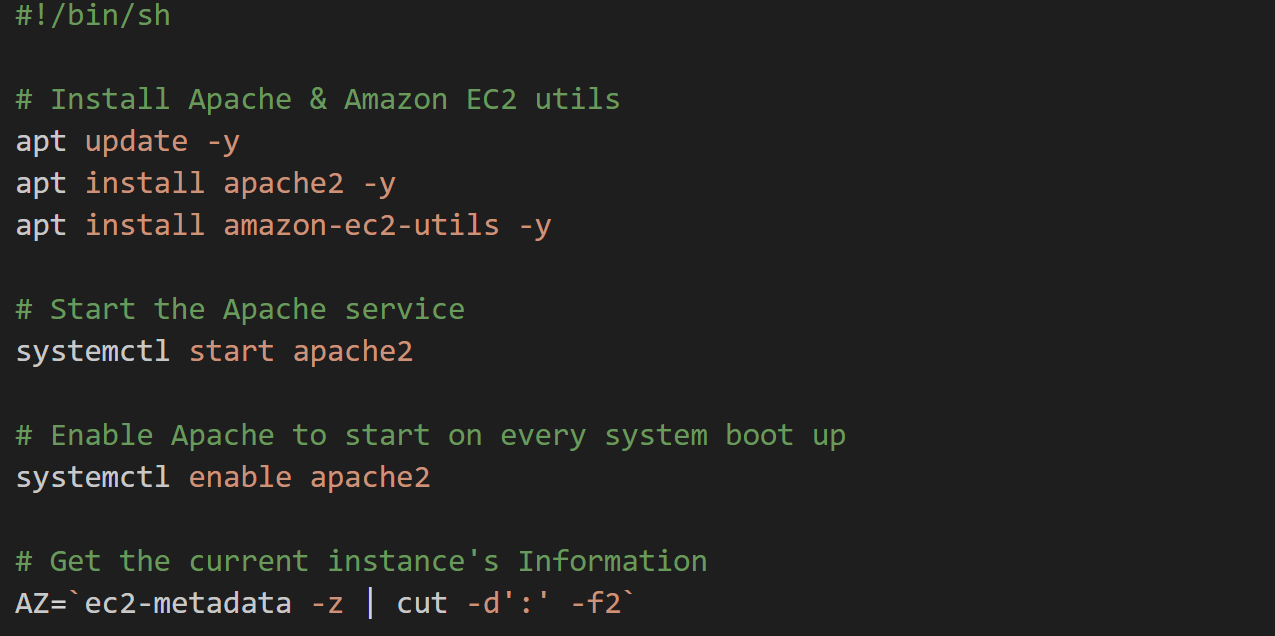
# HW Assignment 4

**Chong Chen NUID: 002833267**

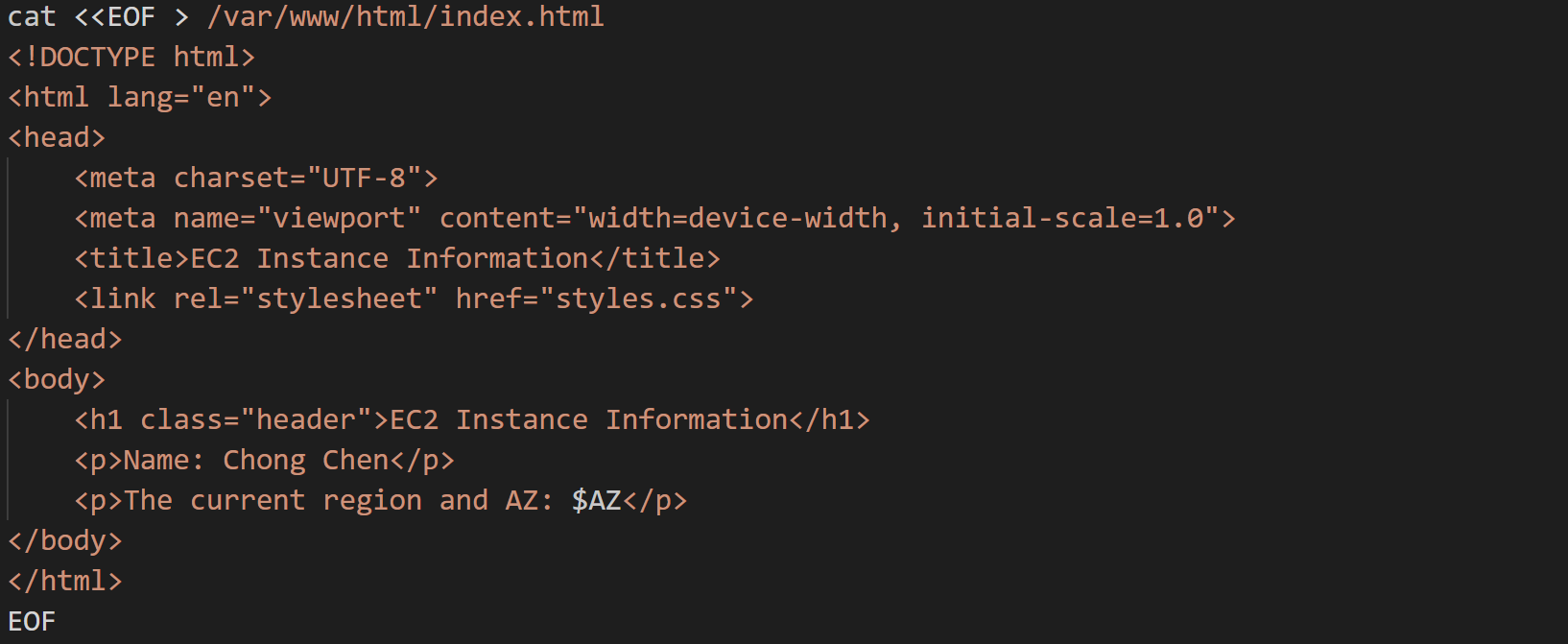
### Step 1 (15 points)

**Write the shell script for the user data that will be executed every time when an EC2 instance is created.**

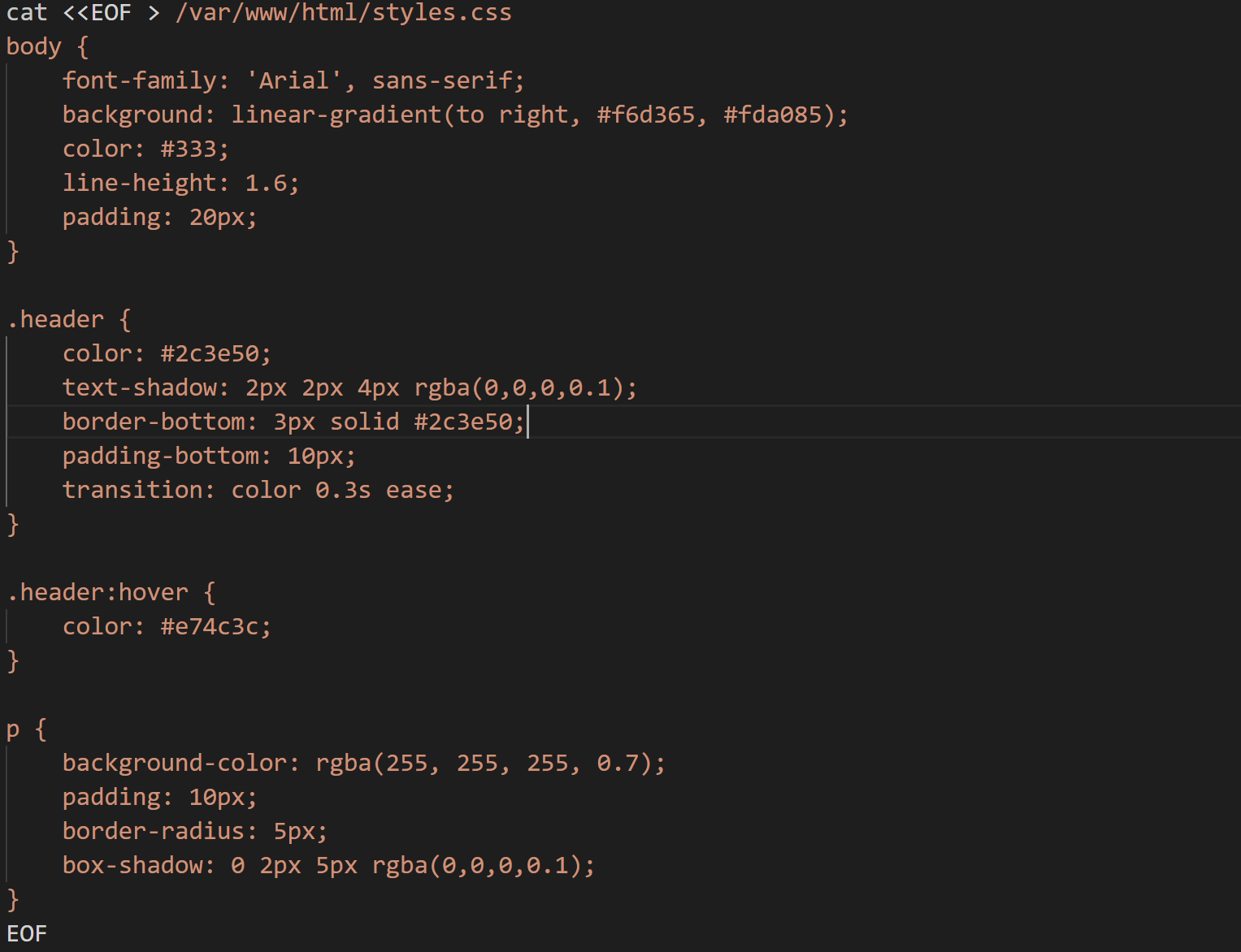
**Install Apache:**



**Create the HTML content:**



**Create a basic CSS file:**



**The detailed script could be seen on user-data-initial-ec2.**

### Step 2 (10 points)

**Create an EC2 instance for a web app server. Make sure that the Security Group allows incoming HTTP traffic. Make sure the user-data-initial-ec2 is used for your User Data.**

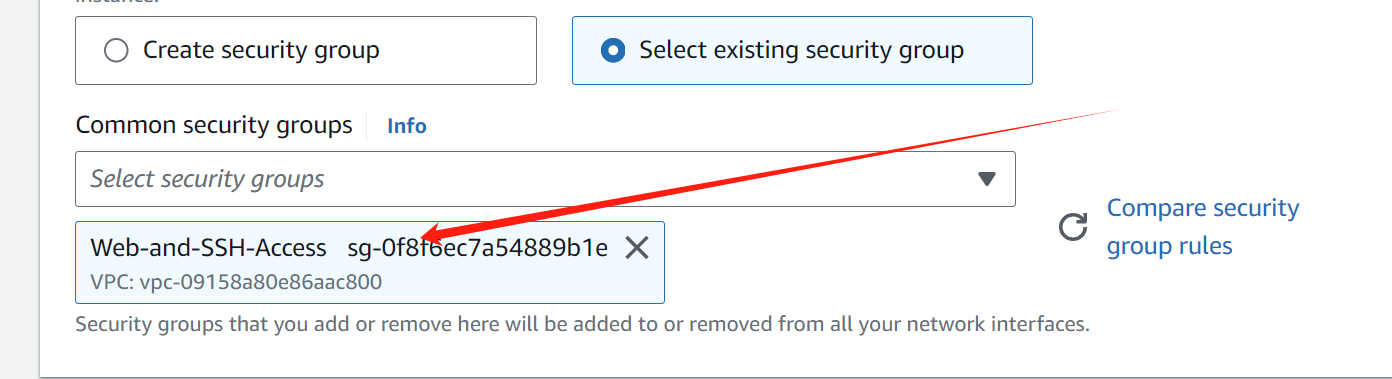
**Set ec2 instance name:**



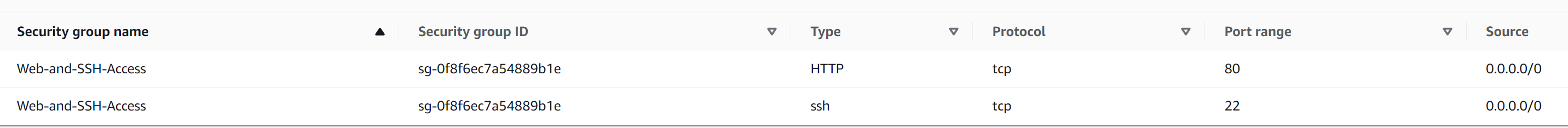
**Set key pair:**



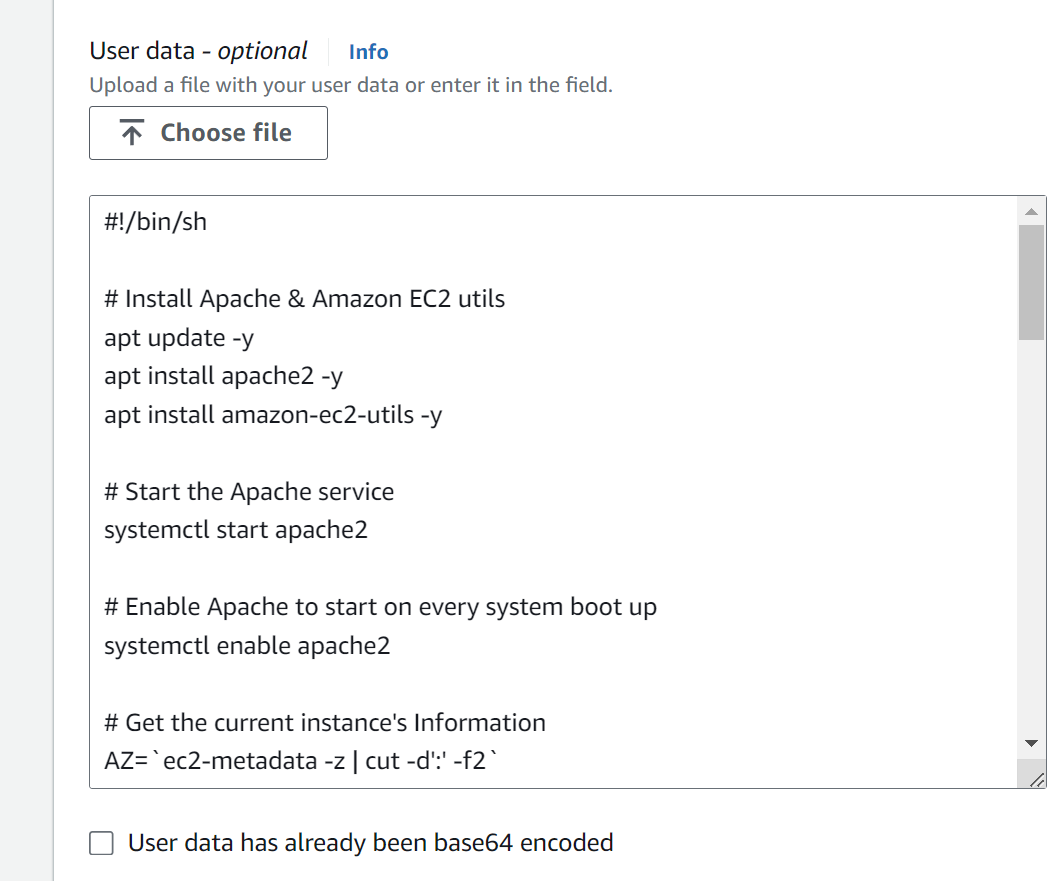
**Configure Security Group:**



**Add a rule to allow HTTP traffic and SSH access:**

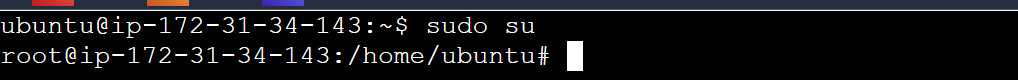


**User data: Paste the contents of the user-data-initial-ec2 script here.**



### Step 3 (15 points)

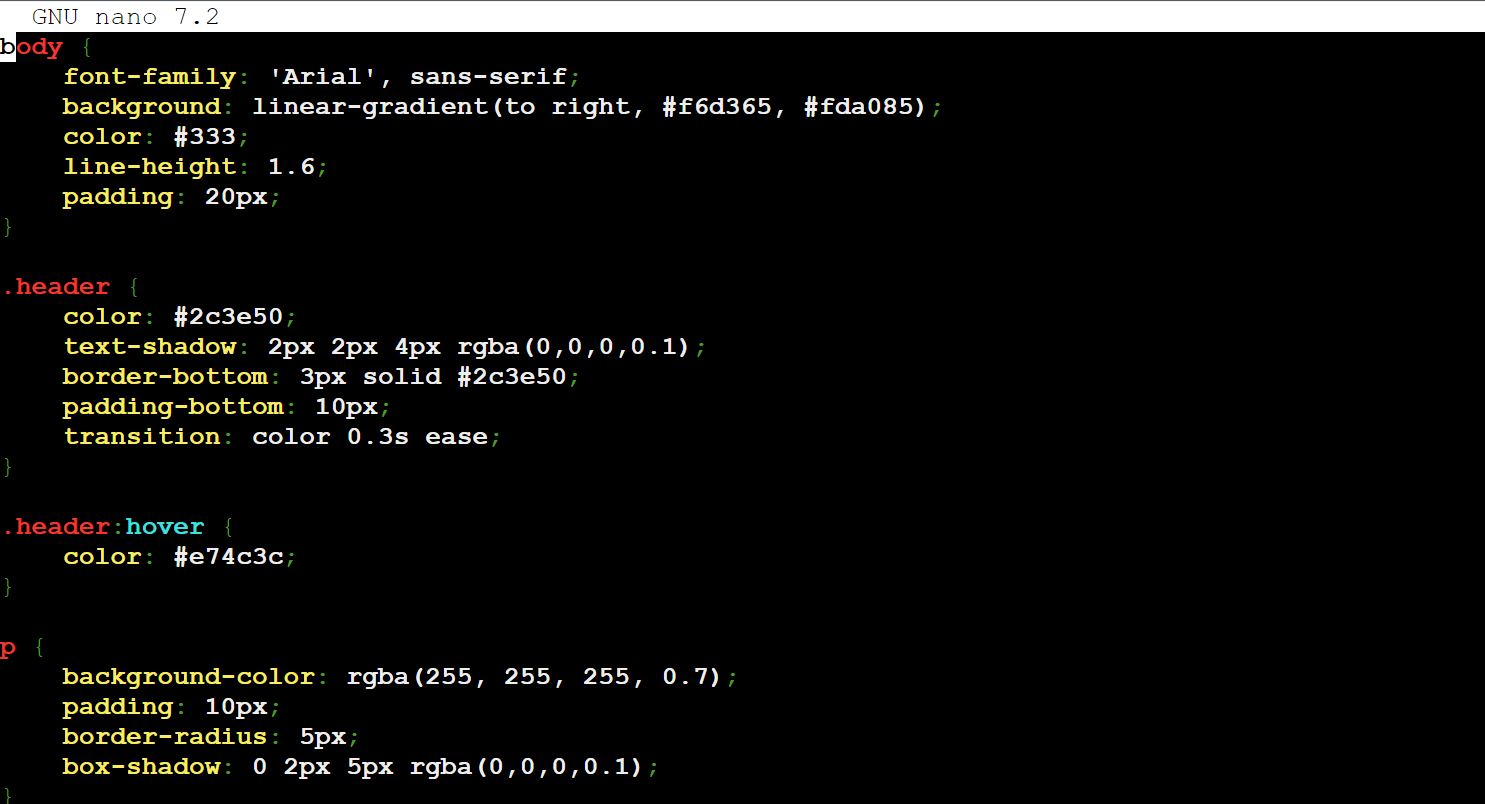
**Login to the EC2 instance. Switch to the root user.**



**Create a CSS file styles.css under /var/www/html/ directory. Add at least one style to the CSS file. Be creative.**

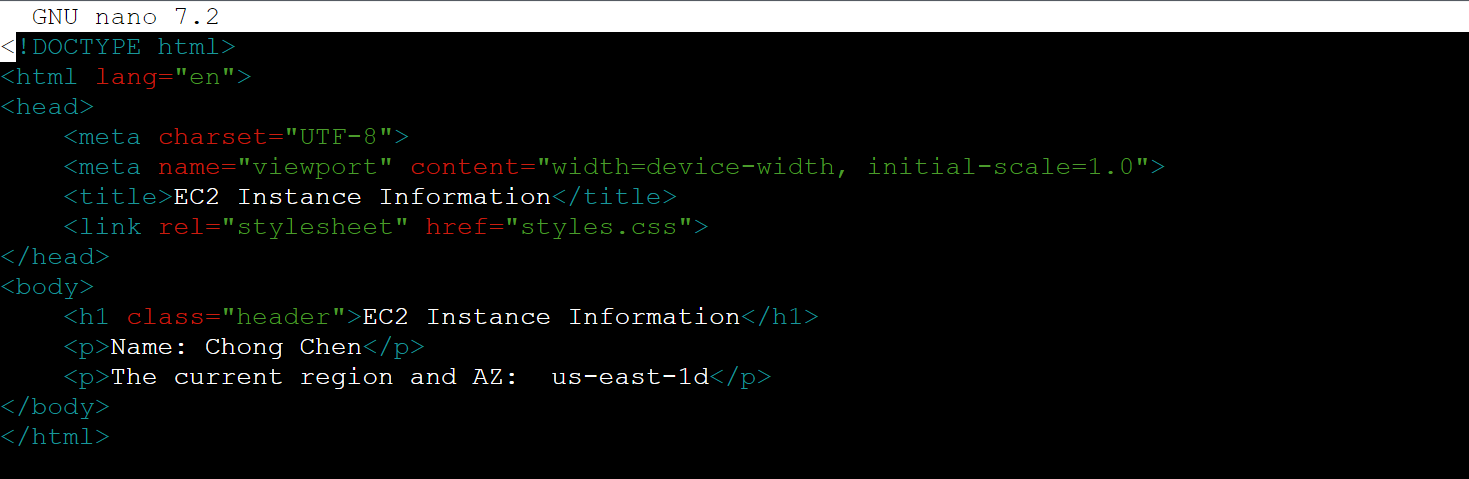


**Create and edit the CSS file:**



**Apply the selector from the CSS to HTML element(s).**



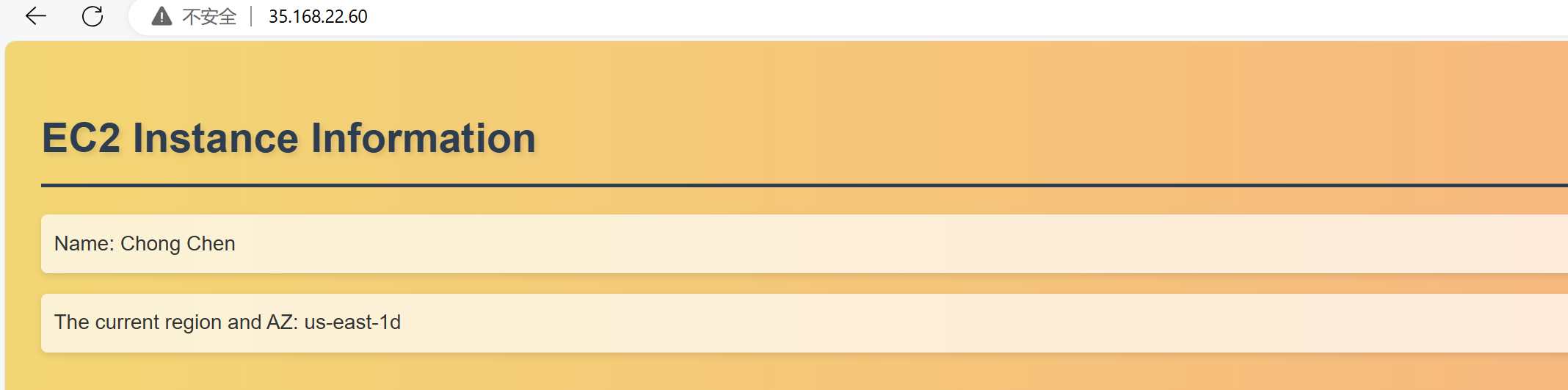


**Restart Apache2 so Apache2 is aware of your code changes: systemctl restart apache2**



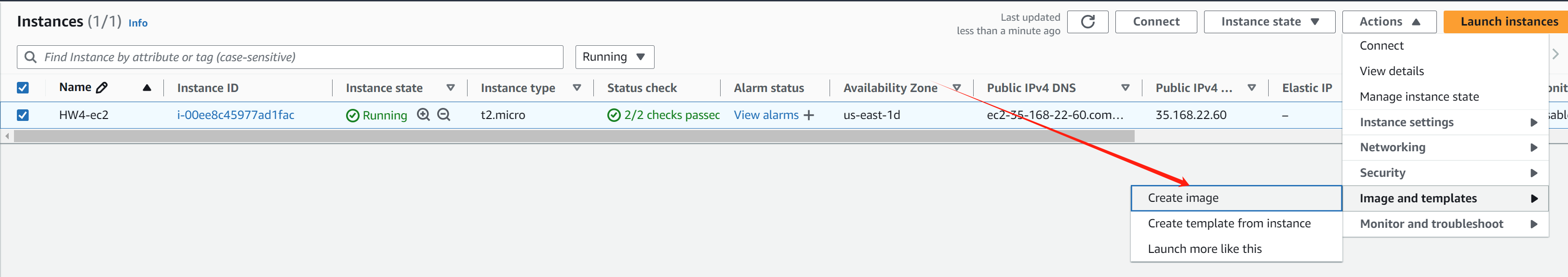
**Use the browser to access your web page through the public IP of your EC2 instance. Verify whether the CSS is applied.**

**Open a web browser and enter the EC2 instance's public IP address. It should show the webpage with the new styles applied.**

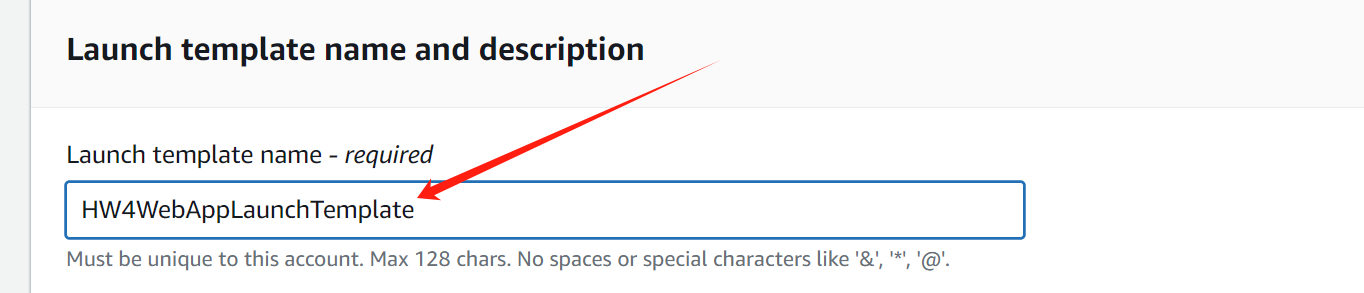


### Step 4 (20 points)

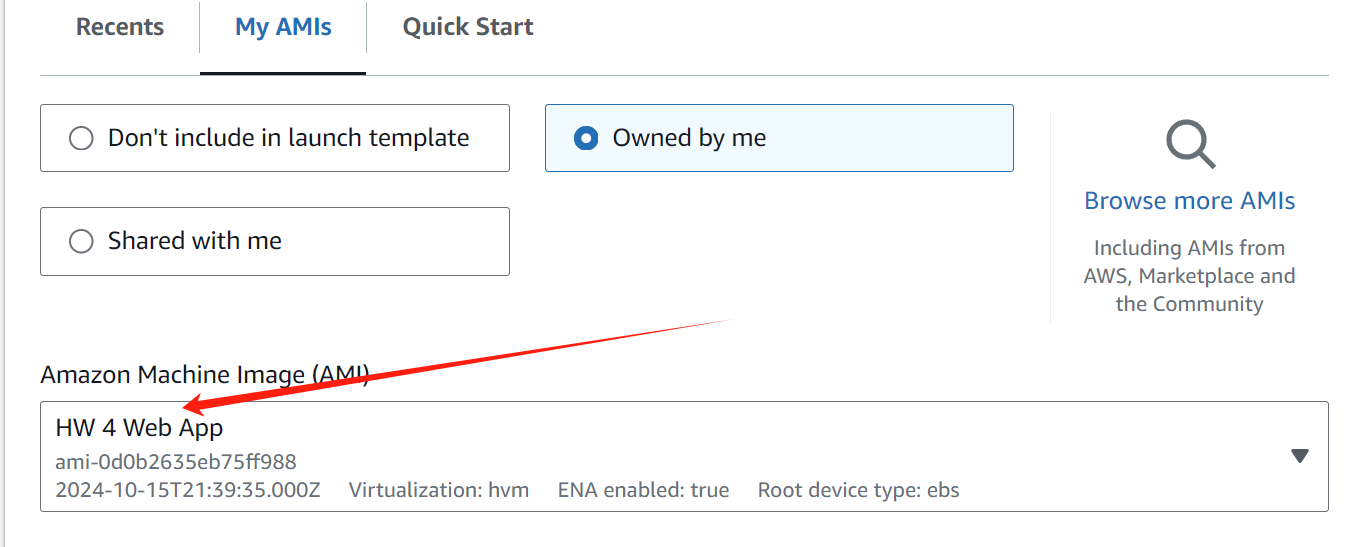
**Create an AMI based on the EC2 you just created. Name your image as "HW 4 Web App".**



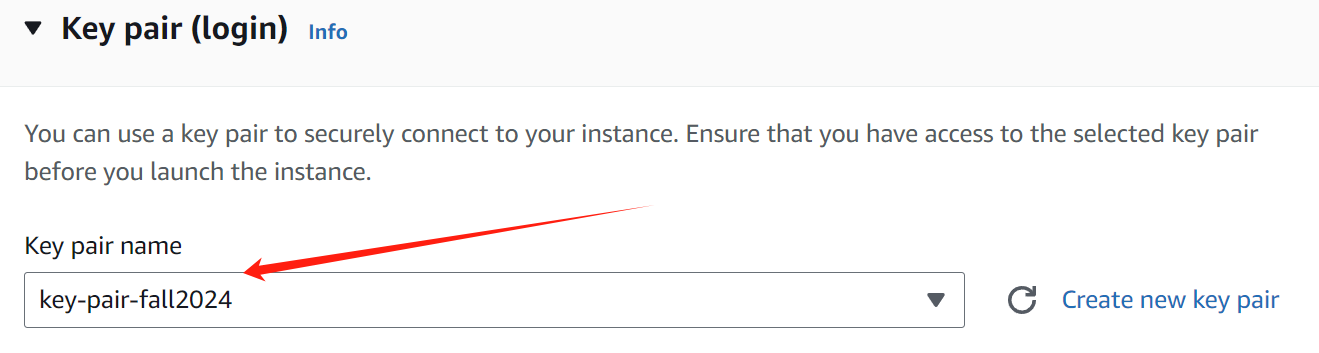
**Create a launch template based on the AMI you created. Name your launch template as "HW4WebAppLaunchTemplate".**



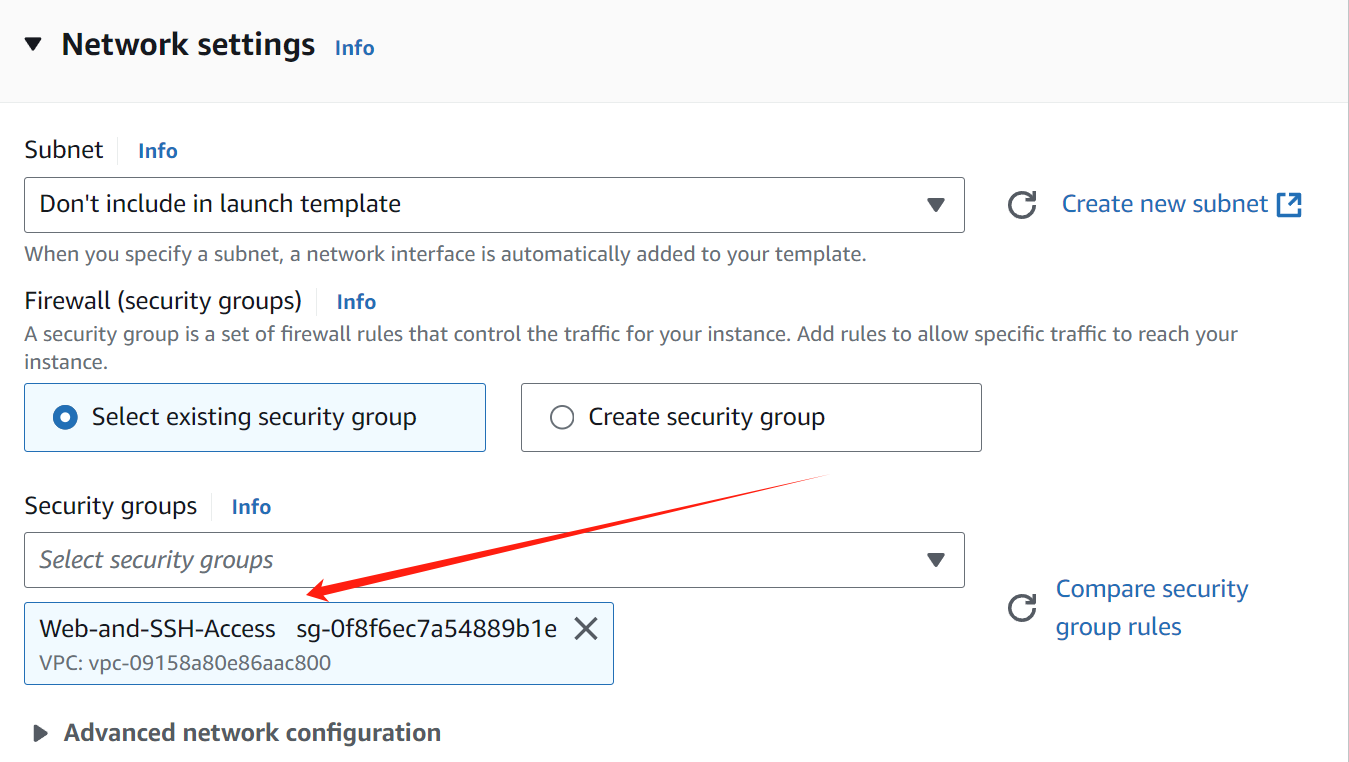
**For the AMI, select the "HW 4 Web App" AMI just created.**



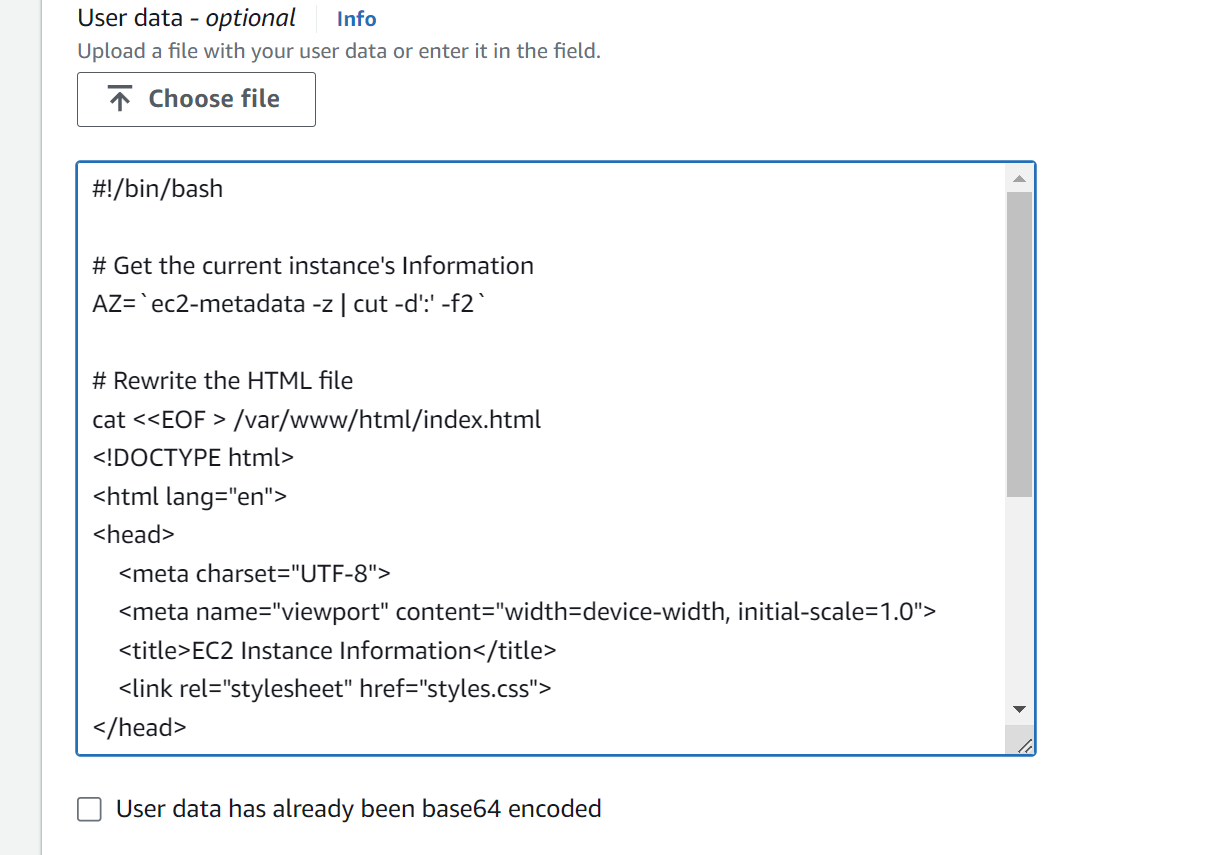
**Select key pair for SSH access.**



**In Network settings, choose a security group that allows HTTP and SSH.**



**Add the user data.**



**Store the user data to another file on your computer.**

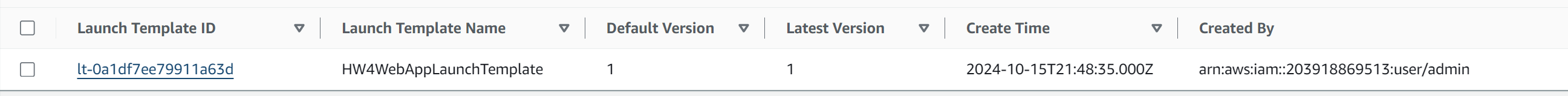
**The full script could be seen on user-data-launch-template.**

**Now we have:**

1. **An AMI named "HW 4 Web App" based on the original EC2 instance.**



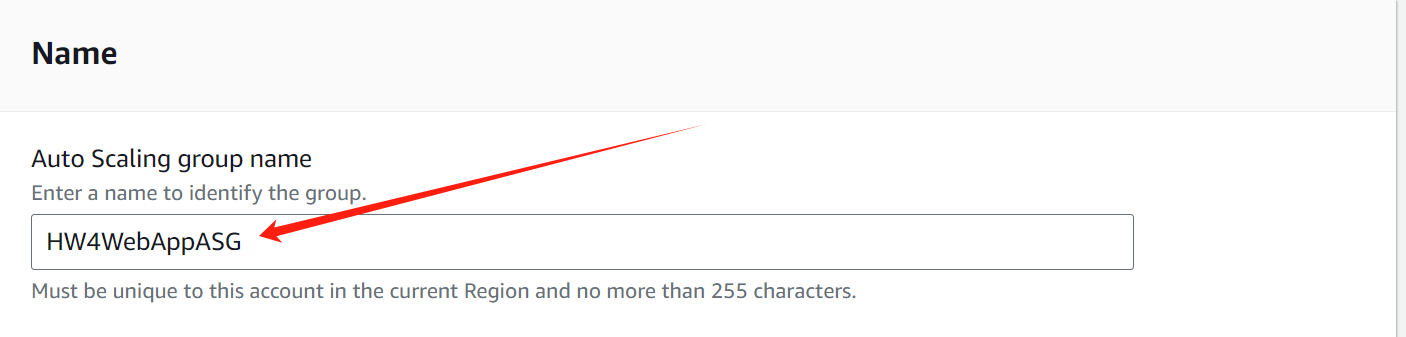
1. **A launch template named "HW4WebAppLaunchTemplate" that uses this AMI.**



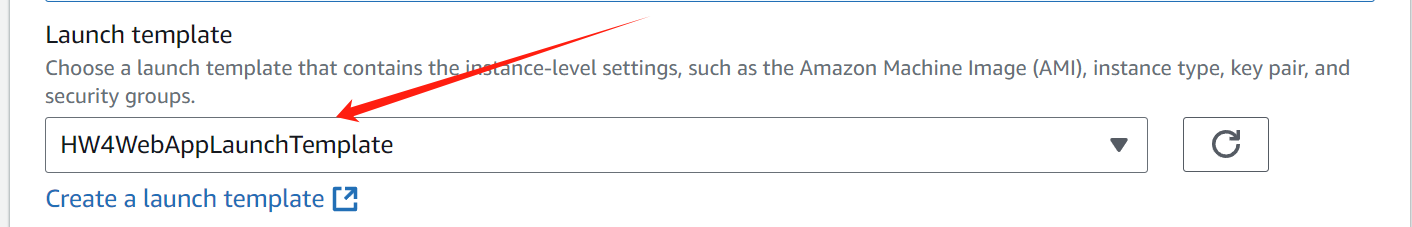
### Step 5 (20 points)

**Create the Auto Scaling Group (ASG) using the "HW4WebAppLaunchTemplate". Name your ASG as "HW4WebAppASG". Select the default VPC with all available subnets. Set your desired capacity, min capacity, and max capacity as 3. No scaling needed.**

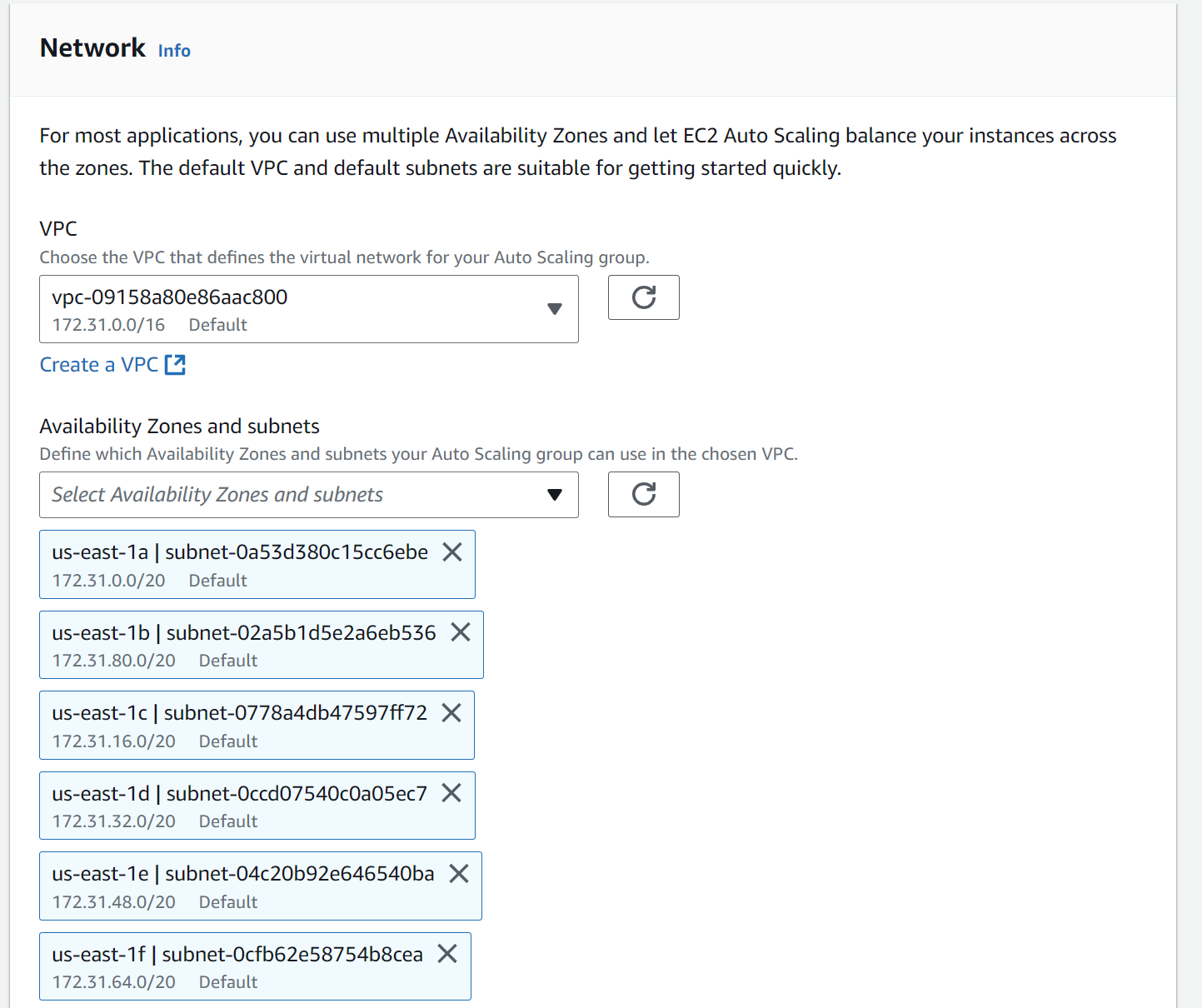
**Create the Auto Scaling Group.**



**Select "HW4WebAppLaunchTemplate" as the launch template.**

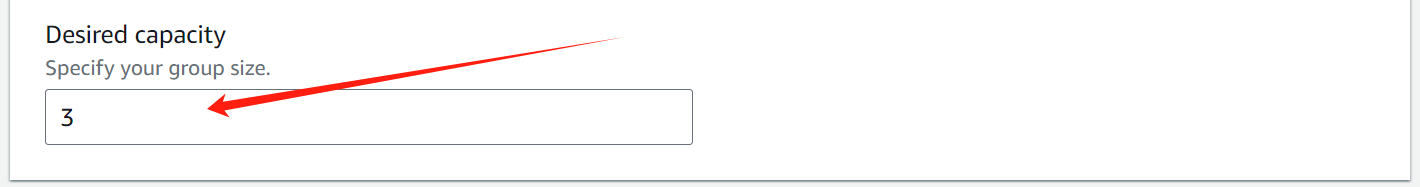


**For Availability Zones and subnets, select all available subnets.**

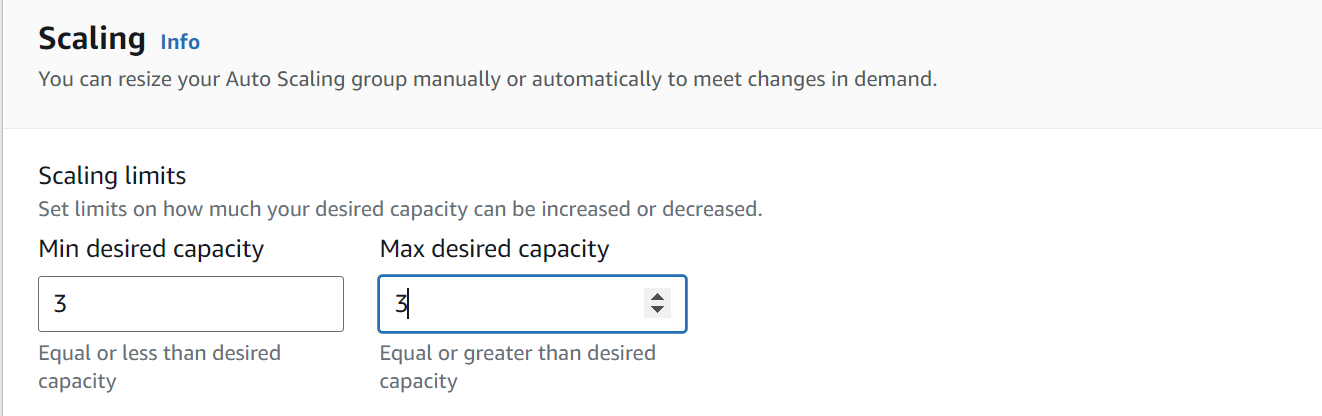


**Configure group size and scaling policies:**

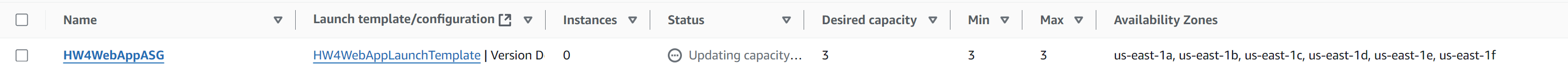
**Set Desired capacity: 3**



**Set Minimum capacity: 3, Set Maximum capacity: 3**

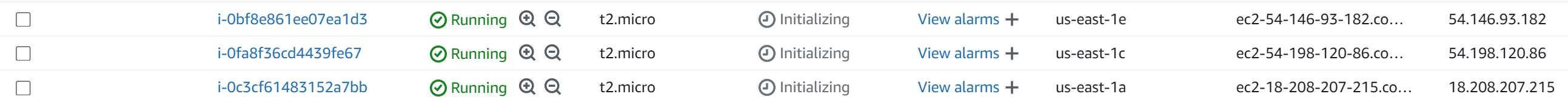


**Review and create. Now, the Auto Scaling Group will start launching three EC2 instances based on the launch template.**



**Test each instance using your browser to access the public IP.**

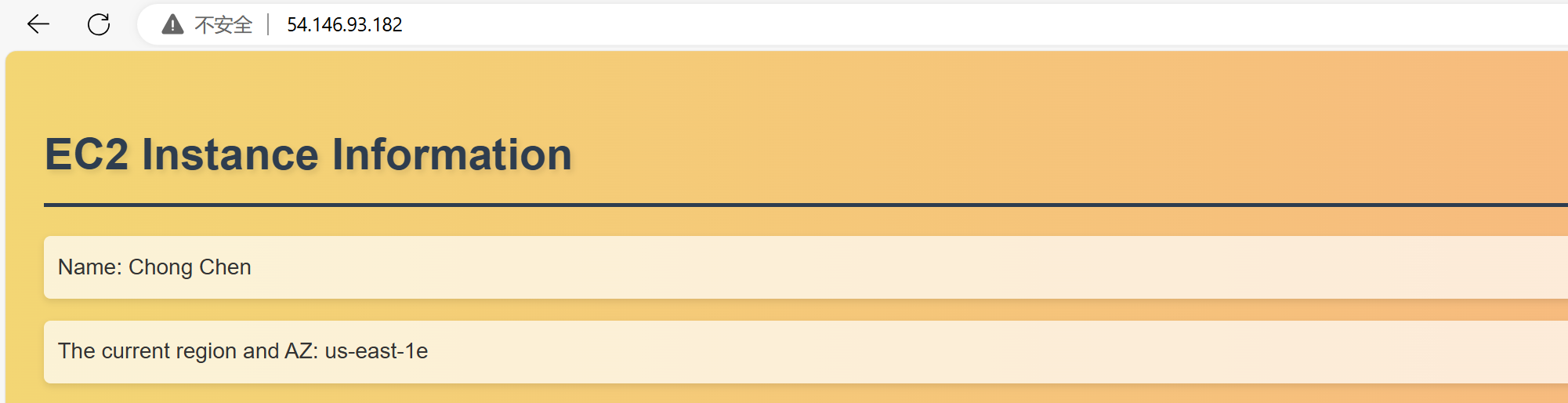
**Go back to the EC2 dashboard and wait for the instances to be in the "Running" state.**

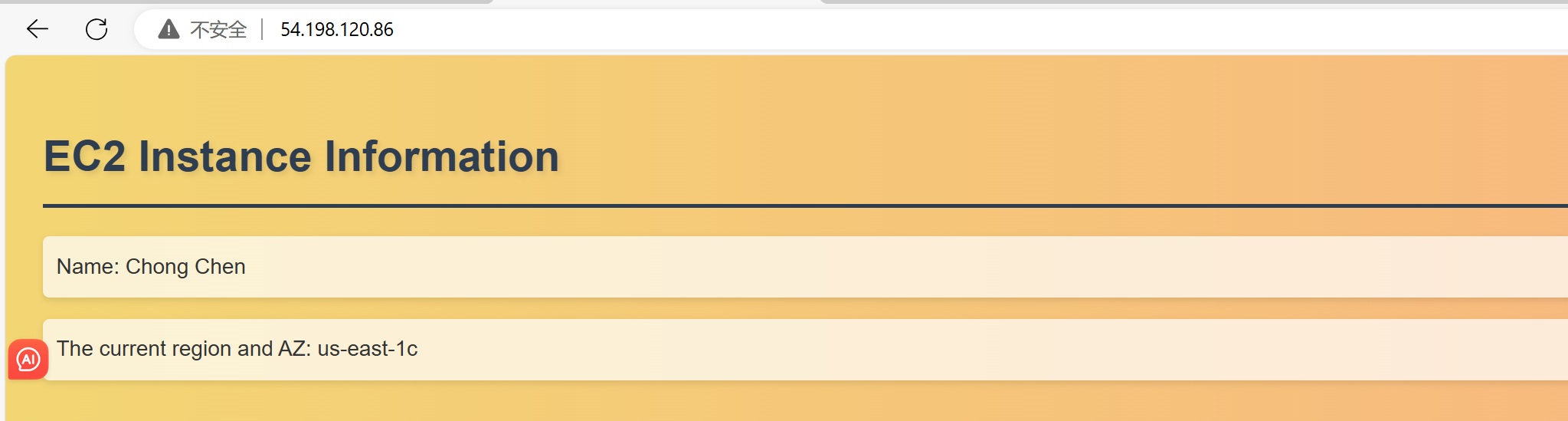


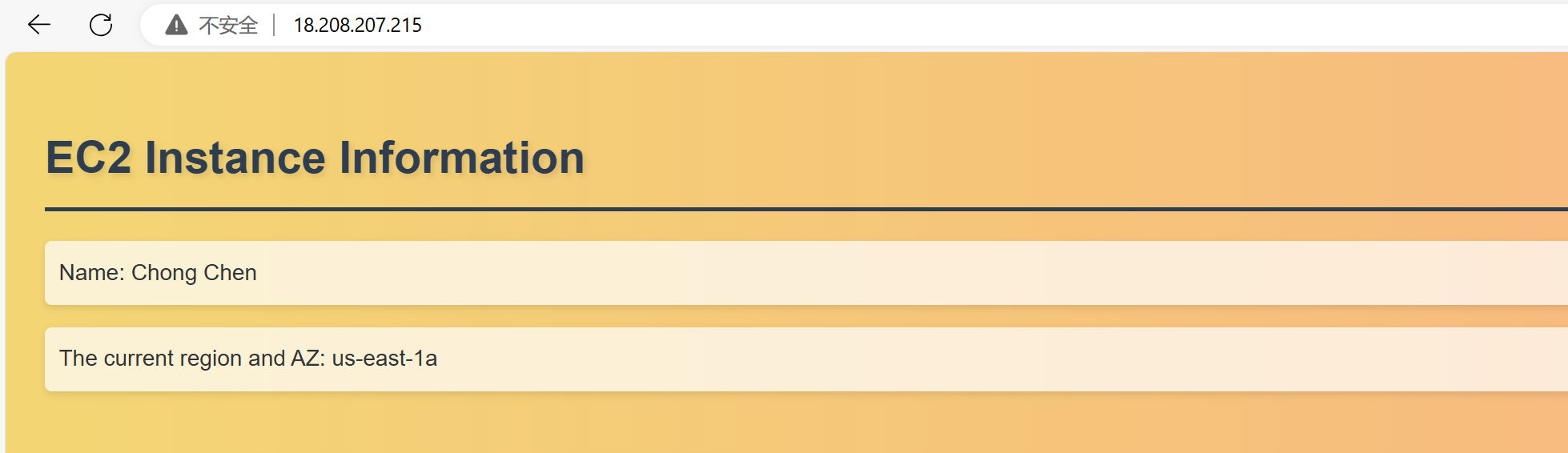
**For each instance, look for the "Public IPv4 address" in the instance details, copy this IP address, open a new tab in the web browser, and paste the IP address into the address bar and press Enter.**

**It should show the webpage for each instance. Verify that:**

* **The page loads correctly.**
* **The CSS styles are applied.**
* **The Region and Availability Zone information is correct and different for each instance.**



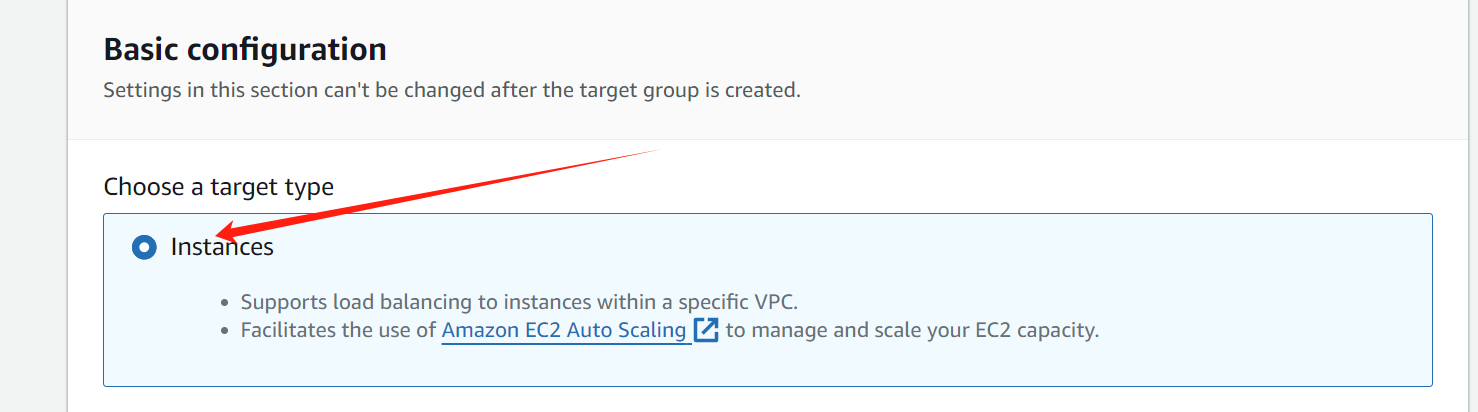




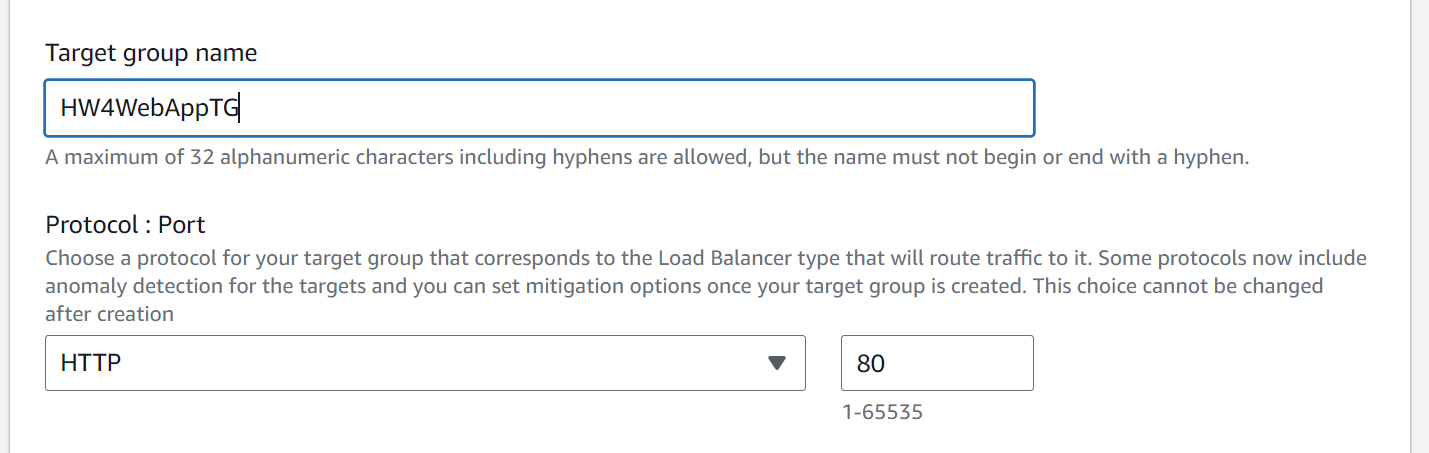
### Step 6 (20 points)

**Setup the Application Load Balancer (ALB) and register it on the target group. Remember, you need to create a Target Group first. Name the target group as "HW4WebAppTG". Name your ALB as "HW4WebAppALB".**

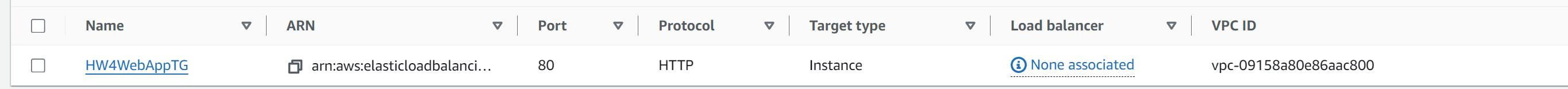
**Create a Target Group,** **choose "Instances" as the target type.**



**Name it "HW4WebAppTG".**

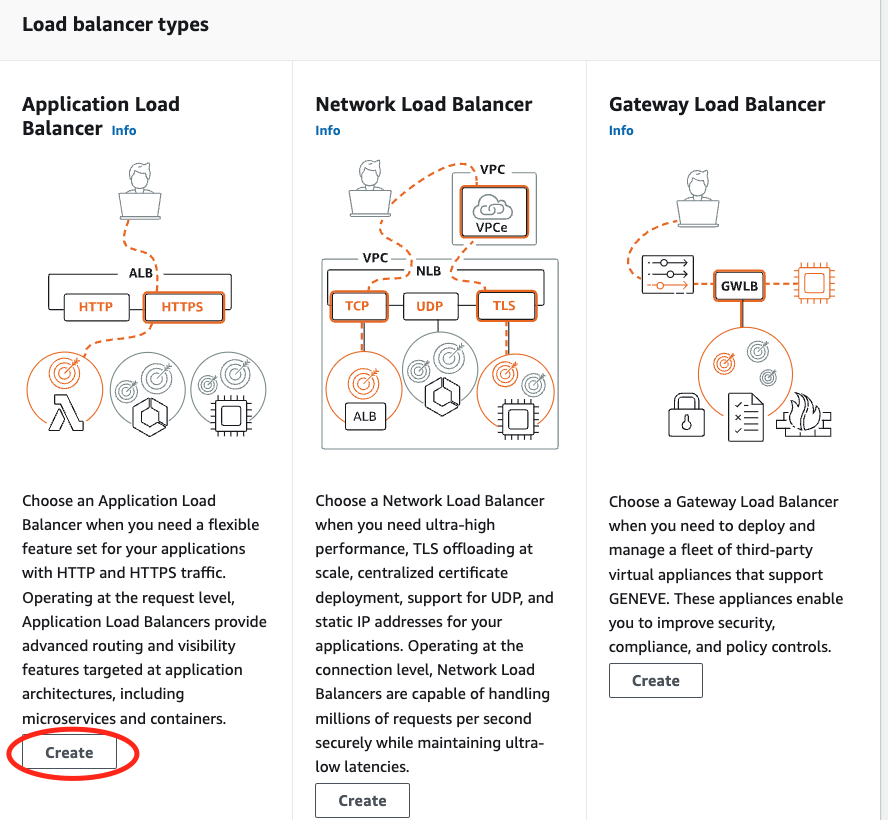


**Click "Create target group" to create.**

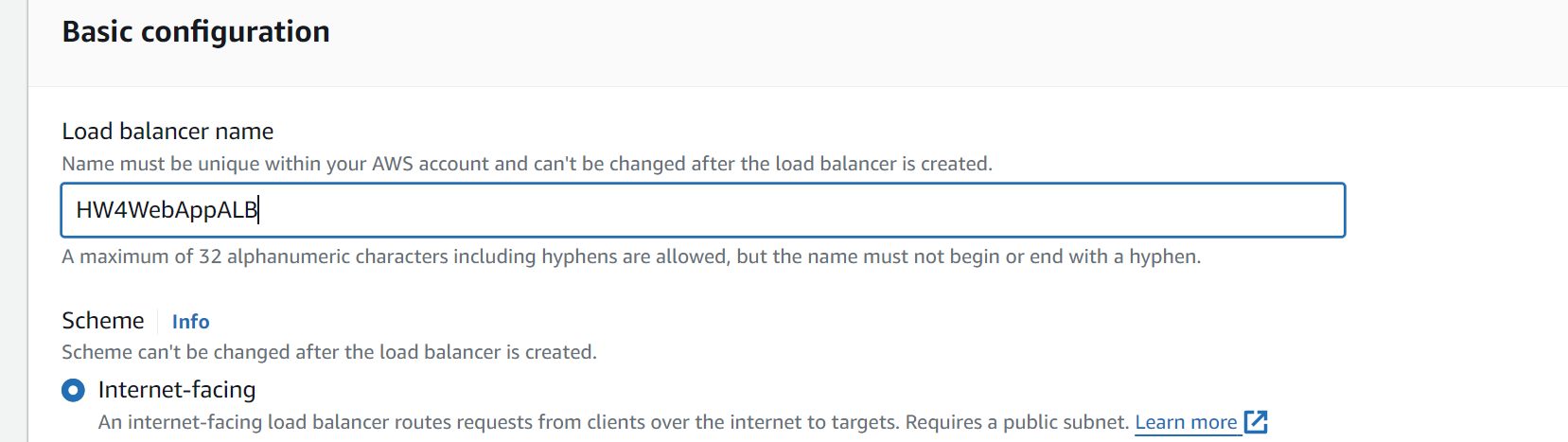


**Create the Application Load Balancer.**

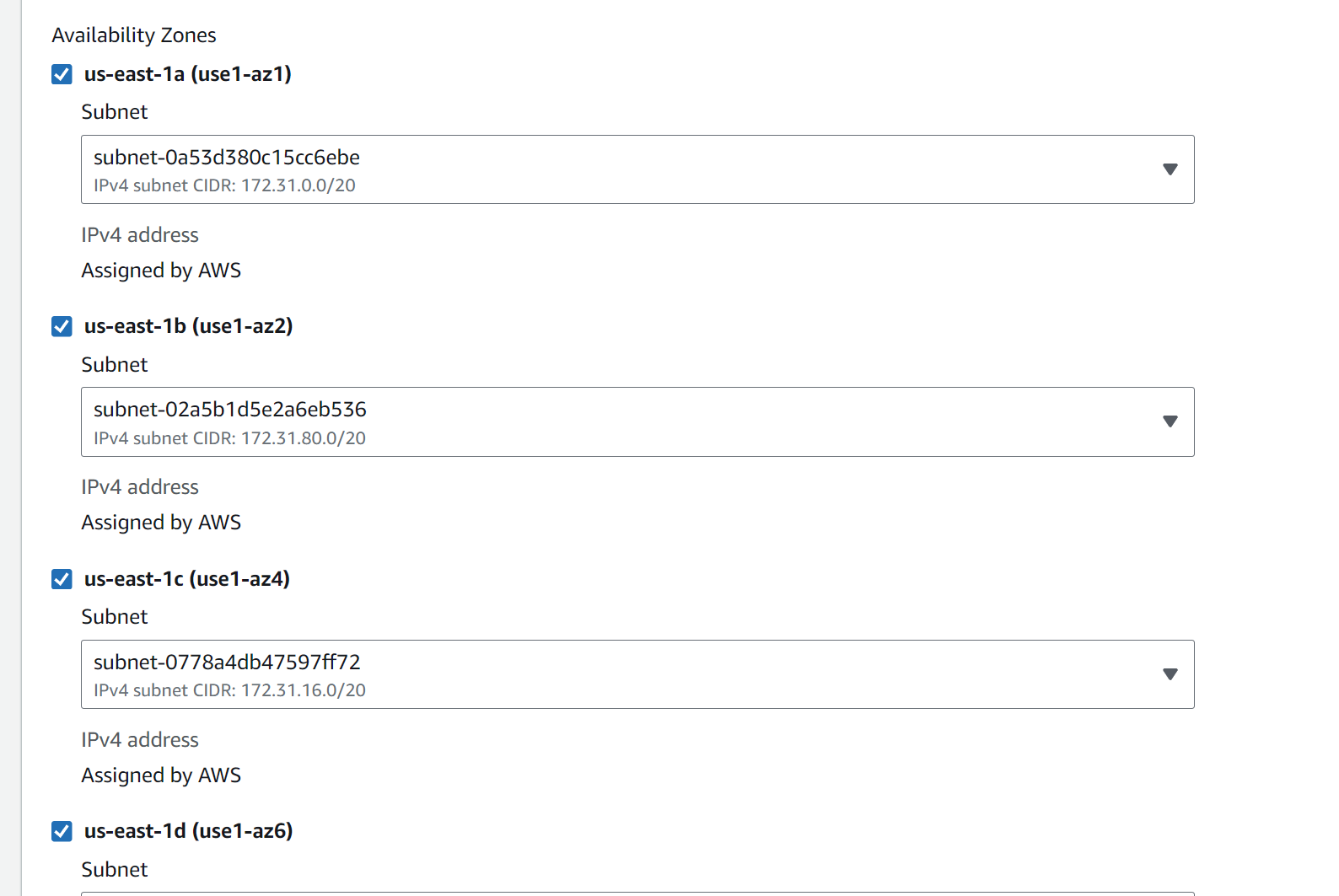
**Click "Create Load Balancer" and choose "Application Load Balancer".**



**Name it "HW4WebAppALB".**



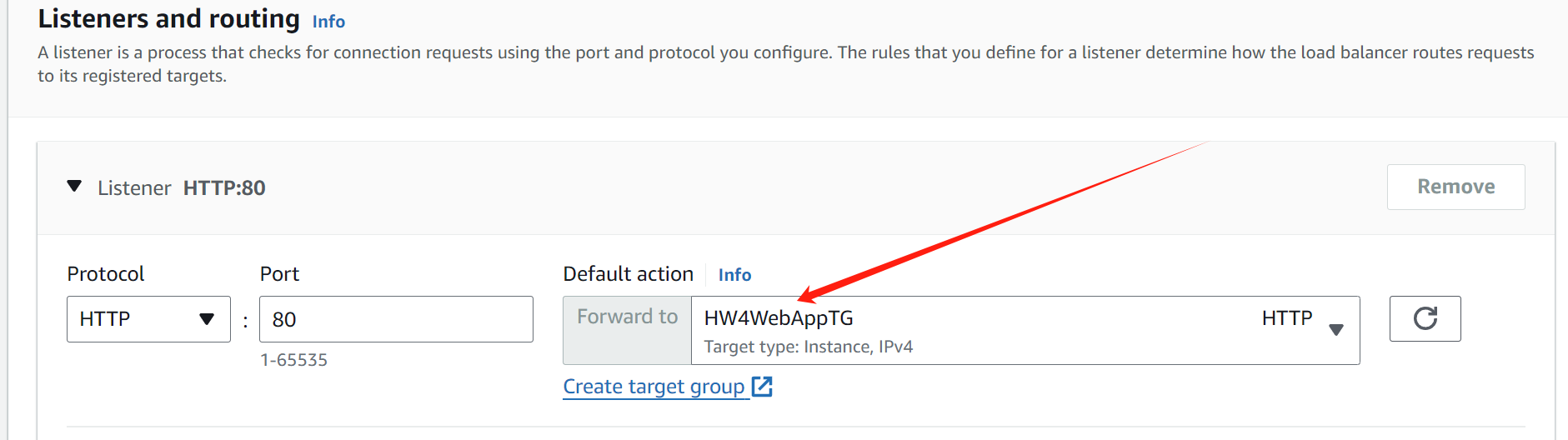
**Select all Availability Zones.**



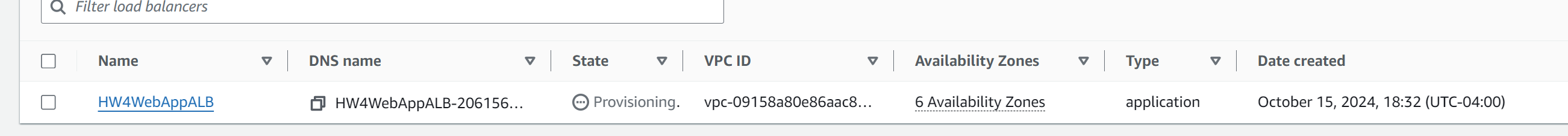
**Select the security group that allows HTTP traffic (port 80) from anywhere.**



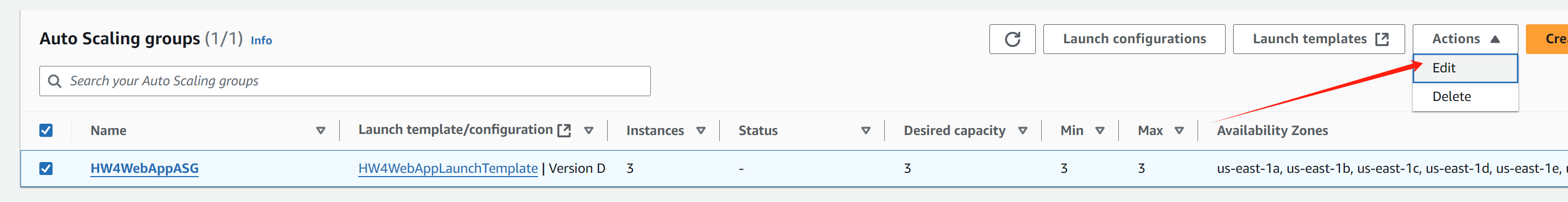
**In the Listeners section, ensure it's configured for HTTP on port 80, select the "HW4WebAppTG" target group just created.**



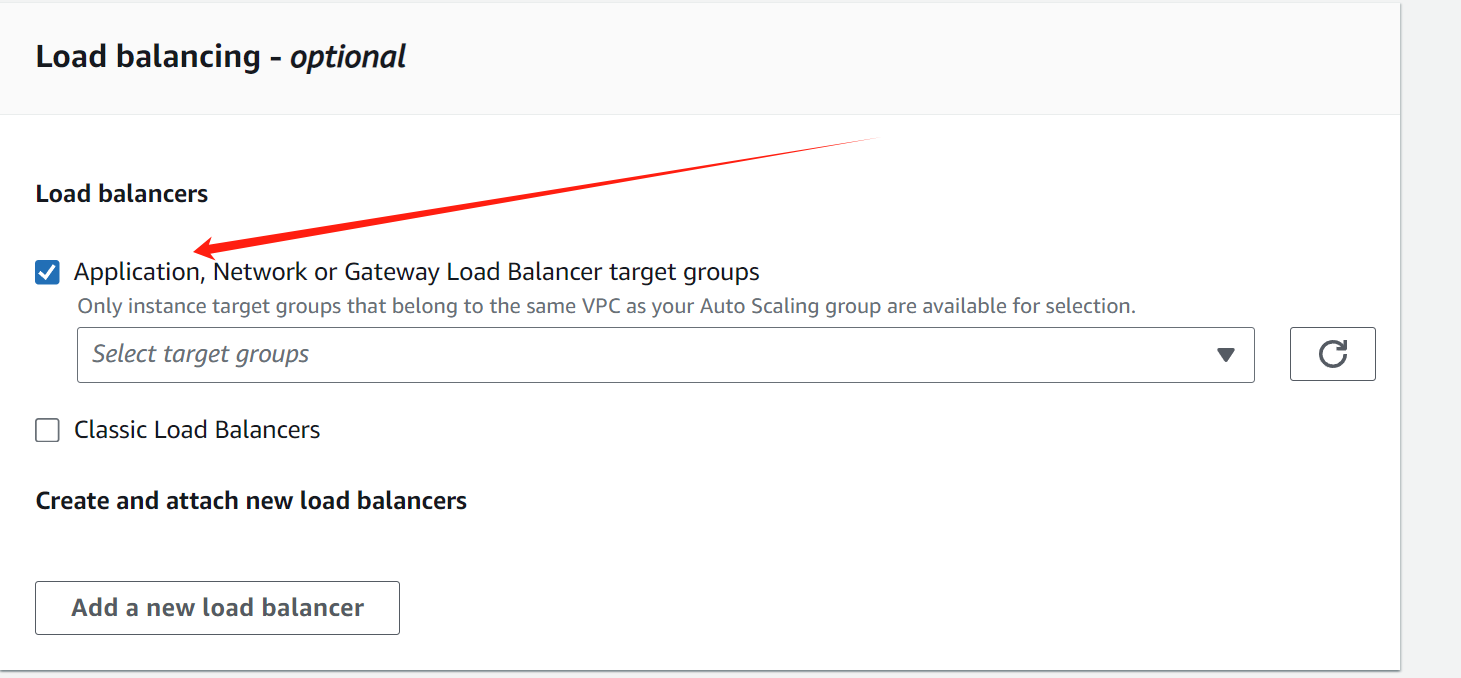
**Review and create the load balancer.**



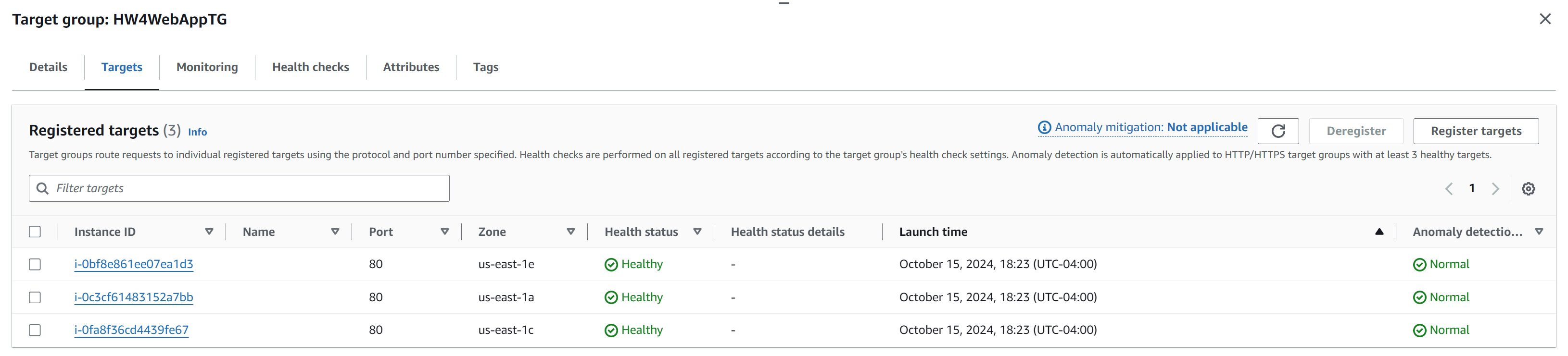
**Update the Auto Scaling Group to use the ALB.**



**In the "Load balancing" section, select the "HW4WebAppTG" target group.**

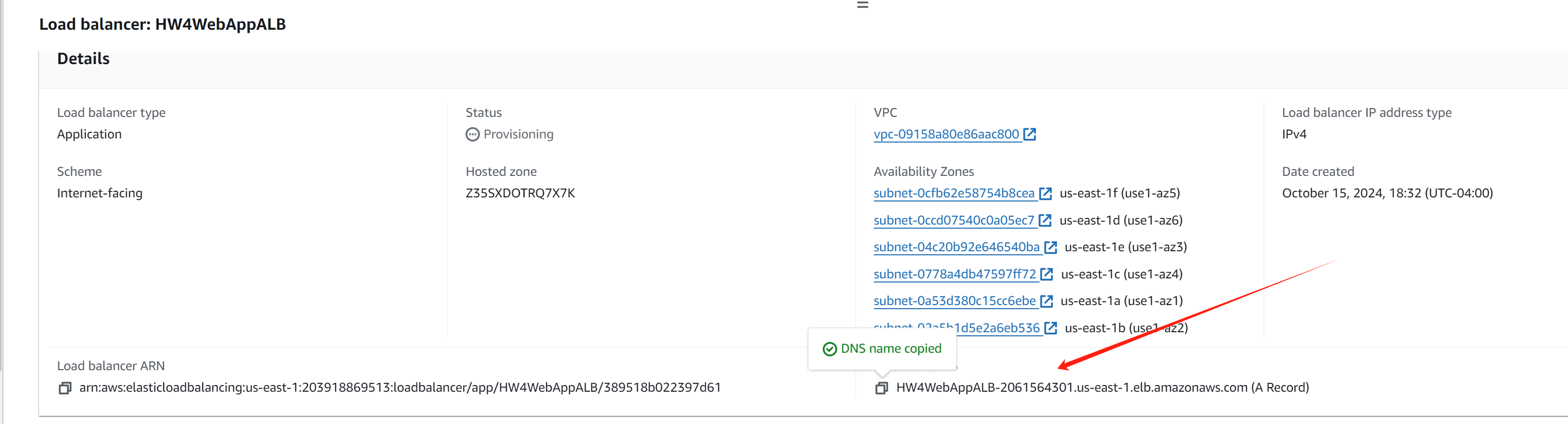


**Save the changes.**



**Test using your browser to open the ALB DNS URL, check if each refresh changes the targets.**

**Once the ALB is provisioned, find its DNS name in the load balancer details.**

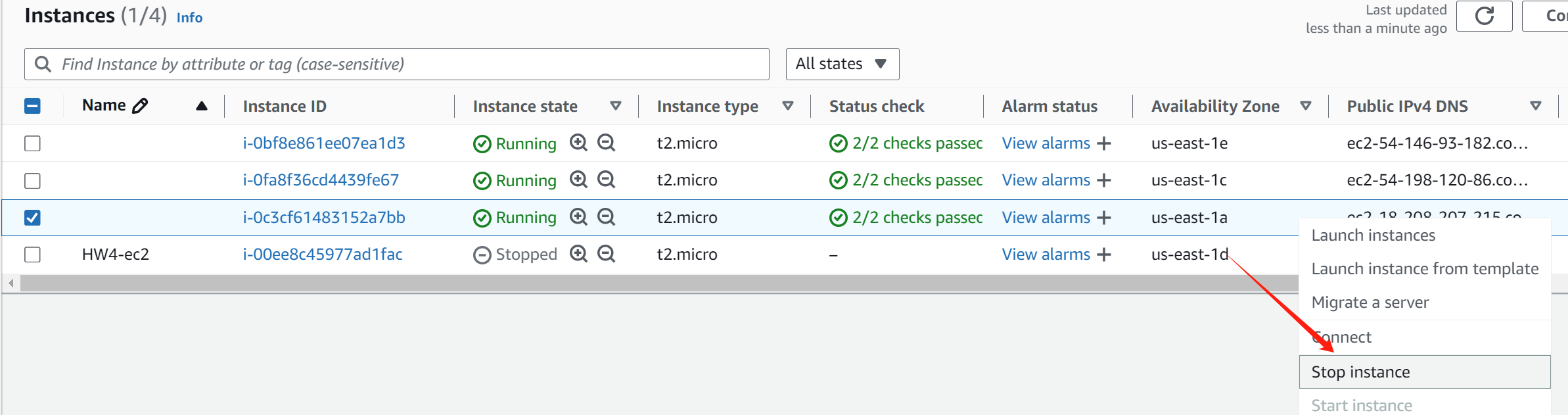


**Open a web browser and paste the ALB's DNS name. Refresh the page several times. It should be seen that the Availability Zone change, indicating that it is being routed to different instances.**

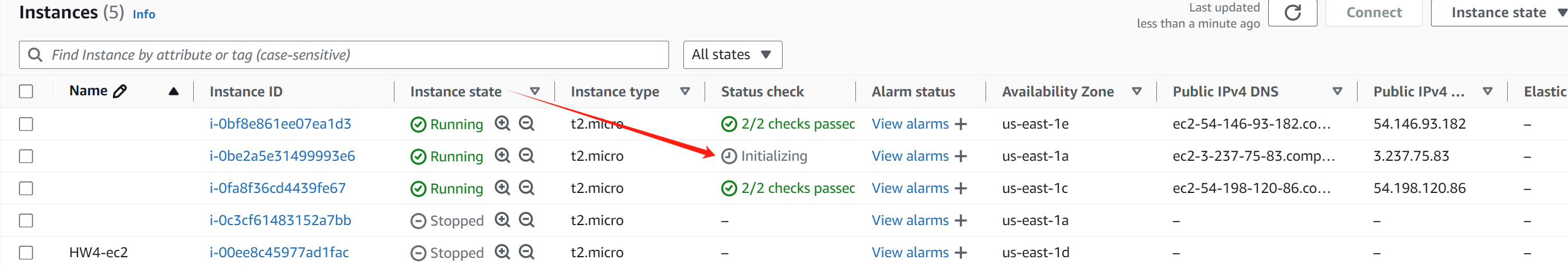




**Shut down one of the instances and check if the ASG creates a new instance for you.**



**After a while,**



**Take screenshots (or include in your video) of the ASG Activity history, which shows the history of the ASG creating/removing EC2 instances.**

