Secure S3 Bucket Runbook

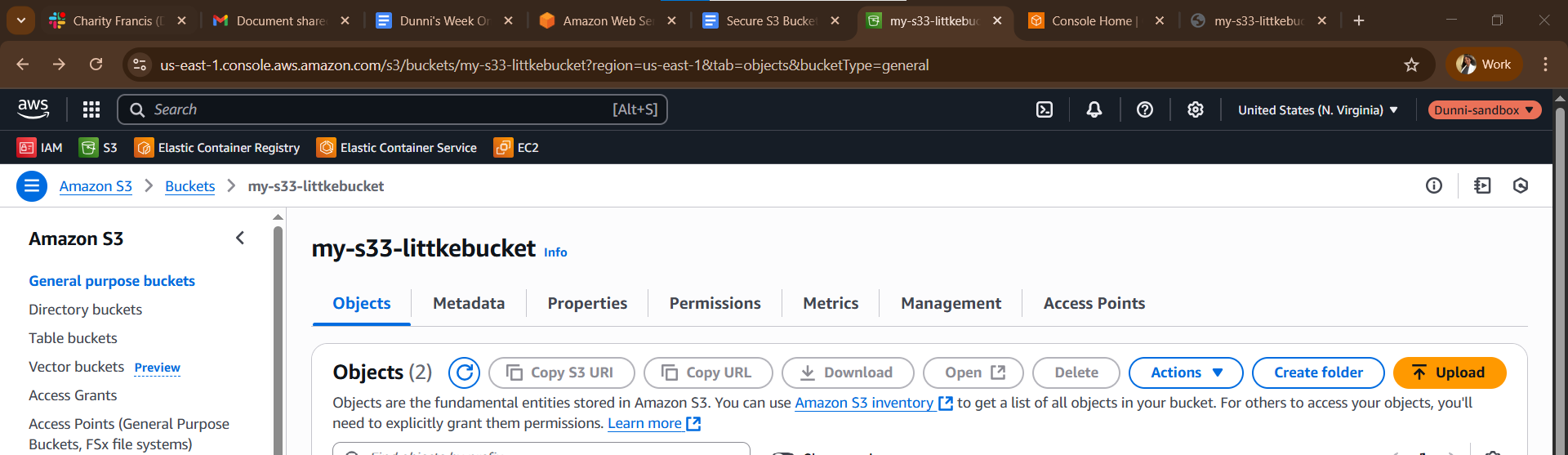
**Secure S3 Bucket Runbook**

Pre-requisites;

* Access to the console
* Network access to the internet

## **1 Create an Encrypted S3 Bucket**

1. Login to S3 in the AWS Console
2. Click on “create bucket”
3. Named my bucket—-”my-s3-littlebucket’’
4. Scrolled down to Default encryption:
   * Select Enable
   * Chose SSE-S3

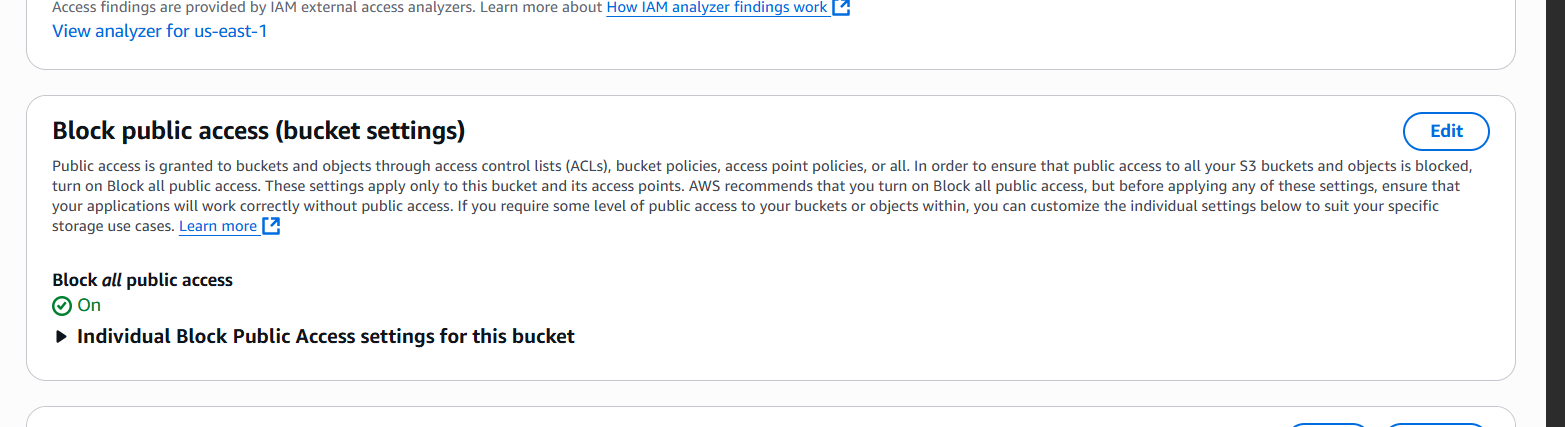
5. Enabled versioning  
 6. Leave other settings as default.  
 7. Click Create bucket.

My “my-s3-littlebucket” bucket is encrypted

## **2. Enable Versioning**

1. Go to the new “my-s3-littlebucket”
2. Click the Properties tab.
3. Scroll to Bucket Versioning.
4. Click Edit, choose Enable, then Save changes.
5. Versioning is now on.

## **3. Block All Public Access**

1. Go to the Permissions tab of the bucket
2. Click Edit next to Block public access.
3. Check all four options :
   * Block public ACLs
   * Block public bucket policies
   * Ignore public ACLs
   * Restrict public bucket policies
4. Click on Save changes.
5. Public access is blocked.

## **4. Add a Bucket Policy for IP Restriction**

1. Go to the Permissions tab.
2. Scroll down to Bucket policy.
3. Click Edit.

Paste this policy, replacing:

* my-secure-bucket with my bucket name; my-s3-littkebucket
* Replaced YOUR\_IP\_ADDRESS with my actual IP; 54.87.233.162

{

"Version": "2012-10-17",

"Statement": [

{

"Sid": "AllowGetFromMyIP",

"Effect": "Allow",

"Principal": "\*",

"Action": "s3:GetObject",

"Resource": "arn:aws:s3:::my-s3-littkebucket/\*",

"Condition": {

"IpAddress": {

"aws:SourceIp": "54.87.233.162/32"

}

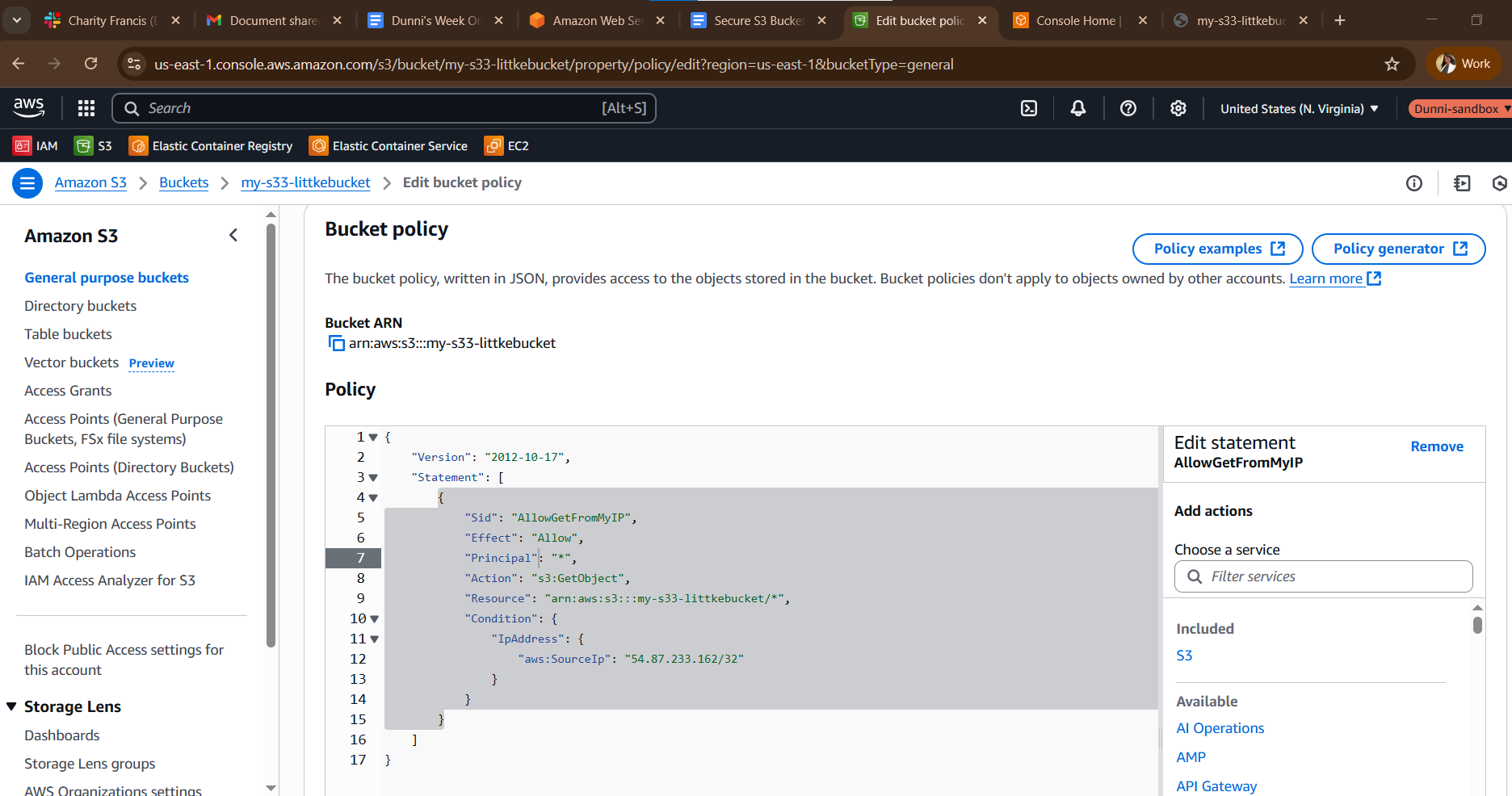
}

}

]

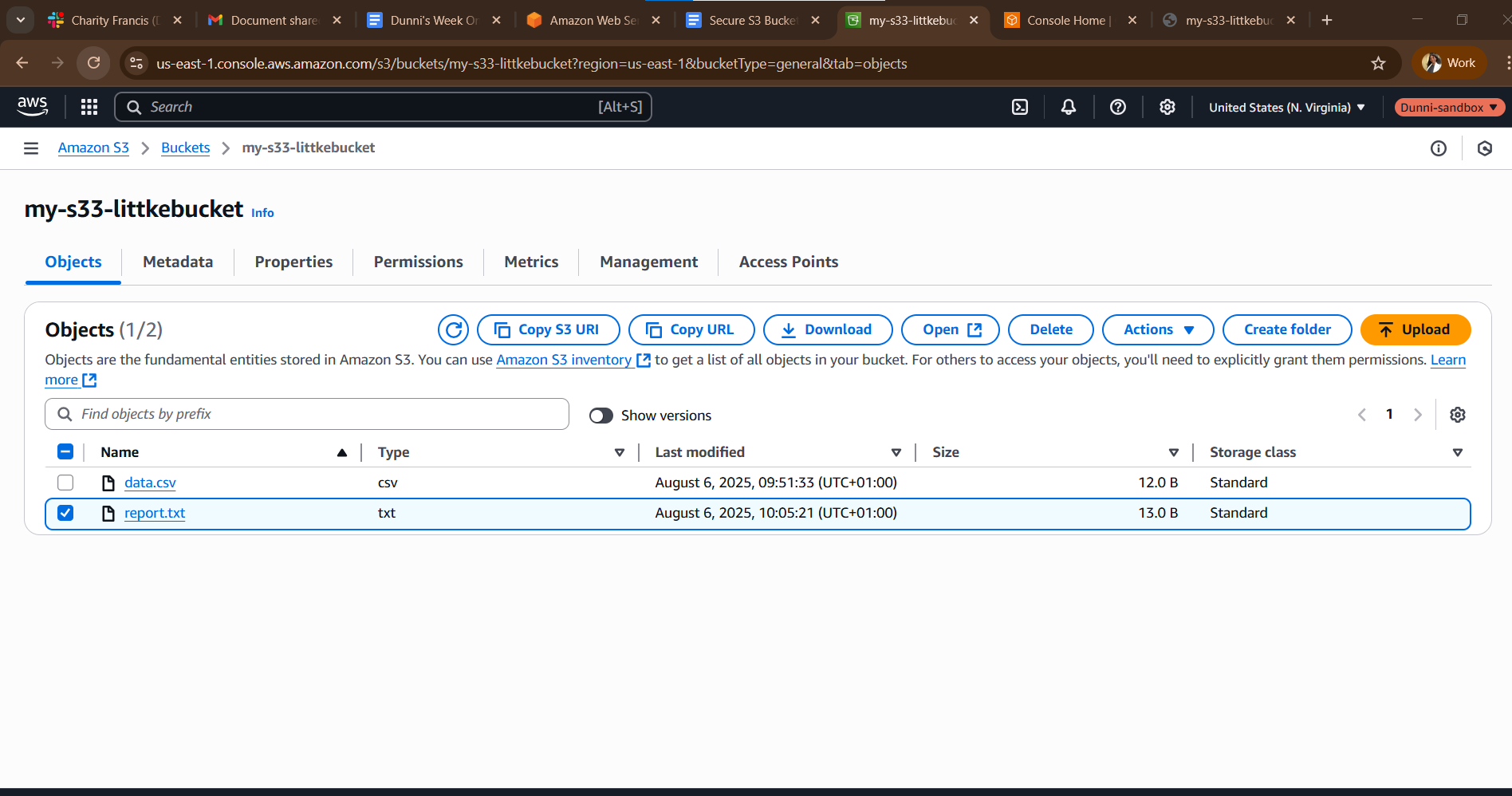
}



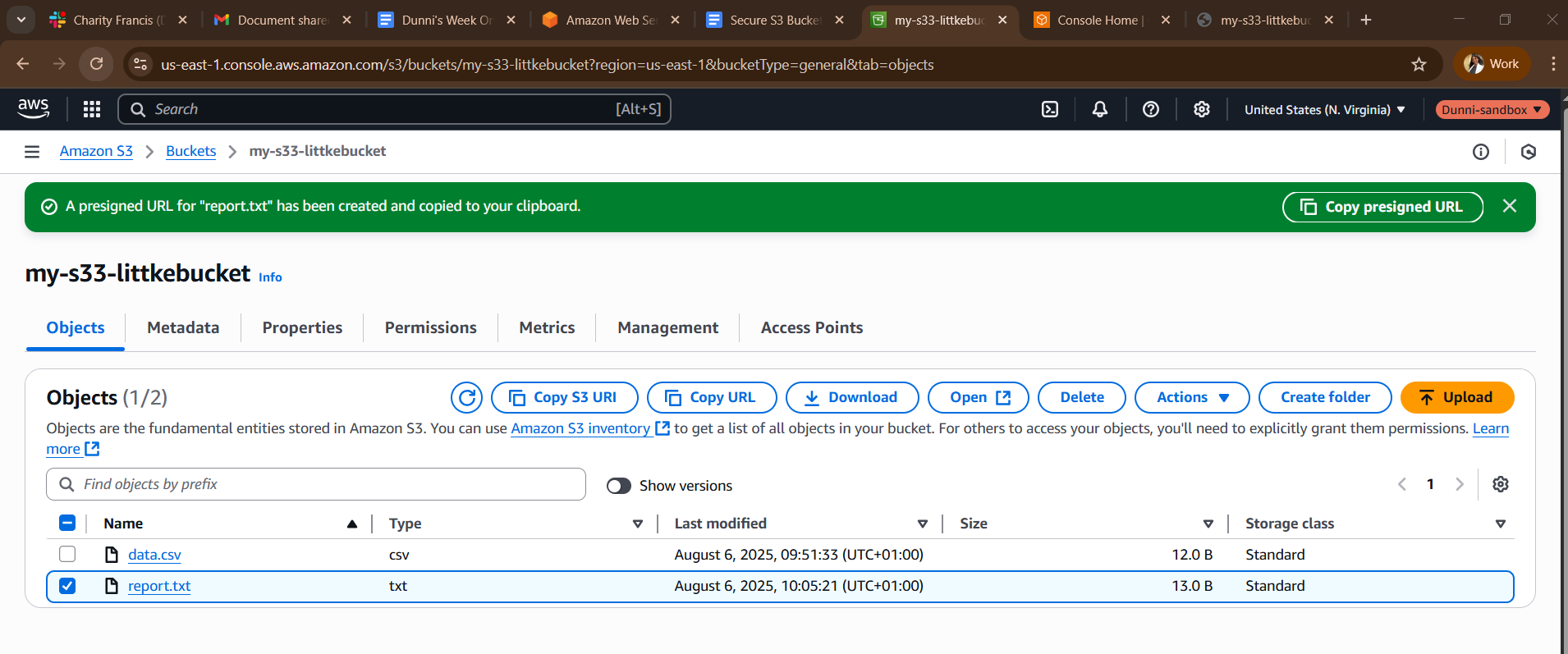
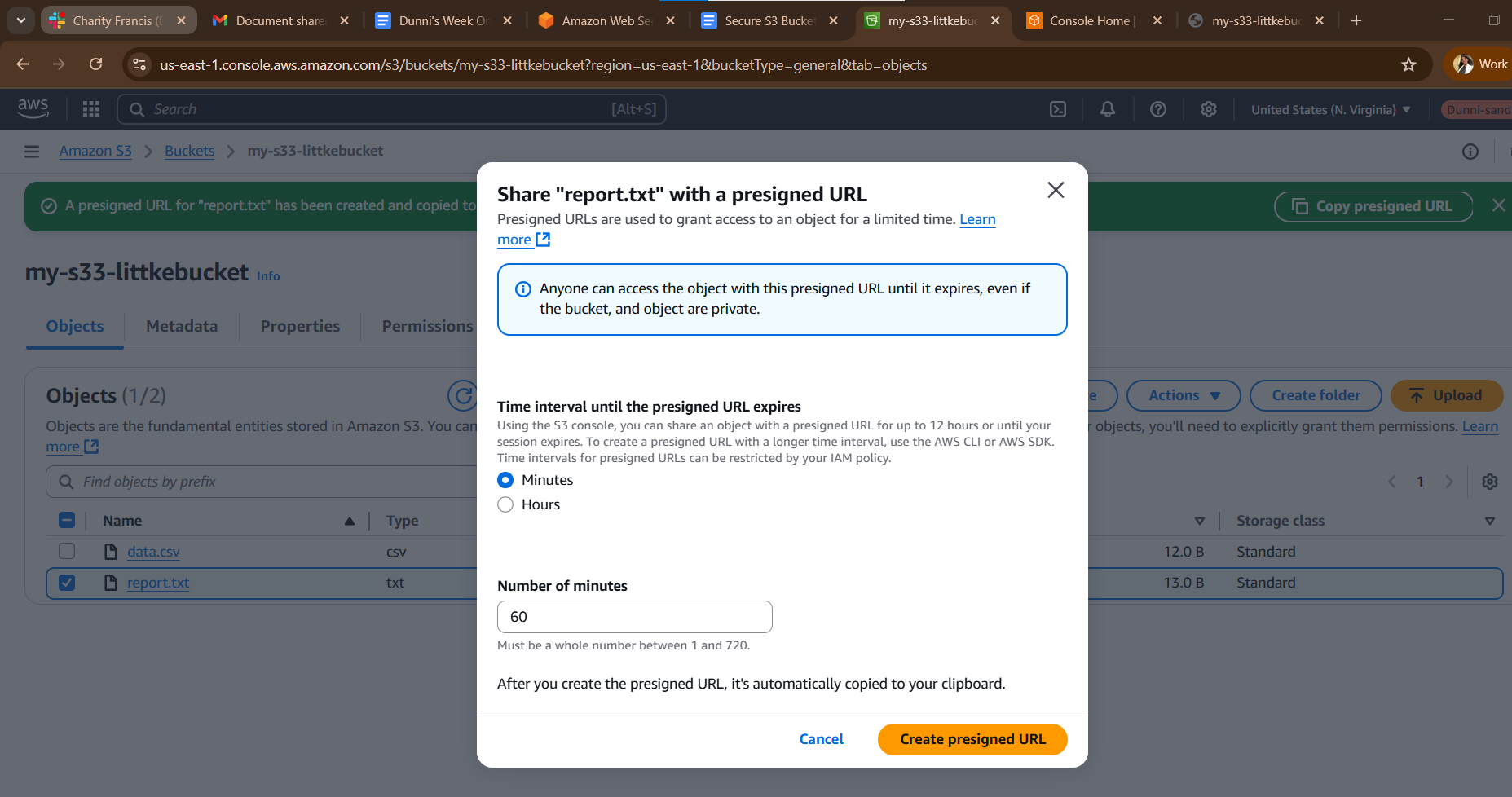
* Click Save changes.
* Successfully edited the policy

## 

## **5. Upload Sample Files via Console**

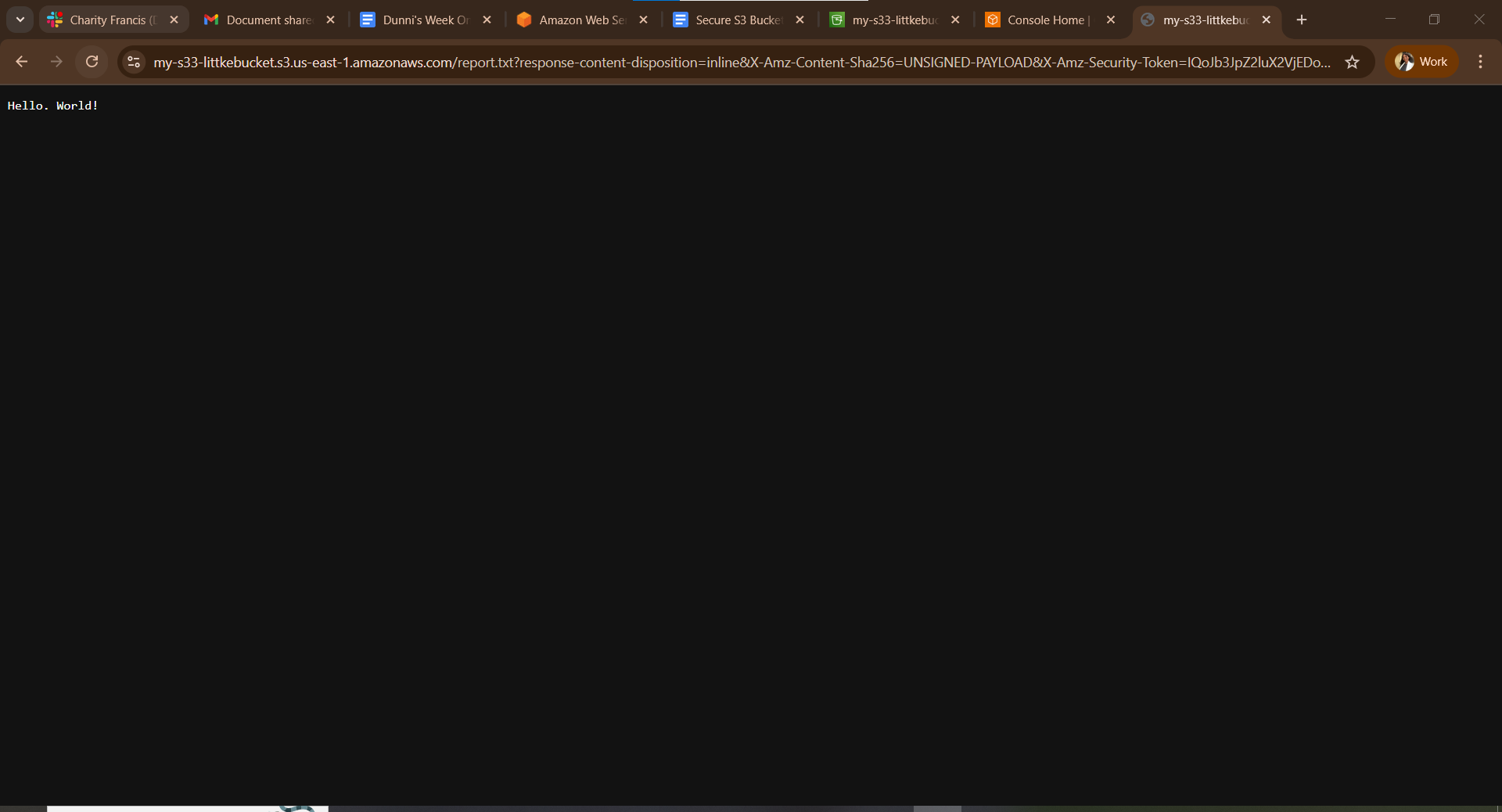
1. Go to the bucket– Objects tab.
2. Click Upload.
3. Upload two files  
   * report.txt
   * data.csv 
4. Click Upload.
5. Files are now in S3.

## **6. Generate a Pre-signed URL**

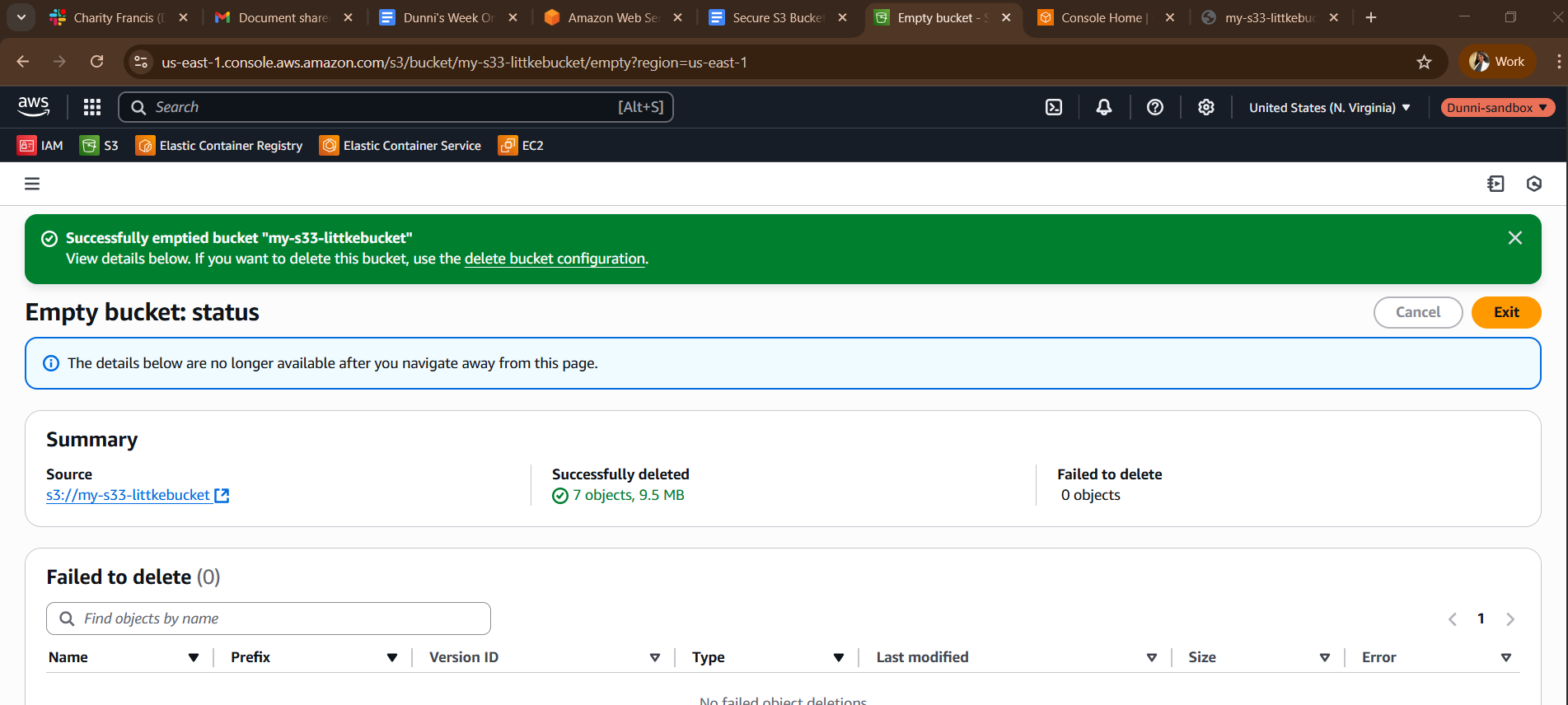
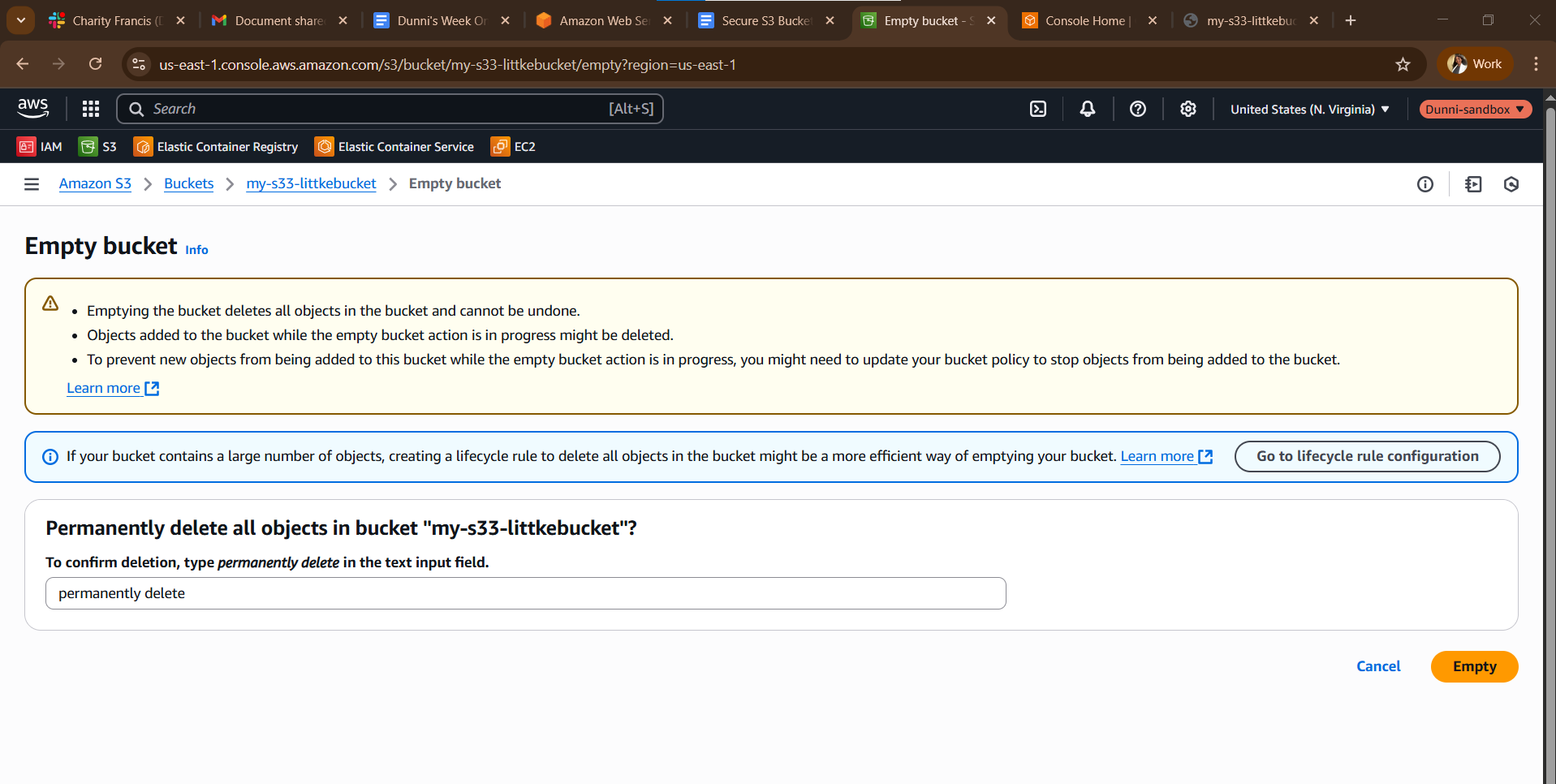
1. In the bucket, find report.txt under Objects.
2. Click the checkbox next to it.
3. Select Actions; share with a pre-signed URL.
4. Set expiration time (60 minutes).
5. Copy the generated URL.

https://my-s33-littkebucket.s3.us-east-1.amazonaws.com/report.txt?response-content-disposition=inline&X-Amz-Content-Sha256=UNSIGNED-PAYLOAD&X-Amz-Security-Token=IQoJb3JpZ2luX2VjEDoaCXVzLWVhc3QtMSJIMEYCIQD9Y%2FLnUKptIEWlYfQe7tmjYRPwRFvP9bMd1UnrFazFYwIhAPXc0PGo38BH%2BvUXhLyQdMZ%2F7JSEMQ7HQew5jBM1TOJKKtcDCHMQARoMNDg5NjczMDIxMDkyIgwUHKplsefkhQPHh28qtANZx%2FuxU0arelWyX%2Fwoy2zHs9ww12AMtrGW1j4ZspQhtORPUI%2B5JuBjKTQbQY3fvUlxqb4FlKolC5TxVYfTP4IraT%2FRnhHpjkmZPaySAjICZzuPG2WsX6lPtHK5va69ntGj0JY5WW%2BScL0tZRreaY3hYlNj7nvWFqXG0l6AiKOndB7BaQCtxOBuAcFLMJX0f4bkBhve7MO3NhnAkxJP2zZjBCirCX89fESHvDx36m1QdhVuEm9Qz7Qbd7QKjLGydflSUSk3rf76I0pfJydUMVmY2iLSwAvq%2BawWBDVxrgbWxaPI9JzCpdBQyeh7cLBuD%2BN3UQDZbC1GycXL6TiY%2FGWNUsAXoG5PWBFDAtsDtiM9ZYLe5pzSq5joyDPTmyX5sQYx%2FNPAdnBGZF91MRgtvIDbHFoqecUxKGfor1kbyLK%2FM02doEUQgZ%2Bv4Ej5QIA7L6SFIVbdpfnEkX41IlO93vJaIolNKsfO%2FeVSBtRIdct7i3ujyDix6voRvbeHPH%2FF52T7pQOzavZWKlg4z4snuRkby0wiLJrrAtMGXKP46djJiMAjGK%2B6iNY3XR7i%2BWeegU4FUOOHMKu5zMQGOrYC2qIv4wOj%2B6it0D5JLoIA5w3i%2BPY6ob1Dzah4%2Bi9deVvSpuWsRhl4xJPKtApAhnE36IheUF6nQ4kfQ%2FELUvsTUOBkXa1bR4iKQj5h7d5oKzsYqn%2FOeC5KwAOevjfMr%2B5%2BMpOkDL7gTikl09Losqr8U6Niz59qFYGXzfNQwn6dNgBKnp7eutc3sO0kDmBpNrugPiA3HT4%2FpZaPIIvgdEUU3uROehe4S8lYgKWkAhzjMHsuBhPPSY6H0%2Bf6EWRx043U189PuDHB6sNLgds9yg%2BA%2B6nvVHP9BZDKYgCo%2B7x1DPZnBWZZnJxupY4%2BQd4u50u8M%2BzIG6xlZPYUJ0cEhoEy2zFaqpaiu7M6Q9L1Y3Q1T59w0Otqj6Y2frQrd%2FKggh346lBvIul%2F2t%2FwY%2FRB8E70UWxhlzTv9w%3D%3D&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=ASIAXEAWJK2SCYVWWX57%2F20250806%2Fus-east-1%2Fs3%2Faws4\_request&X-Amz-Date=20250806T092441Z&X-Amz-Expires=3600&X-Amz-SignedHeaders=host&X-Amz-Signature=5b2f786940355c019aabc2822327453245e9338ba176152783235f50f0ac73be



6. Test it by opening the URL in your browser.

**7. Clean Up**

1. Go to the bucket
2. Empty to delete bucket
3. Back to the S3 dashboard, select the bucket.
4. Then delete

EC2(Instance Deployment)

Instance Deployment

# **1. EC2 Instance Deployment Runbook**

# **Goal:**

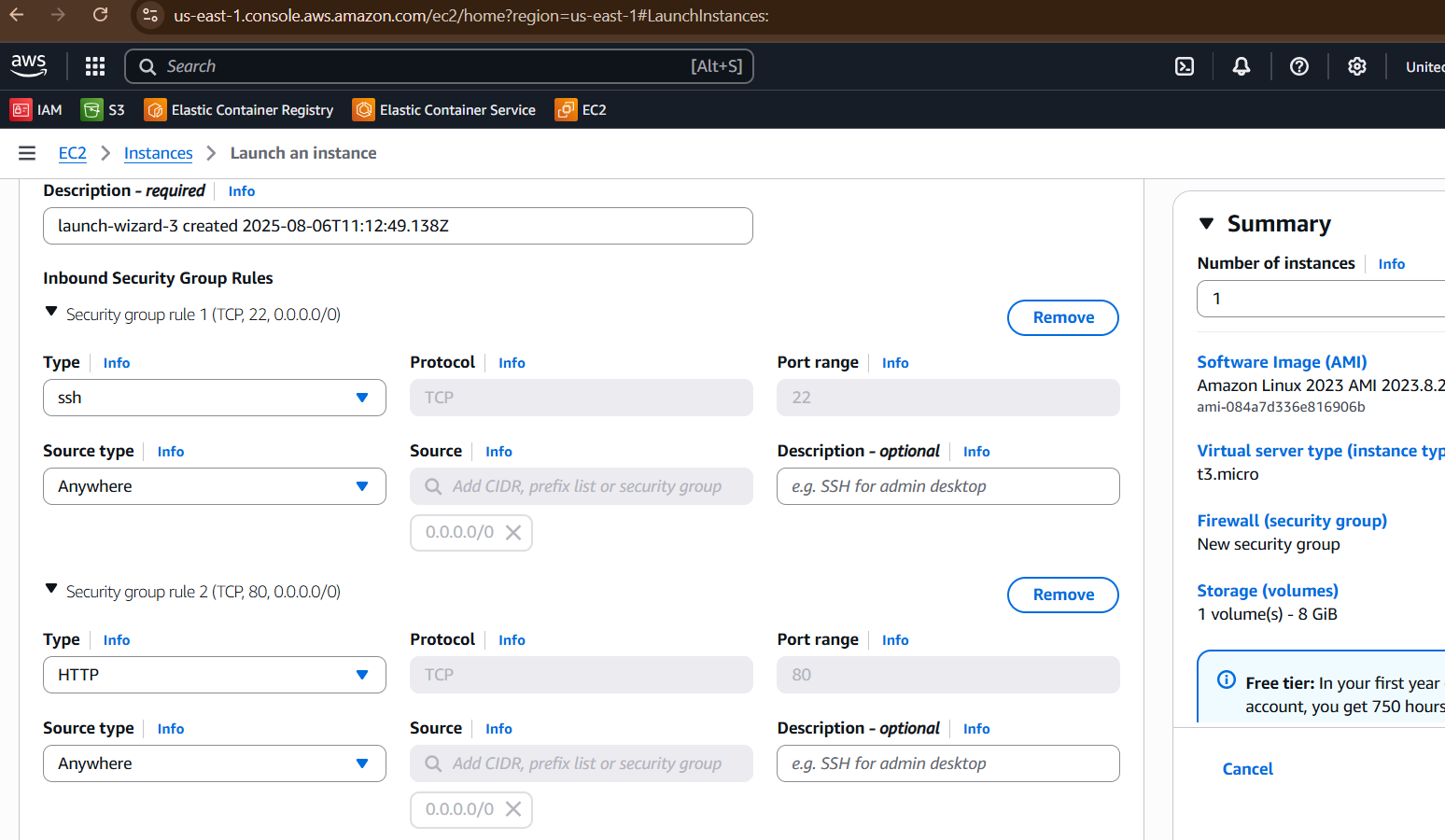
* Launch a t3.micro EC2 instance (Amazon Linux 2023)
* Set up Apache web server via user-data
* Connect via EC2 Instance Connect
* Validate web page in browser

## **1. Launch Virtual Server (EC2)**

### **Steps:**

1. Log in to AWS Console –EC2 – Instances
2. Click Launch instance

### **Fill in the details:**

* Name: My-instance
* AMI: Choose Amazon Linux 2023
* Instance type: Select t3.micro
* Key pair: Existing keypair “dun-dun-doon
* Network settings:
* Allow SSH (port 22) and HTTP (port 80)****
* Storage: Keep default (8 GiB)

**Install Apache using user-data script**

* Scroll to Advanced details – paste this in User data:

#!/bin/bash

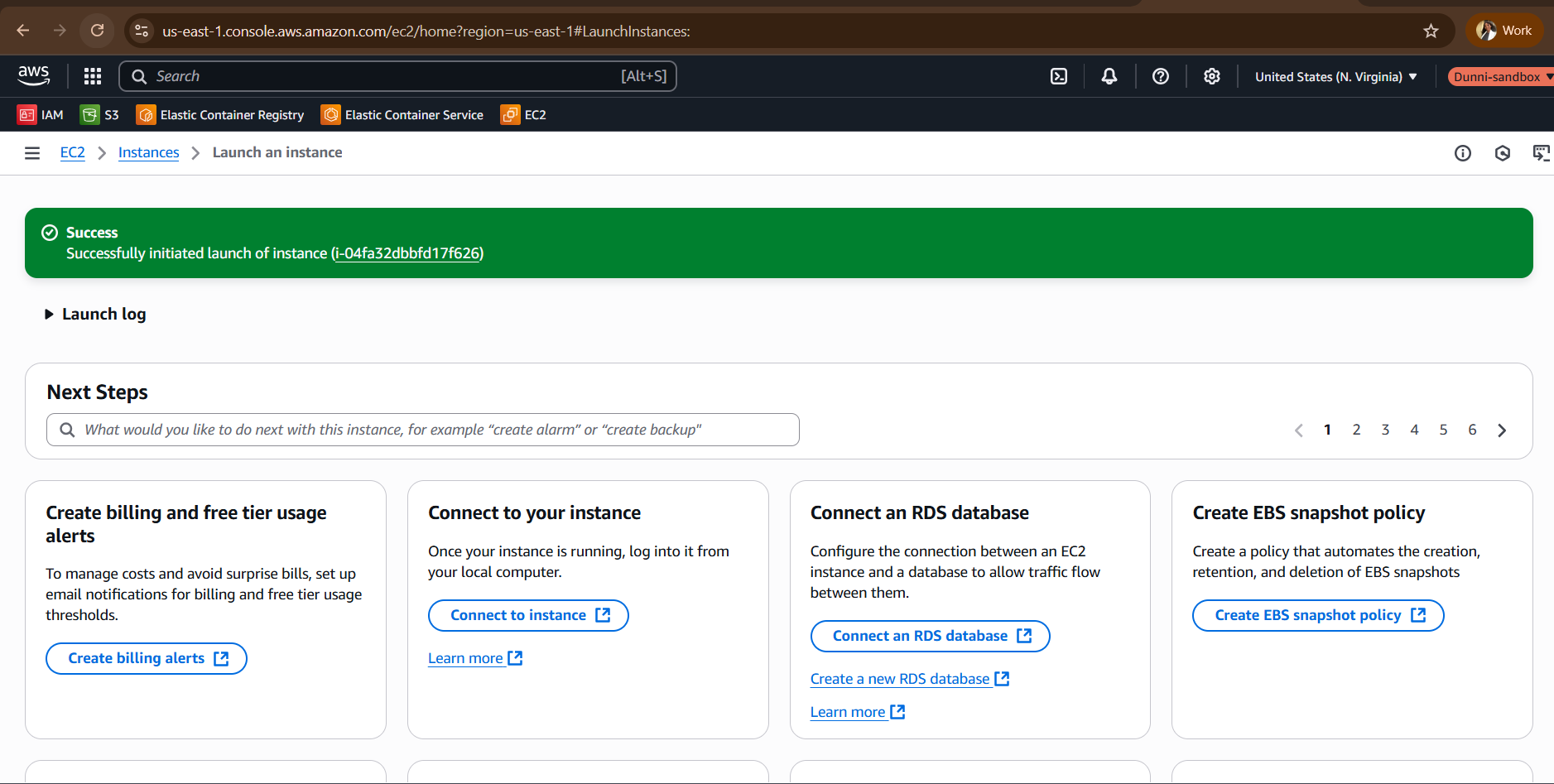
yum update -y

yum install -y httpd

systemctl start httpd

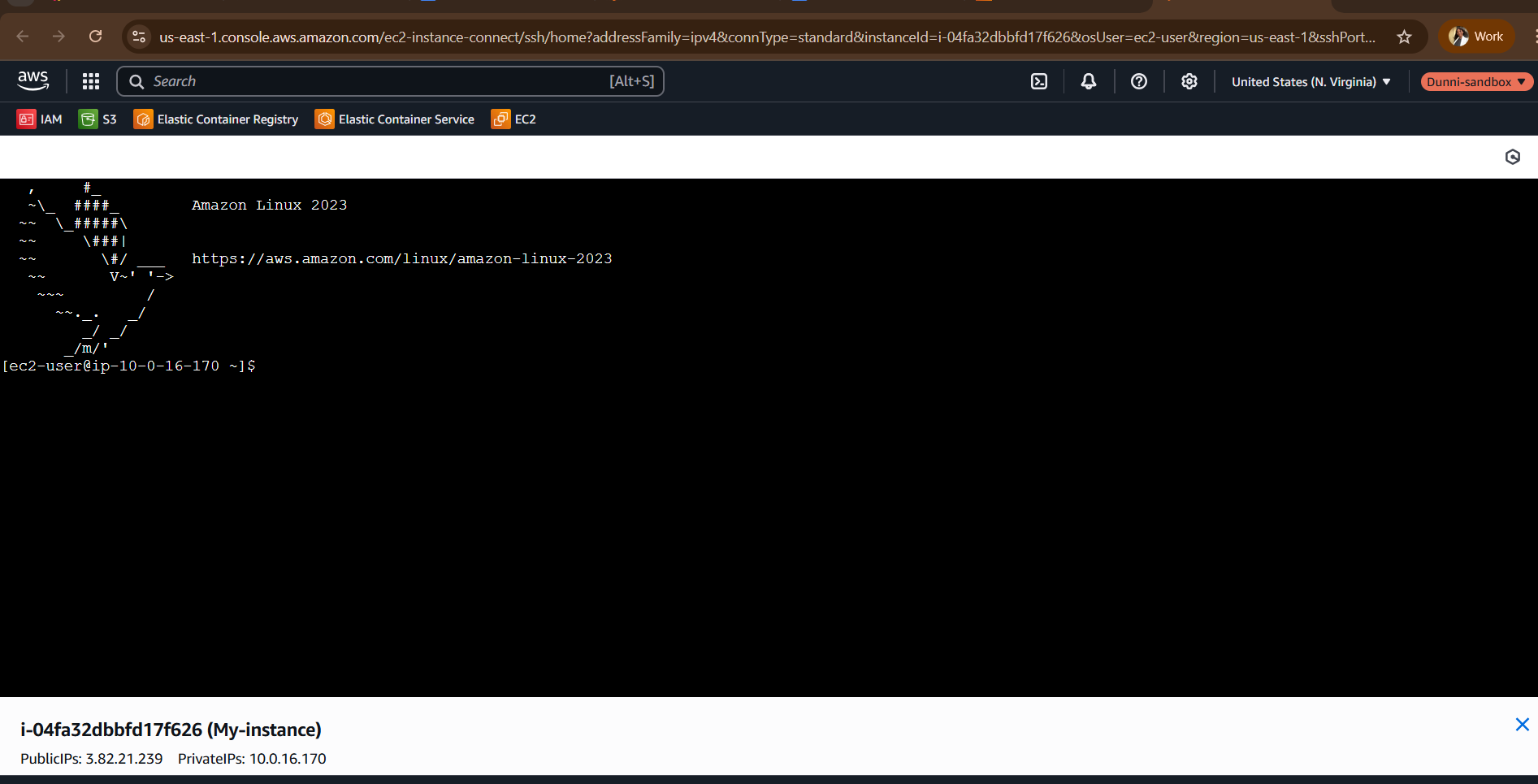
systemctl enable httpd

echo "<h1>Welcome to my EC2 Web Server!</h1>" > /var/www/html/index.html

* Click Launch Instance
* Instance successfully launched

## **2. Connectivity Test**

### **Connect to instance:**

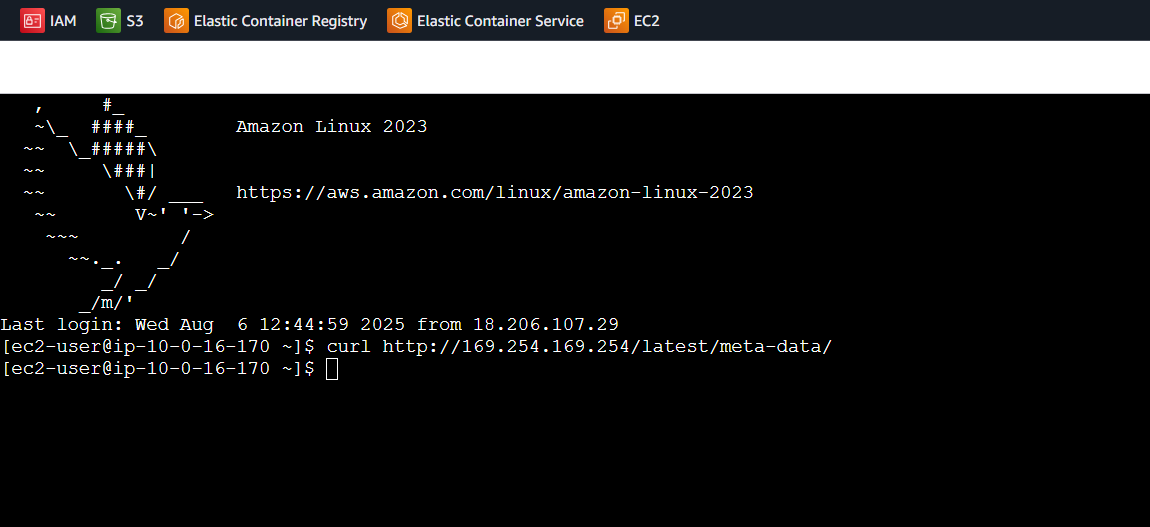
1. Go to EC2 > Instances
2. Select My-instance
3. Click Connect
4. Choose EC2 Instance Connect → Connect****

Now connected to the instance!

### **Test metadata access:**

Run this command:

curl http://169.254.169.254/latest/meta-data/



## **3. Web Server Setup**

Apache was installed and started via the **user-data script**.

**4. Content Validation**

1. In EC2 > Instances, copy the Public IPv4 address (3.82.21.239)

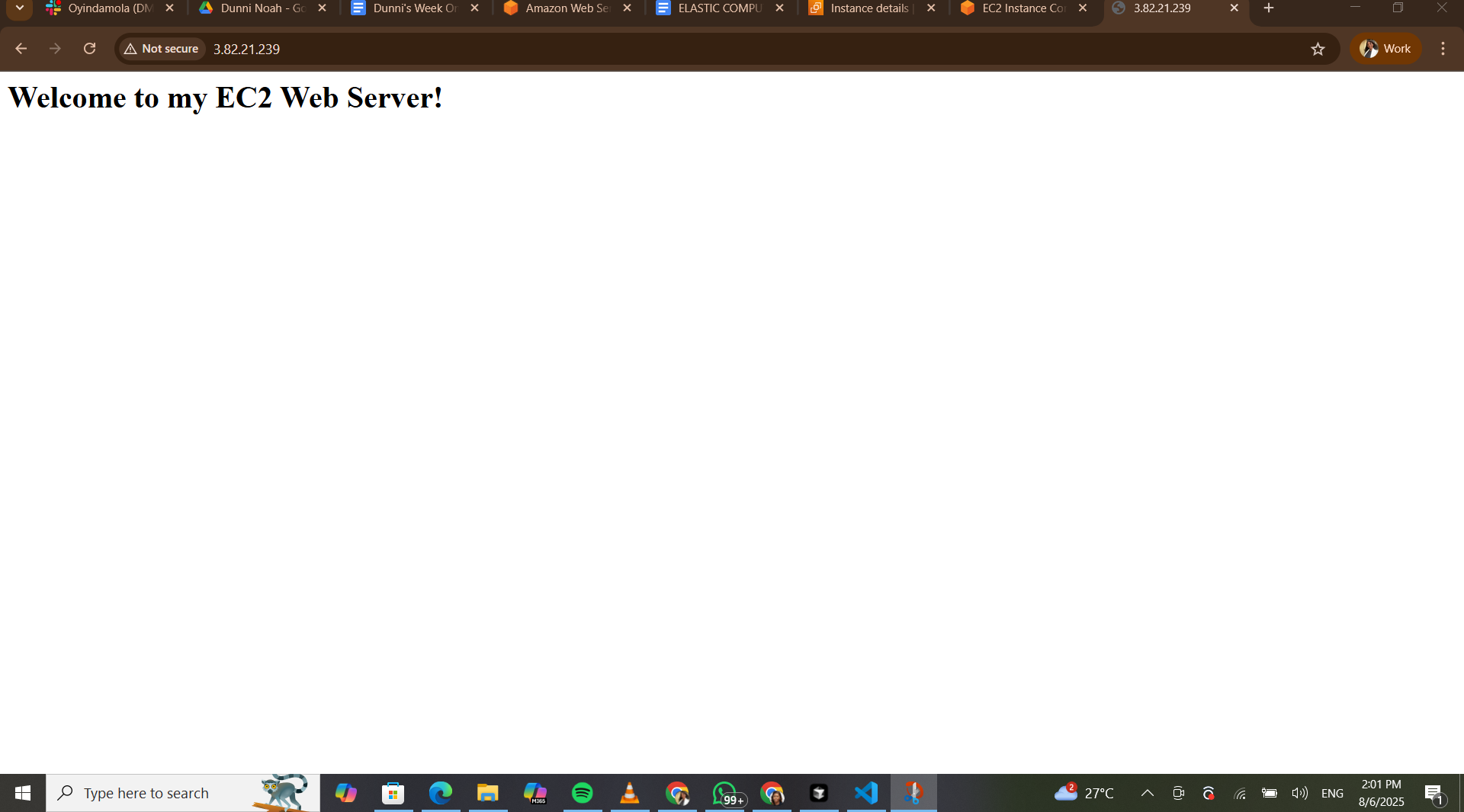
### **Open browser**

Copy and paste this to the browser

http://3.82.21.239

It will bring this:

Welcome to my EC2 Web Server!



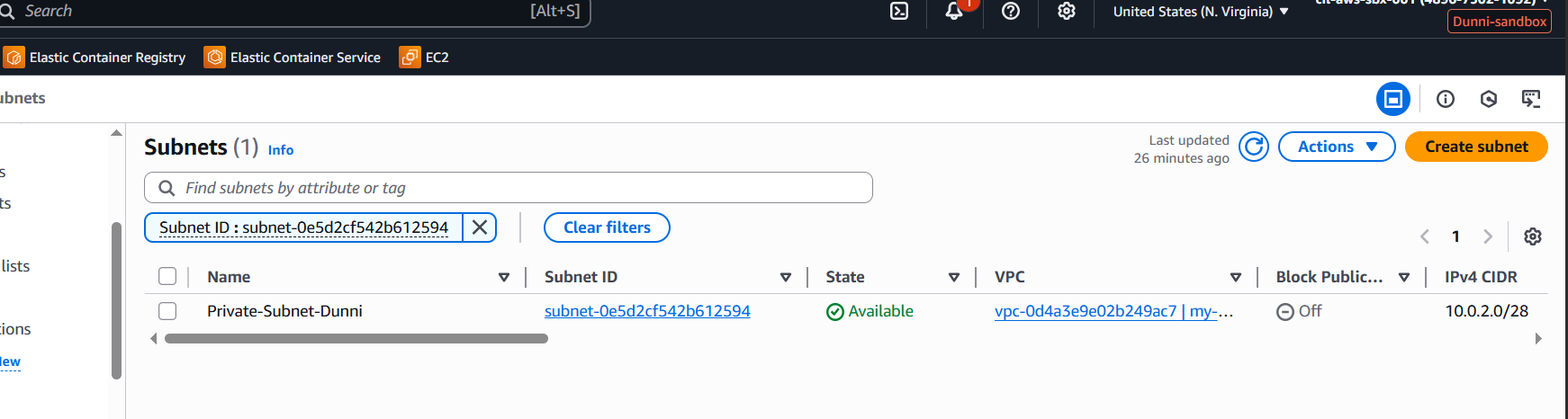
VPC(Network Isolation)

## **Network Isolation**

### **1. Subnet Creation**

**Goal**: Create a private /24 subnet in the default VPC.

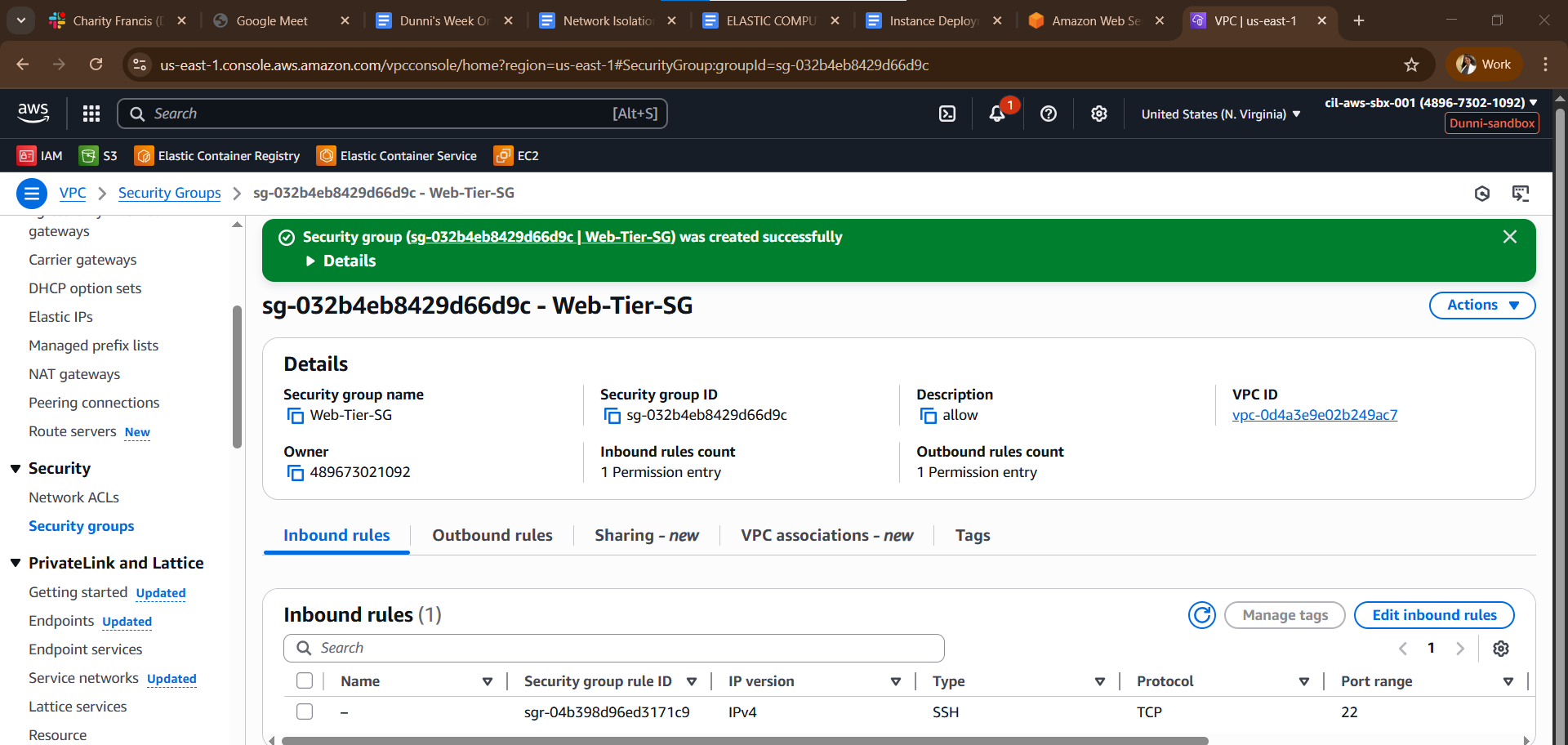
#### **Steps:**

1. Go to VPC Dashboard in the AWS Console.
2. Click Subnets > Create subnet.
3. Choose:
   * VPC: Select your default VPC.
   * Subnet name: Private-Subnet-Dunni
   * Availability Zone: us-east-1a
   * IPv4 CIDR block: Use a /24 block 10.0.2.0/24
4. Click Create subnet.  
   

### **2. Security Hardening**

#### **Goal: Create and configure a security group for web servers.**

#### **Steps:**

1. Go to EC2 Dashboard > Security Groups > Create security group.
2. Name: Web-Tier-SG
3. Description: "Allow HTTP from anywhere, SSH from My IP"
4. VPC: Select the default VPC
5. Inbound rules:
   * HTTP | TCP | Port 80 | Source: 0.0.0.0/0
   * SSH | TCP | Port 22 | Source: My IP 102.88.110.250/32
6. Click Create security group  
   

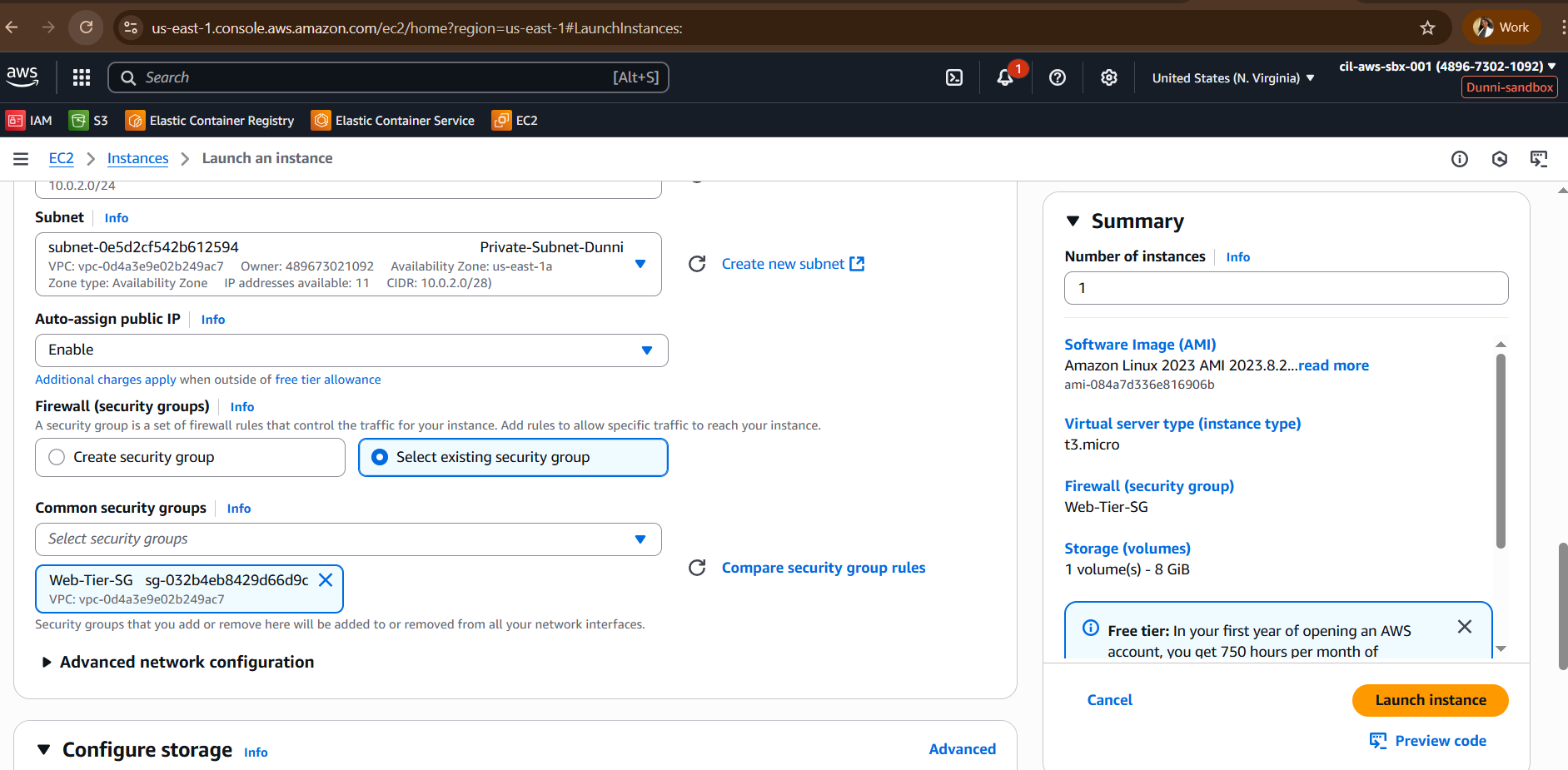
### **3. Resource Association**

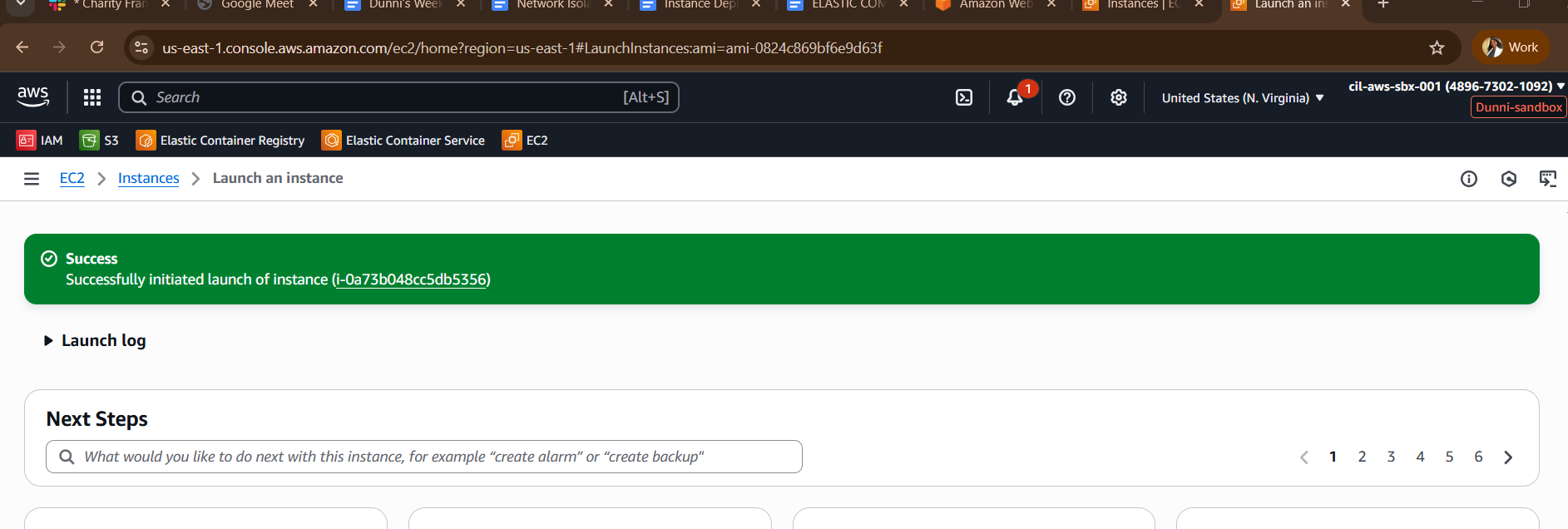
#### **Goal: Move Wednesday's EC2 instance to new subnet and apply new security group.**

#### **Steps:**

#### Note: You cannot directly move a running instance to a new subnet. You'll need to stop the instance, create an AMI, and launch a new one in the new subnet.

1. Identify Wednesday's EC2 instance -My-instance
2. Stop the instance.
3. Click Actions > Create Image (AMI)
4. After AMI is ready:
   * Go to AMIs, select the AMI
   * Click Launch instance
   * Choose:
     + Subnet: MY-DEFAULT-VPC-SUBNET
     + Security Group: Web-Tier-SG



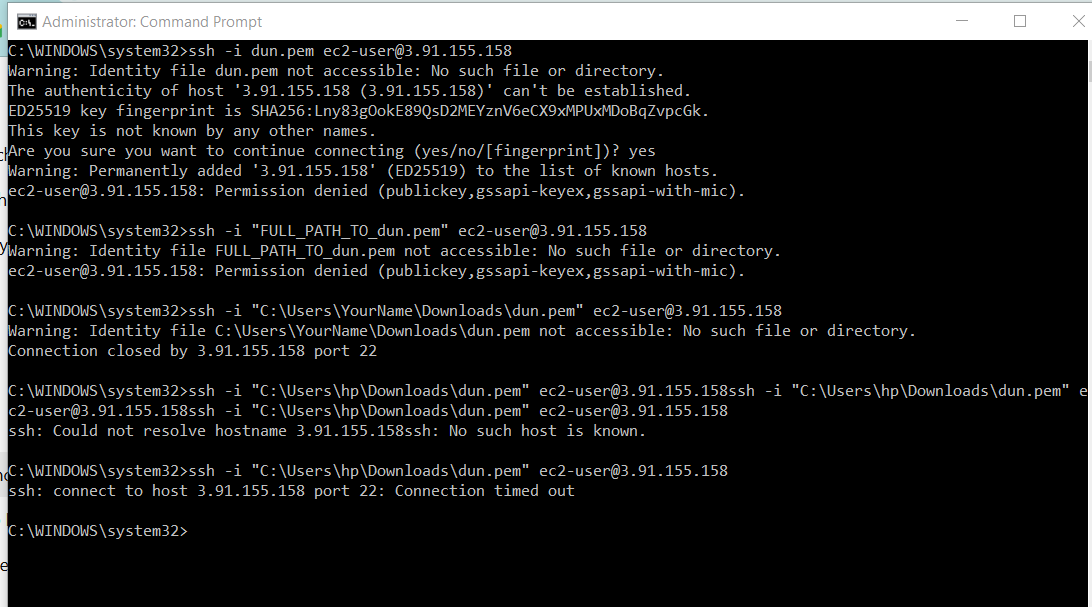
5. Launch the instance.

### **4. Connectivity Verification**

#### **a) Check HTTP (Apache) access:**

1. Get the Public IP of the new EC2 instance.
2. Open browser and go to: http://3.91.155.158/
3. the Apache test page will appear

#### **b) Test SSH restriction:**

1. Try SSH from an unauthorized IP (MyIP).
2. Firstly, i had to change my inbound rules to shh into My-IP-102.88.110.250/32
3. This means:
   * Only My IP can SSH in.
   * No one else can connect via SSH including myself
4. If You See: Permission denied (publickey)
   * That means:
   * You reached the server over SSH so the IP restriction is working
5. You should not get access.

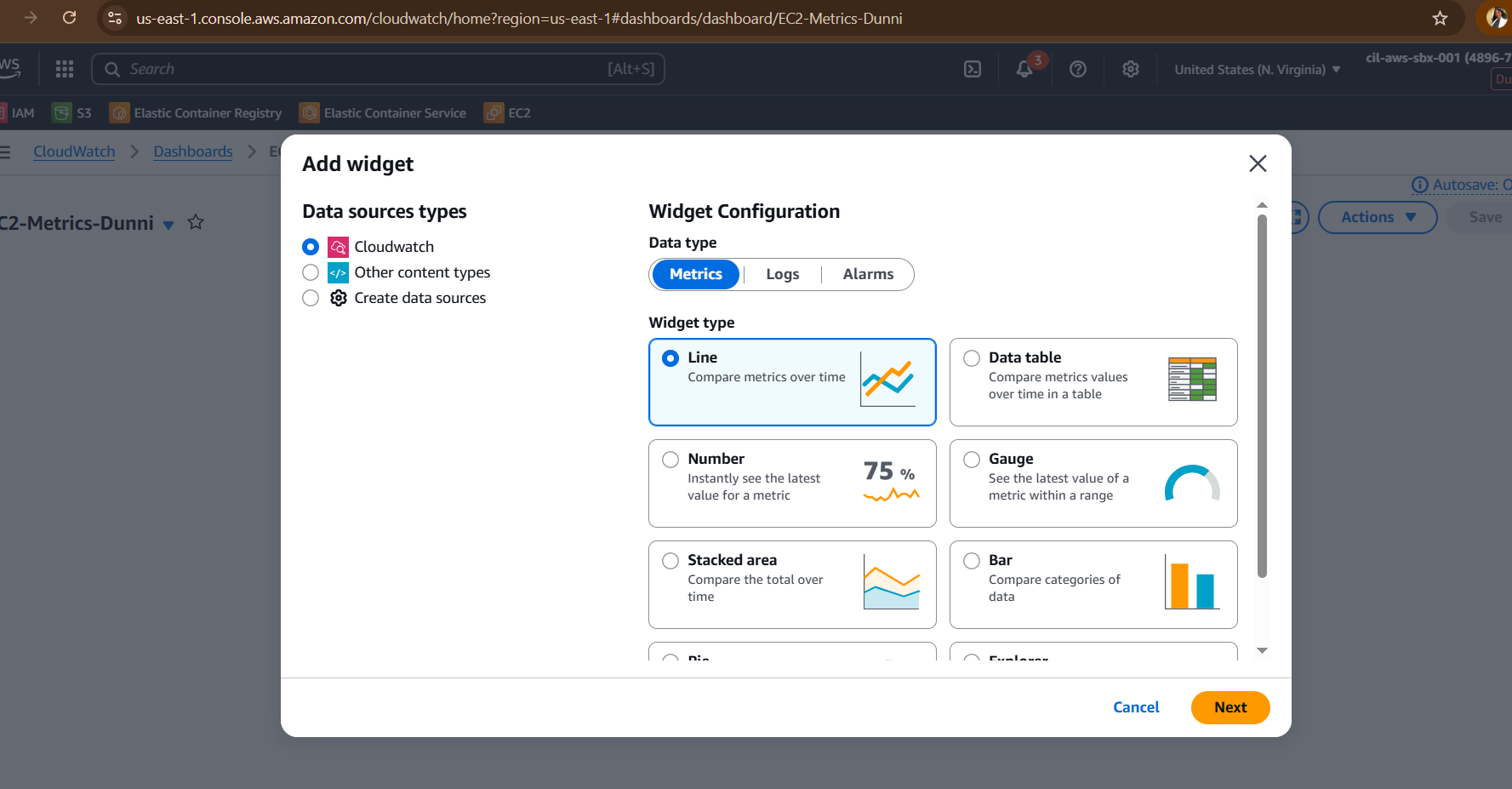
Monitoring & Automation Ob

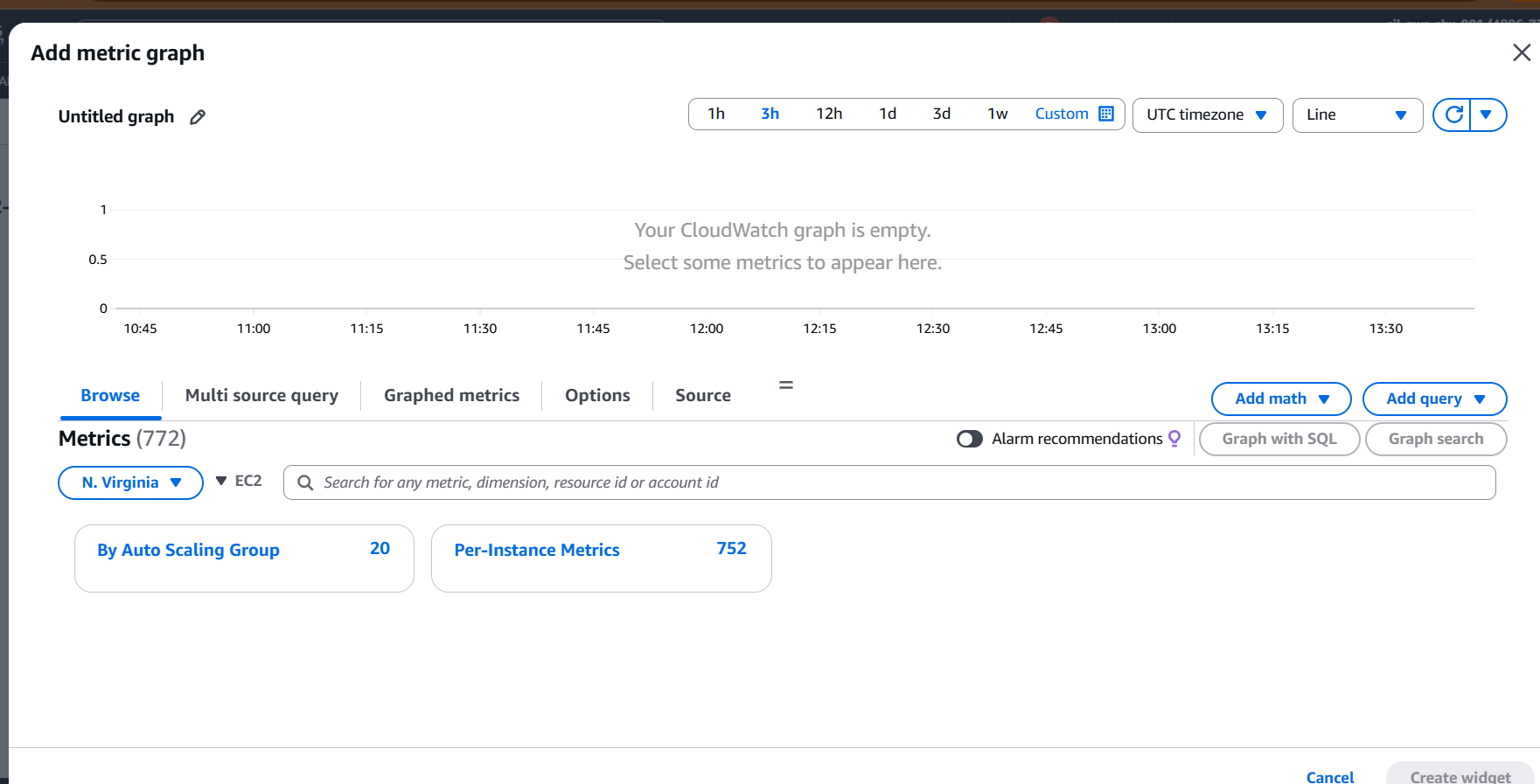
OBSERVABILITY BASICS RUNBOOK

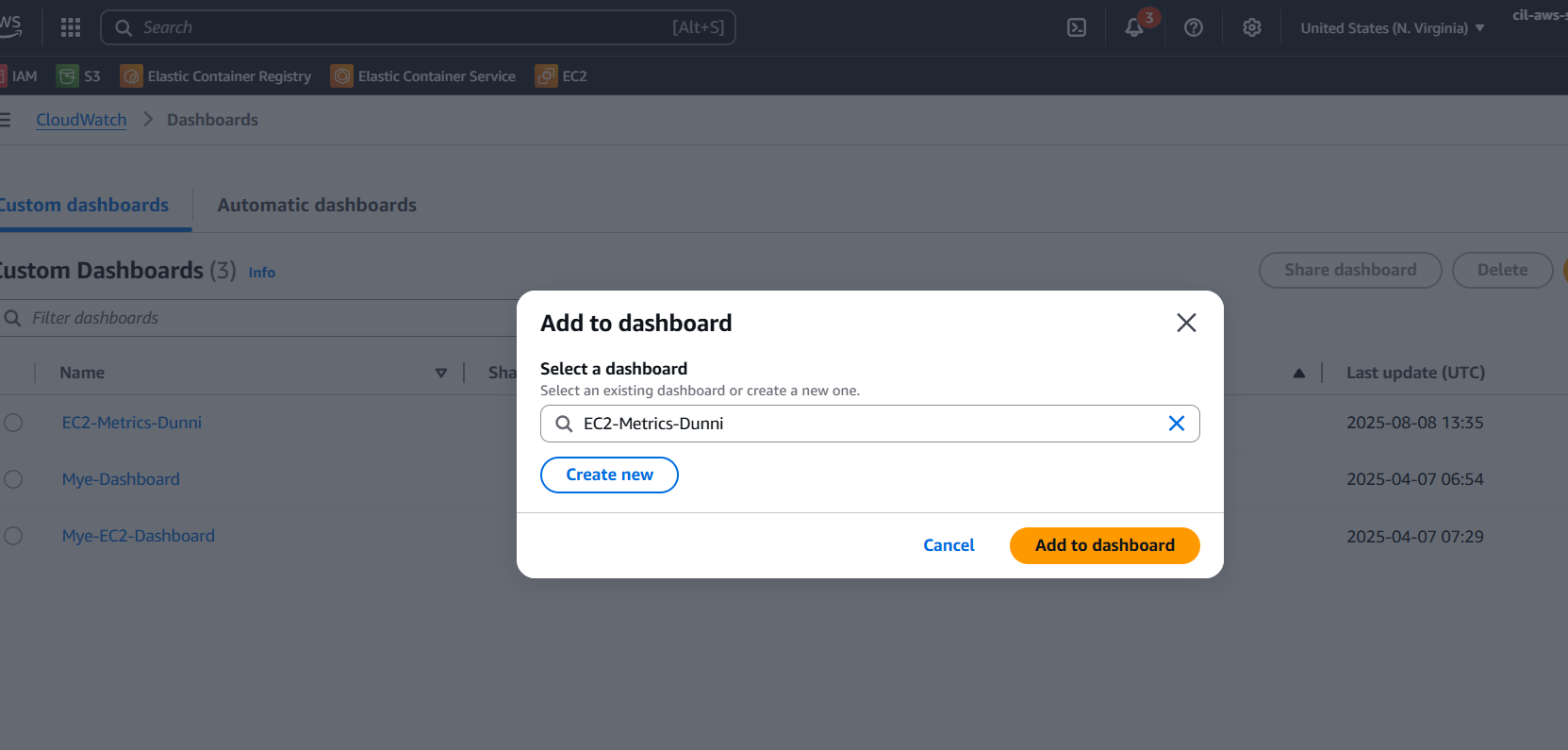
### **1. Create a CloudWatch Dashboard**

**Goal**: Create a dashboard to monitor EC2 metrics

Steps:

1. Go to CloudWatch Console → Dashboards
2. Click Create dashboard
3. Name it: EC2-Metrics-Dunni
4. Choose Line widget → Next
5. Select:

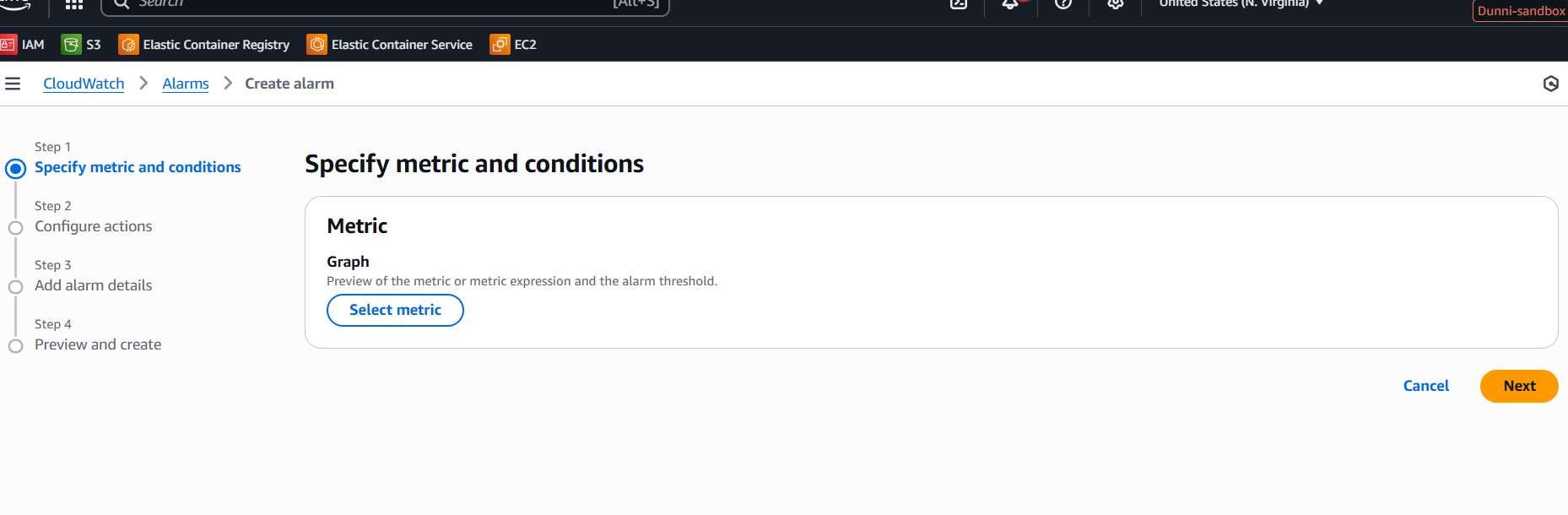
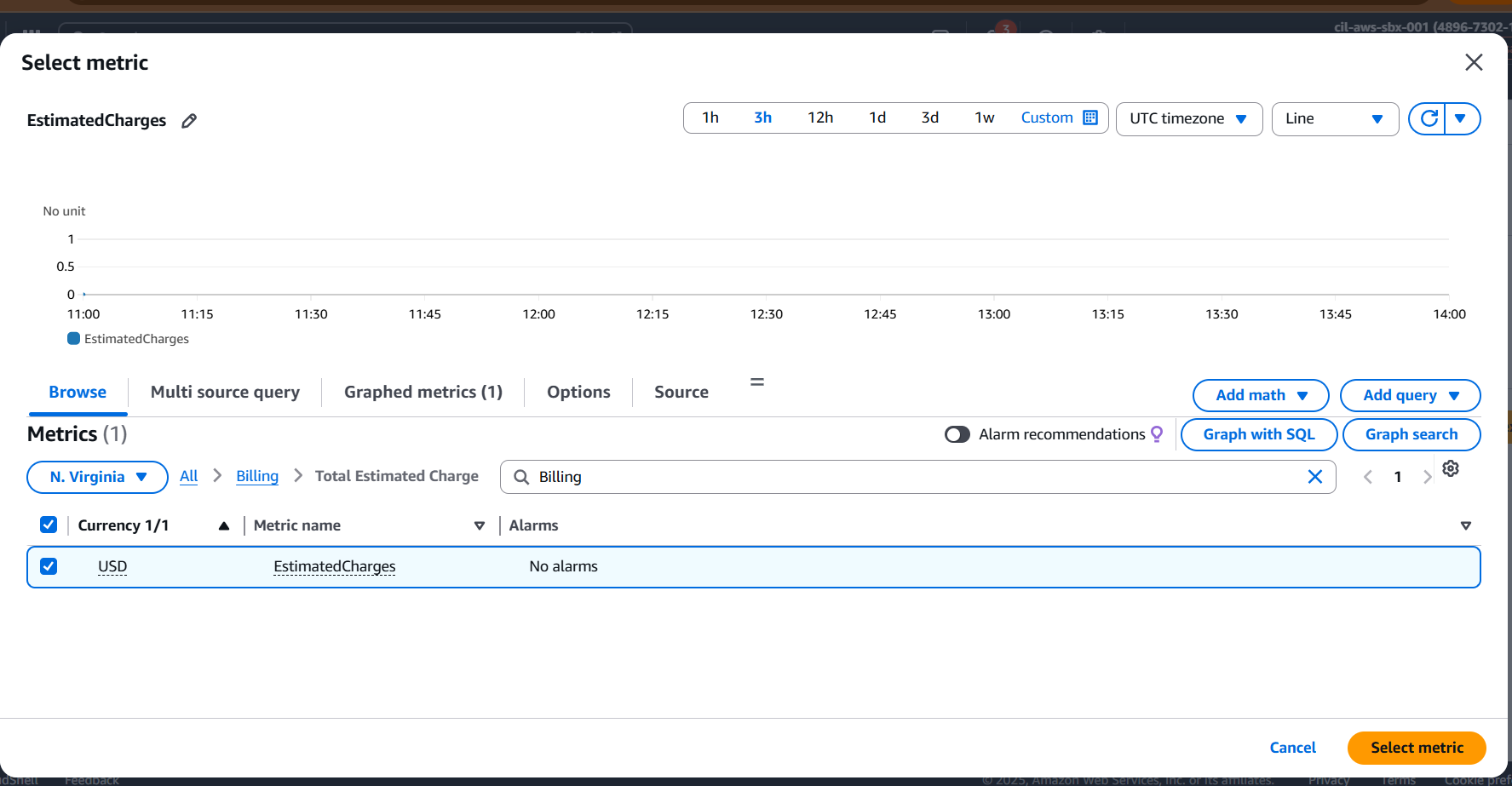
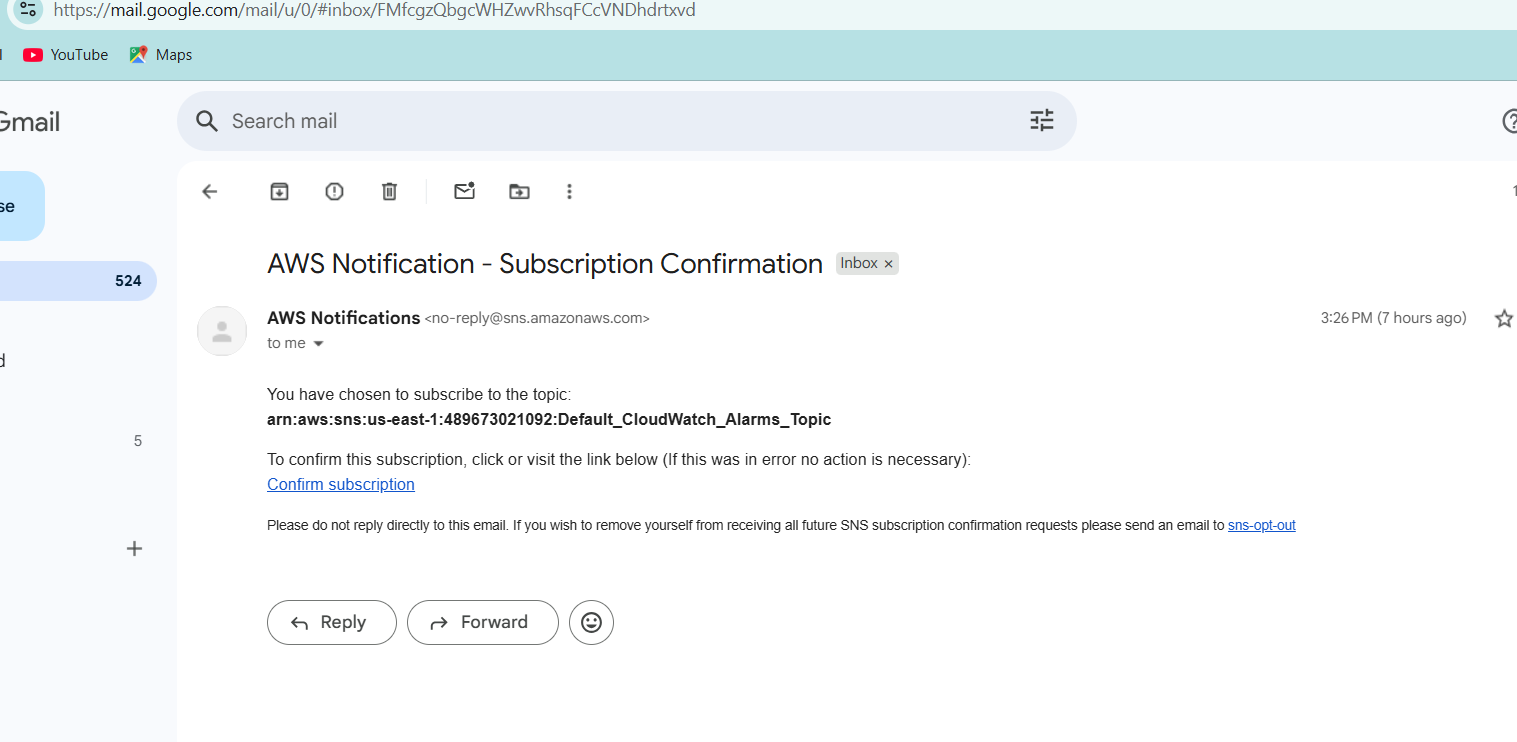
* Browse → EC2 → Per-Instance Metrics  
  + Go to MY-instance–click on instance and —- check CPU-Utilization

1. Click Add to dashboard

### **2. Create Billing Alarm ($10)**

**Goal**: Get email alerts if AWS bill exceeds $10

Steps:

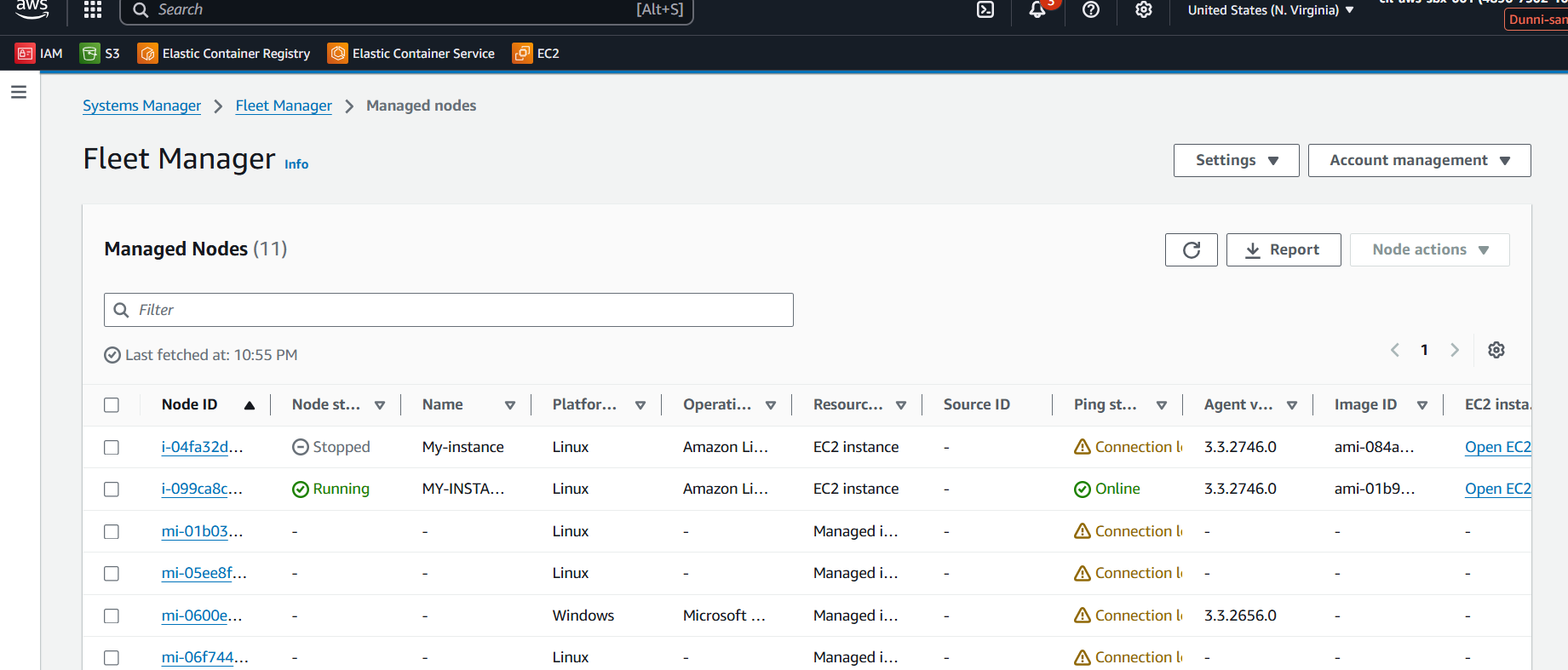
1. Go to CloudWatch → Alarms
2. Click Create alarm
   * Click Select metric → Search:Billing
3. Choose:  
    EstimatedCharges – Currency: USD
4. Click Select metric
5. Set Threshold:
   * Whenever EstimatedCharges > 10
6. Notification:
   * Create SNS topic
   * Enter the email address
   * Confirm the email in the inbox

### **3. Verify Systems Manager Agent (SSM)**

**Goal**: Make sure EC2 can be managed remotely

**Steps**:

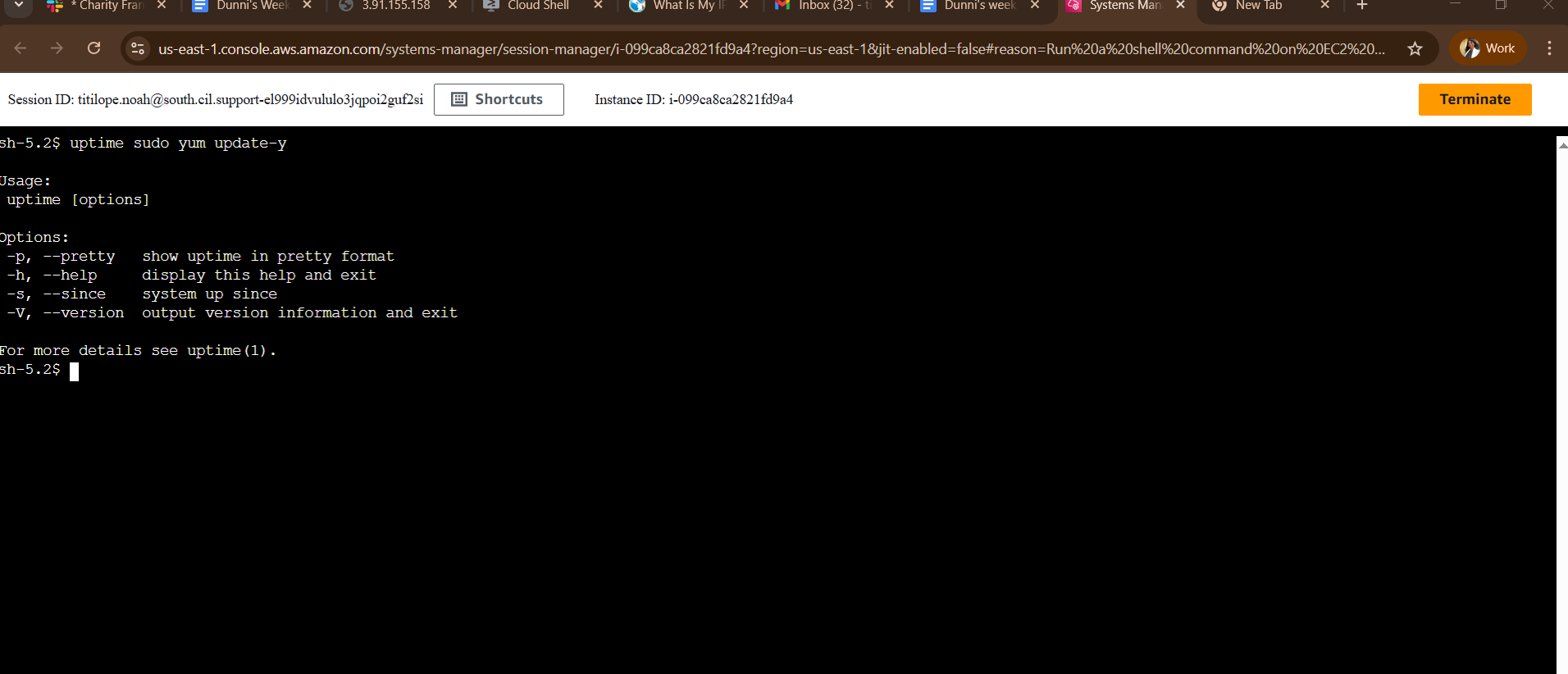
1. Go to EC2 → Instances
2. Select your instance
3. Scroll down to Agent Status section
   * Check fleet manager,
4. look for:
   * SSM Agent running

Instance managed by Systems Manager****

**4. Run Command via Session Manager**

**Goal**: Run a shell command on EC2 without SSH

**Steps**:

1. Go to Systems Manager — Session Manager
2. Click Start session
3. Choose your instance — Start

In terminal, run:

uptime

sudo yum update -y



### **5. Install and Use AWS CLI v2**

Goal: Run AWS commands from local machine

#### **A. Install AWS CLI v2**

#### Verify install, by using this command

aws --version



#### **B. Configure Named Profile**

Run:

aws configure --profile myprofile

Enter:

* AWS Access Key ID—{AKIAXEAWJK2SN6GBE65K}
* AWS Secret Access Key–{IW24JYrV3hqiGsBjJWbJSVDbES/srVPjjDsySzcv}
* Region–us-east-1
* Output format—-json  
  

#### **C. List S3 Buckets**

Run:

aws s3 ls --profile myprofile



