**AWS Task-3**

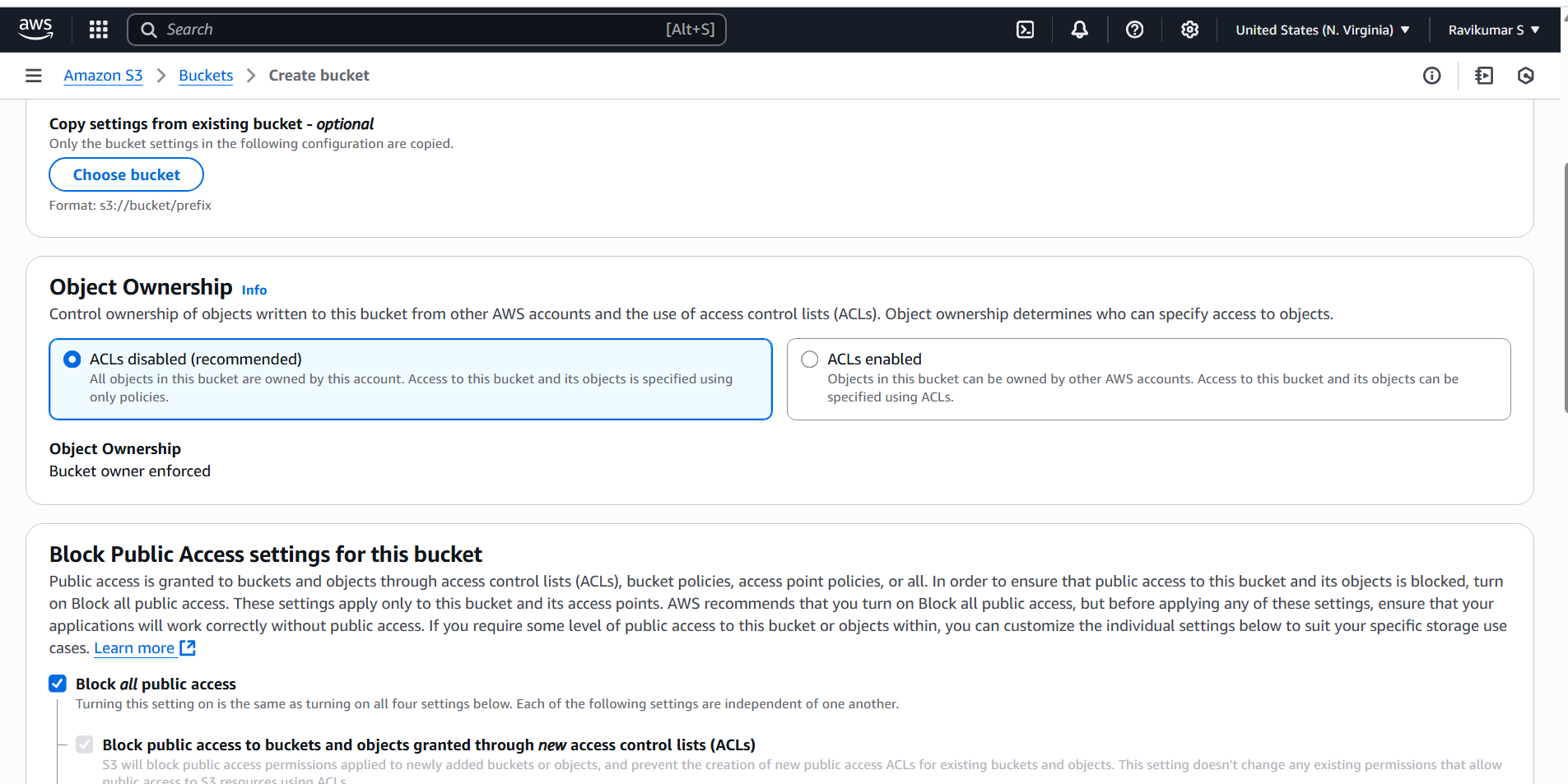
**Task Description:**

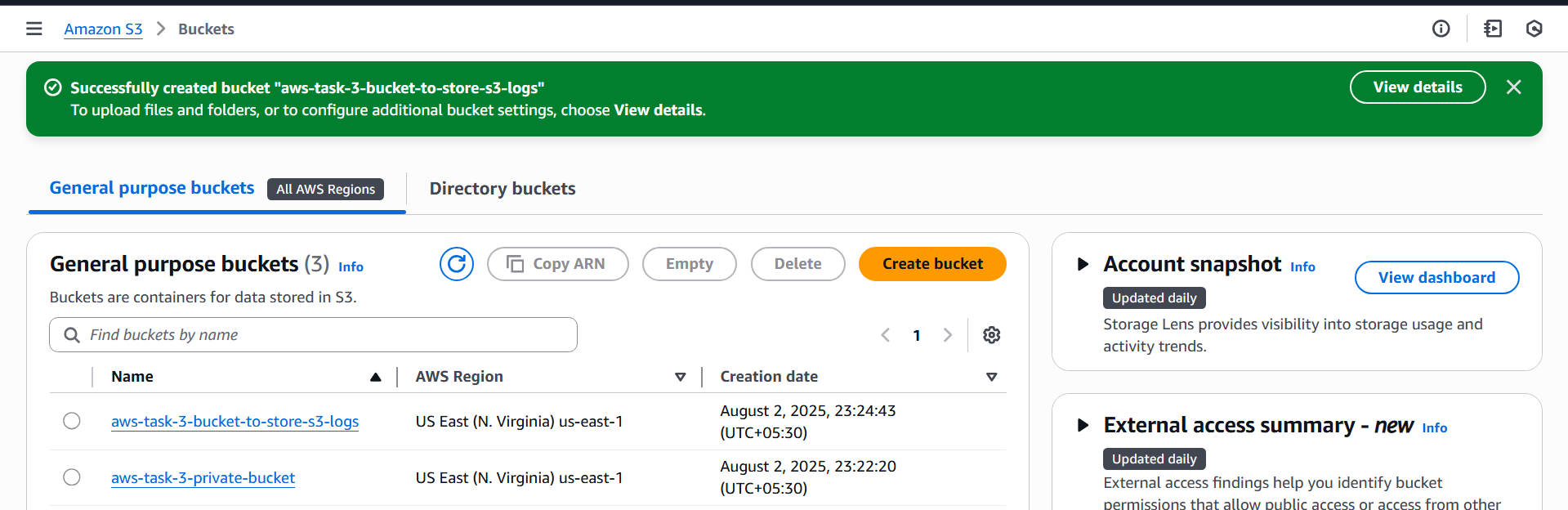
1. Create a S3 bucket, with no public access and upload files to the bucket & view the logs using cloudwatch for the uploaded files.

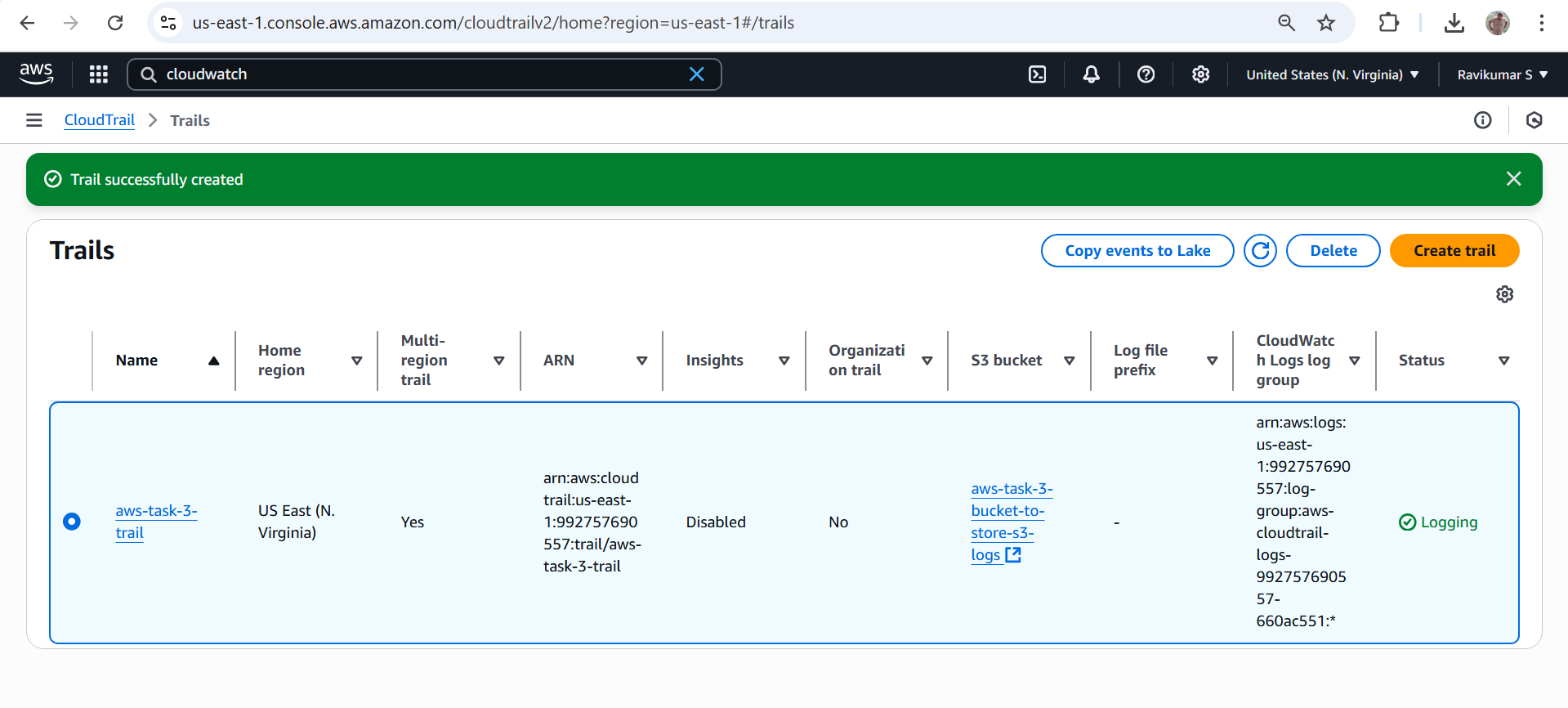
Steps:

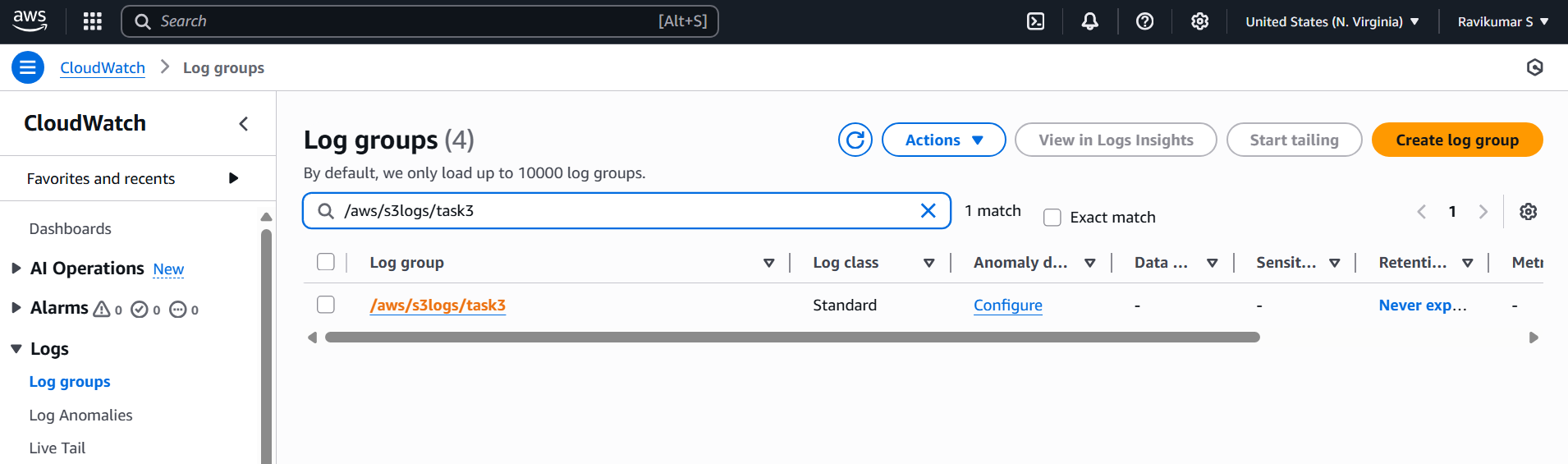
S3 bucket doesn’t push logs to cloudwatch directly. We can make it store them in another s3 using cloud trail and once the logs come to another s3 bucket we create a event notfication with lambda and extract the lgos from s3 location and push them to cloudwatch

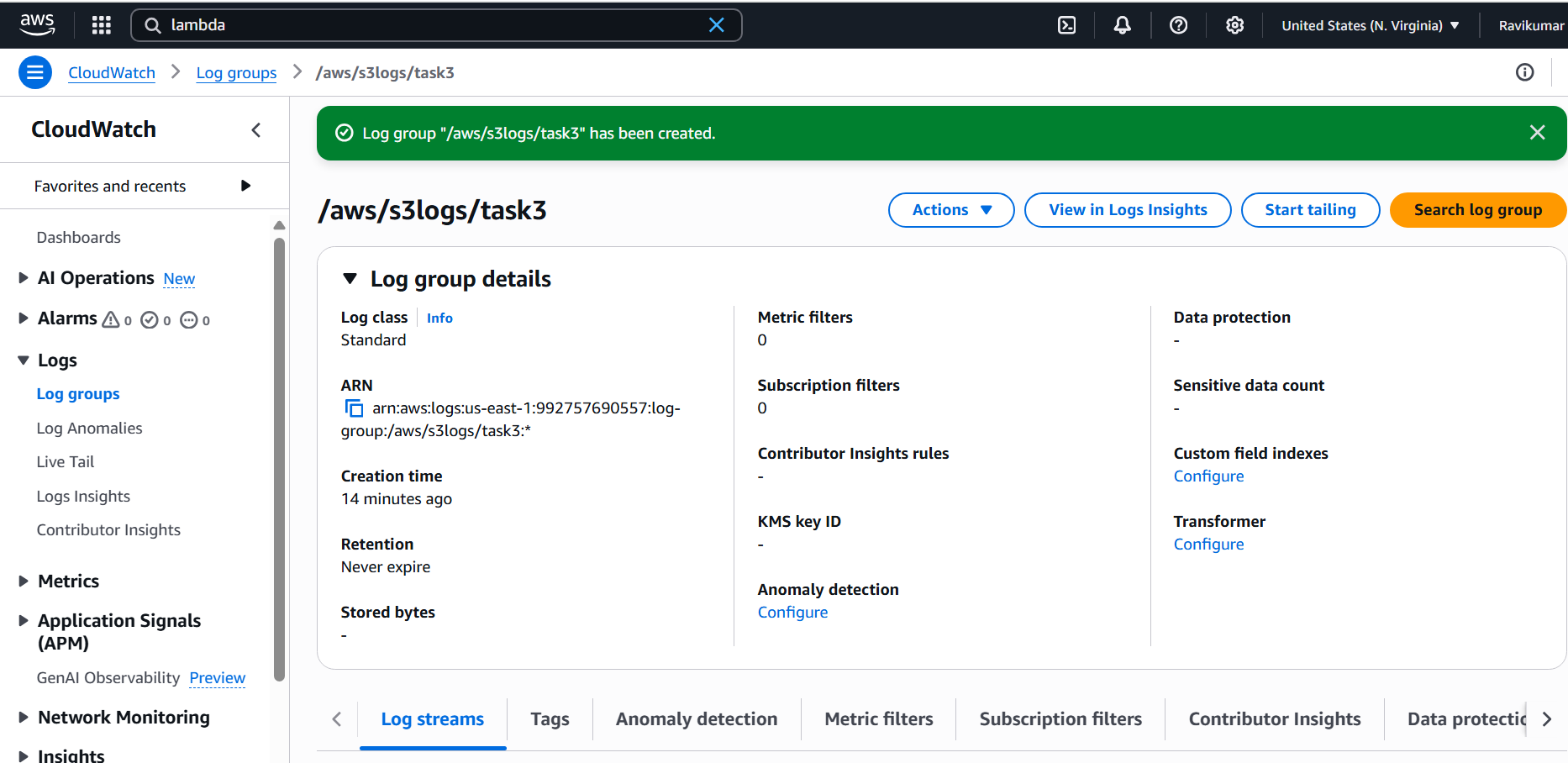
Creating 2 buckets – 1 private bucket with no public access where file will be uploaded and another to store the logs

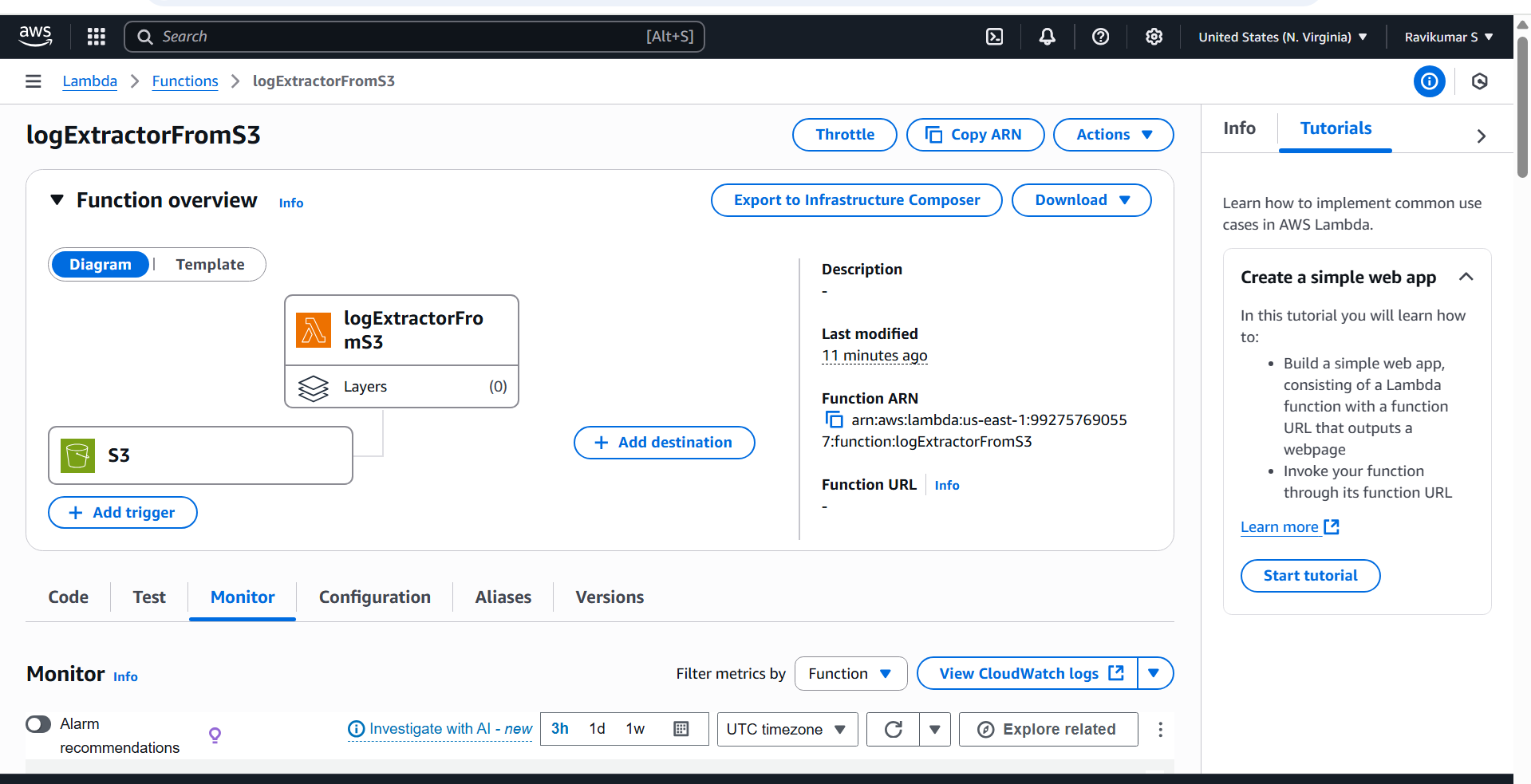




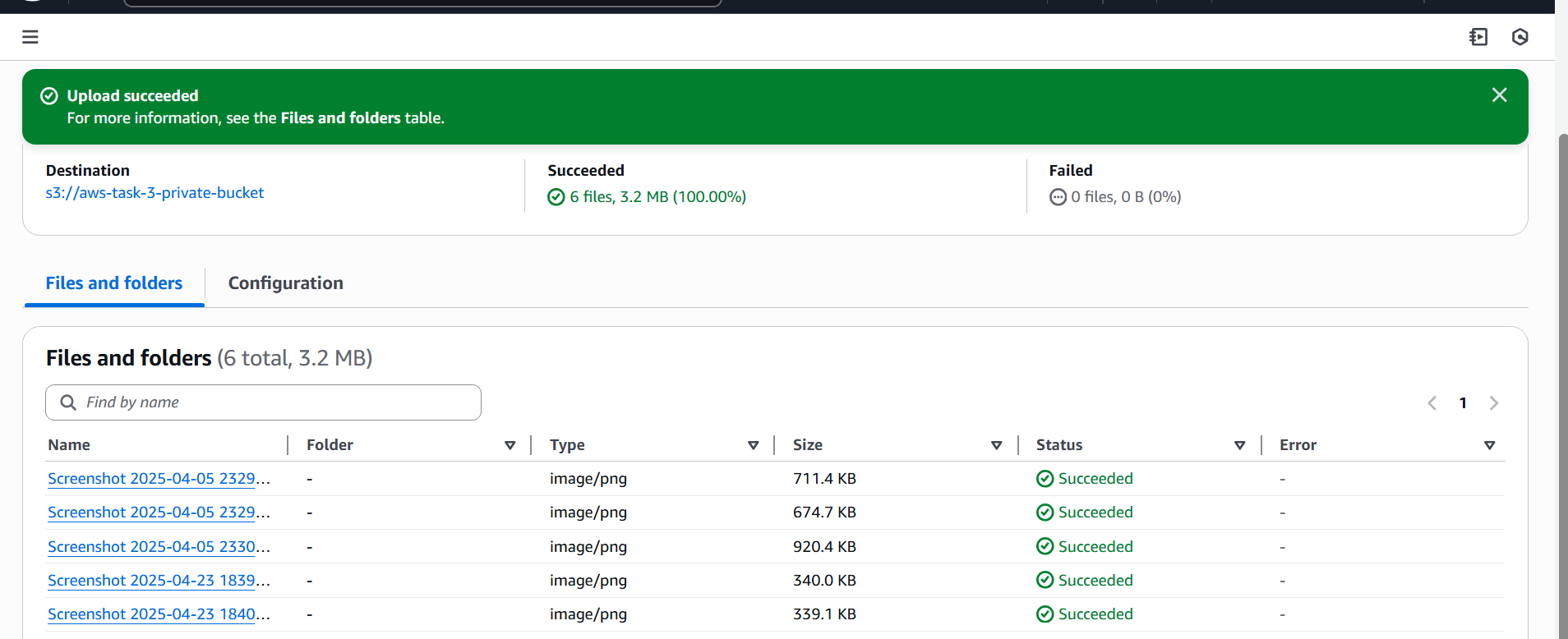


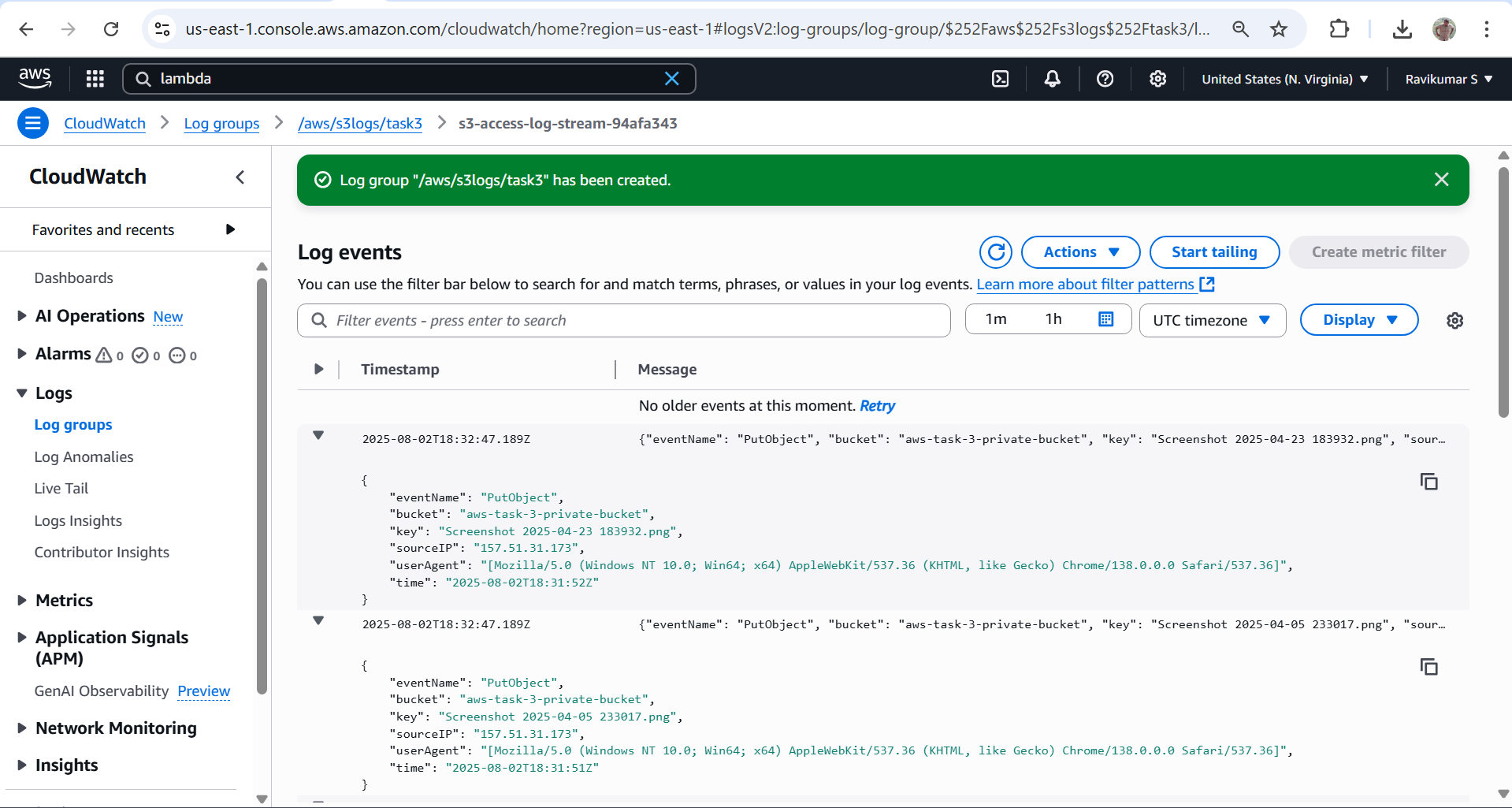


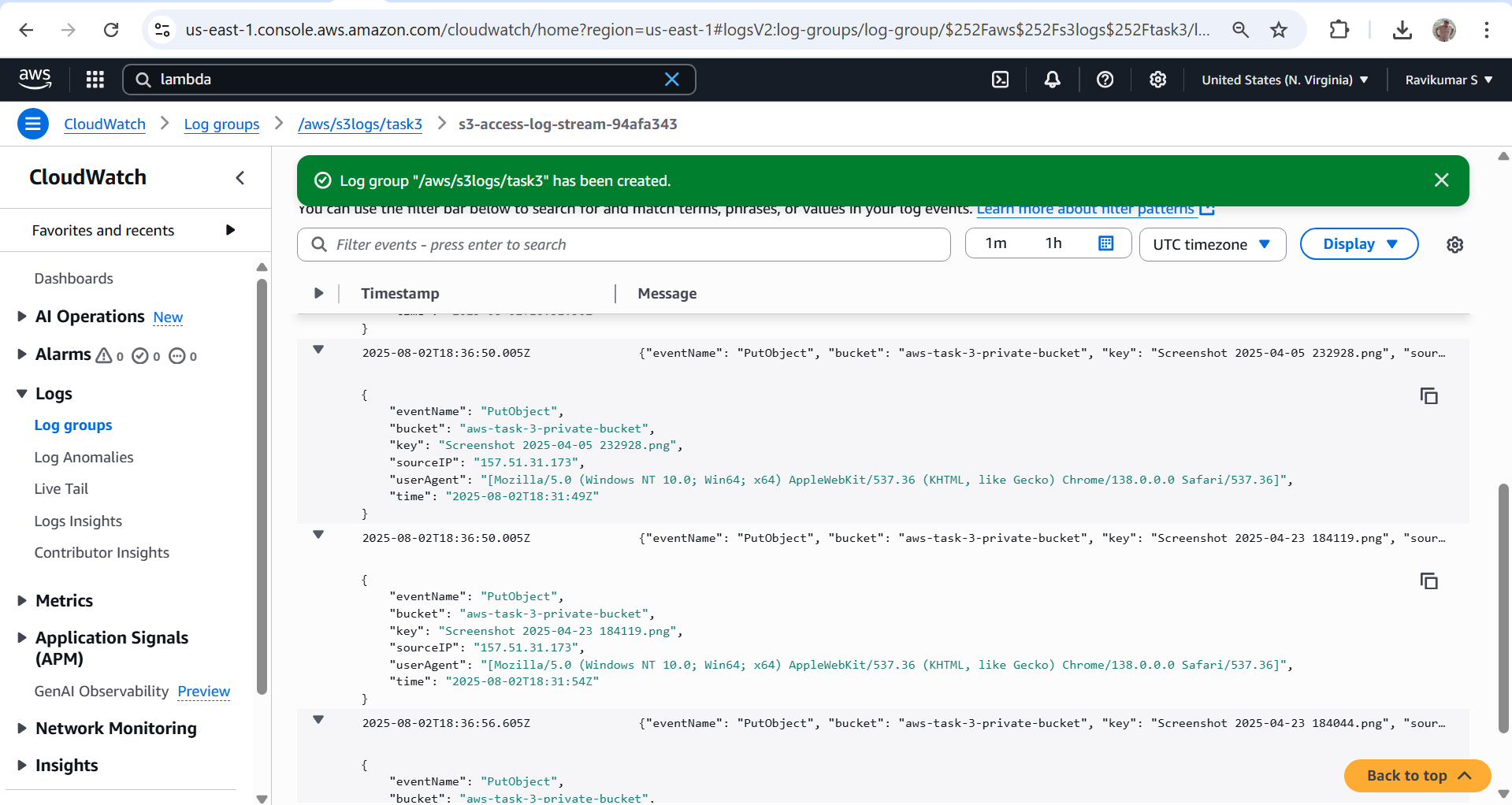




Uploading files to private s3 bucket







One such log

{

"eventName": "PutObject",

"bucket": "aws-task-3-private-bucket",

"key": "Screenshot 2025-04-05 232928.png",

"sourceIP": "157.51.31.173",

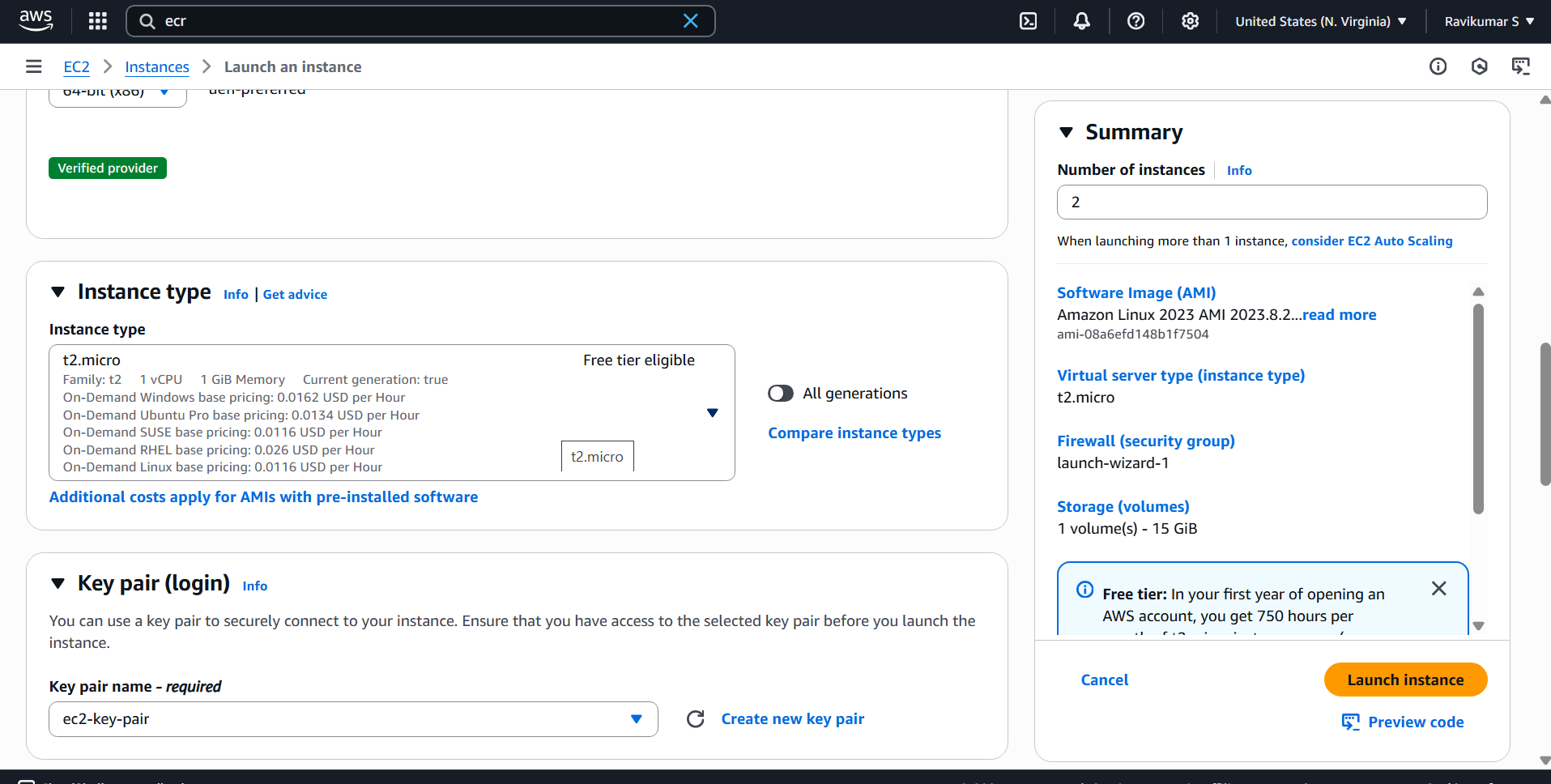
"userAgent": "[Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36]",

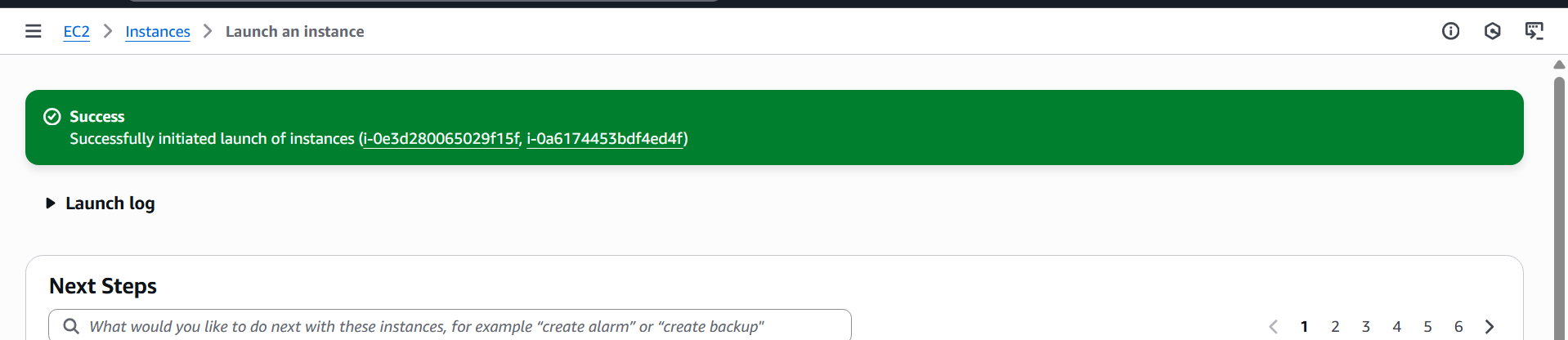
"time": "2025-08-02T18:31:49Z"

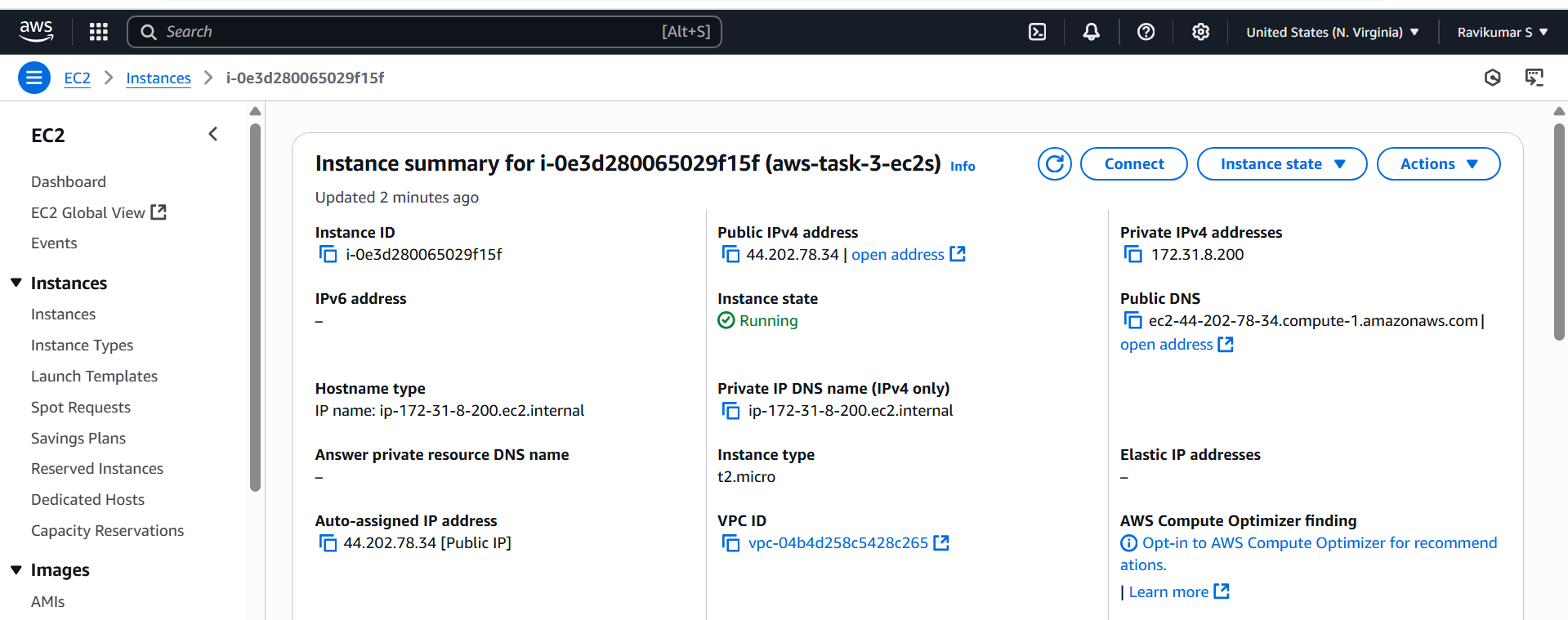
}

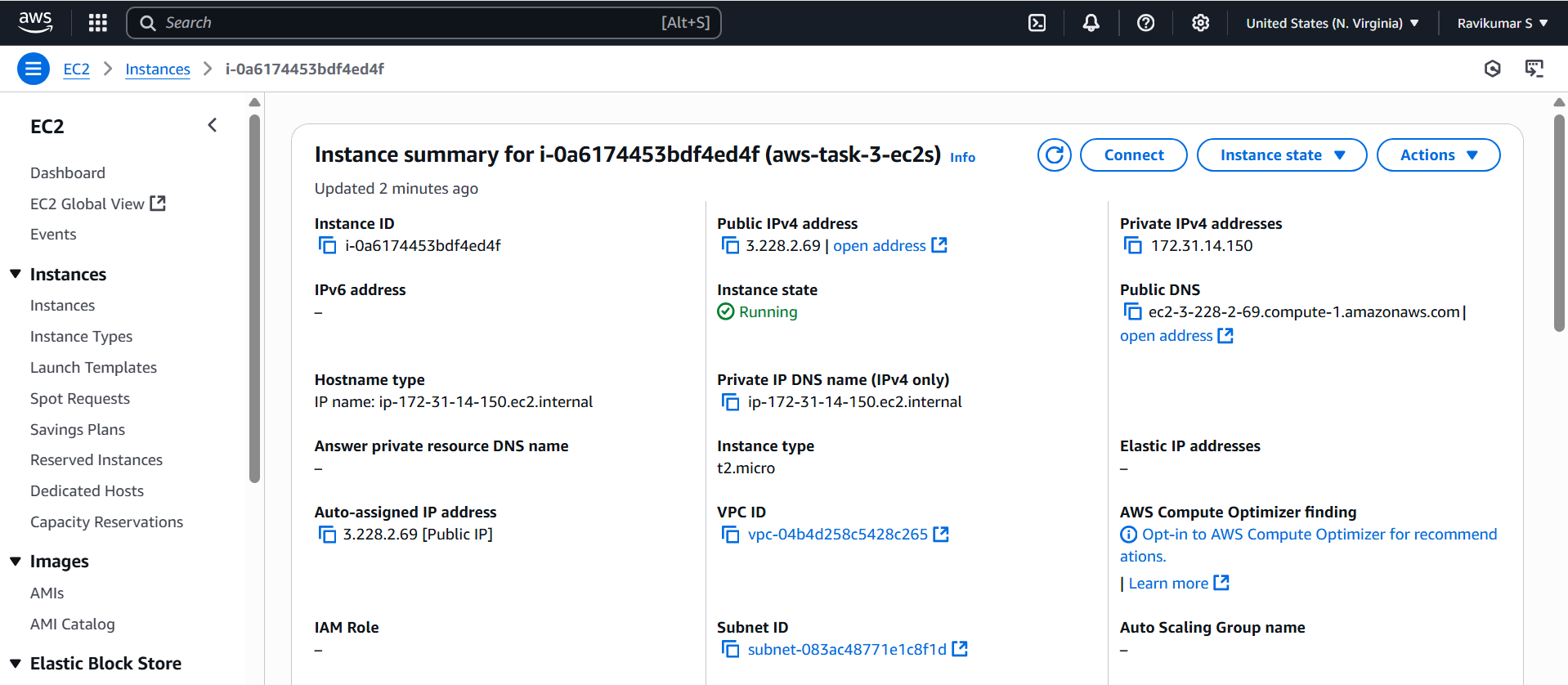
1. Launch two ec2-instances and connect it to a application load balancer, where the output traffic from the server must be an load balancer IP address

Launching 2 ec2 instances

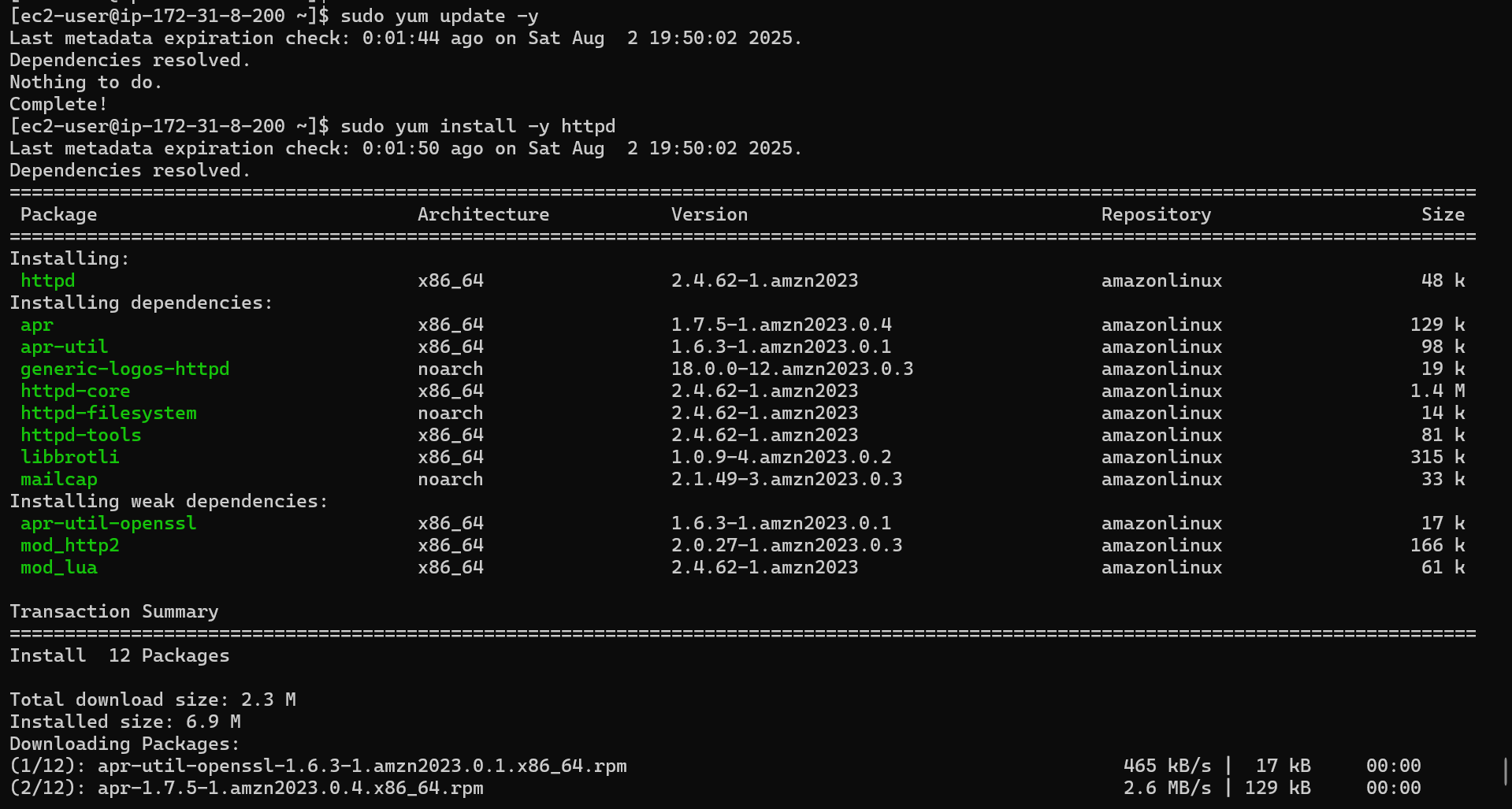


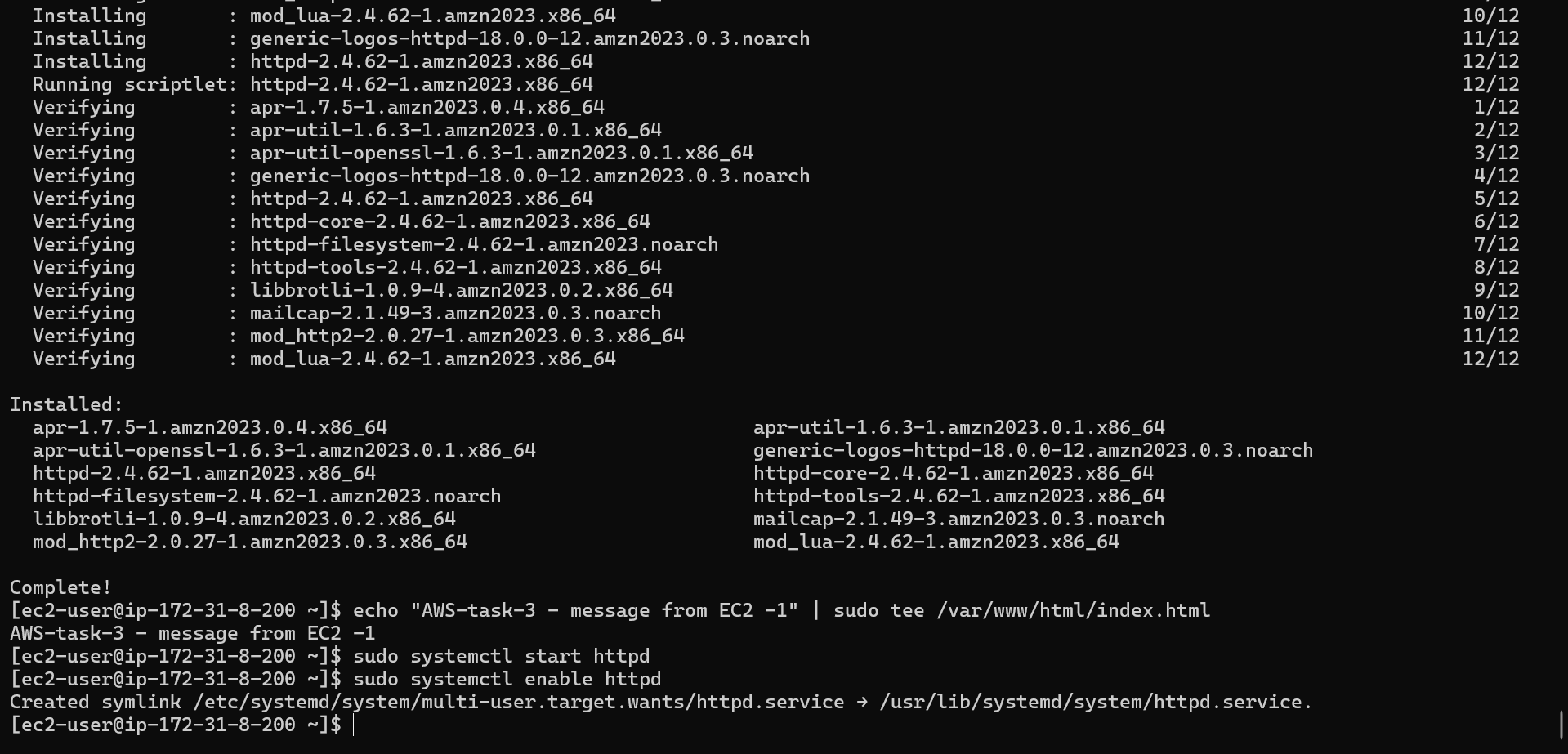


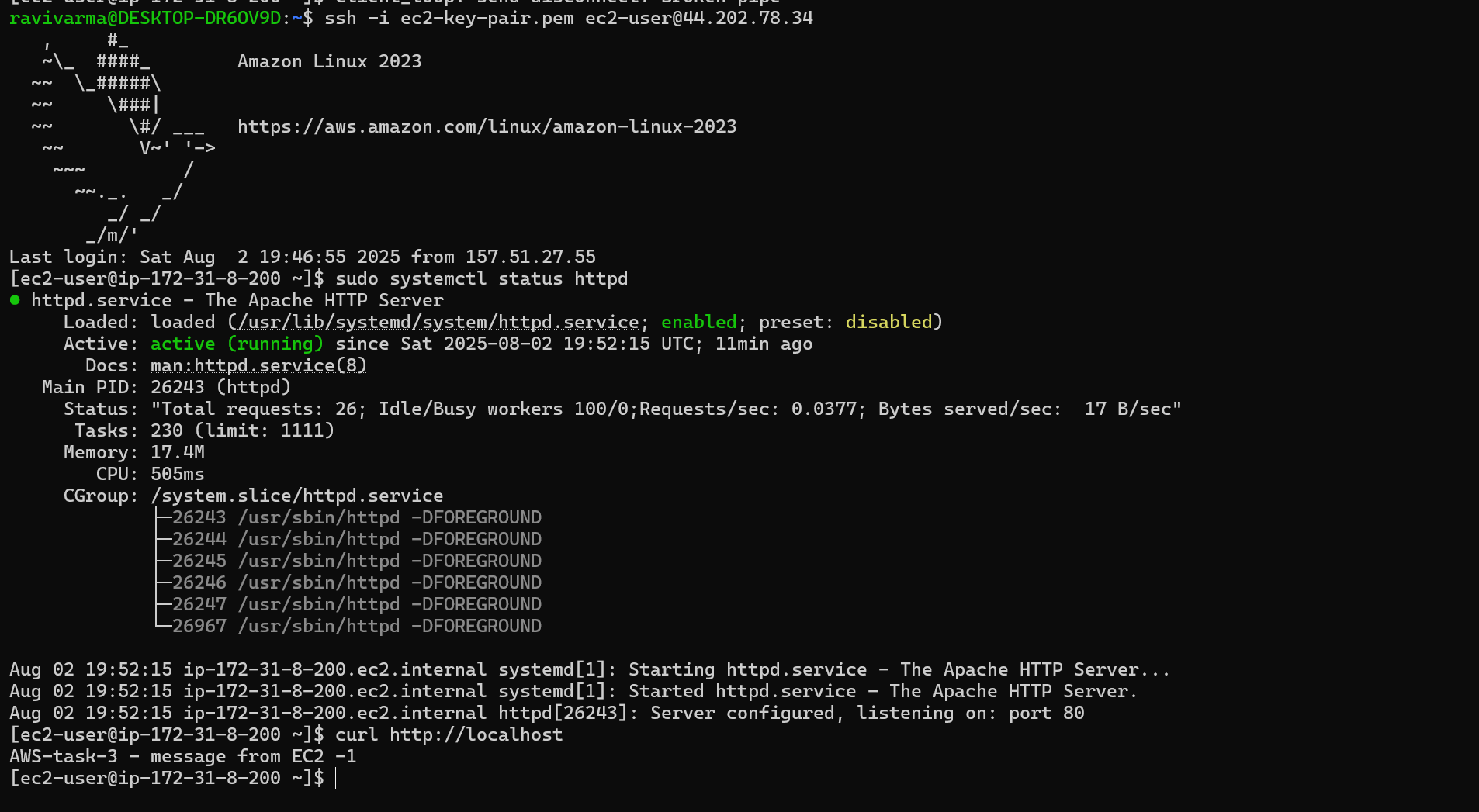


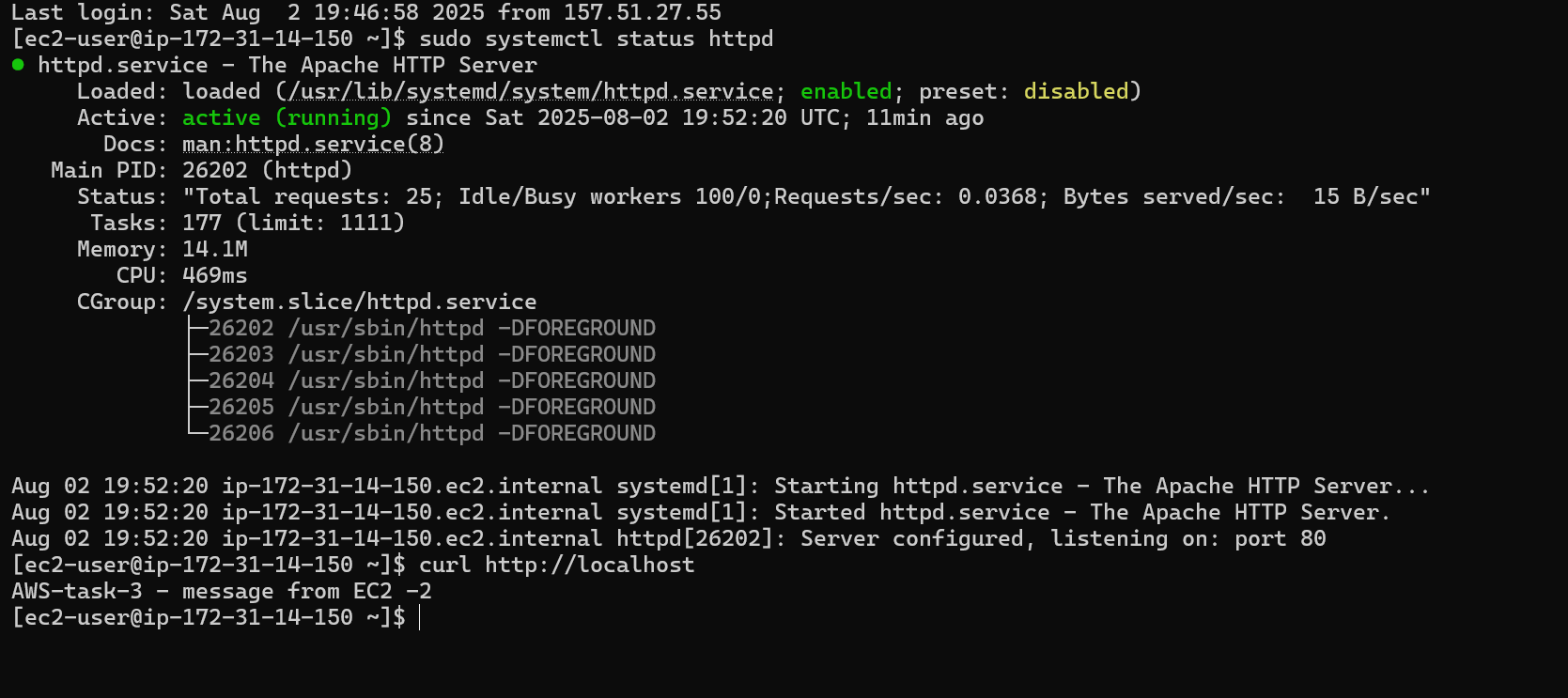


Installing simple web server in both ec2s



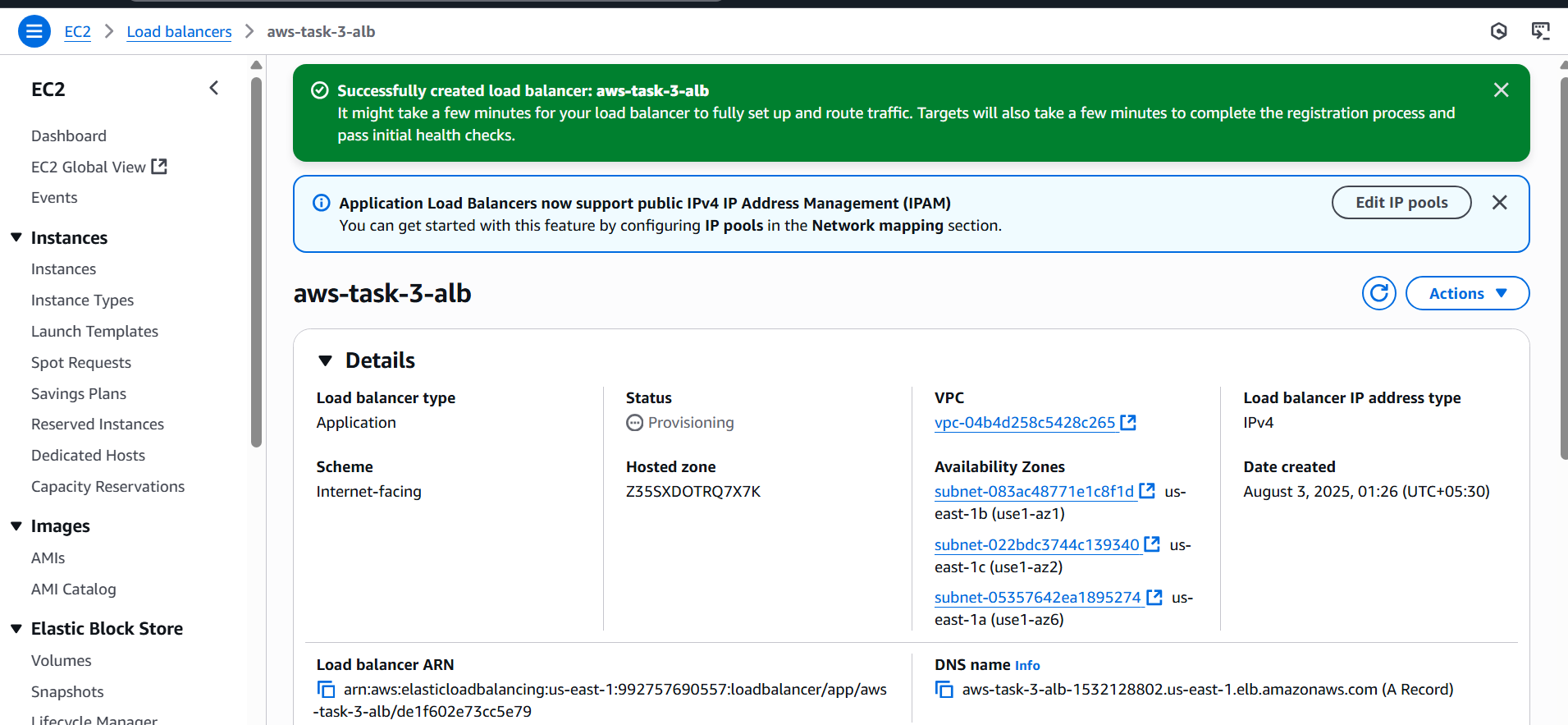


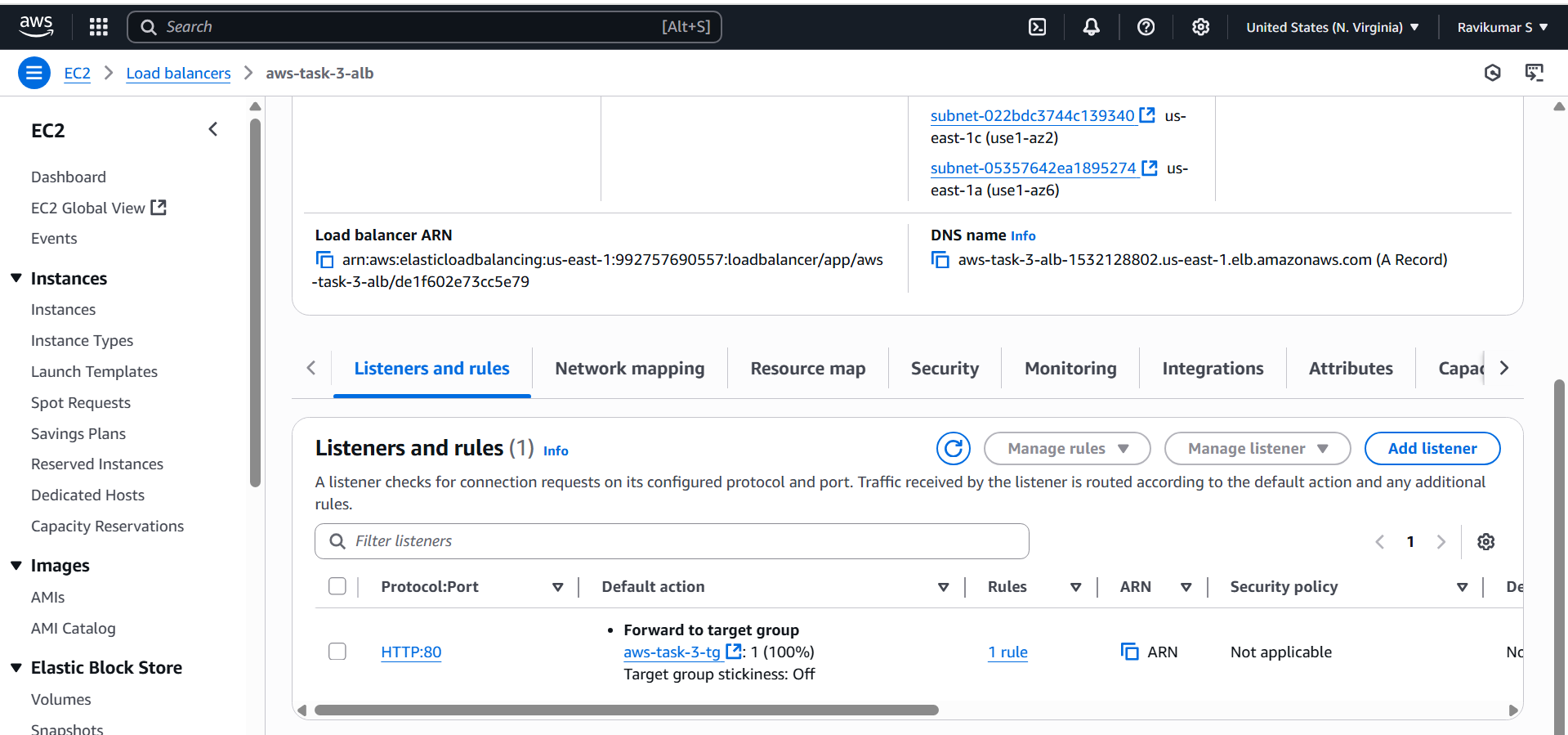




Now simple web server is running in both the EC2s.

Creating ALB with Target group with 2 EC2s





First hit



Next hit

