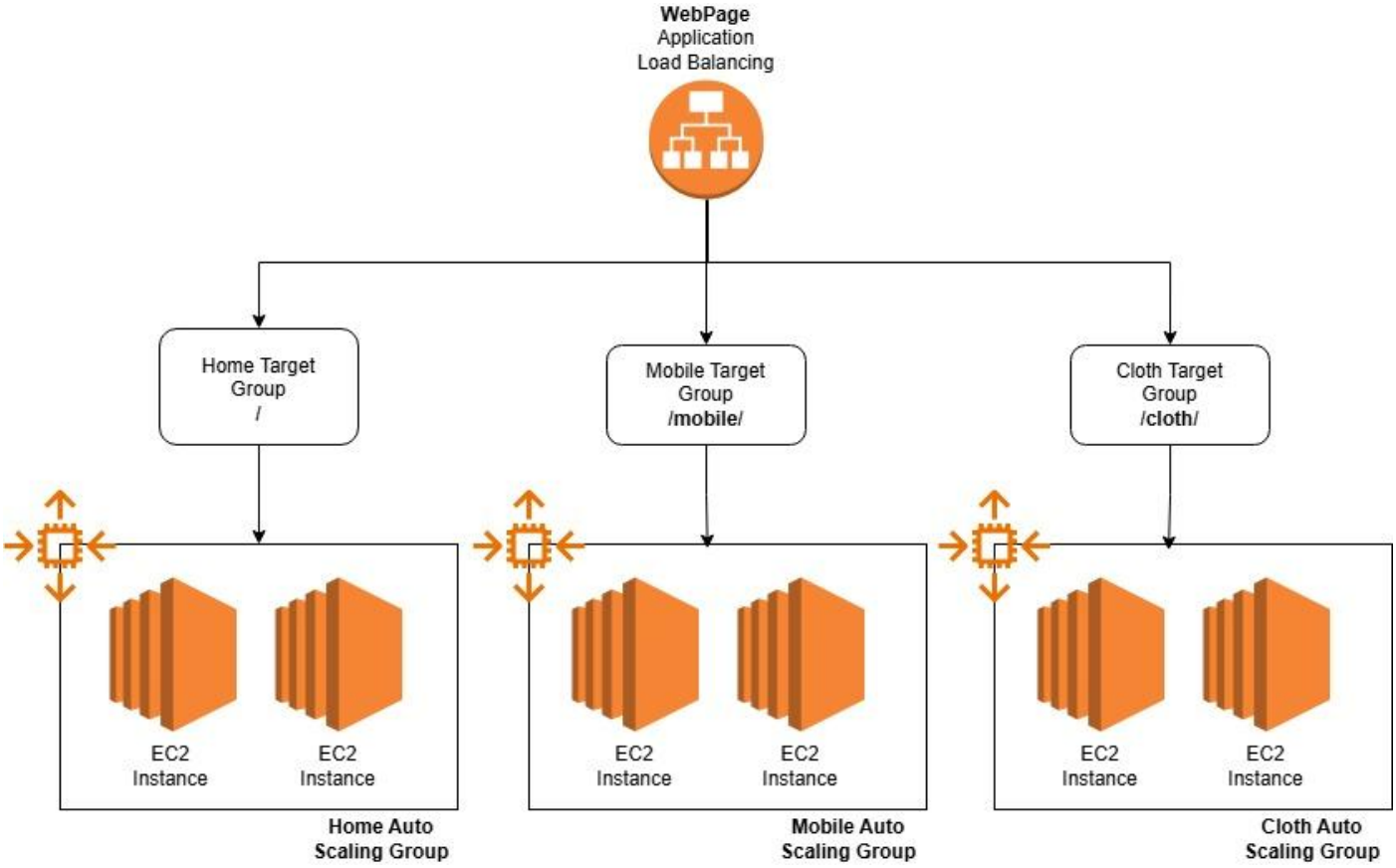
Path-Based Web Application on AWS using EC2 Auto Scaling and Application Load Balancer

By: Nikhilesh Sakhare

Overall layout

The entry point is an internet-facing Application Load Balancer (ALB) that exposes a single HTTP endpoint to users. Behind the ALB, there are three independent Auto Scaling Groups, each running a fleet of Amazon EC2 instances that serve a specific part of the website: home, mobile, and cloth. All resources are deployed in the same VPC across multiple subnets and Availability Zones to improve availability and fault tolerance.



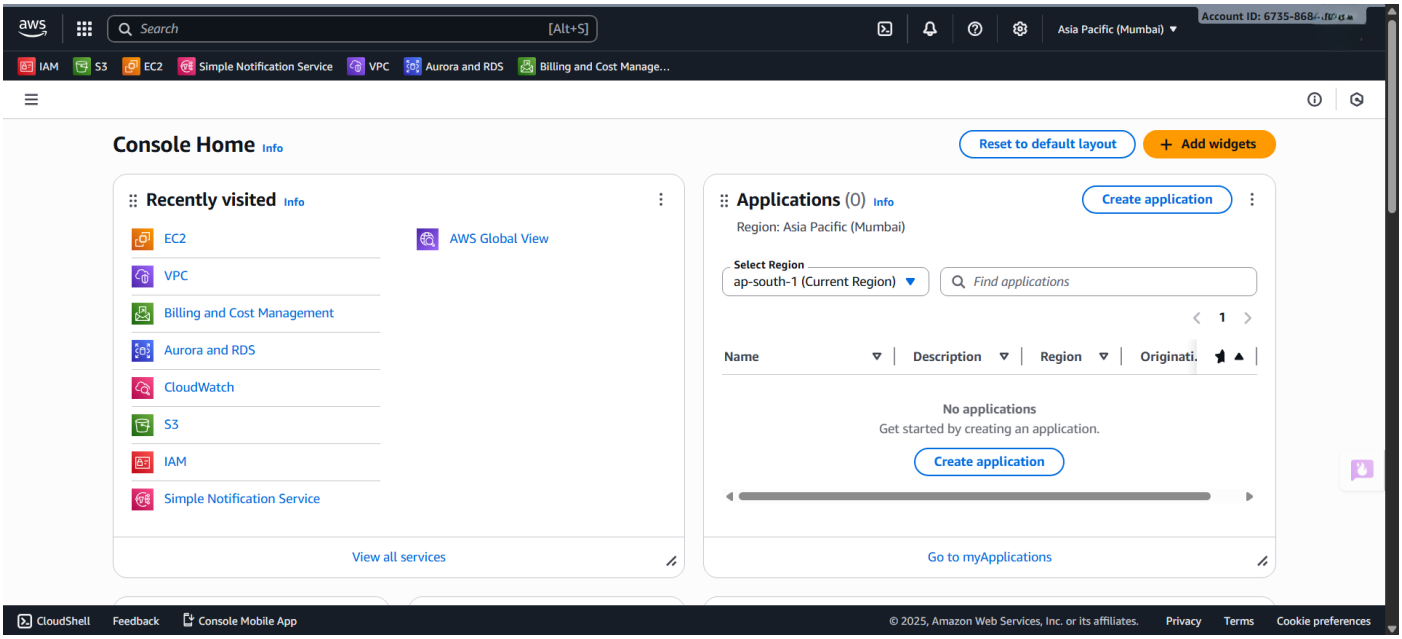
Auto Scaling Groups and instances

Each Auto Scaling Group (home-as, mobile-as, cloth-as) is created from its own EC2 launch template, which defines the Amazon Linux AMI, instance type (t3.micro), security group, and user-data script. The user-data on each template installs Apache and writes a different HTML page so that the response clearly identifies the section being served (home, mobile, or cloth). The groups are configured with minimum, desired, and maximum capacities and can use target-tracking scaling based on average CPU utilization to automatically adjust the number of running instances under load.

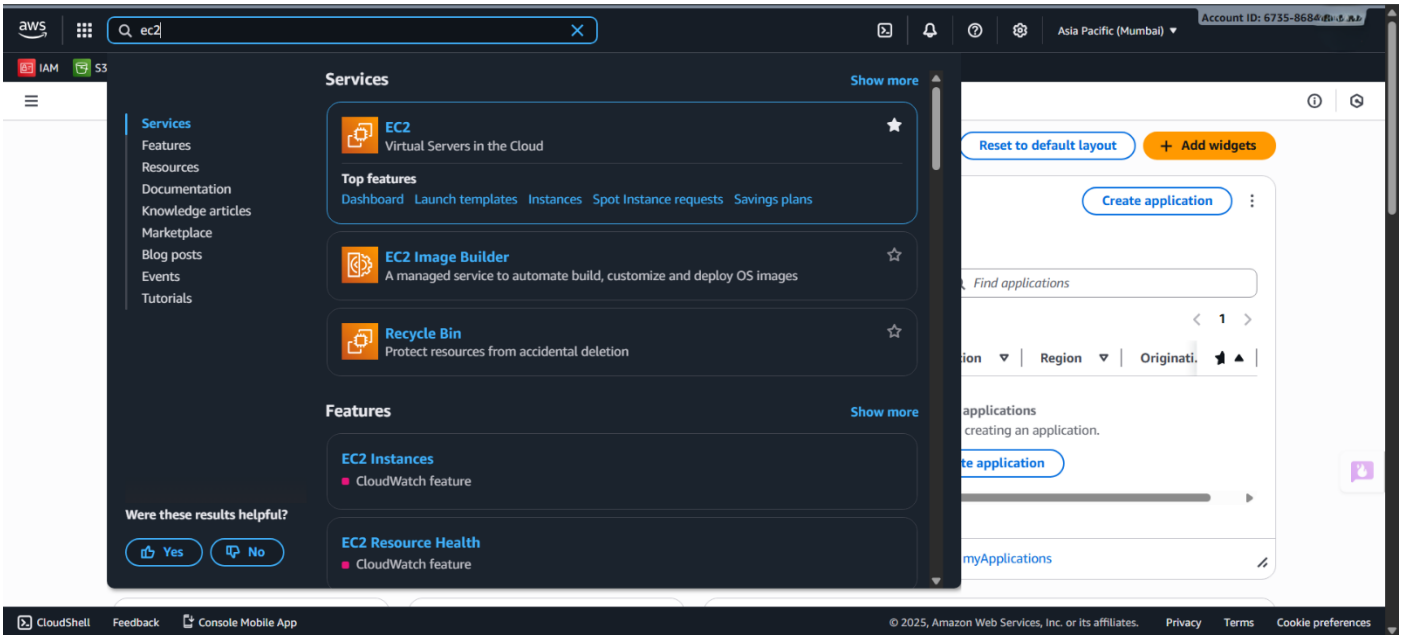
Load balancer, target groups, and routing

The ALB is attached to the same VPC and spread across three public subnets, and it uses a shared security group (MyWebSG) that allows HTTP traffic on port 80. Three HTTP instance-target groups (home-tg, mobile-tg, cloth-tg) register the EC2 instances from their respective Auto Scaling Groups and handle health checks. The ALB's HTTP:80 listener uses a default rule that forwards root path *'/'* traffic to home-tg, and additional path-based rules that forward *'/mobile/*'* to mobile-tg and *'/cloth/*'* to cloth-tg. This allows a single DNS name from the load balancer to serve all three logical applications while keeping their compute layers isolated and independently scalable.

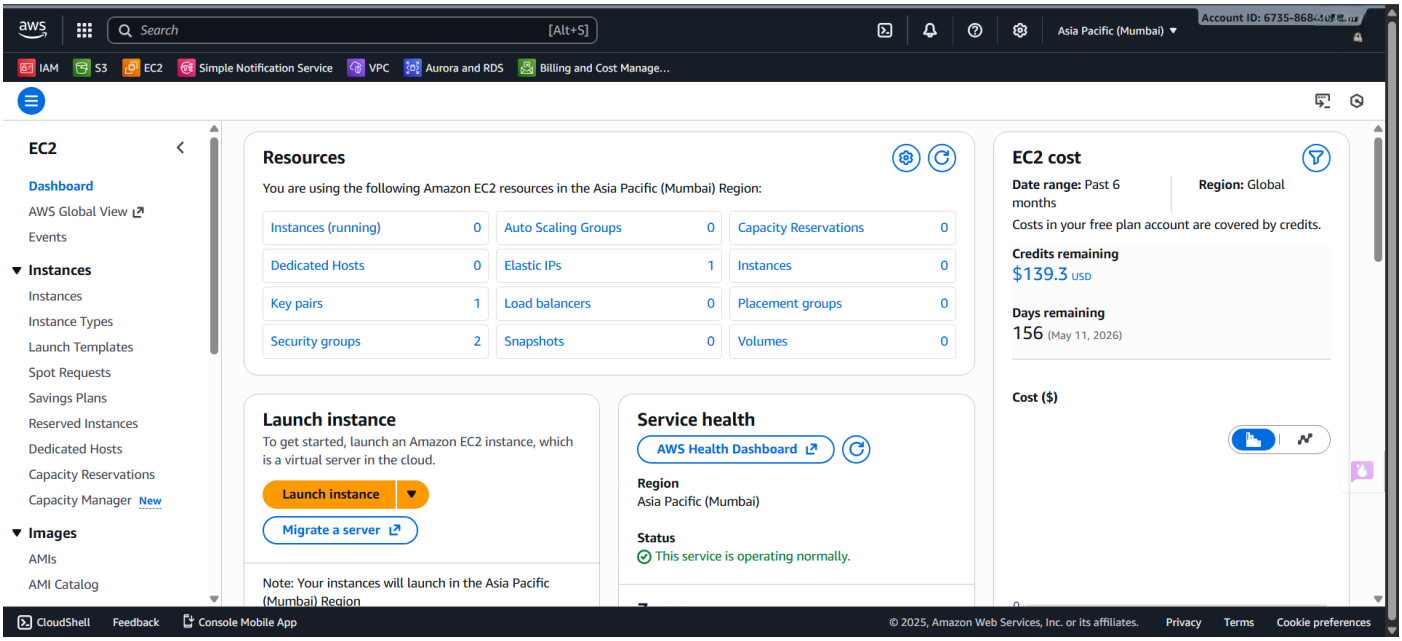
1. Logged in to the AWS Management Console and verified the working region as Asia Pacific (Mumbai) from the console home page.



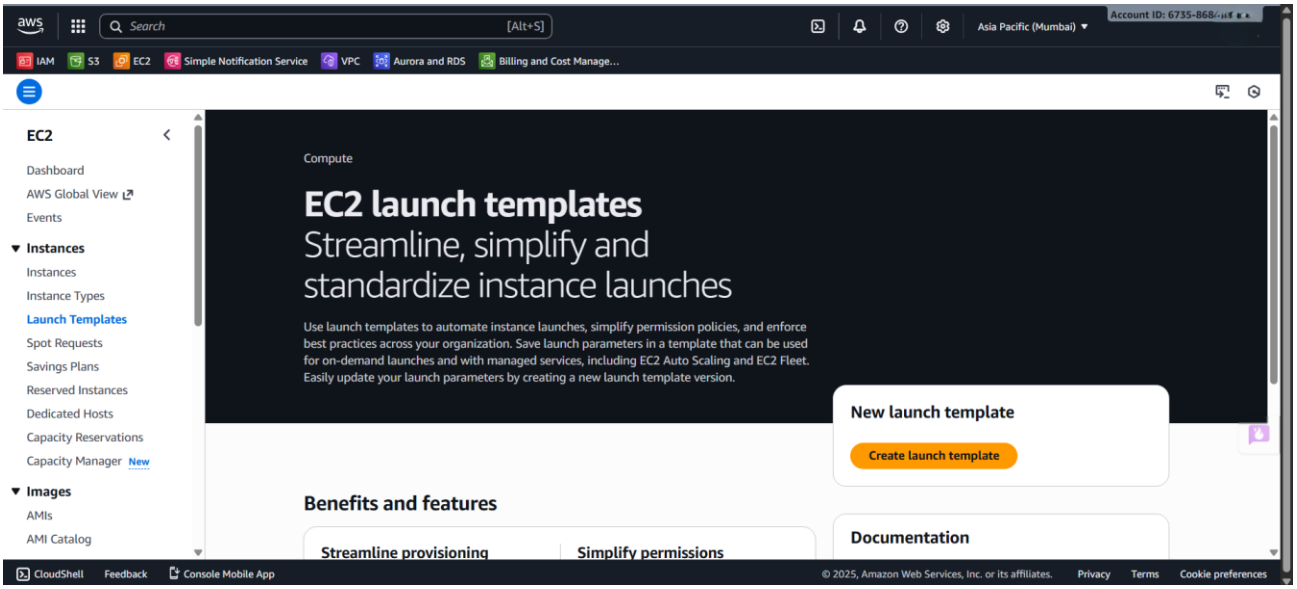
2. Opened the EC2 service from the search bar and reviewed the EC2 dashboard showing zero running instances and no existing load balancers or Auto Scaling Groups.

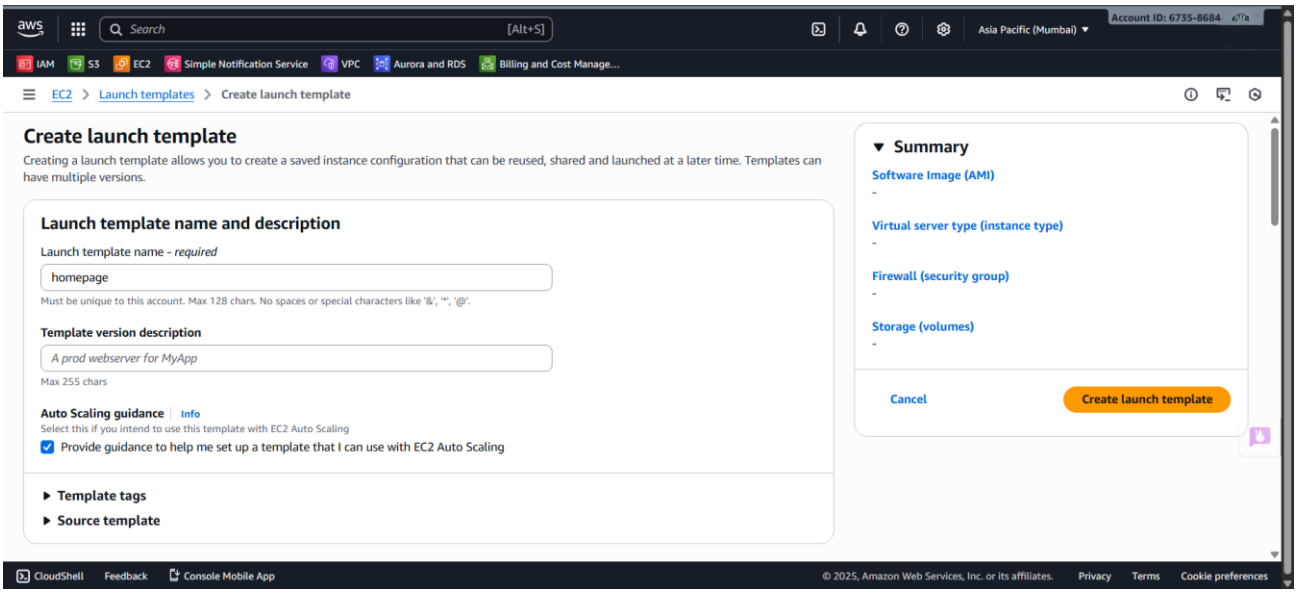


3. From the EC2 left navigation pane, selected “Launch Templates” to start defining reusable instance configurations.

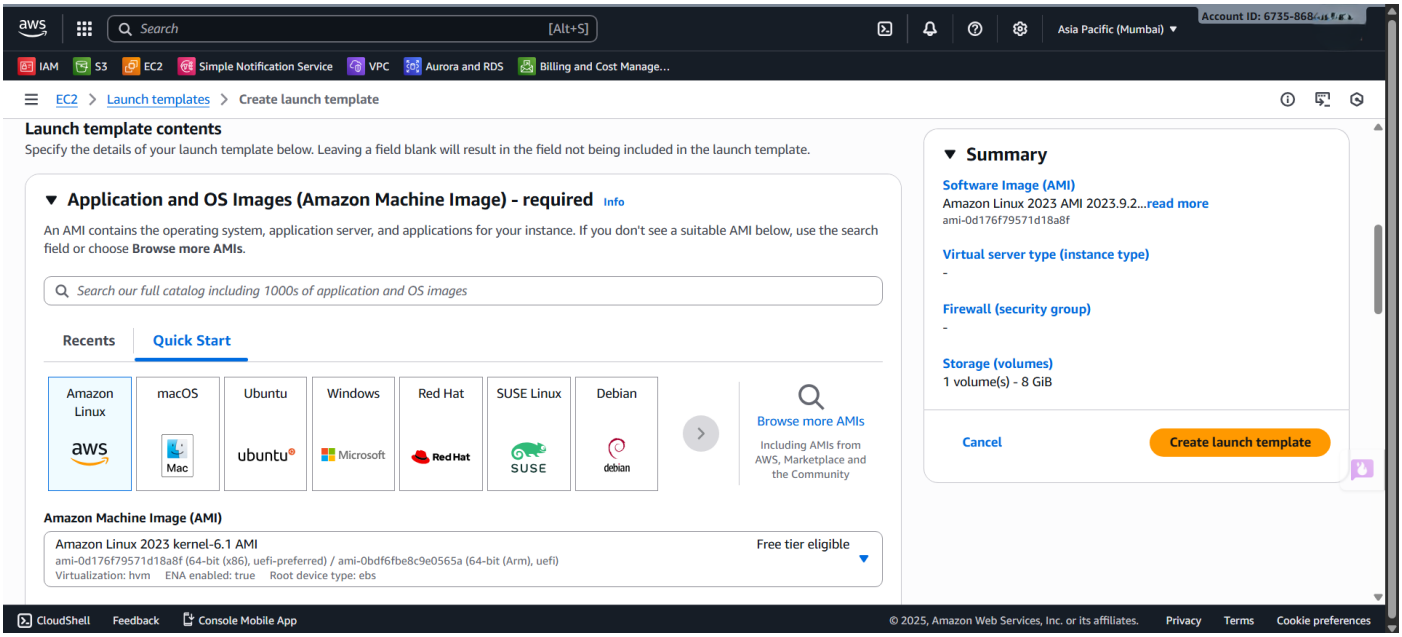


4. Clicked “Create launch template” and entered the template name homepage with an appropriate description, enabling Auto Scaling guidance.

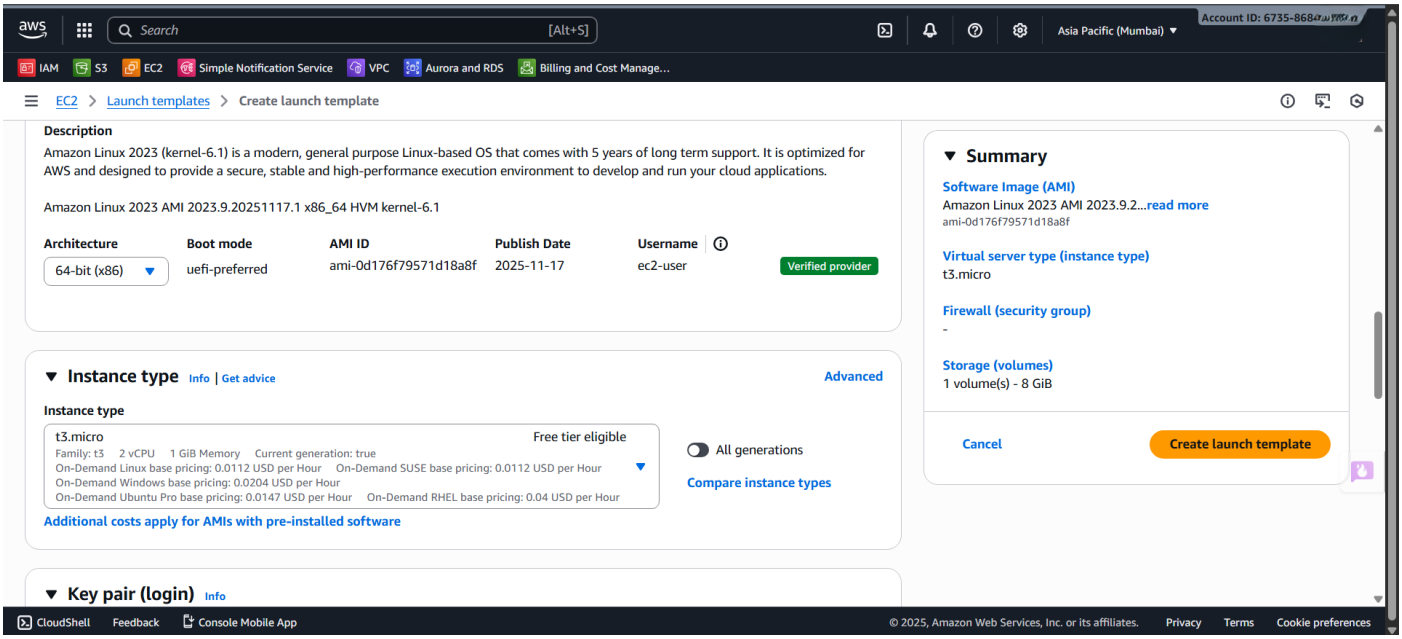




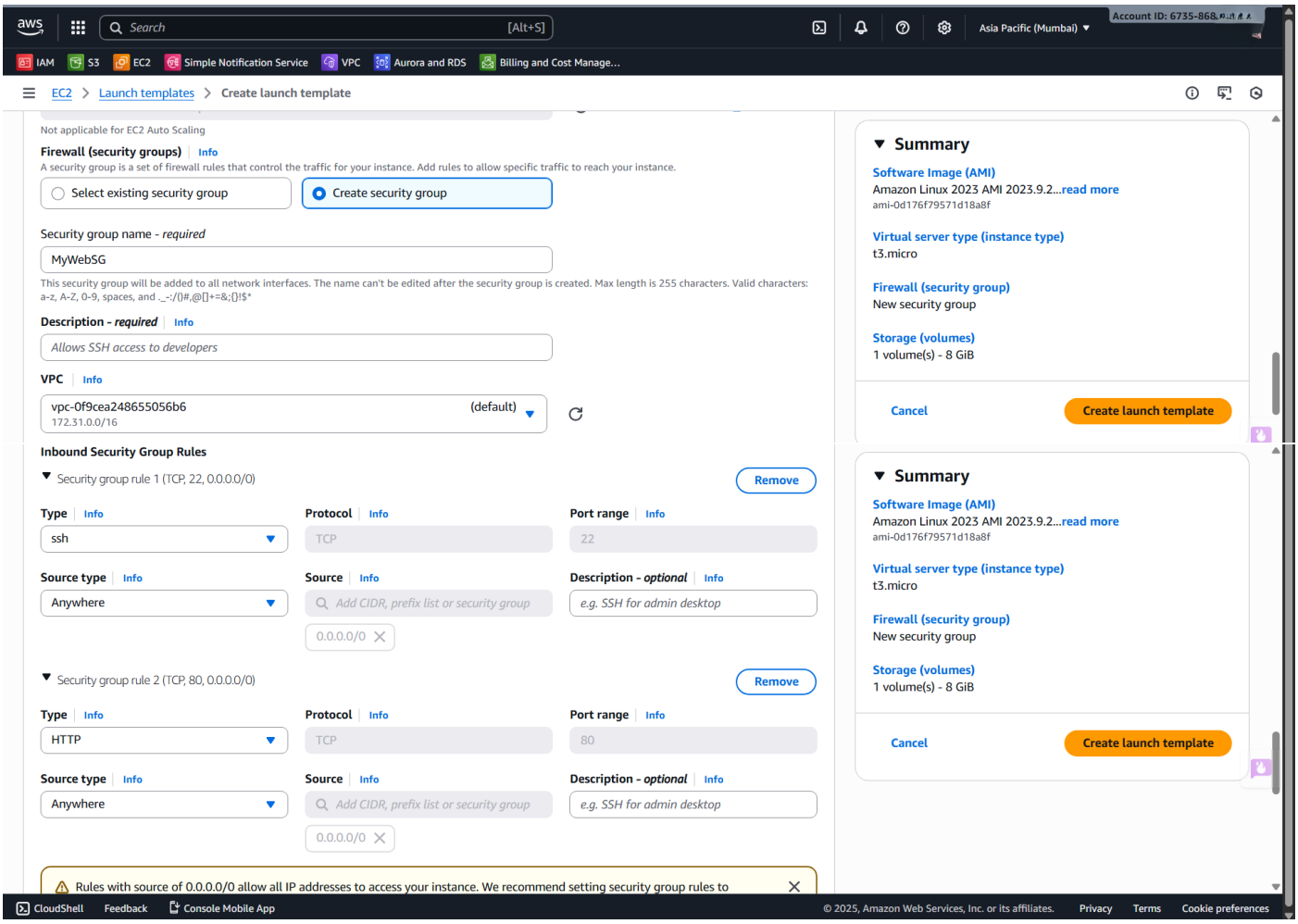
5. In the “Application and OS Images” section, selected the Amazon Linux 2023 AMI from the Quick Start list.



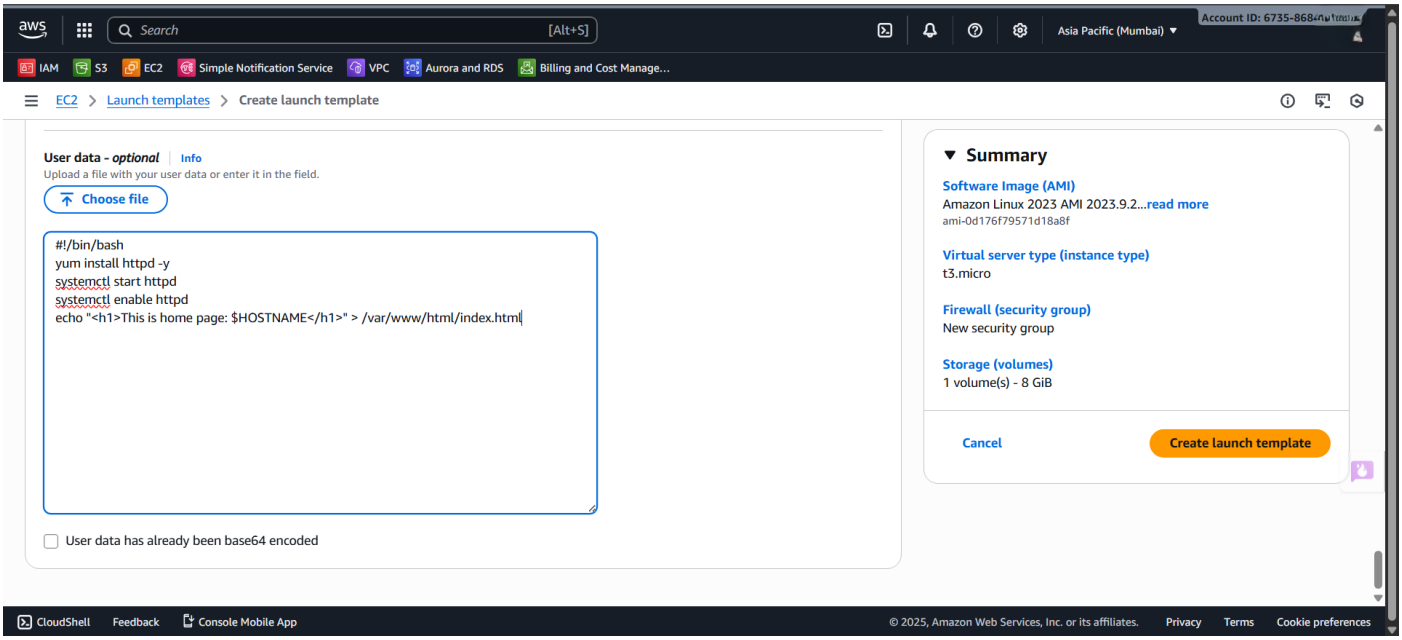
6. Chose instance type t3.micro, which is free-tier eligible, for all web servers.



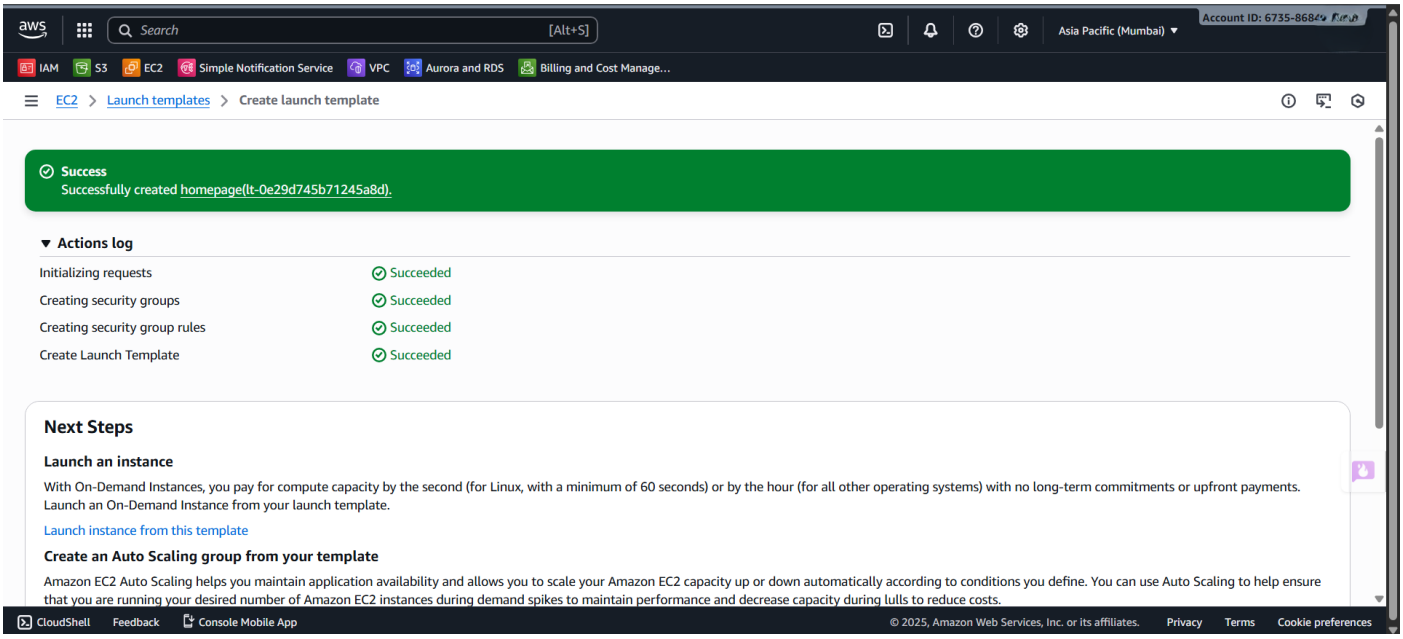
7. In the key pair section kept the default selection as per the account, and then moved to the security group configuration where “Create security group” was selected.
   Created a new security group named MyWebSG, added a description, associated it with the default VPC, and configured two inbound rules: SSH (port 22) from Anywhere and HTTP (port 80) from Anywhere so that the web pages are reachable over the internet.



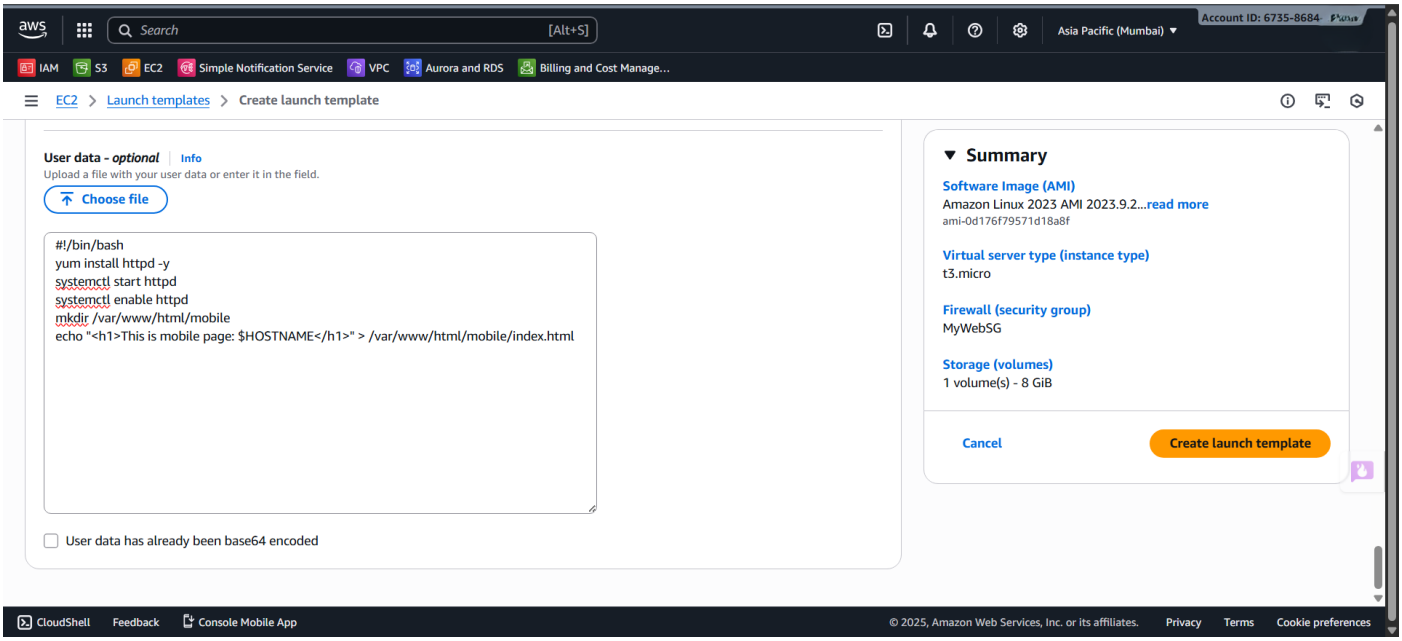
8. Scrolled to the “User data” section and pasted a shell script to install and start Apache (httpd) and to create /var/www/html/index.html with a simple “This is home page: $HOSTNAME” message.



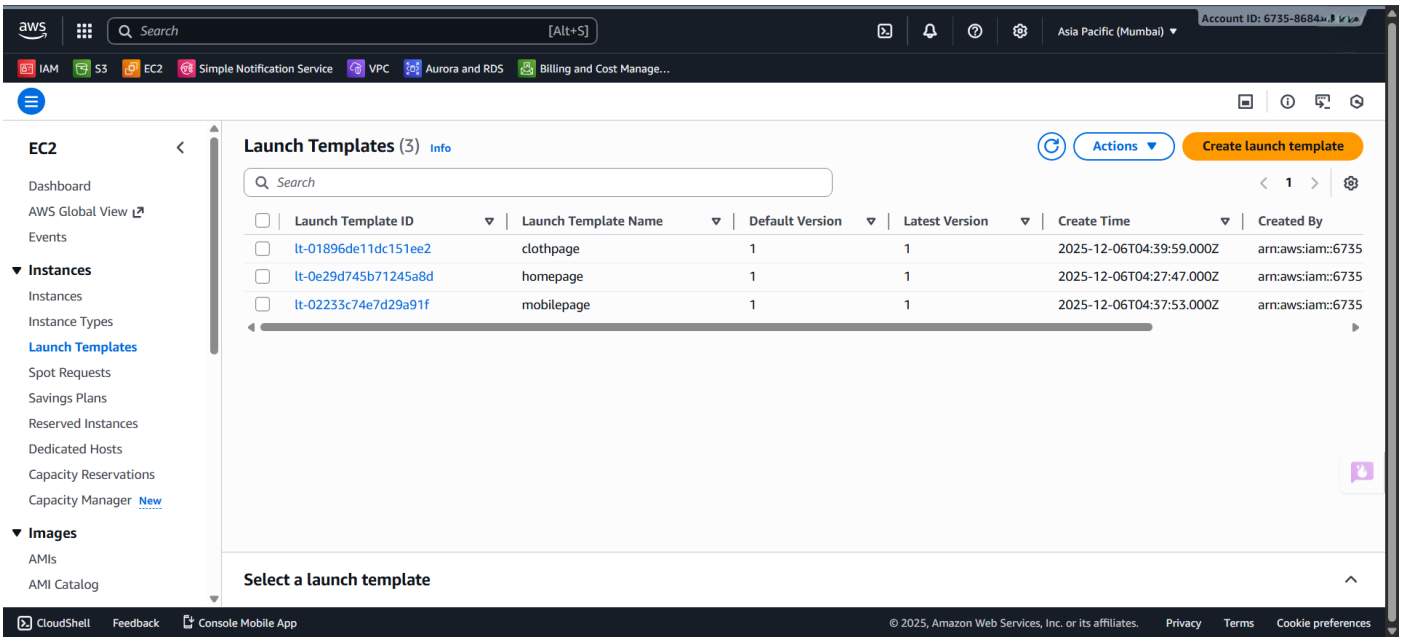
9. Reviewed the summary on the right-hand side and clicked “Create launch template”, confirming that the homepage launch template and associated security group were created successfully.



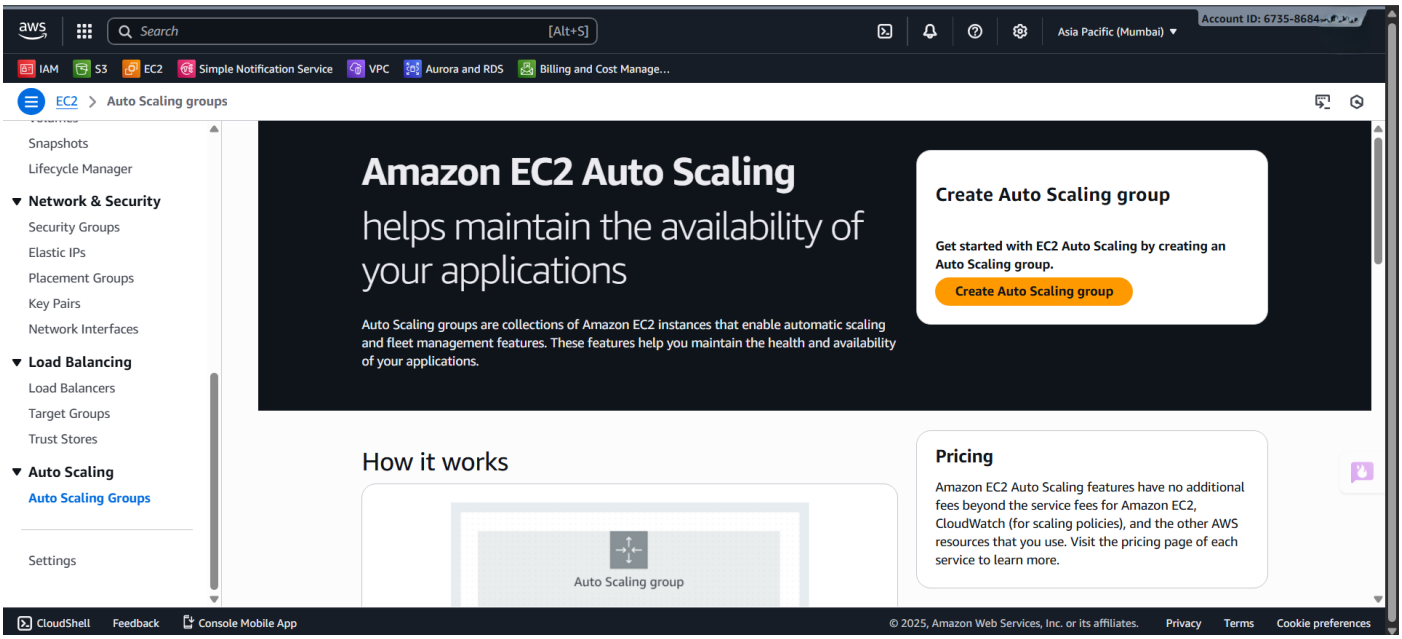
10. Repeated the “Create launch template” workflow to create a second launch template named mobilepage. Reused the same AMI, instance type, and security group MyWebSG but changed the user data script to create /var/www/html/mobile/index.html with “This is mobile page: $HOSTNAME”, creating the /mobile directory first.



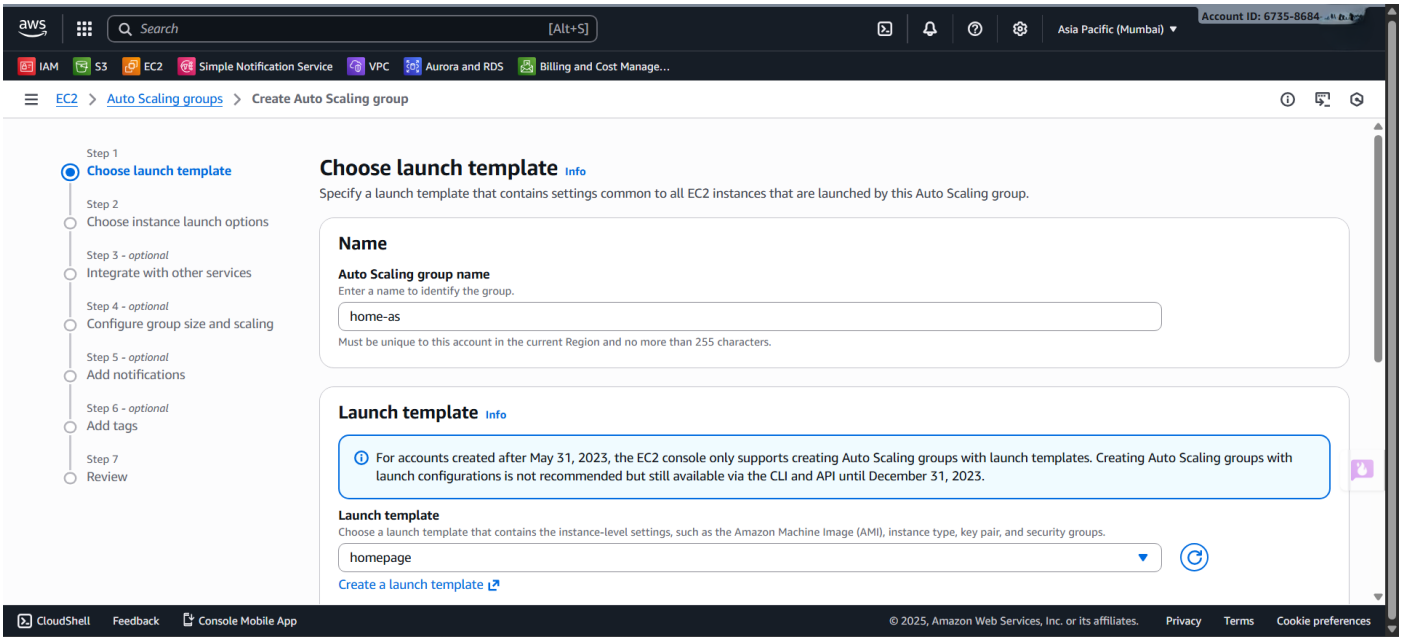
11. Performed the same steps again to create a third launch template named clothpage, with user data that installs Apache and creates /var/www/html/cloth/index.html to serve the “cloth” page.
    After all three templates were created, the “Launch Templates” page showed homepage, mobilepage, and clothpage as separate launch templates ready to be used by Auto Scaling Groups.



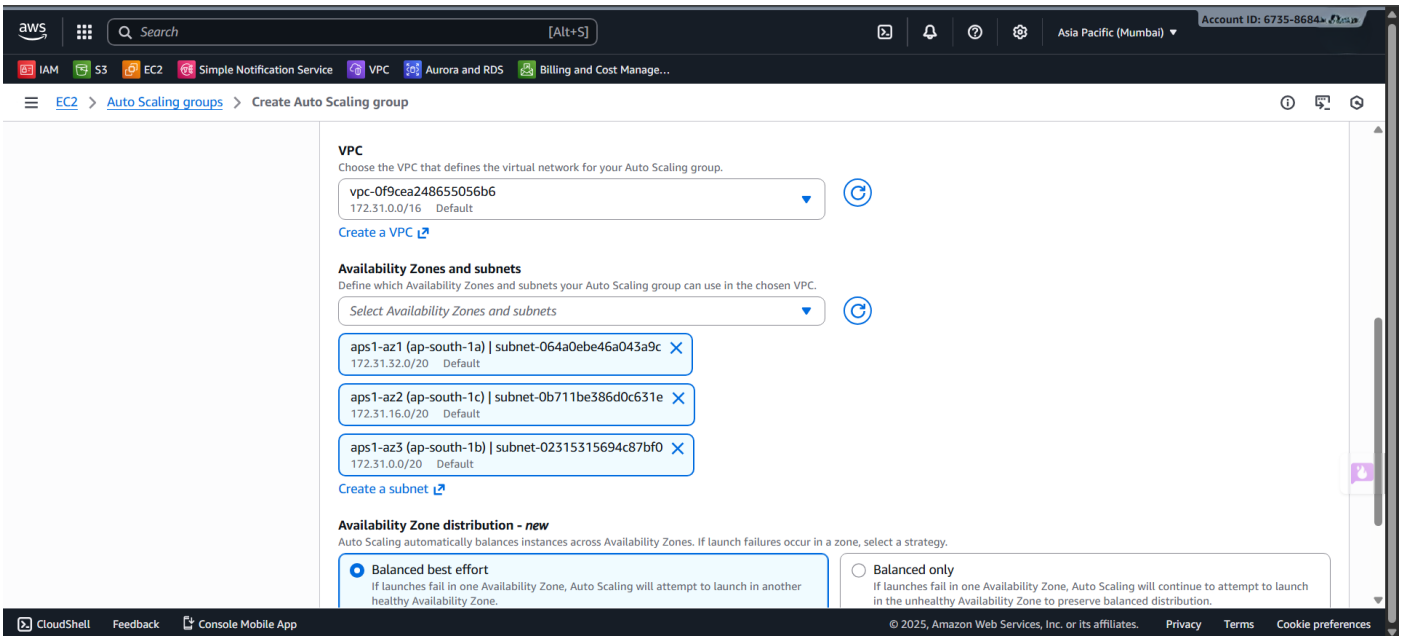
12. From the EC2 navigation pane, opened “Auto Scaling groups” and clicked “Create Auto Scaling group”.



13. Provided the Auto Scaling group name home-as and selected the homepage launch template that was created earlier.

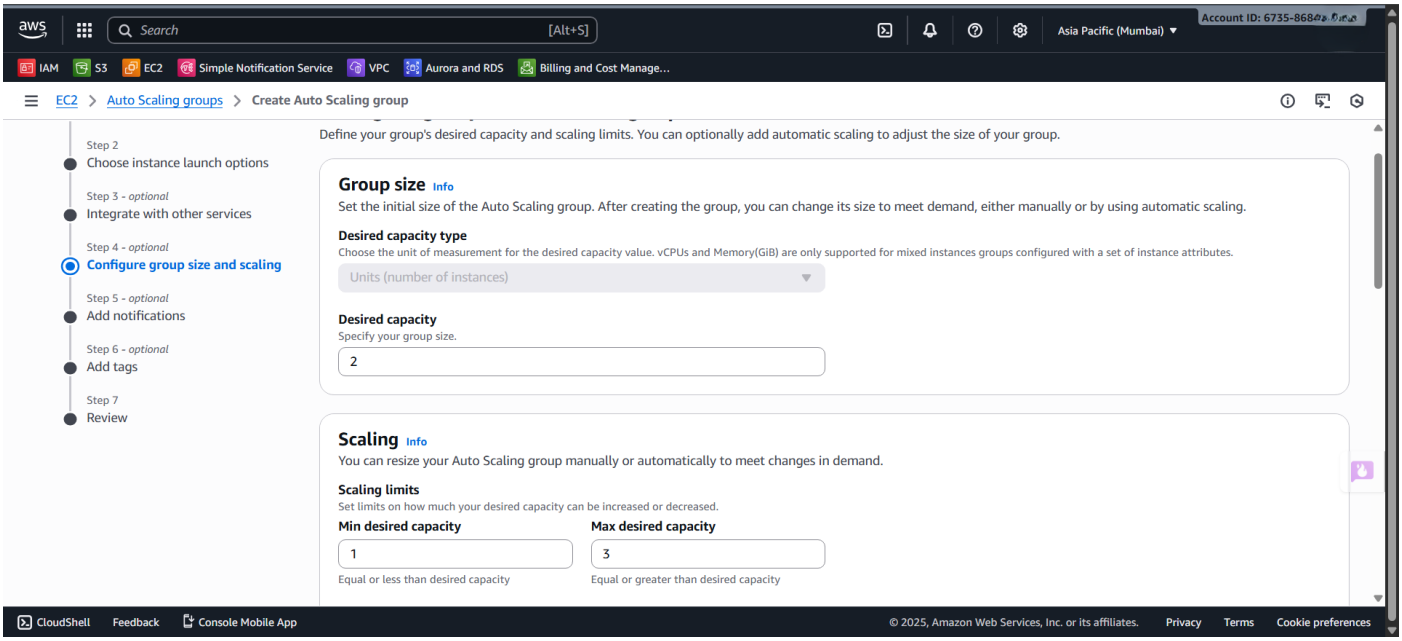


14. On the “Choose instance launch options” step, selected the default VPC and enabled all three available subnets in different Availability Zones (ap-south-1a, ap-south-1b, ap-south-1c) to achieve high availability for the home servers.

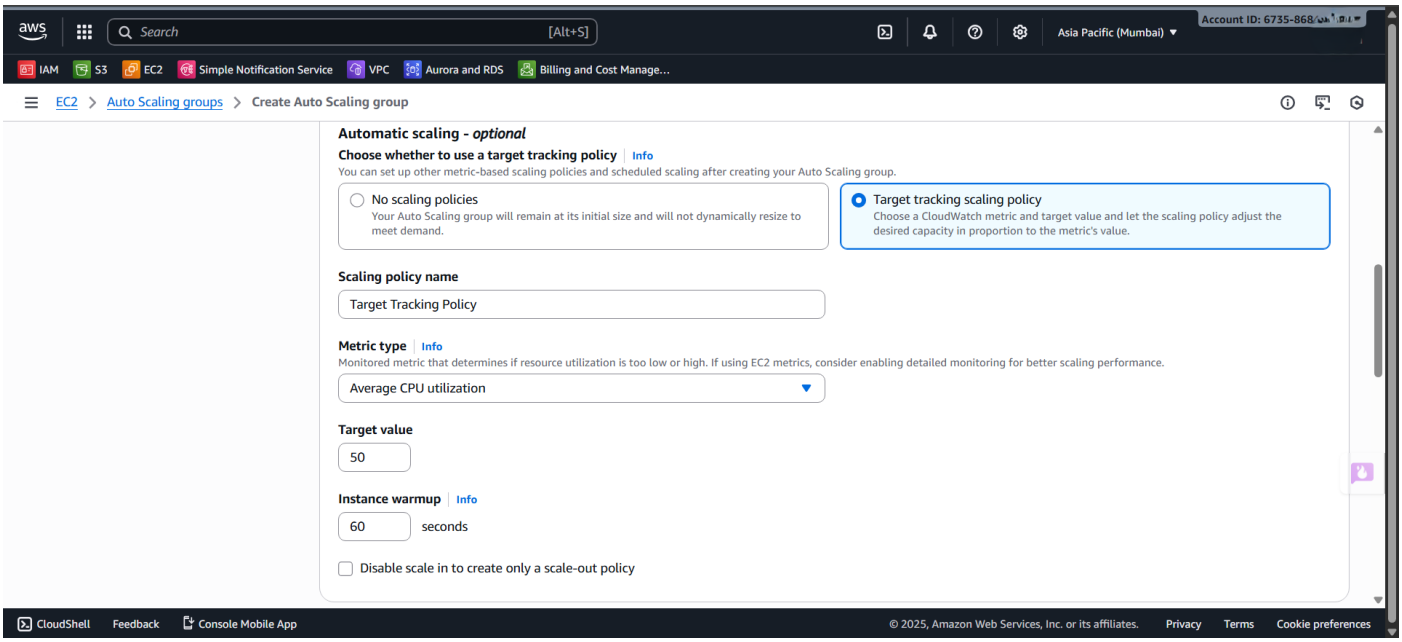


On the same wizard, ensured that the instances launched by this group would inherit the security group MyWebSG from the launch template.

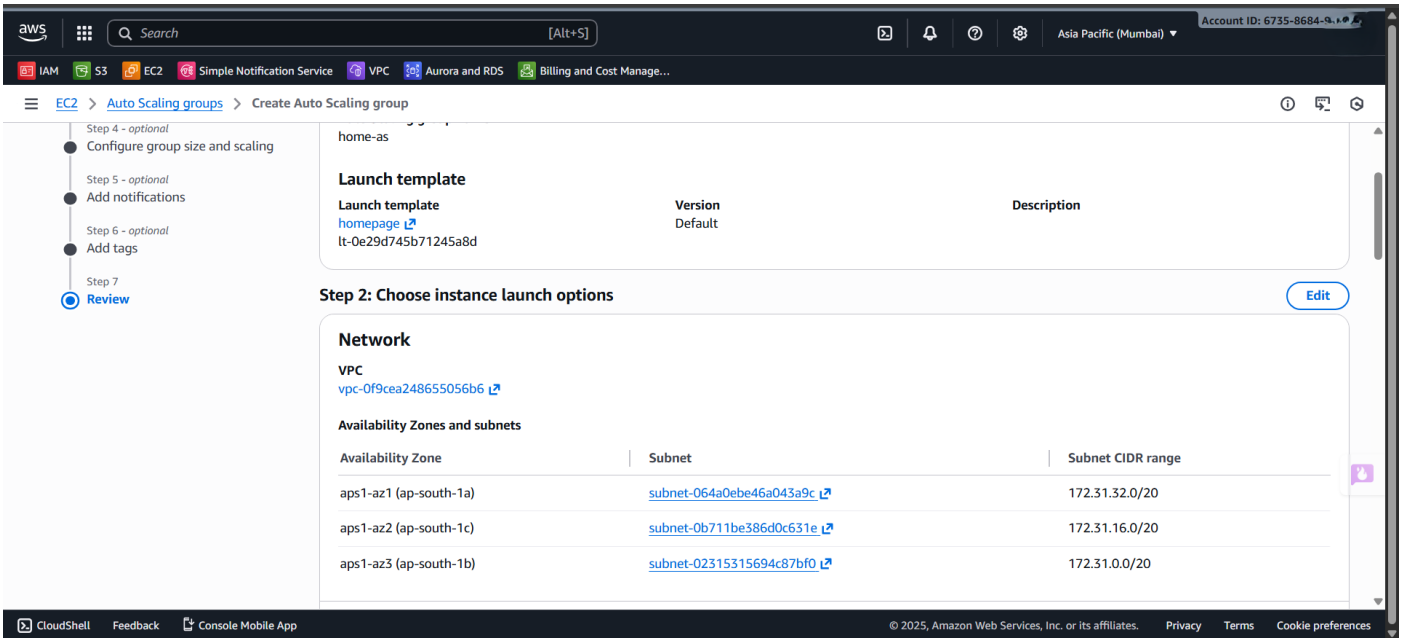
15. On the “Configure group size and scaling” step, set the desired capacity to 2, the minimum capacity to 1, and the maximum capacity to 3 so that the group can scale in and out within that range.



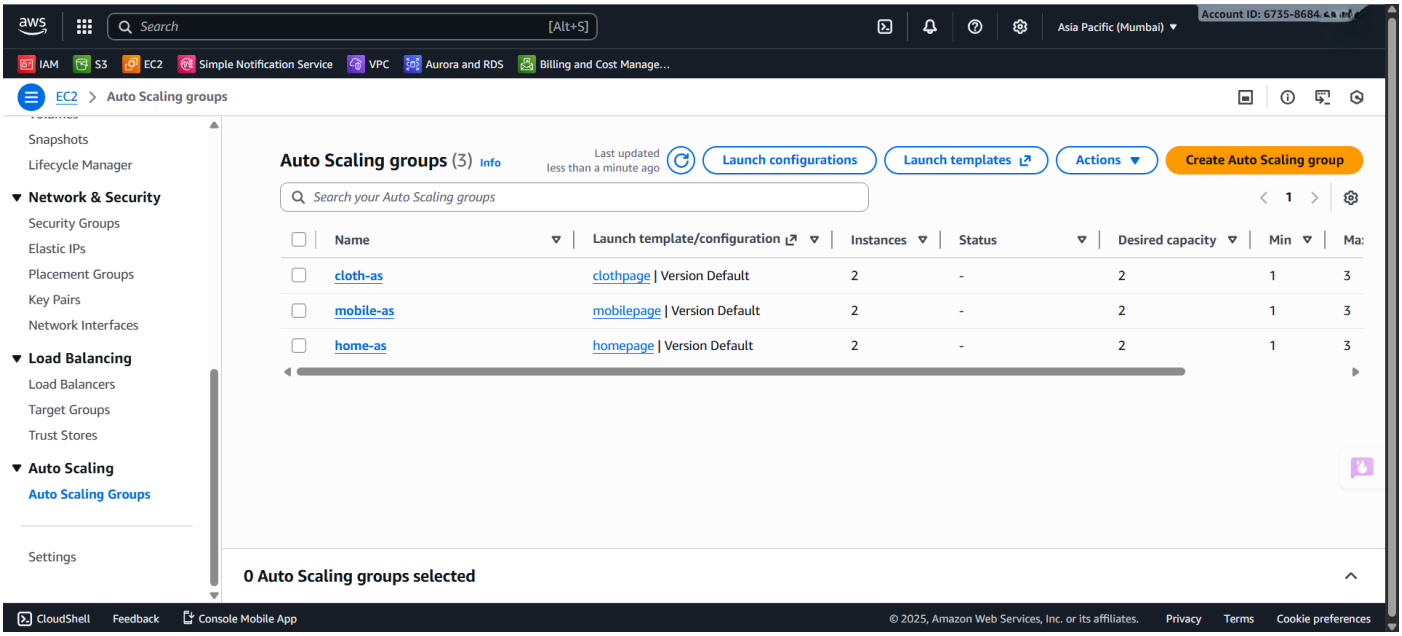
16. Enabled a target tracking scaling policy based on Average CPU utilization with a target value of 50 percent and a default instance warm-up time of 60 seconds, allowing the group to add or remove instances automatically based on CPU load.



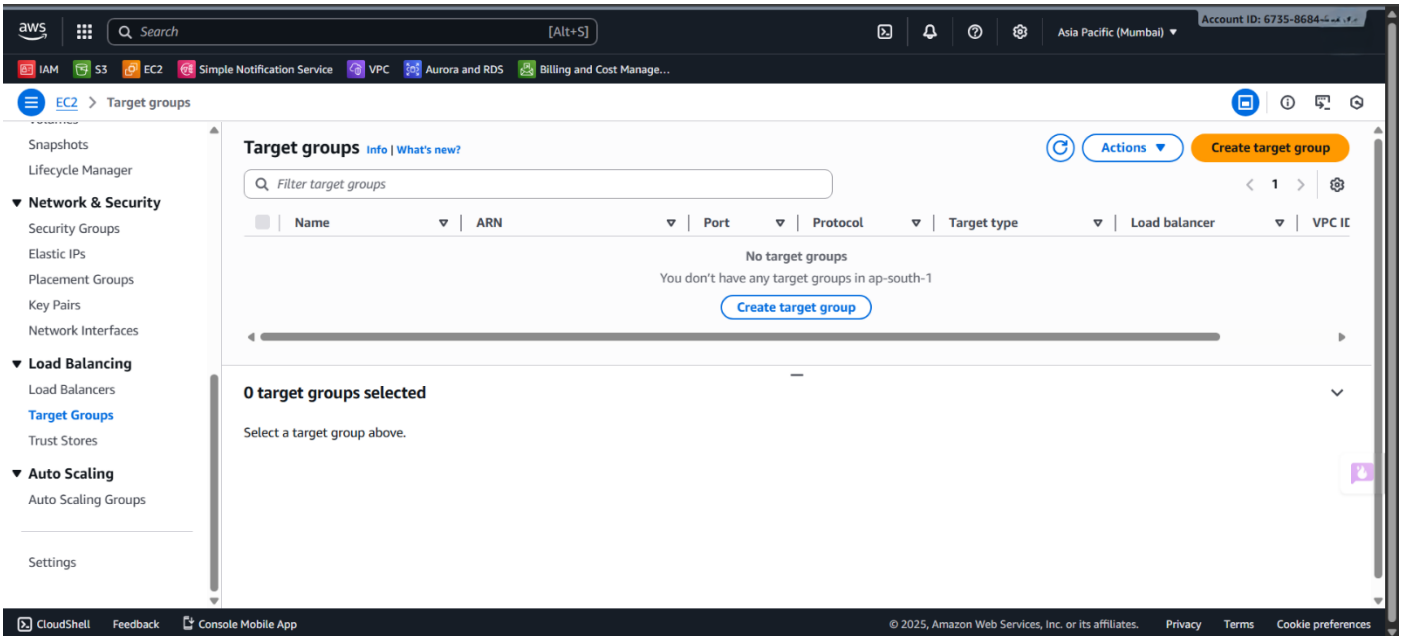
17. Reviewed the configuration summary, confirming the launch template, network, subnets, and scaling settings, then created the home-as Auto Scaling Group.



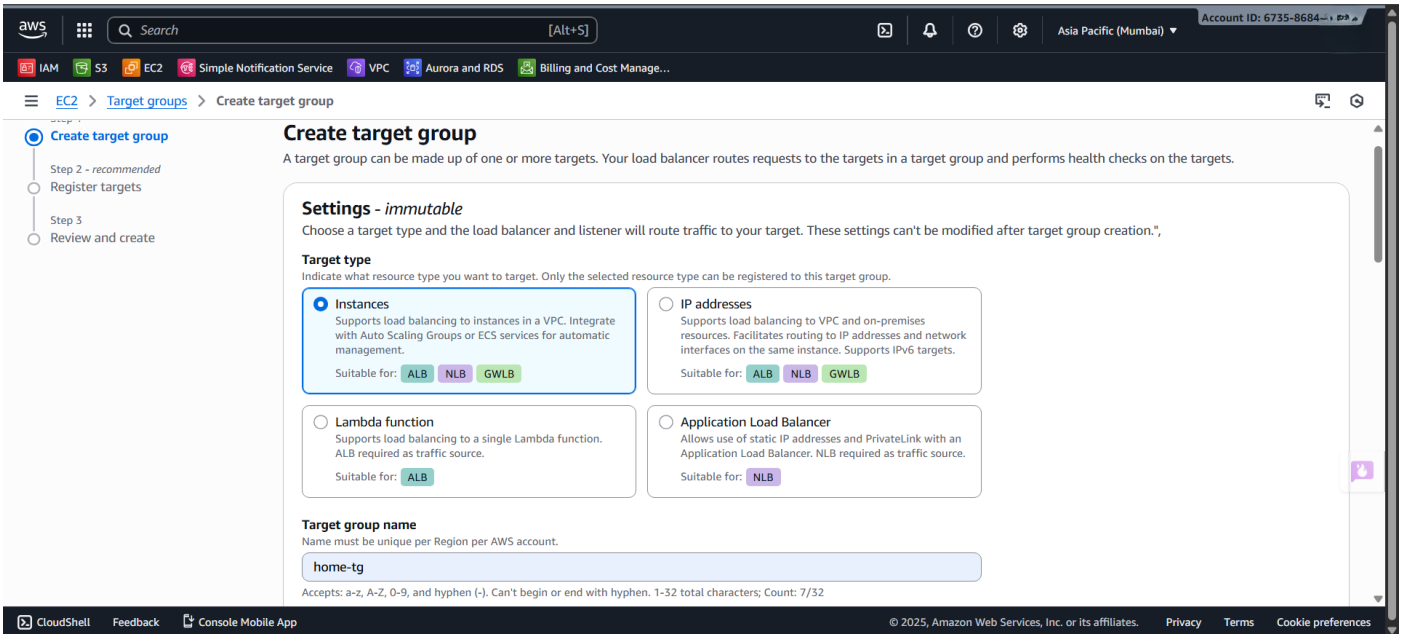
18. Repeated the Auto Scaling Group creation process to create mobile-as using the mobilepage launch template and cloth-as using the clothpage launch template, each with the same network, subnet, and scaling configuration.
    Returned to the “Auto Scaling groups” page and verified that three groups (home-as, mobile-as, cloth-as) were present, each with a desired capacity of 2 instances.



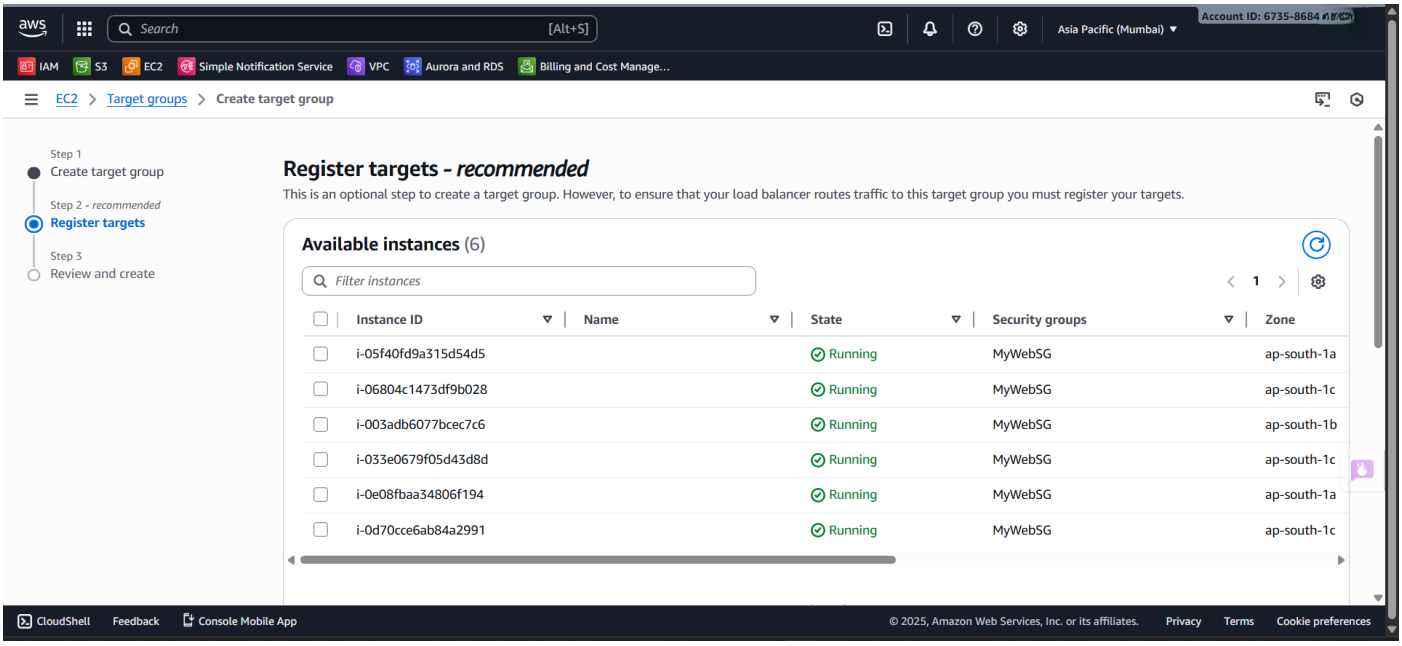
19. From the EC2 navigation pane under “Load Balancing”, selected “Target Groups” and confirmed that there were initially no target groups.



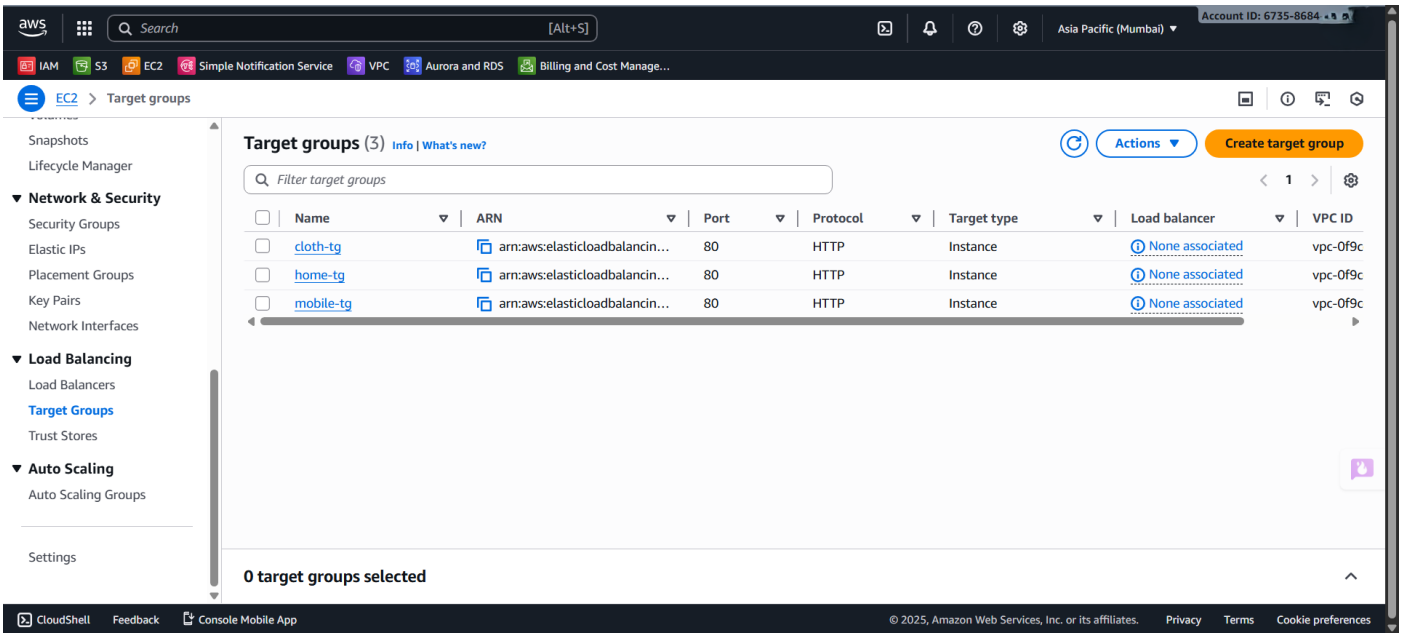
20. Clicked “Create target group”, kept the target type as “Instances”, and entered the first target group name home-tg.



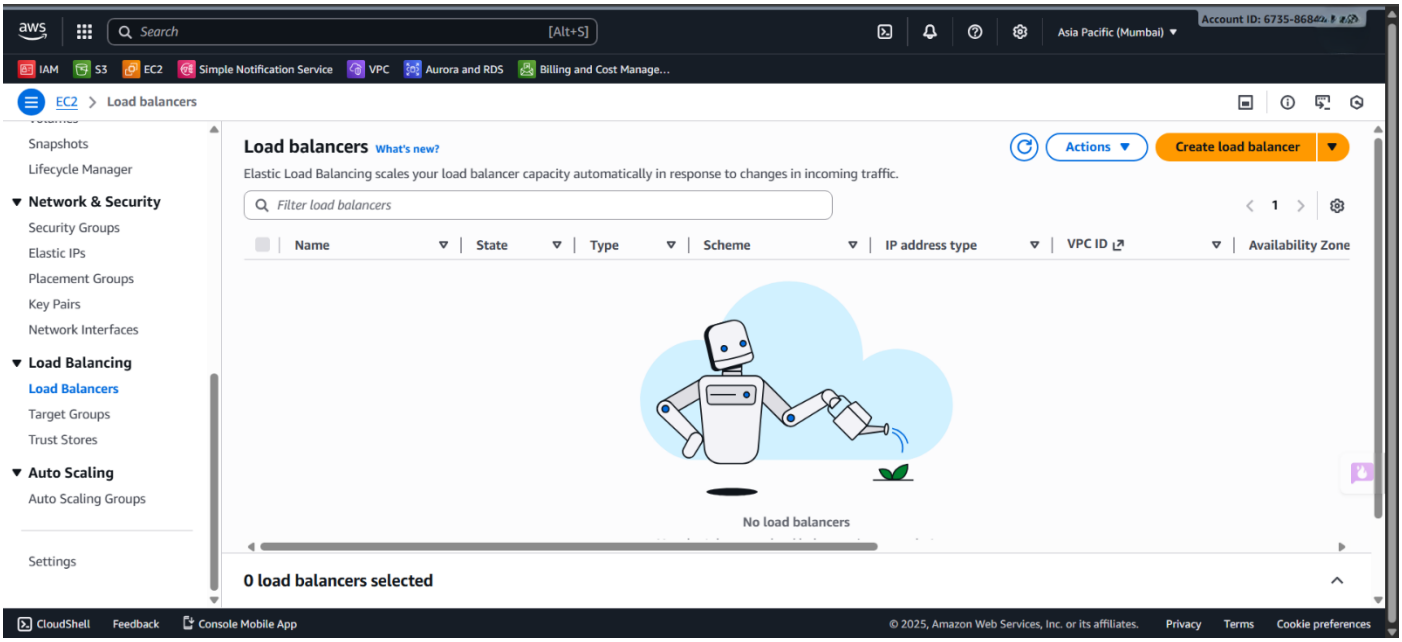
21. Completed the remaining settings (protocol HTTP, port 80, and default health check path) and moved to the "Register targets" step, where running EC2 instances created by the Auto Scaling Groups were visible.



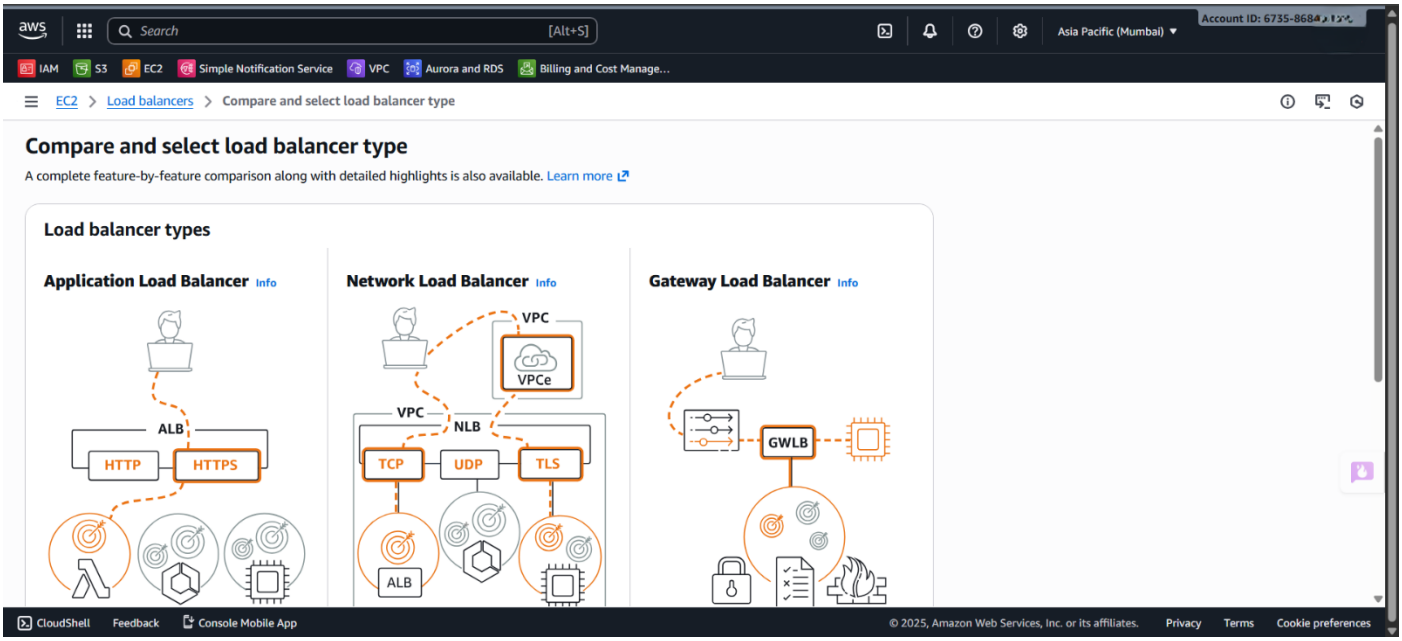
22. Don't select any instances for the home page target group now and then created additional target groups mobile-tg and cloth-tg by repeating the same steps.
After all three target groups were created, the Target Groups page showed home-tg, mobile-tg, and cloth-tg, each configured on HTTP port 80 and currently not yet associated with any load balancer.



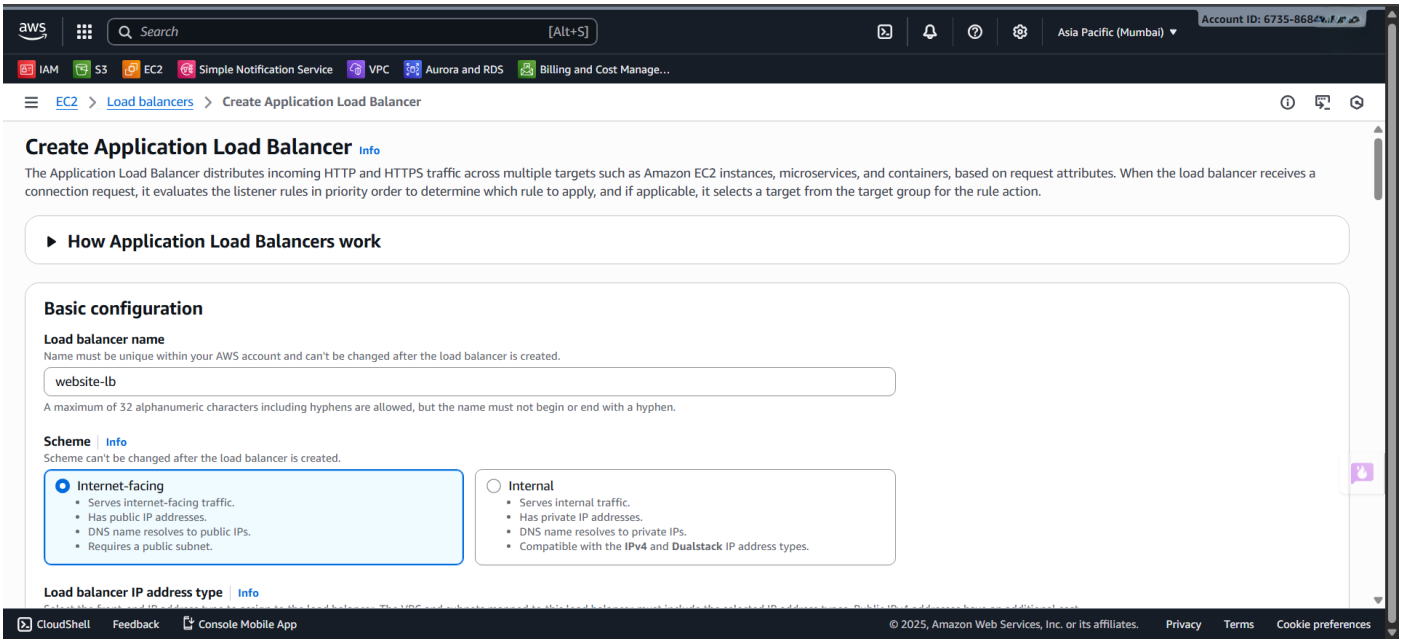
23. Navigated to “Load Balancers” under the EC2 “Load Balancing” section and verified that there was no existing load balancer.



24. Clicked “Create load balancer” and chose “Application Load Balancer” from the list of load balancer types.

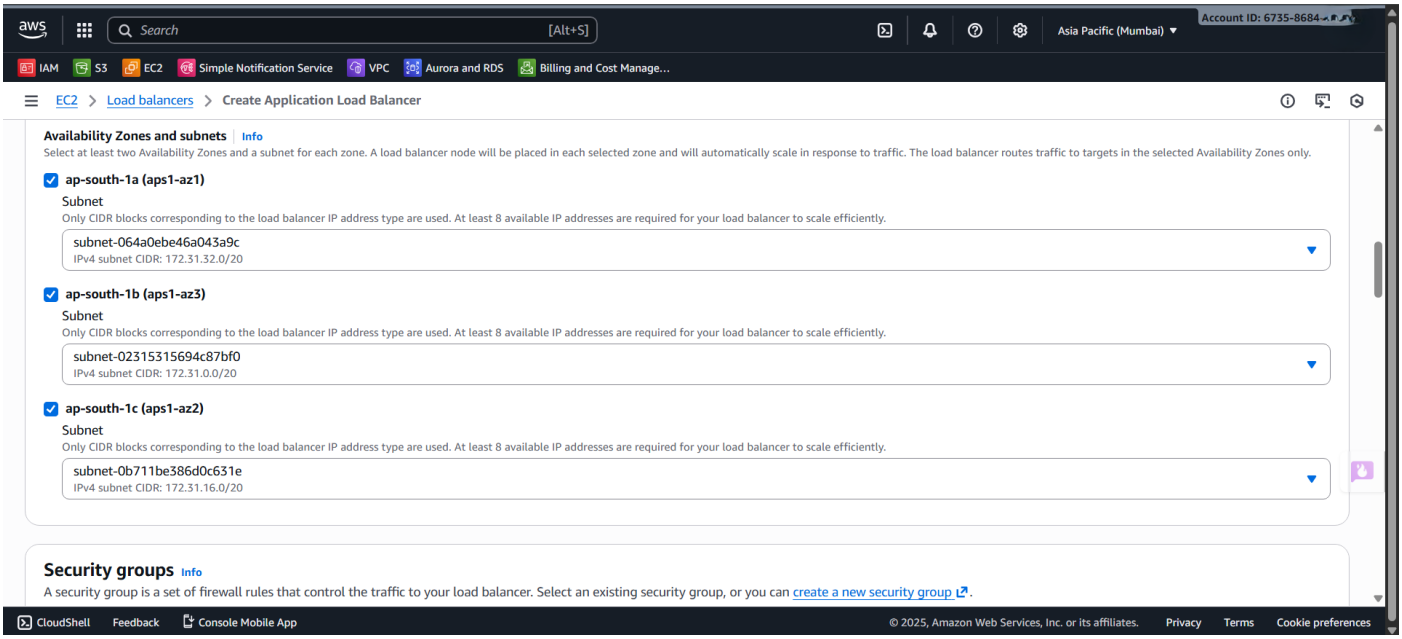


25. On the “Create Application Load Balancer” page, entered the load balancer name website-lb and selected the “Internet-facing” scheme so that the ALB can receive requests from the public internet.

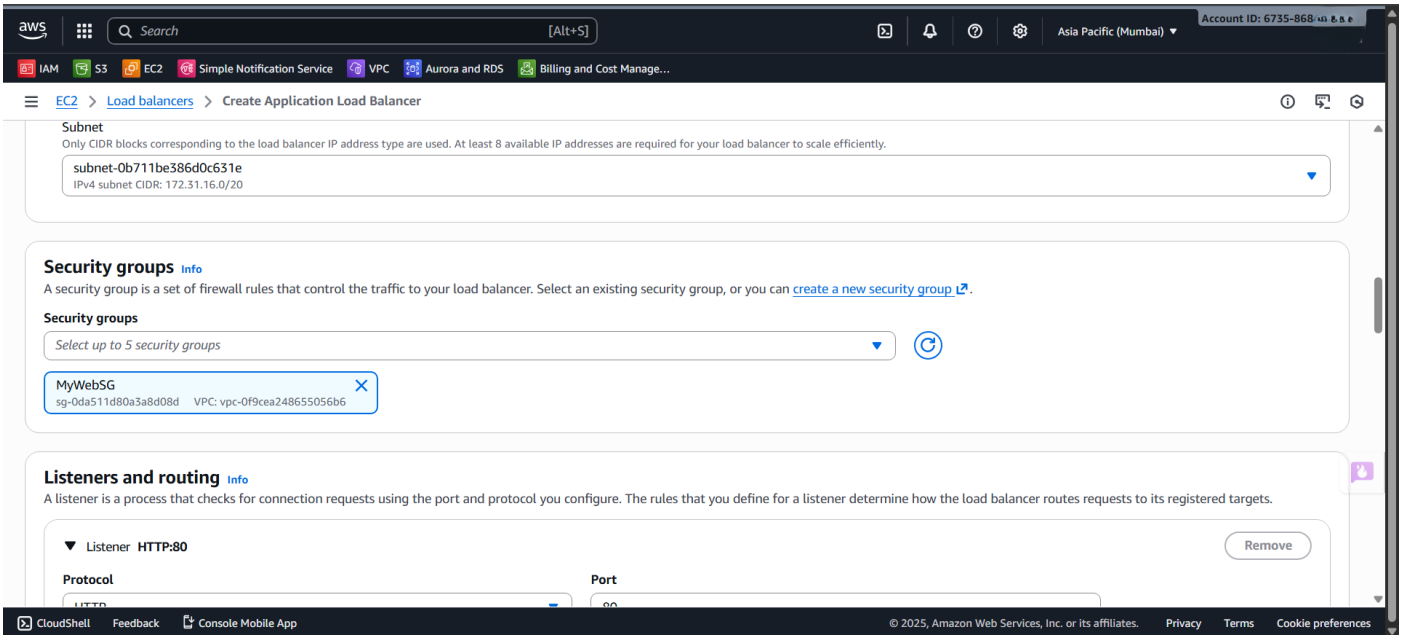


Selected IPv4 address type and kept the default VPC.

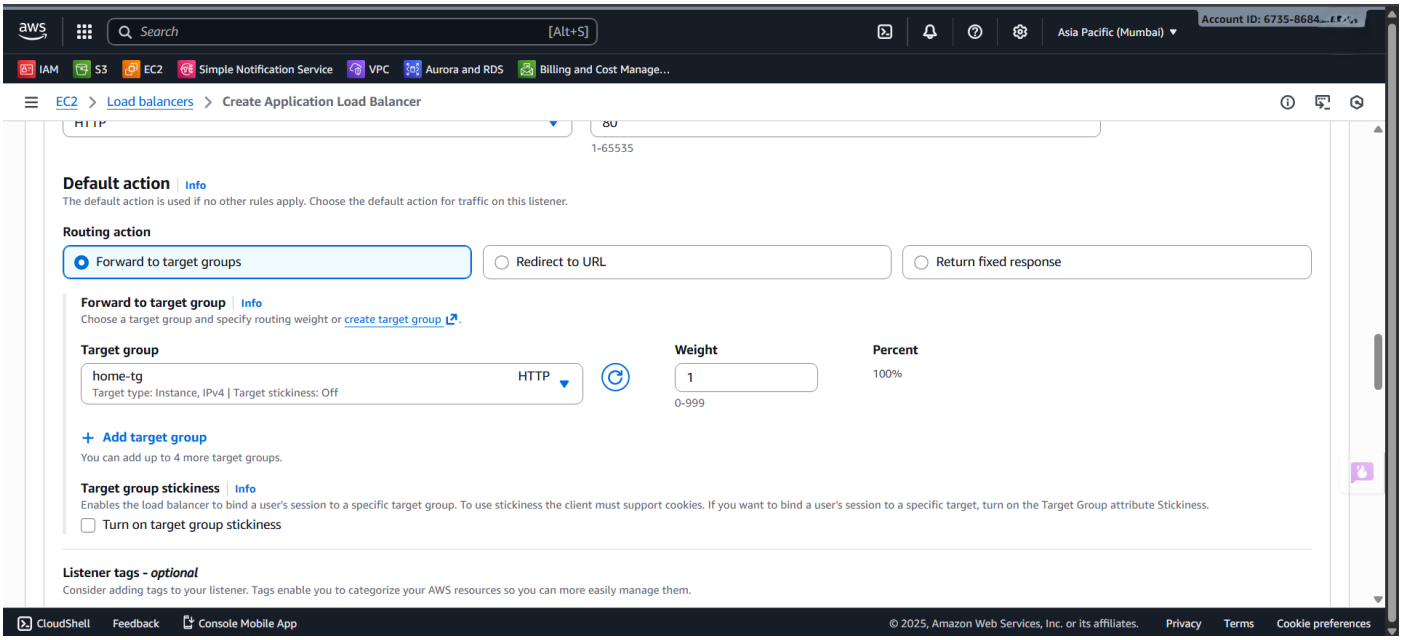
26. In the “Availability Zones and subnets” section, selected three public subnets across Availability Zones ap-south-1a, ap-south-1b, and ap-south-1c so that the ALB is distributed across all zones used by the Auto Scaling Groups.



27. Under “Security groups”, attached the existing security group MyWebSG to the load balancer, allowing HTTP access to the ALB.

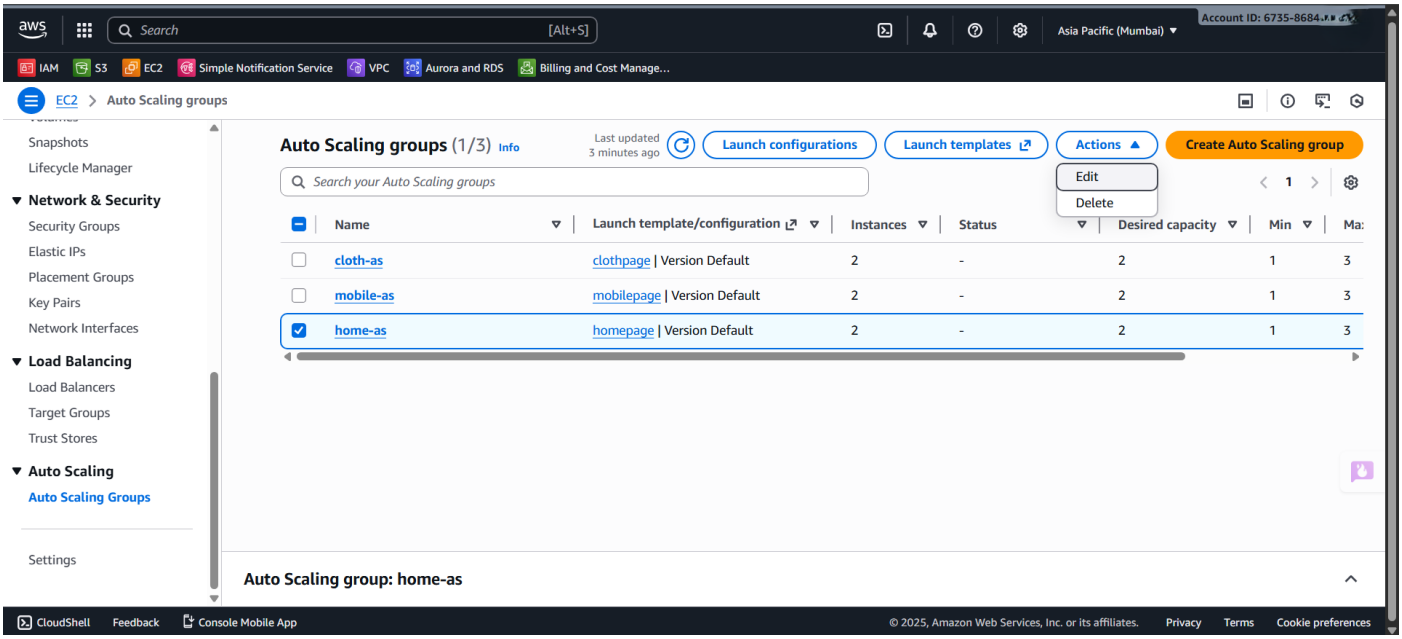


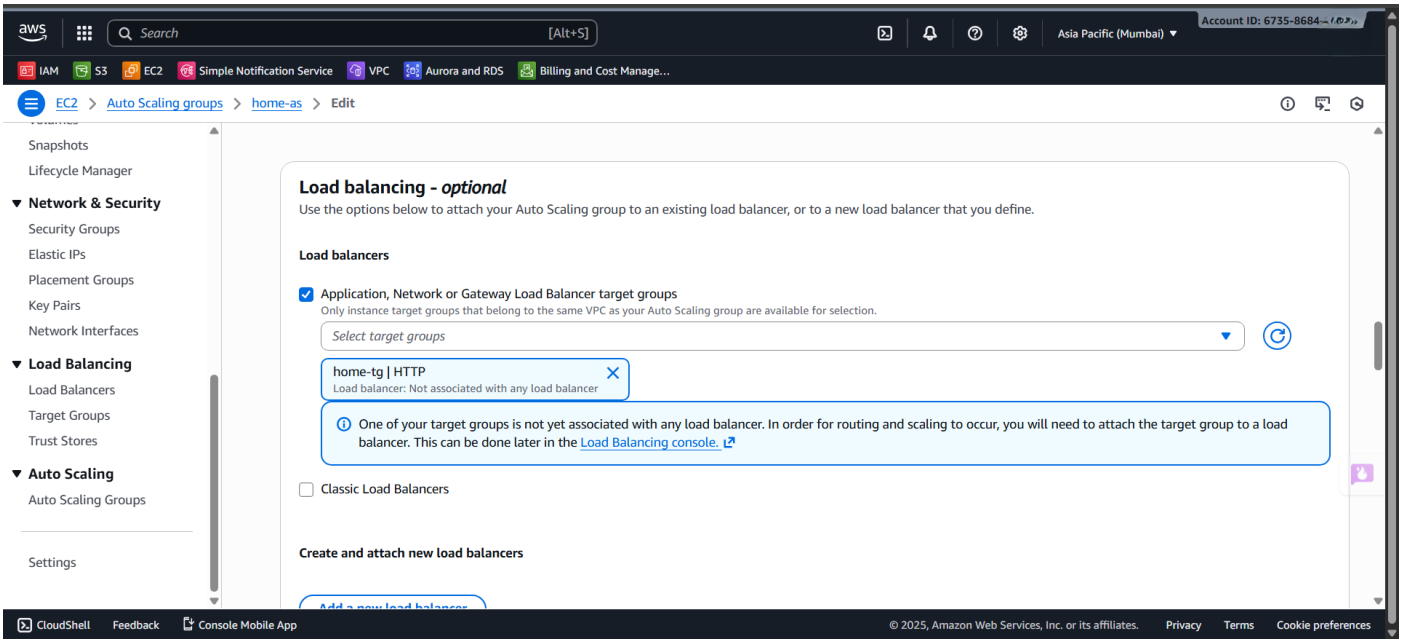
28. In the “Listeners and routing” section, configured the default HTTP listener on port 80 and set the default action to forward traffic to the home-tg target group so that root path / requests go to the home page.



Reviewed all settings and created the Application Load Balancer website-lb, waiting until its state became “Active” and noting the DNS name shown on the Load Balancers page.

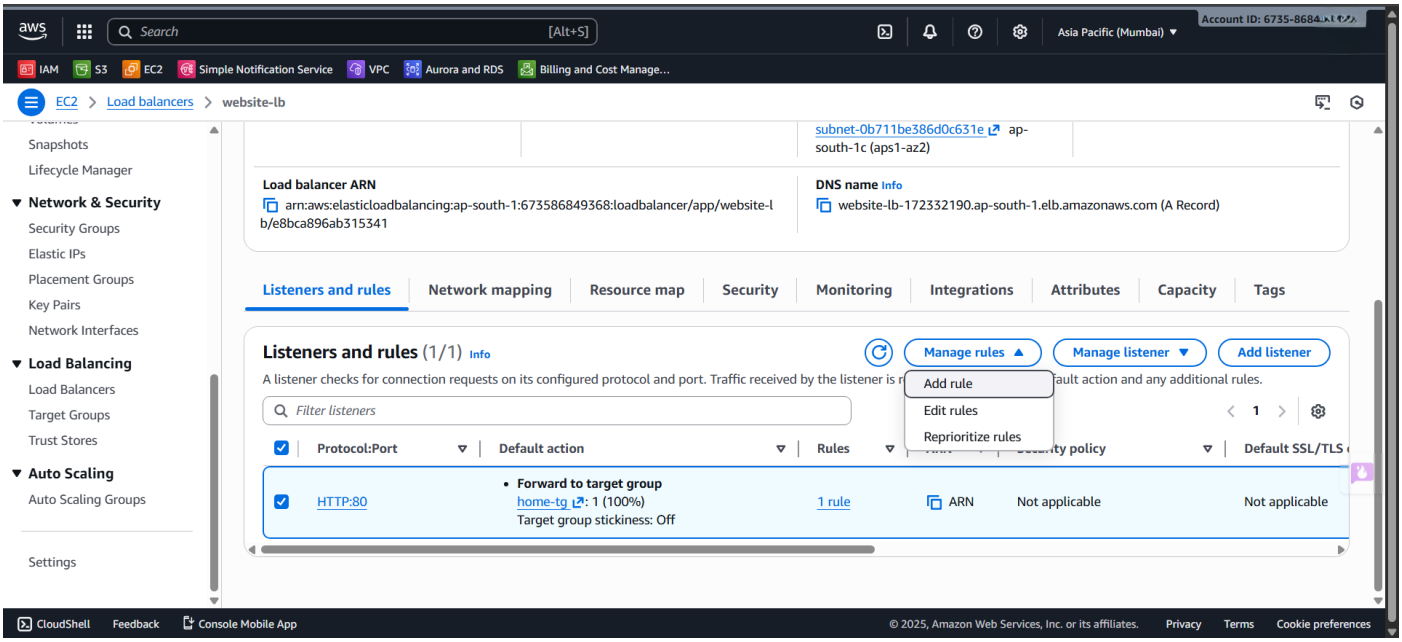
29. Returned to the “Auto Scaling groups” page, selected home-as, chose “Edit” from the Actions menu, and, in the “Load balancing – optional” section, attached the home-tg target group to this Auto Scaling Group.





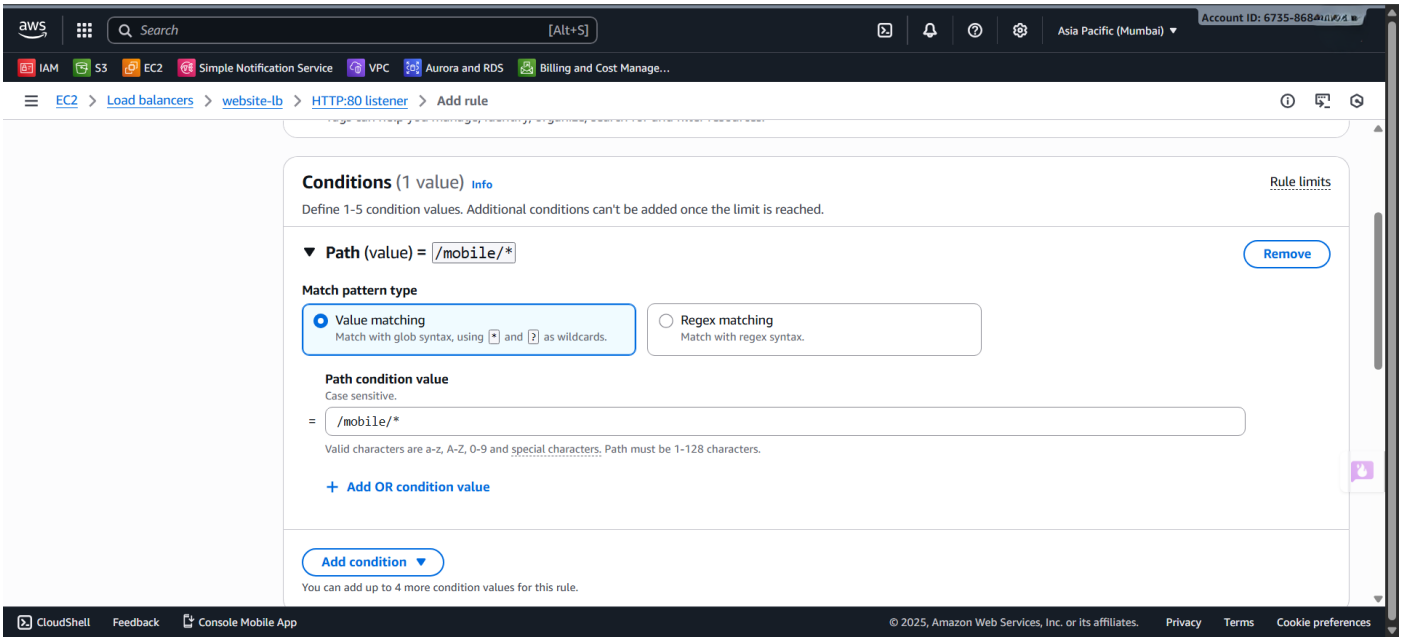
Performed equivalent edits for mobile-as and cloth-as so that each Auto Scaling Group was registered with its respective target group.

30. Opened website-lb in the Load Balancers list and went to the “Listeners and rules” tab, which initially showed a single HTTP:80 listener forwarding all traffic to home-tg.

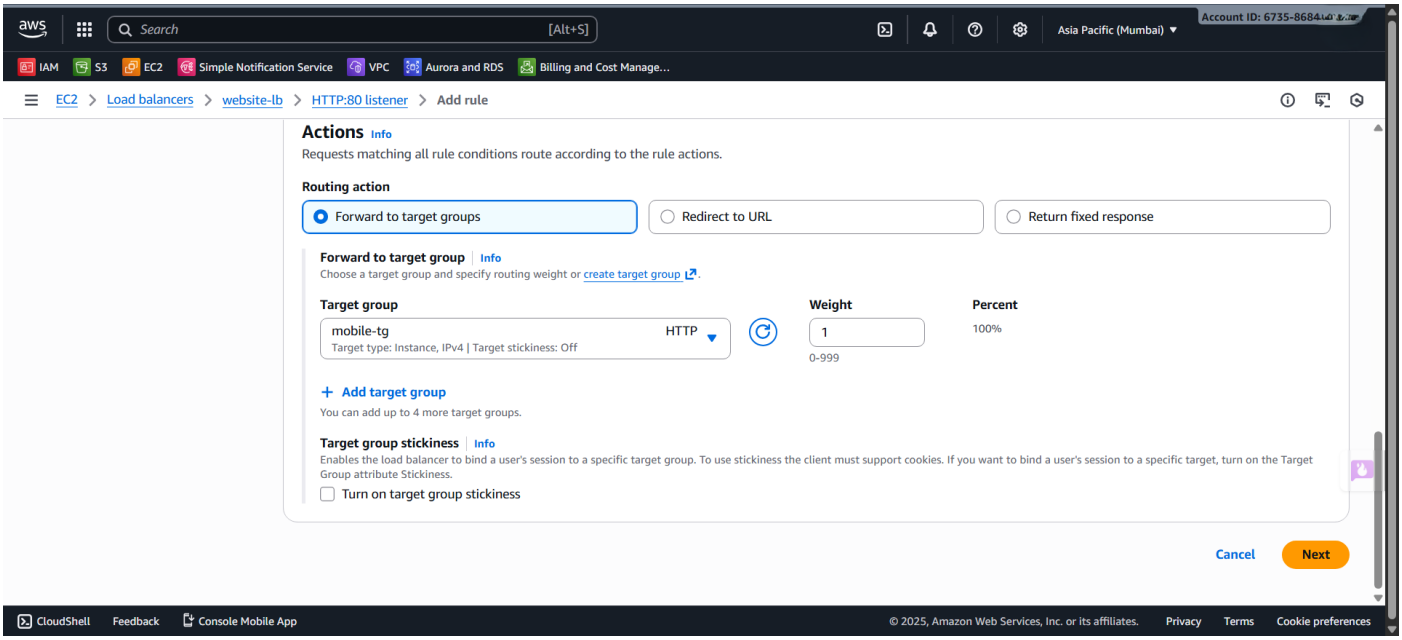


Clicked “Manage rules” and selected “Add rule” to define additional listener rules for path-based routing.

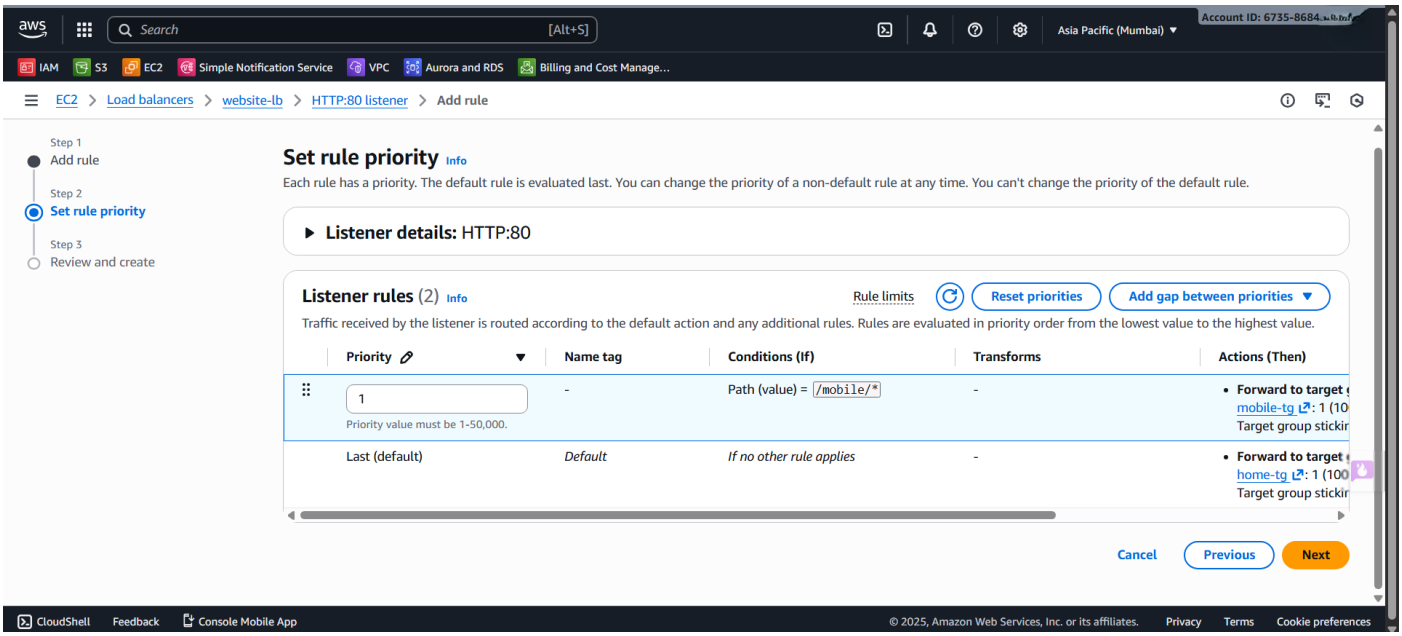
31. In the “Add rule” wizard, set the rule condition to “Path” with the match value /mobile/*, using value matching syntax with wildcard support.

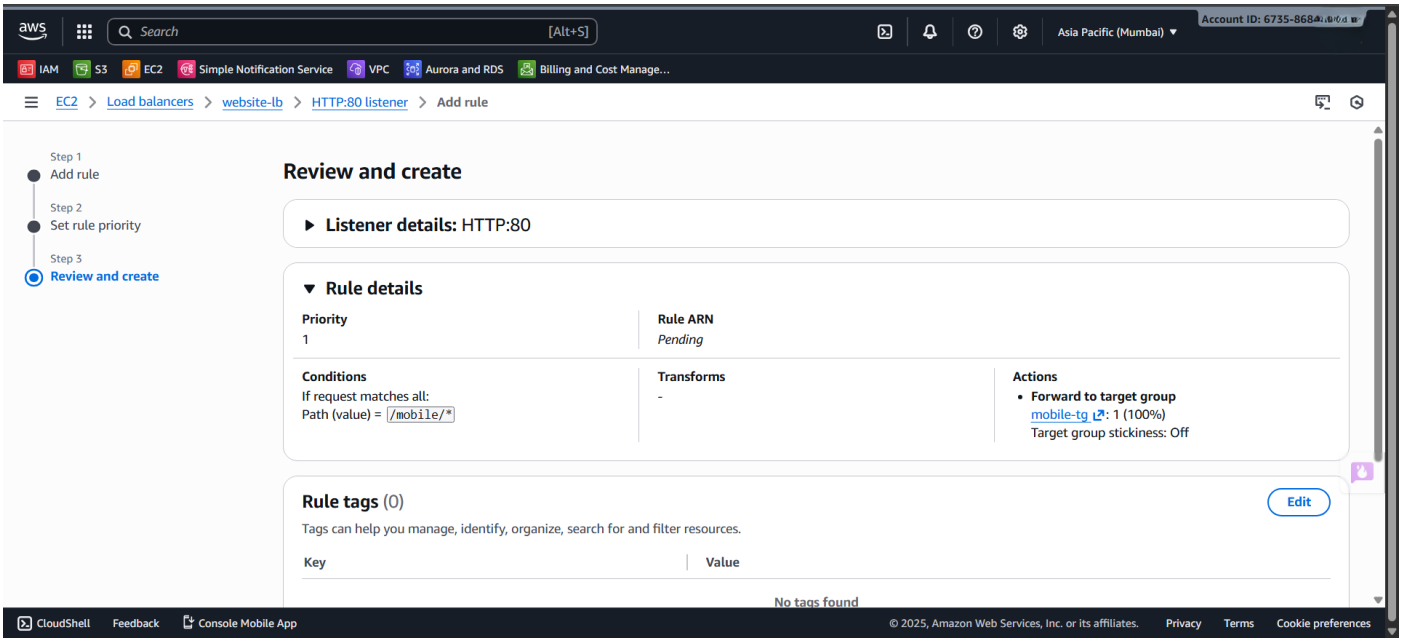


32. For the same rule, configured the action “Forward to target groups” and selected mobile-tg as the target group, leaving weight at 1 (100 percent of traffic for that rule).



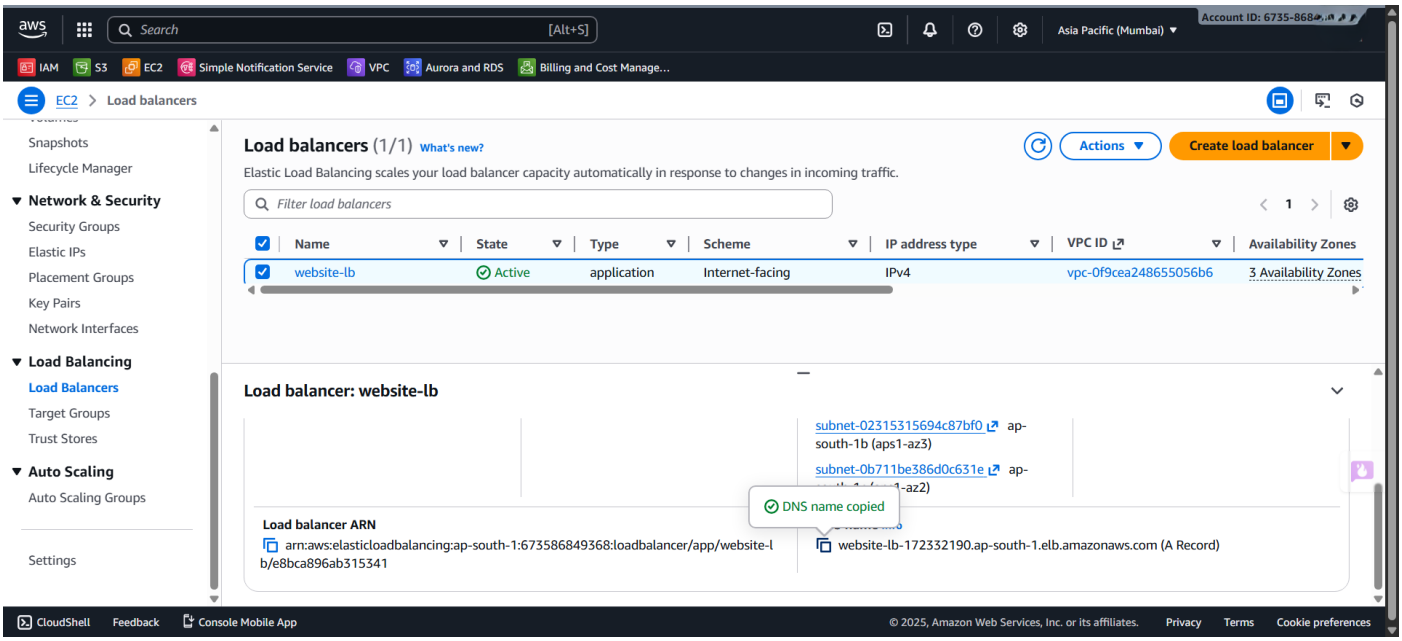
33. Assigned this rule a higher priority value (for example, priority 1) so that it is evaluated before the default rule, then reviewed the rule details showing the path condition /mobile/* and the forward action to mobile-tg, and created the rule.



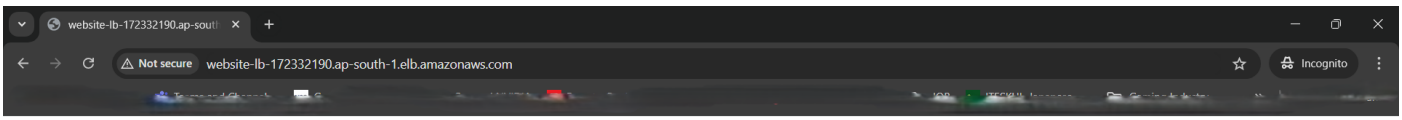


Similarly, created another rule with a path condition /cloth/* and an action forwarding to cloth-tg, giving it an appropriate priority, while keeping the default rule forwarding remaining traffic to home-tg.
After saving the rules, the HTTP:80 listener on website-lb had multiple rules: one default rule forwarding to home-tg and additional rules routing /mobile/* and /cloth/* paths to mobile-tg and cloth-tg respectively.

34. From the Load Balancers page, copied the DNS name of website-lb once its status was Active.

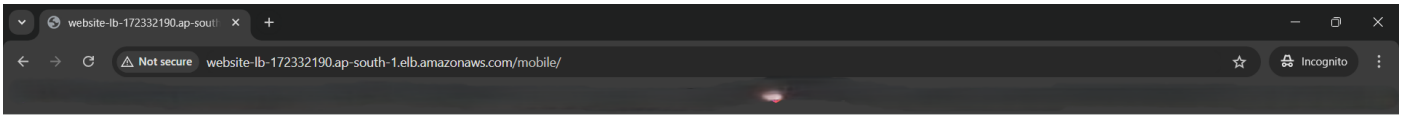


35. In a browser, access http://website-lb-…/mobile/ and verify that the “This is mobile page” HTML content is served through the ALB from one of the mobile Auto Scaling instances.



This is home page: ip-172-31-33-100.ap-south-1.compute.internal

36. In a browser, access http://website-lb-…/ (root path) and confirm that the “This is home page” content is returned from the home Auto Scaling Group.



This is mobile page: ip-172-31-2-166.ap-south-1.compute.internal

37. In a browser, access http://website-lb-…/cloth/ and verify that the “This is cloth page” content is served from the cloth Auto Scaling Group.

This is cloth page: ip-172-31-23-110.ap-south-1.compute.internal

38. On the EC2 “Instances” page, view all running and terminated instances and confirm that multiple t3.micro instances across different Availability Zones are running as part of the three Auto Scaling Groups.