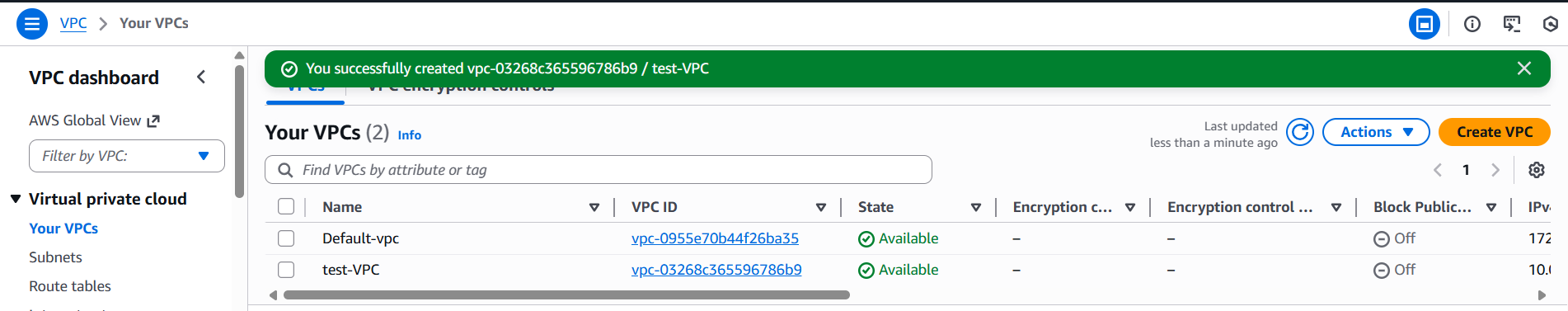
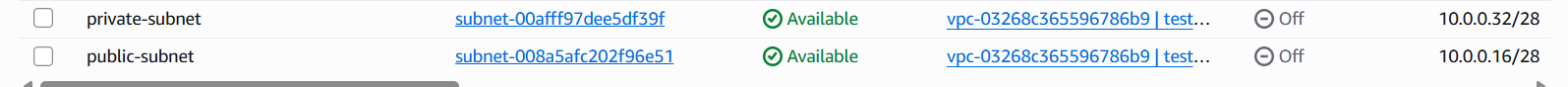
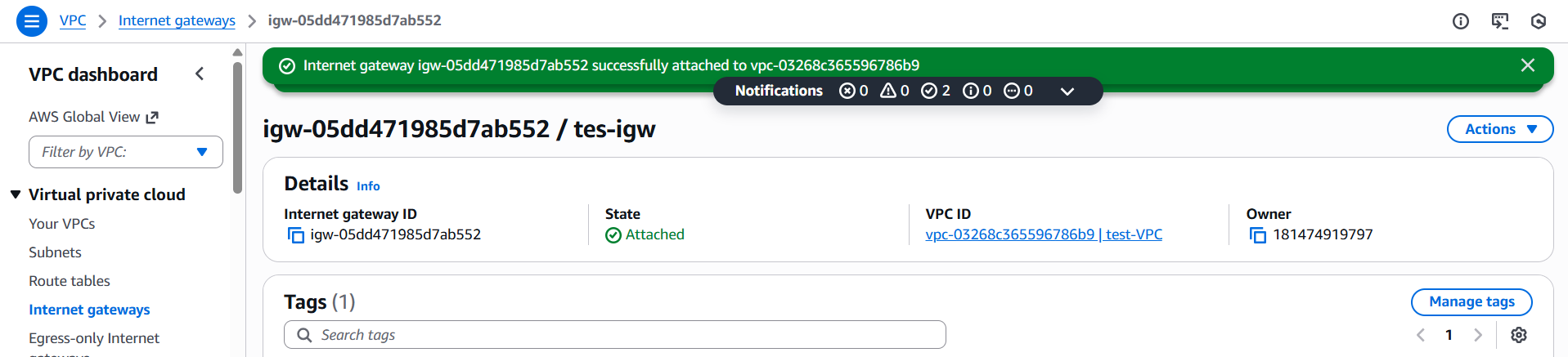
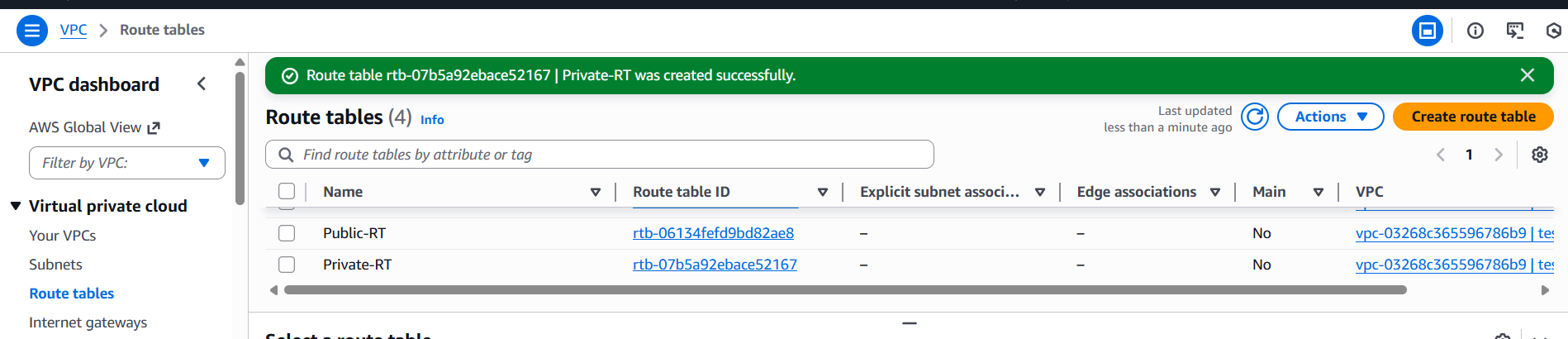
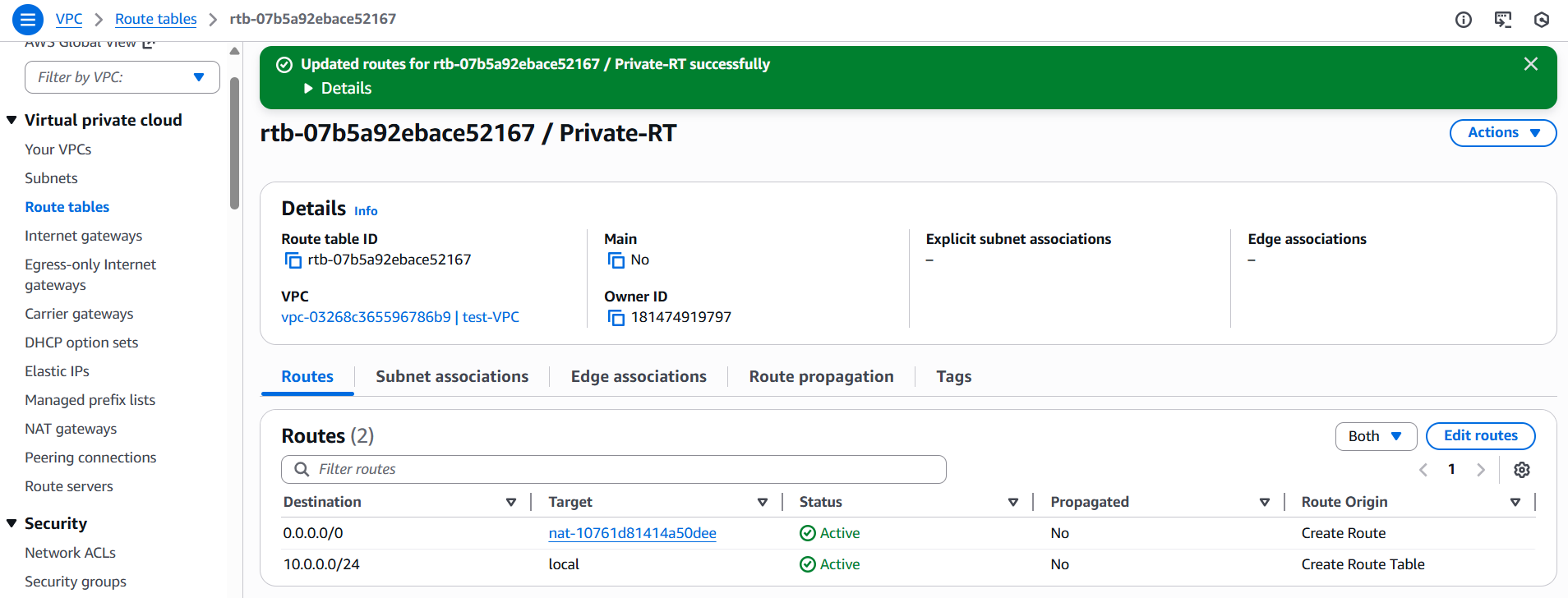
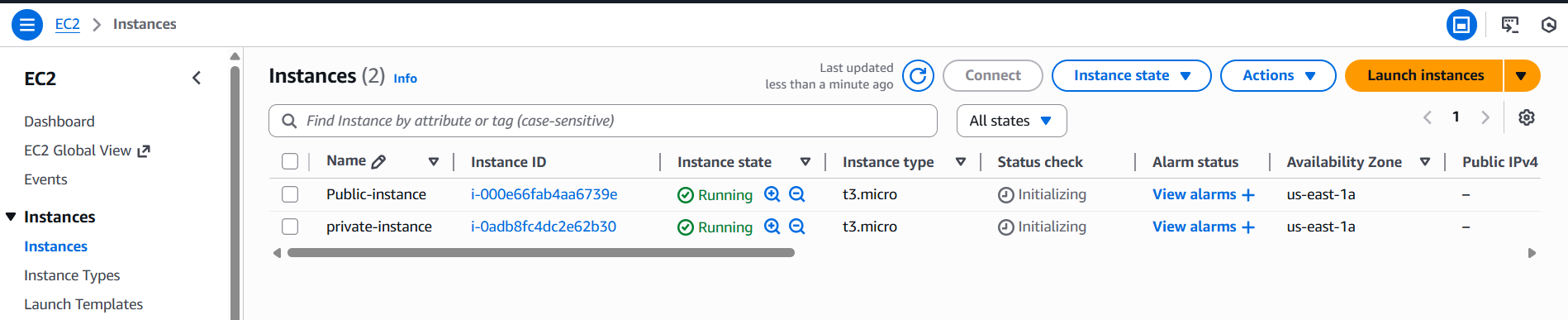
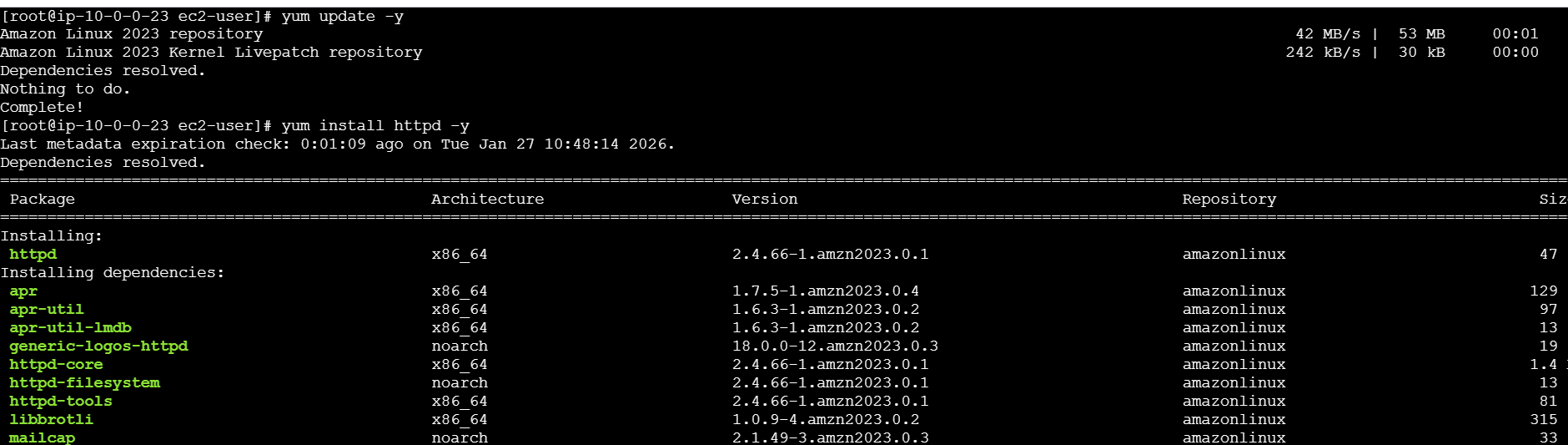
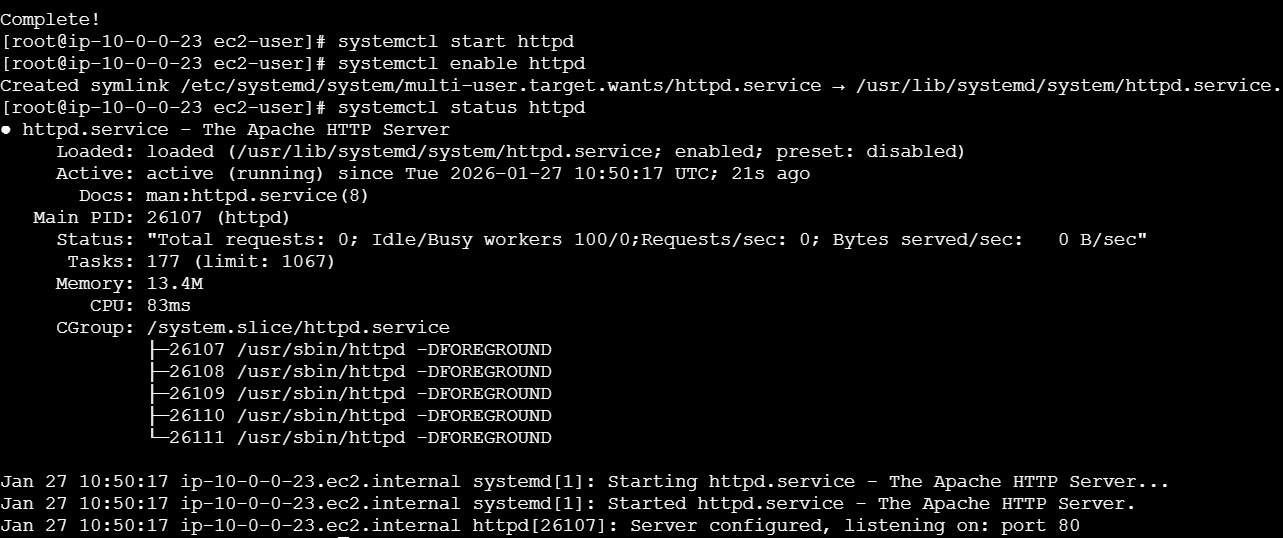
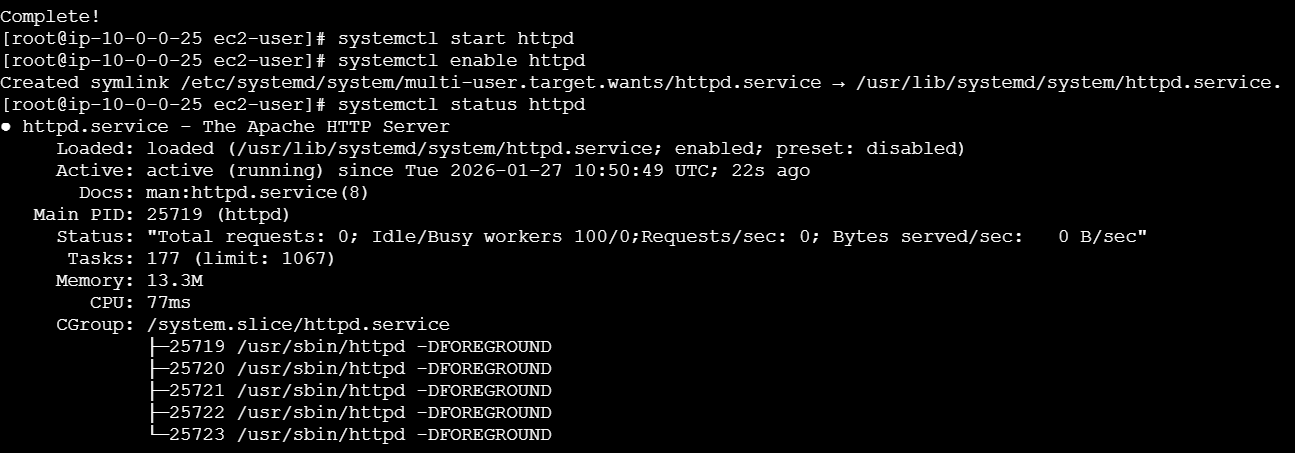
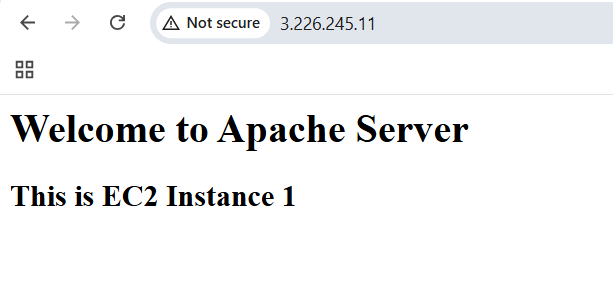
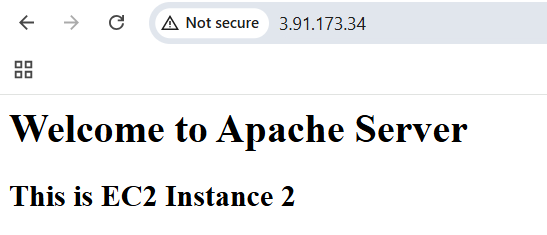
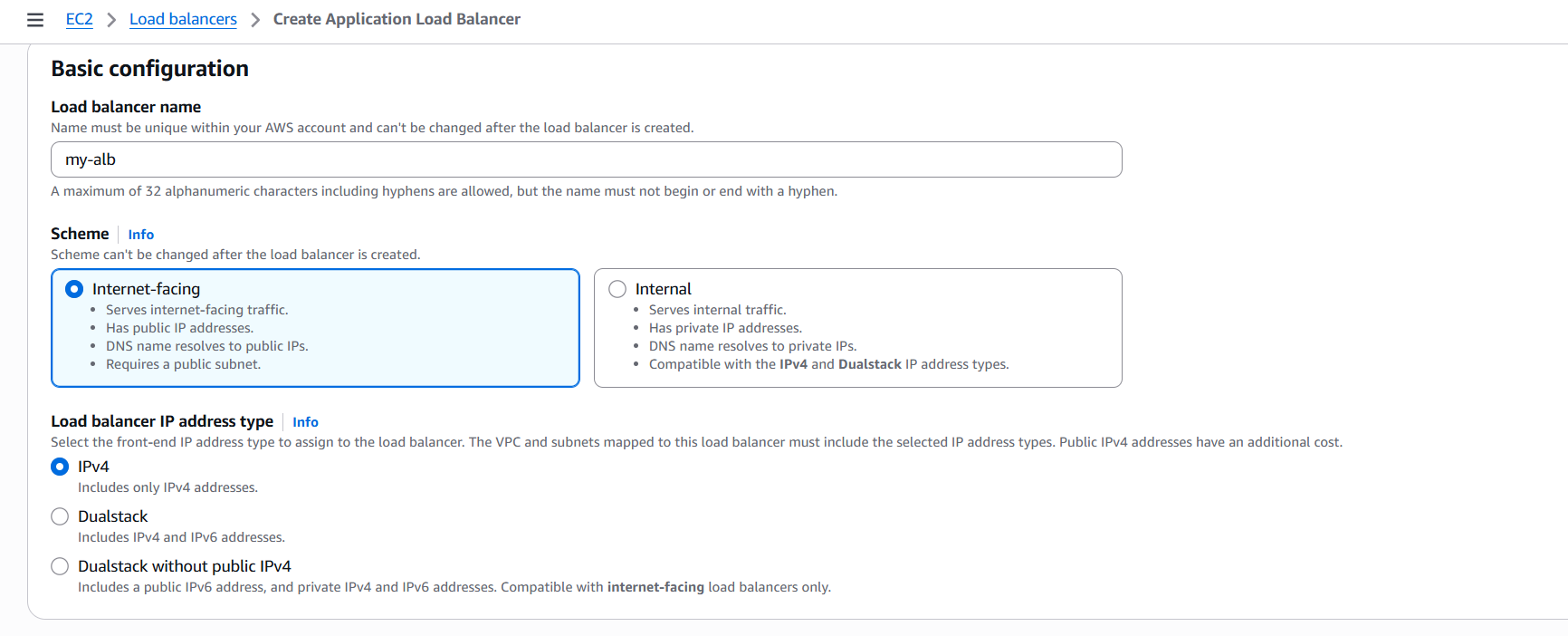
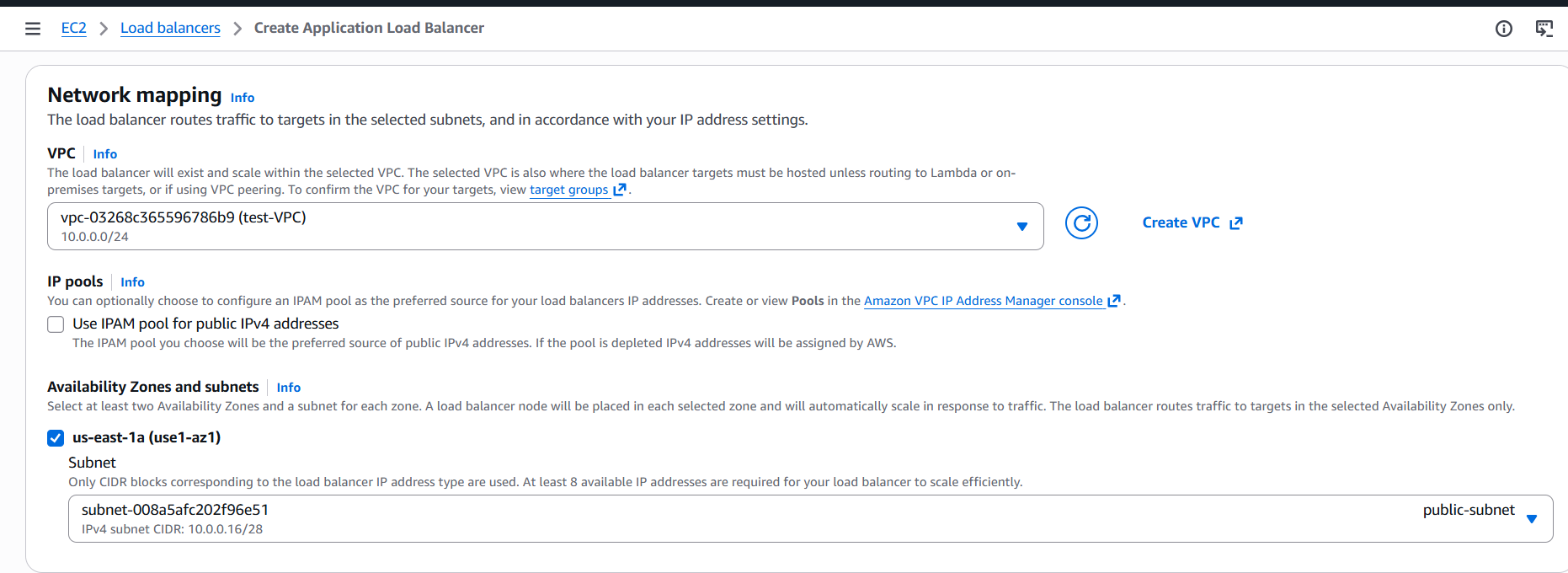
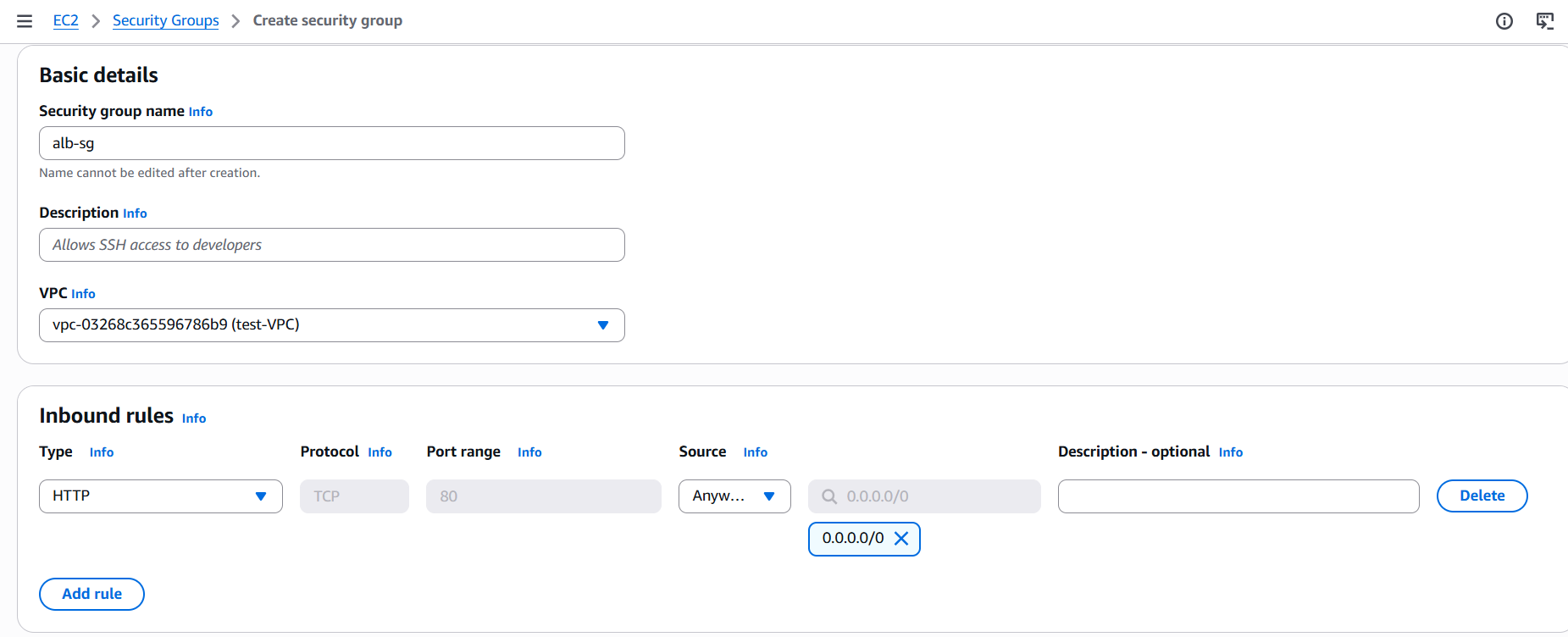
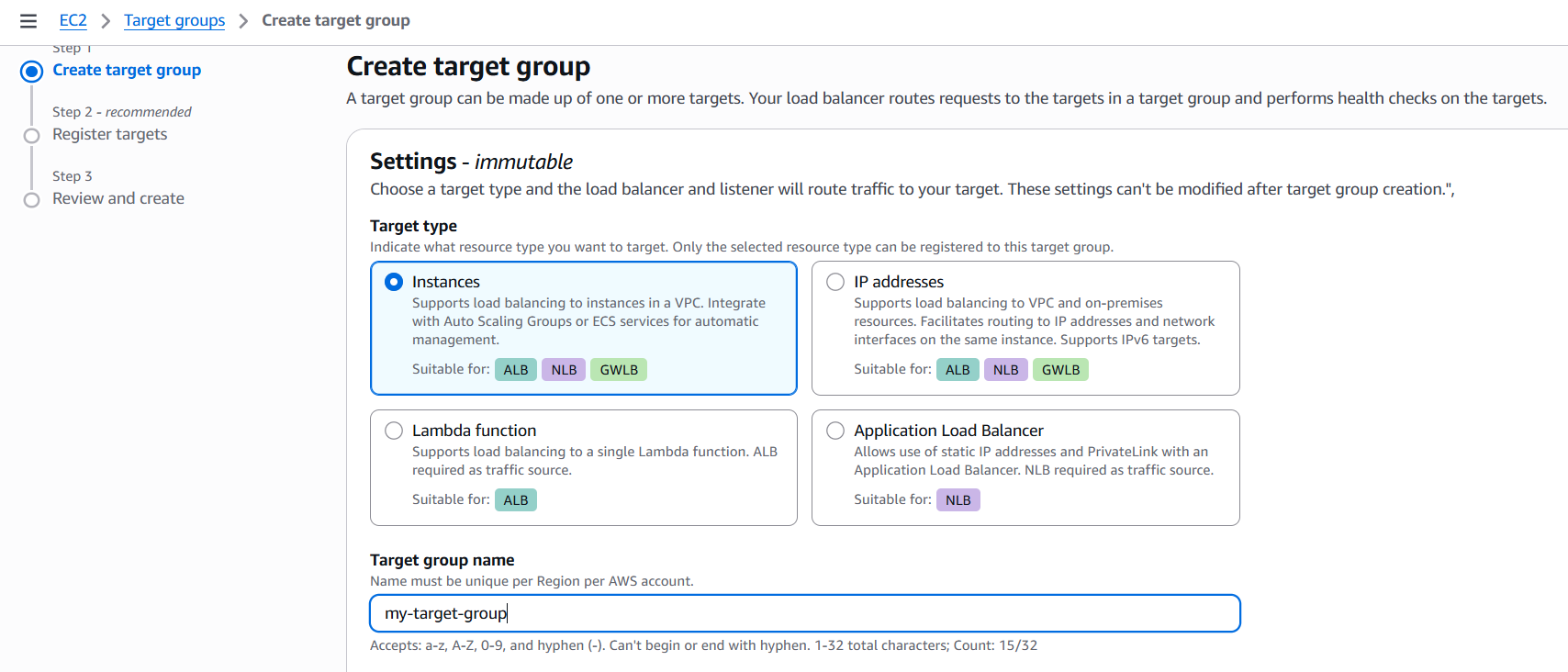
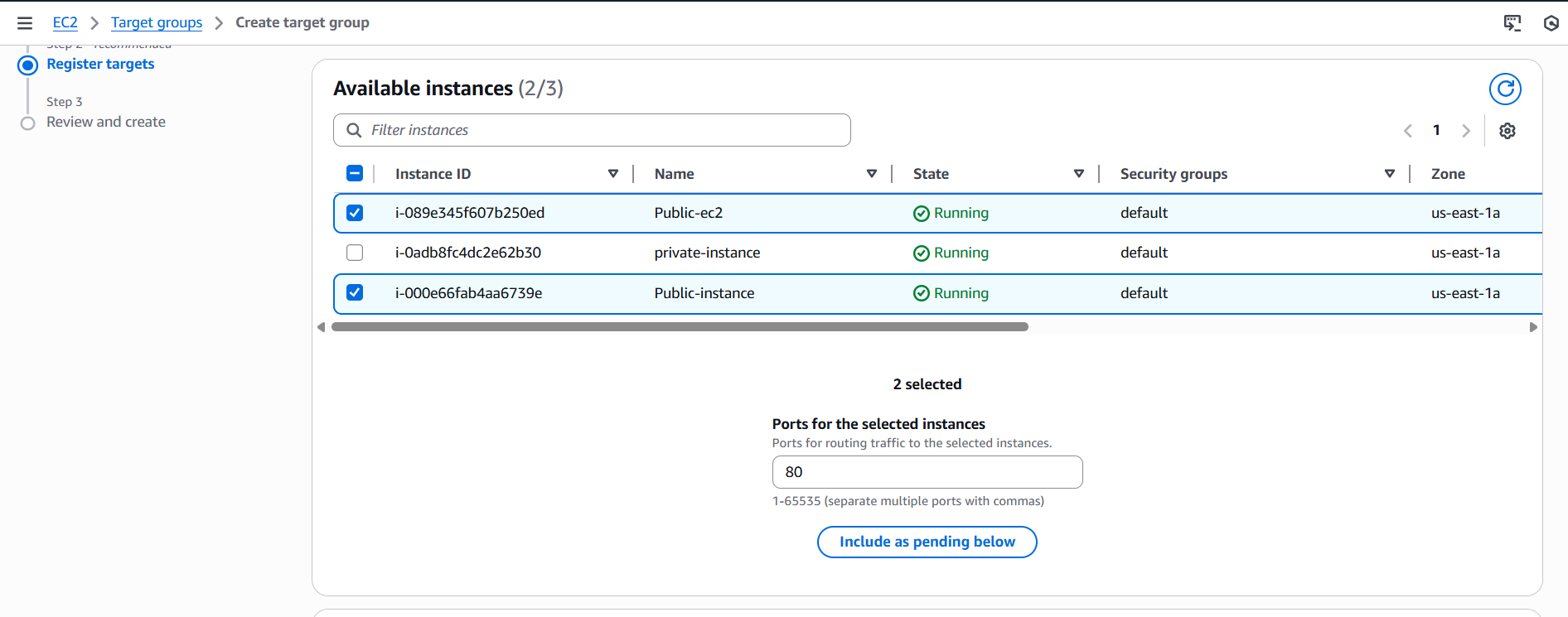
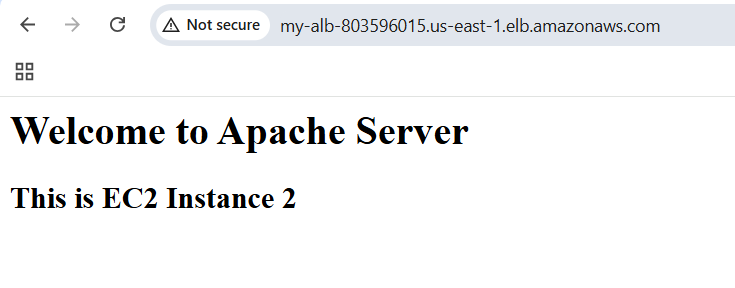
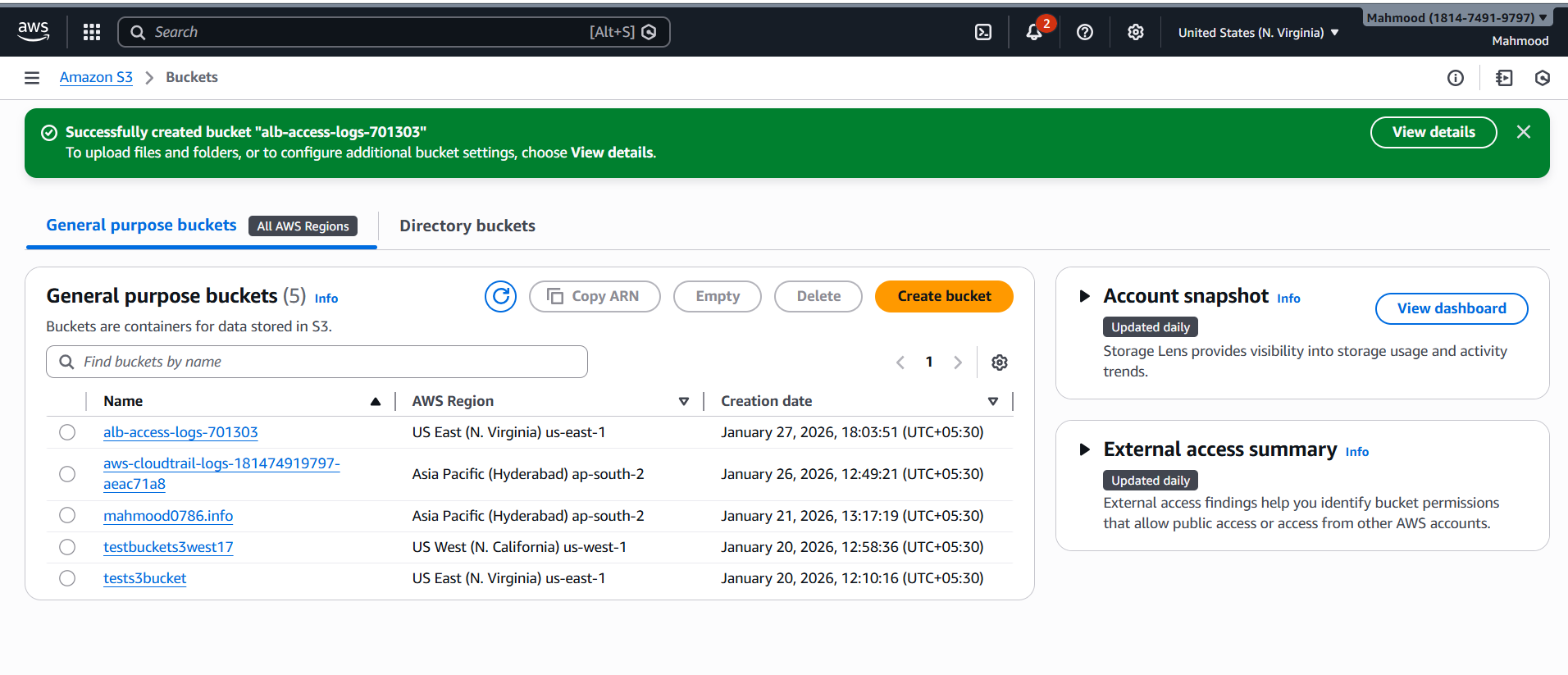
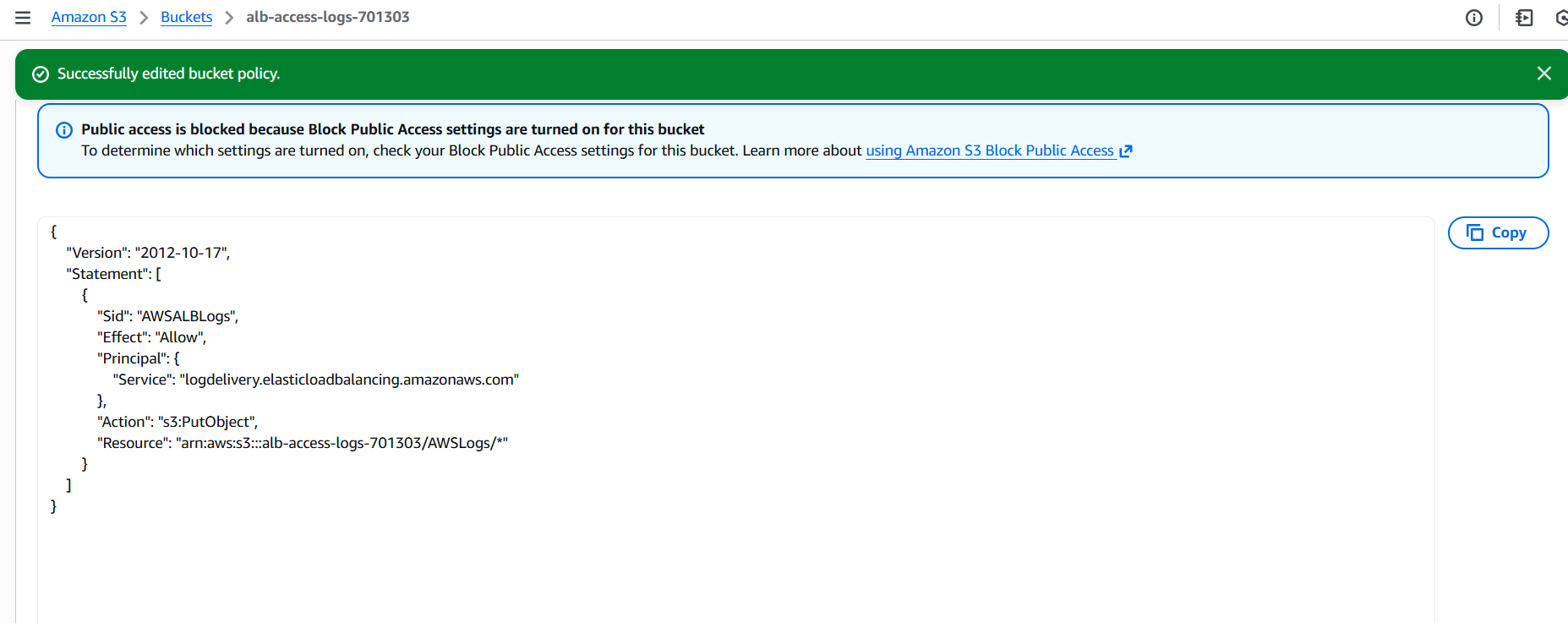
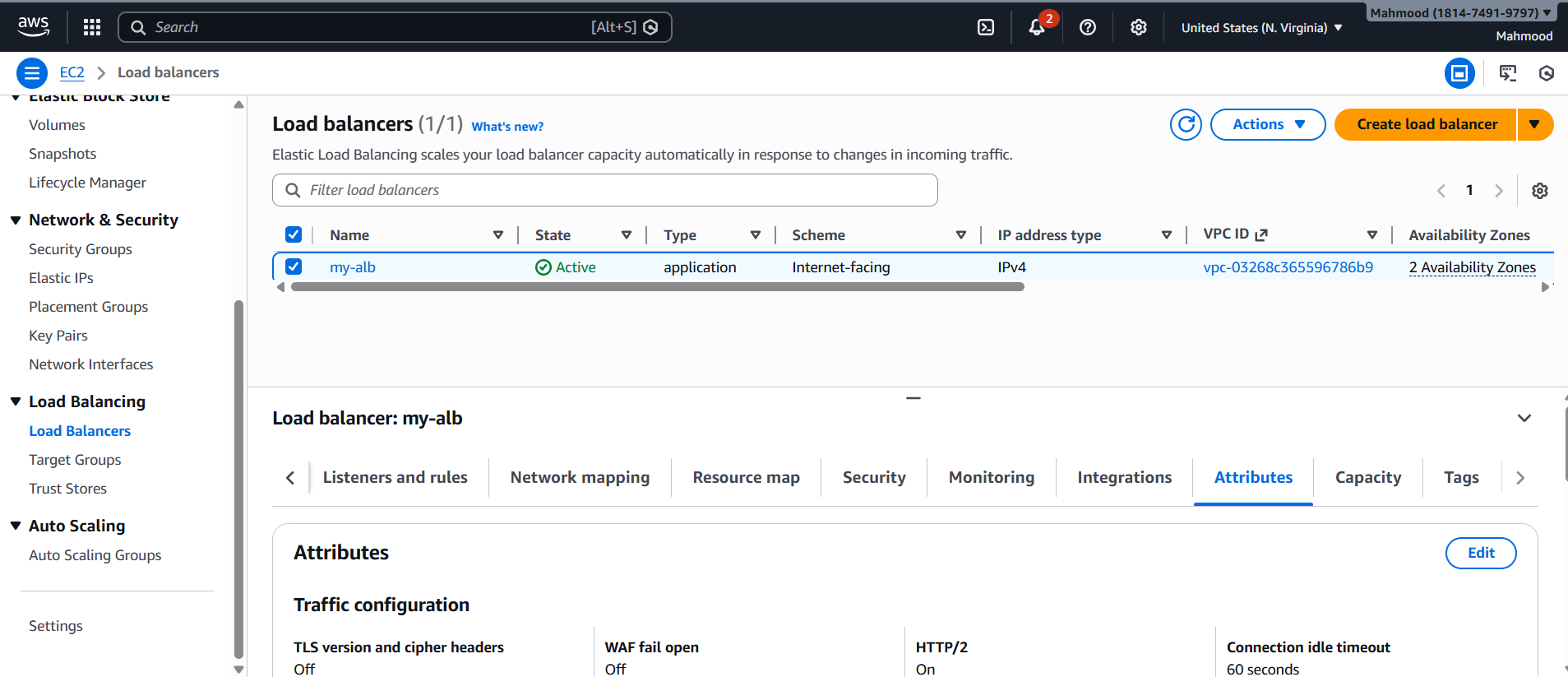
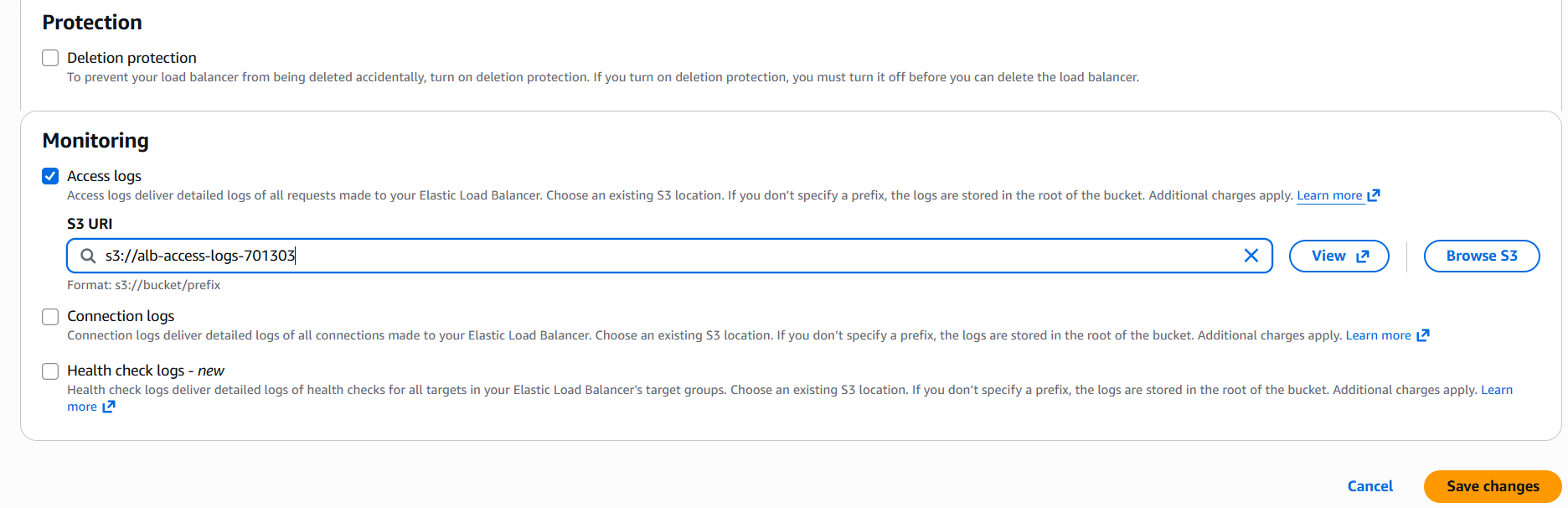
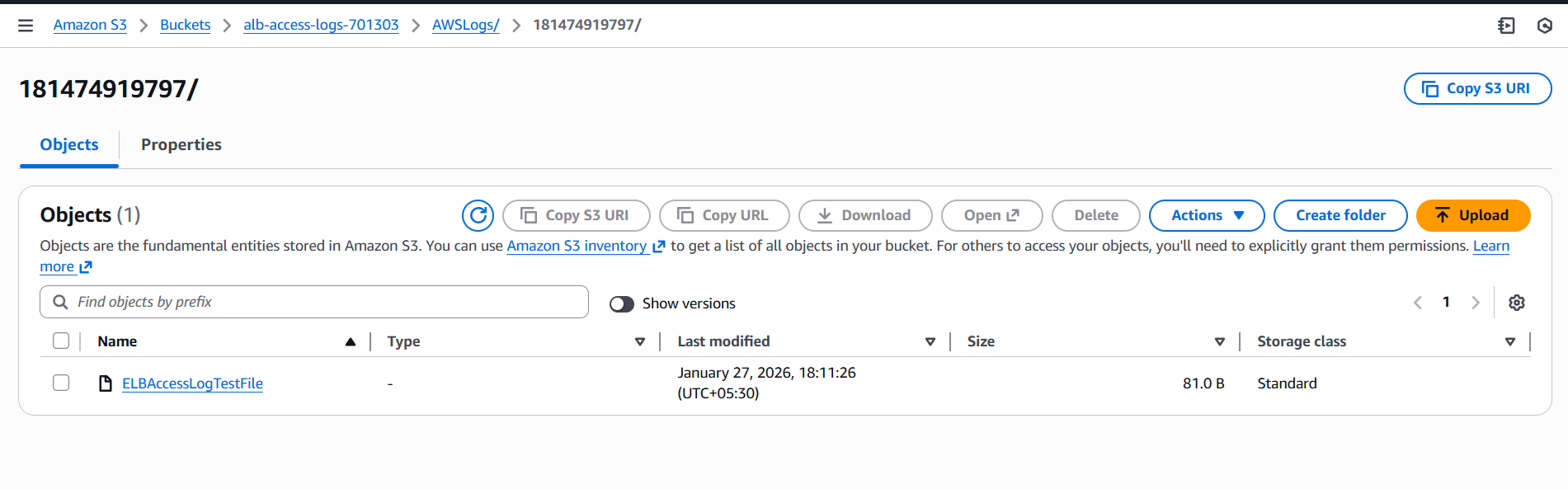
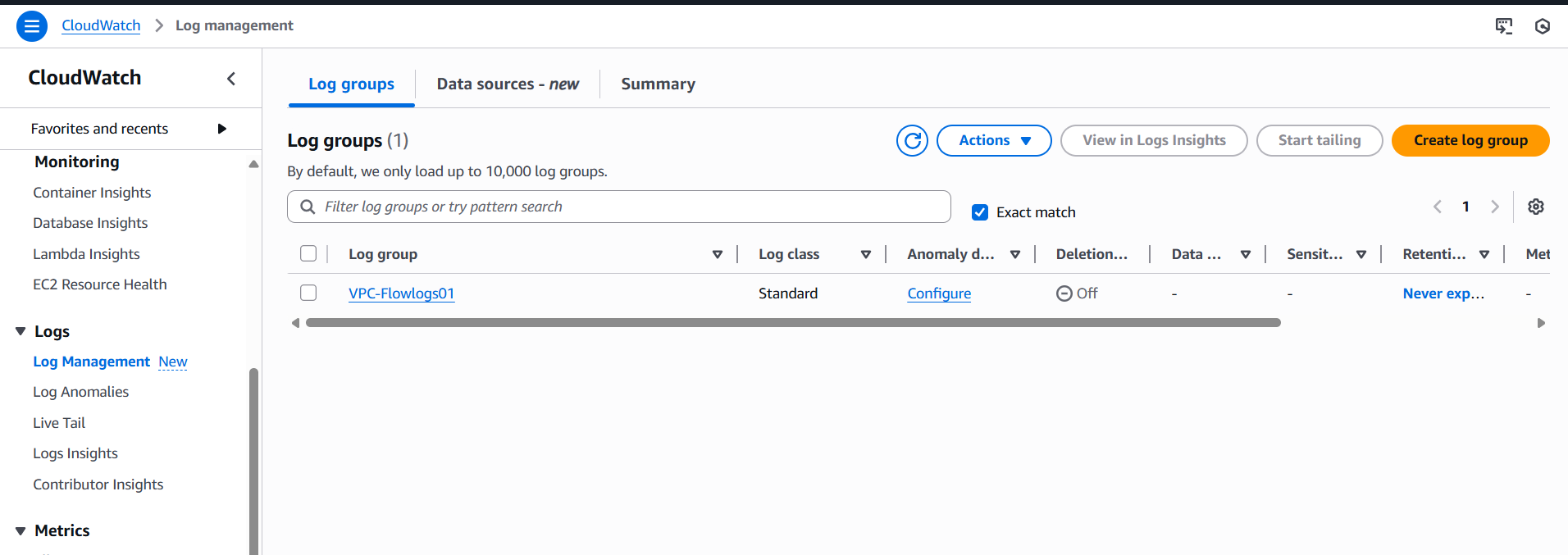
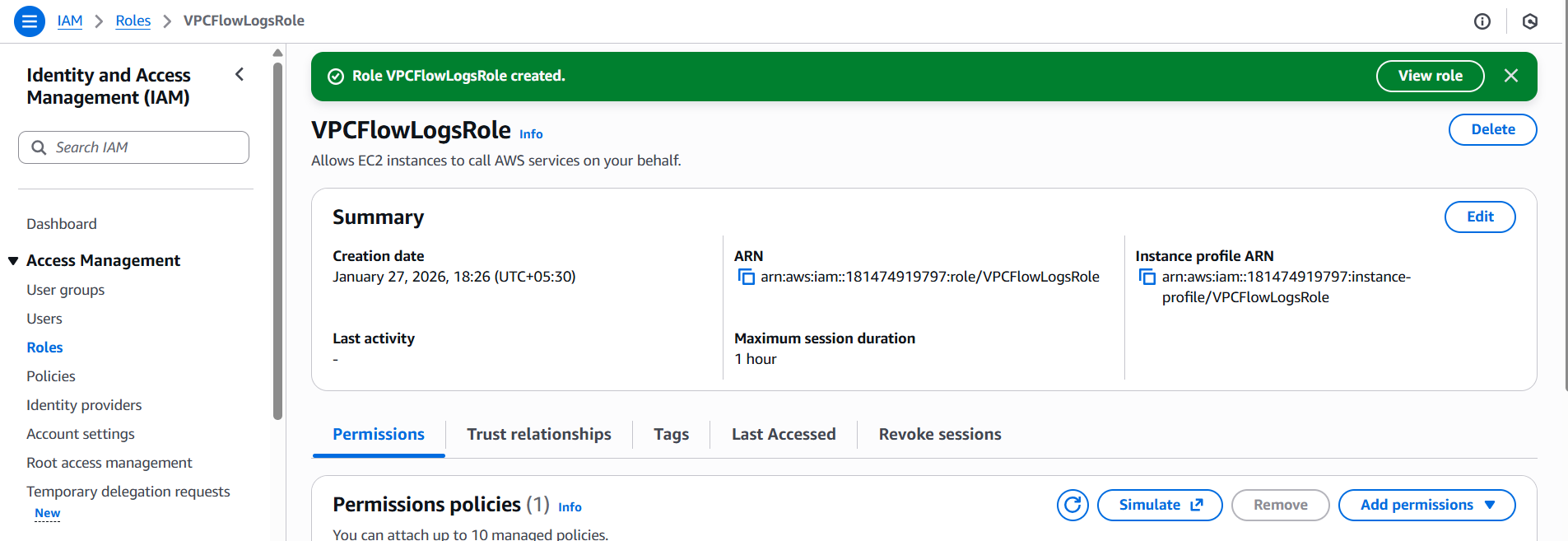
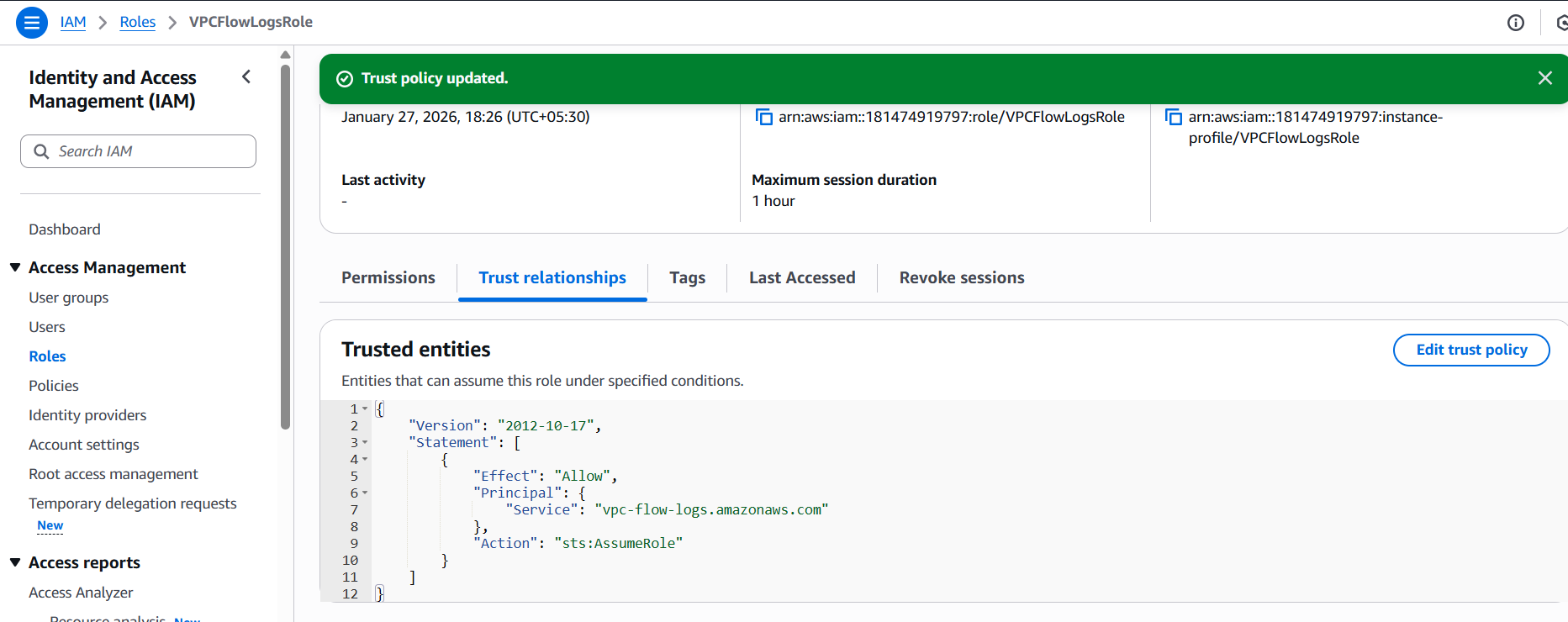
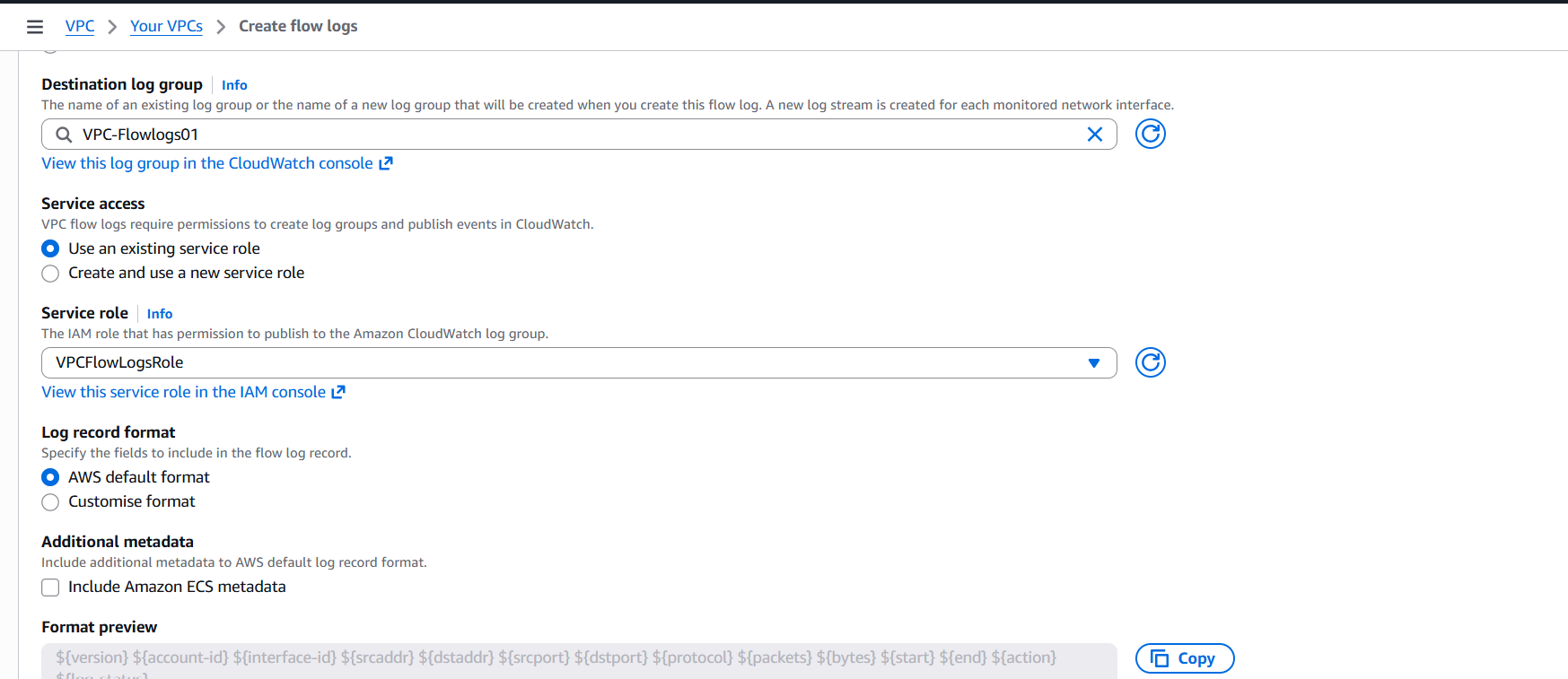
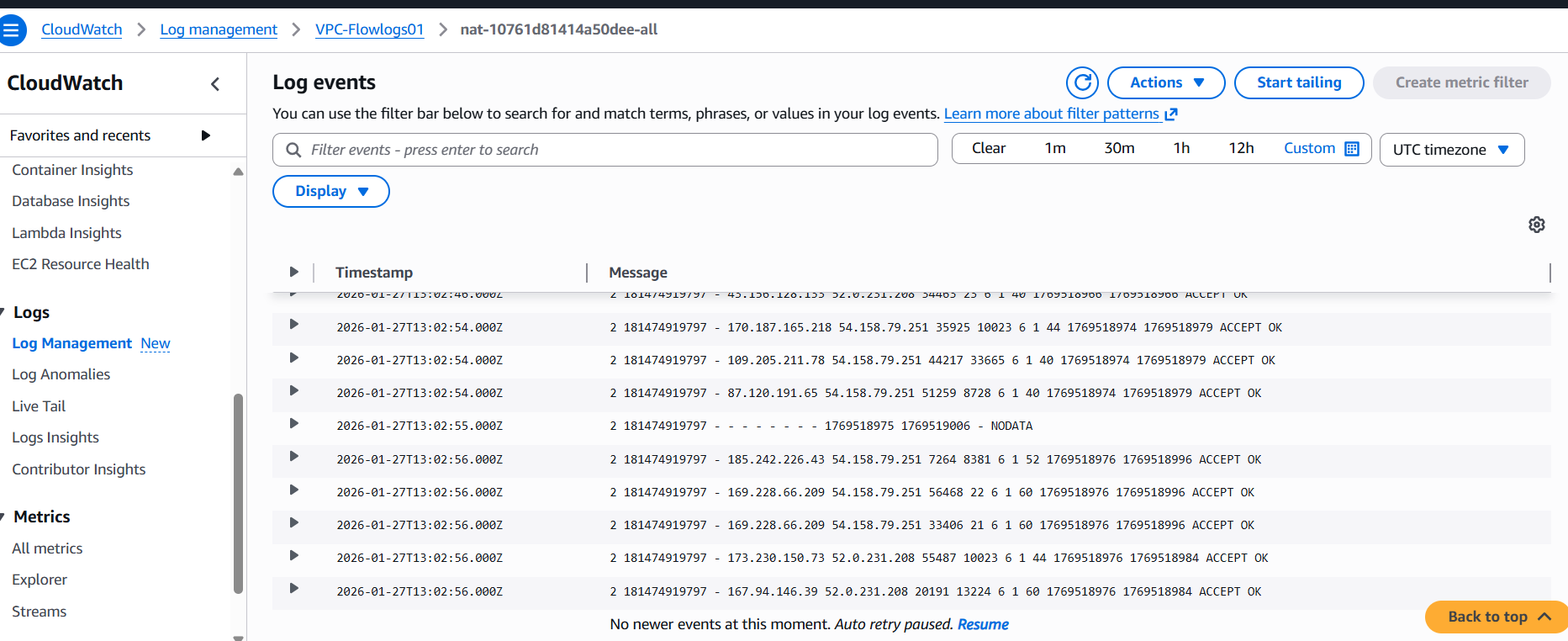
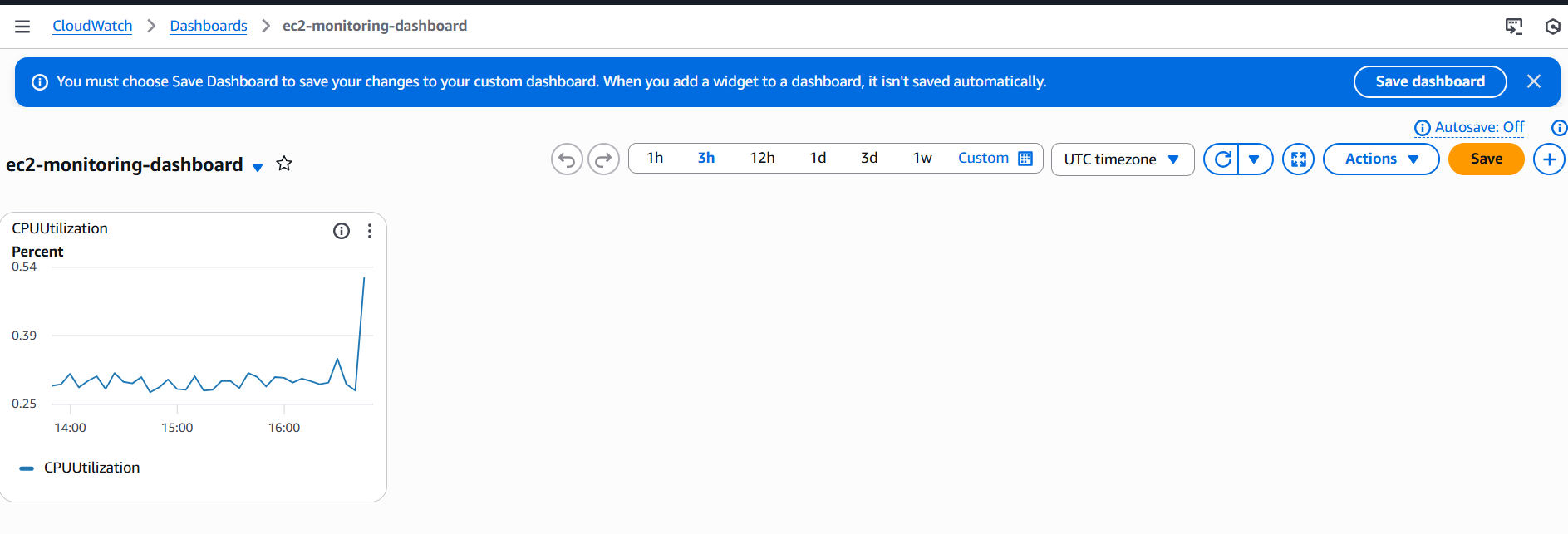
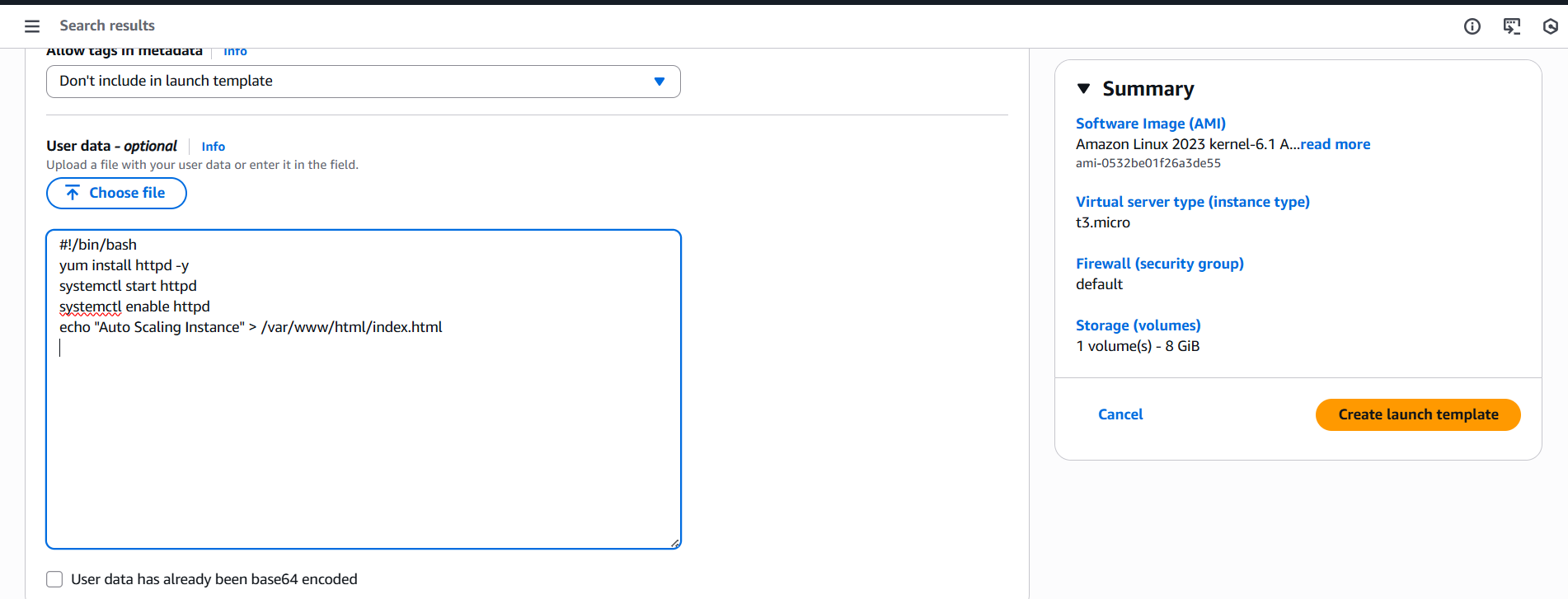
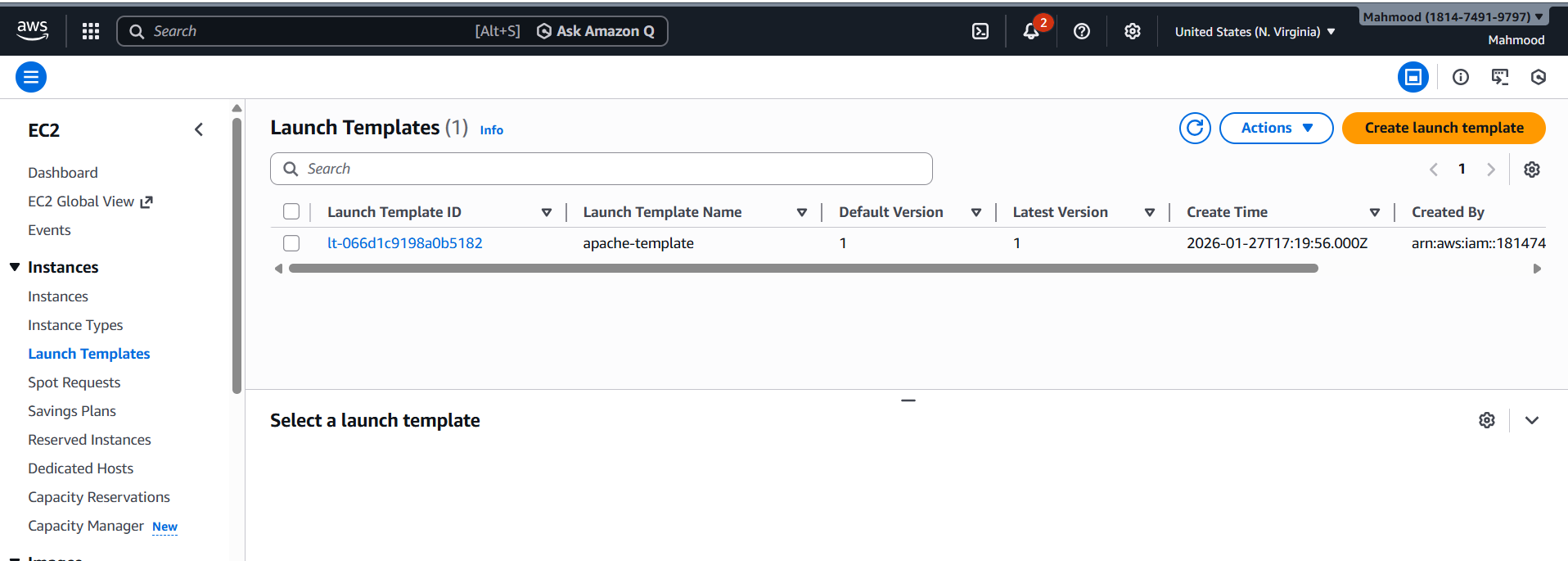
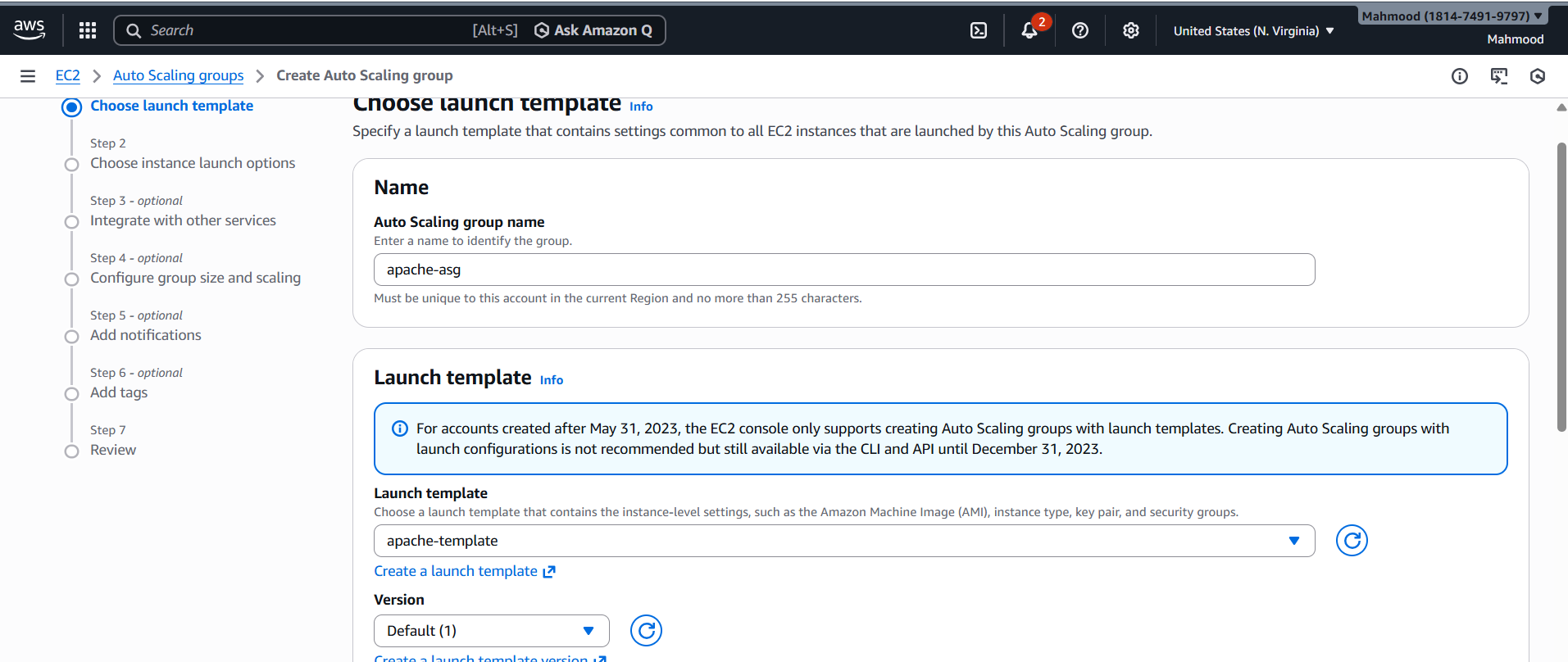
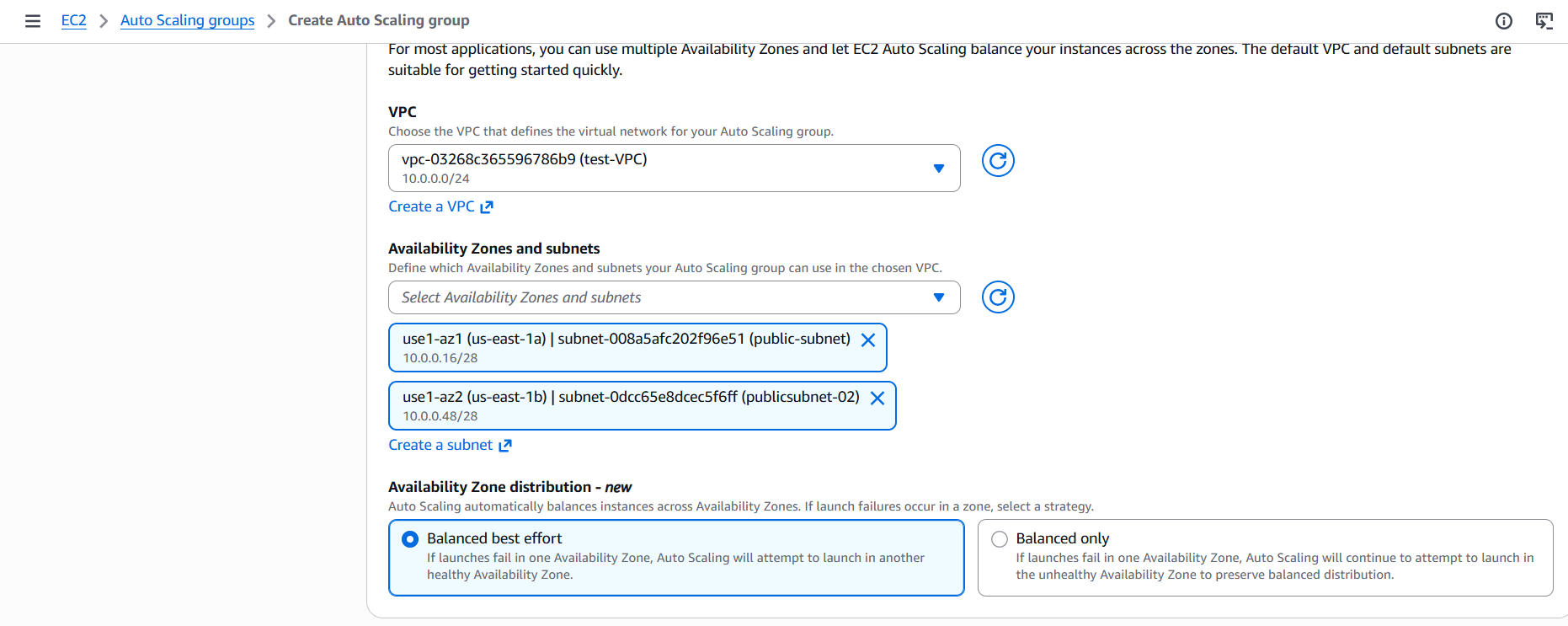
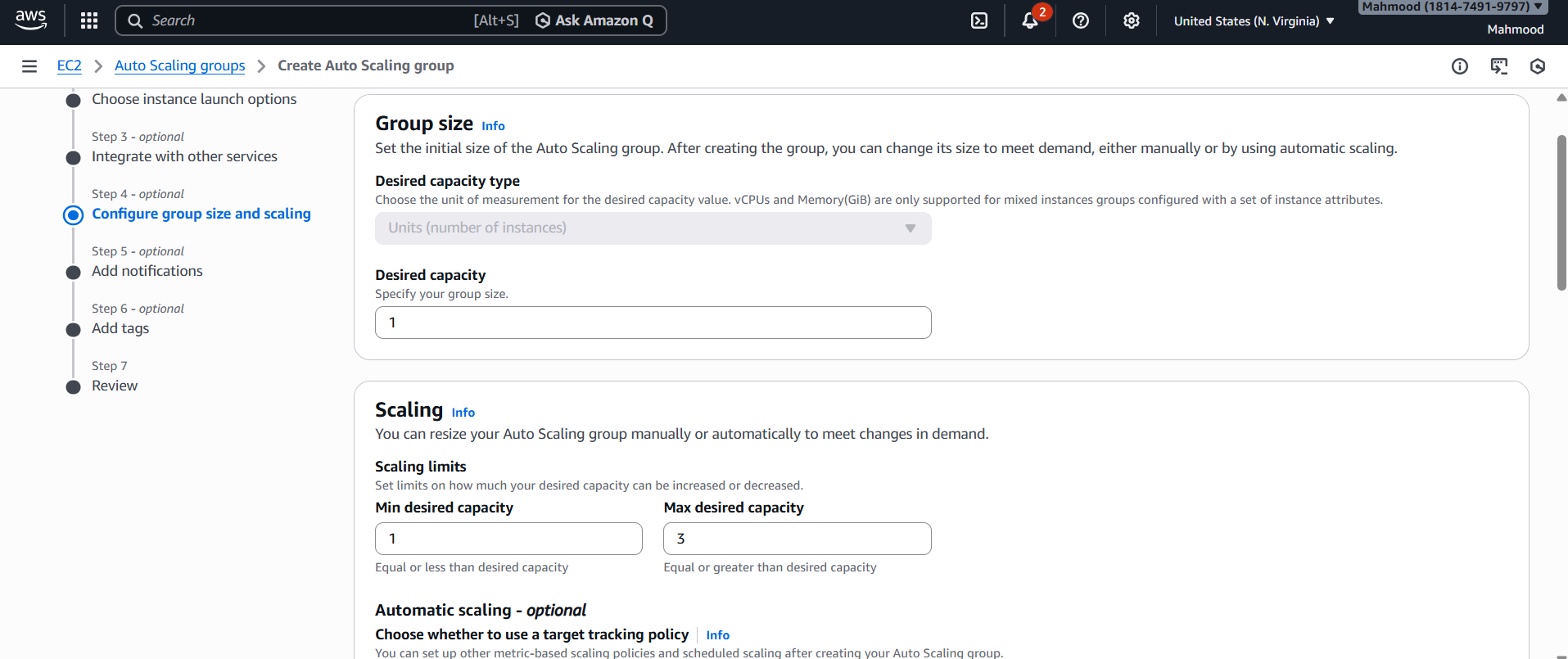
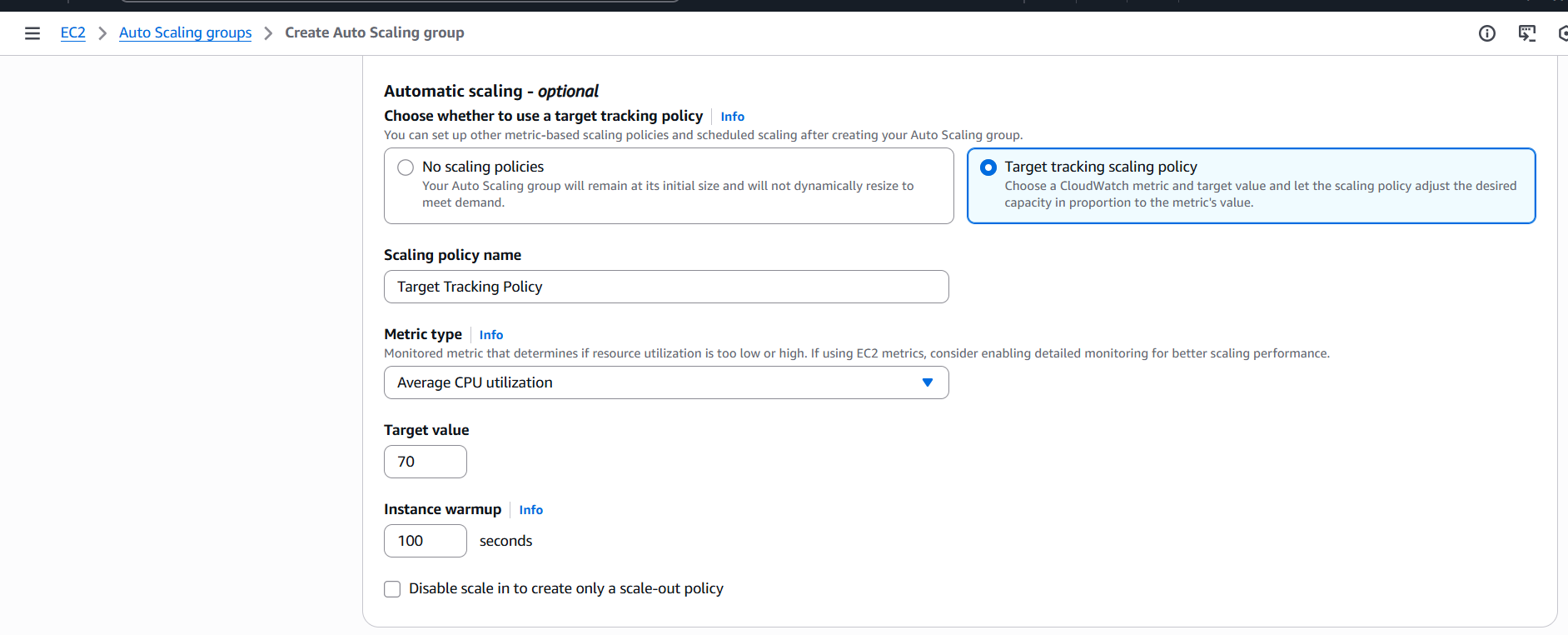
**AWS – ASG**

**Summary :** AWS Auto Scaling Group (ASG) automatically adjusts the number of EC2 instances based on demand to maintain application performance and availability. It replaces unhealthy instances and scales resources up or down using defined policies, helping optimize cost and ensure reliability.

ASG is used to scale up and scale down instances based on certain metrics. It automatically add one new instance if number users or requests are high

* Create one VPC in N. Virginia region  
  
* Create two subnets: one public subnet and one private subnet  
  
* Attach an IGW to the VPC  
  
* Create one public route table (RT) and one private route table  
  
* Deploy a NAT gateway in the public subnet and attach the NAT gateway to the private subnet  
  
* Create two instances, one in the public subnet and one in the private subnet  
  
* Deploy Apache server on both EC2 instances with a sample index.html file  
  Take 2 public instances and connect both instances and install Apache in both instances  
  Instance 1   
    
  Instance 2  
    
  And start httpd and enable it and check the status in both instances  
  Instance 1  
    
  Instance 2  
    
  After installing Apache, create sample index.html on both instances  
  Instance 1  
    
  Instance 2  
    
  Now copy the public ip and paste it on browser  
    
  Instance 2  
  
* Create one application load balancer and attach it to both EC2 instances  
  Go to EC2 on left we can see load balancer click on create load balance and set application load balance and create  
    
  And provide name as my-alb, scheme as internet-facing and load balance type ipv4  
    
  Now select same vpc as ec2 and availability zone select subnet  
    
  And now create new security group and provide http 80  
    
  Now click on select create target group as instance and give group name and next  
    
  Select instances and click on including pending below and create target group  
    
  Load balancer successfully created   
    
  Now to load balancer and click on my-alb and copy the DNS name and paste it on browser  
    
  And refresh it to check the instance 2  
  
* Store application load balancer logs in S3  
  Go to aws console and search s3 and create a bucket  
    
  click on bucket and go to permissions and click bucket policy and paste the policy then click save changes  
    
  And go to load balancer, click on attributes and click on edit  
    
  And enable access logs and browse the bucket name and save changes  
    
  And go to s3 open aws logs and we can see logs are creating  
  
* Store the VPC flow logs in a CloudWatch log group  
  Go to aws console -> cloud watch and create log group  
    
  Then create IAM role for vpc flow logs, click on role select aws service and ec2 then next and select cloudwatchlogsfullaccess then click on create role   
    
  Open role and go to trust policy and update the policy   
    
  And go to vpcs and select the vpc and create flow log. Filter will be all and click on create flow log  
    
  And go to cloudwatch and click vpc flow log and we can see the log streams  
  
* Create monitoring dashboards to monitor CPU utilization and to monitor the Apache service  
  Create a Dashboard and select line and select metrics ec2 and- preinstance metrics- instanceid-click on cpu utilization and click on create wedget and save dashboard  
  
* If CPU utilization is more than 70%, then it should trigger auto scaling and launch new instance  
  Go to ec2, launch templates and click on create launch templates then provide name and browse Amazon Linux 2023 kernel-6.1 AMI, and instance type as t3.micro and add existence key pair and add security groups the add the user data  
     
  Click on create launch template  
    
  And go to auto scaling group and create auto scaling group and then choose apache template and click on next  
    
  Select Vpc and select 2 subnets  
    
  And provide group size  
    
  And provide scaling policy   
    
  Auto scaling group is created   
  