated using Amazon EC2 instances. The **EC2 console** shows a total of **9 instances**, with multiple instances being launched and terminated automatically according to the scaling policies.

- **Running Instances:**
  - 4 instances (`c6a.large` and `c5a.large` types) are currently in the *Running* state.
  - Two of them (`i-0789ed...` and `i-040cce...`) have passed the **3/3 status checks**, indicating they are fully operational.
  - The other two (`i-00a9cb...` and `i-0a87ce...`) are in the **Initializing** state, showing that Auto Scaling has just launched them as part of the scaling activity.
- **Terminated Instances:**
  - 5 instances have been terminated (`c6a.large`, `c5a.large`, and `t2.micro`), which confirms that Auto Scaling removed excess capacity when it was no longer needed.
- **Availability Zones:**
  - Instances are distributed across multiple AZs (`ap-south-1a`, `ap-south-1b`, and `ap-south-1c`) for high availability and fault tolerance.
- **Public IPs:**
  - Some running instances have public IPv4 addresses assigned (`13.127.48.191` and `13.200.21.130`), making them accessible over the internet.

This file contains the full HTML code for the webpage. You can use the guide provided earlier to place this code on your EC2 instance. This should complete the test and confirm that your infrastructure is working correctly.

Linkedin : https://www.linkedin.com/in/uppala-venkata-sai/
Github : https://github.com/UppalavenkataSai



Linkedin : https://www.linkedin.com/in/uppala-venkata-sai/
Github : https://github.com/UppalavenkataSai