Here are the goals for this challenge:

1. Create a Linux EC2 Instance in any region (US-East-1 via Terraform + AWS)
2. Ensure that the EC2 Instance is publicly available with the Instance Public IP
3. Allow access to the server from the public internet
4. Use the Terraform destroy command to shut down OR use terraform destroy —target as a way to target destroy certain services

Here's my goal:

1. Automate the creation of an EC2 Instance with the goal to keep it protected. This will require me to set up:
- Security Group to allow port 80 (HTTP) and port 22 (SSH) That way I can access the Instance via Browser and the Terminal with the public key (for now)
- VPC and IGW with route tables, subnets. These few settings that I've assigned to the services allows me to have more control on the traffic to the instance and allow the VPC access to the internet (IGW).
- To create an EBS and EBS Snapshot, to have a snapshot of the instance as soon as it's created which I can revert back to if needed for disaster recovery.

2/3. Configuring the Security Group to allow port 80 allows this to be "reached" via Web Browser and I've set up port 22 to be used as well. In this case, I will provide the public key as it's been coded in the script as well.

Using the AWS Console to find the Public IP:

- After the instance is created, **navigate** to the EC2 dashboard
- **Click** "Instances" along the left column
- **Select the check box** to the left of the Instance name OR click the Instance ID HyperLink
- The "Public IPv4 Address will be located near the very top, regardless of your selection
- **Copy** the Public IPv4 Address
- **Open a new Tab**, then paste the Public IPv4 Address into the Web Browser URL Box. Press "Enter"
- You'll be redirected to a page that states ""Hello World from IPxxxxxxxx"

Now you've connected!

Using Terraform to find the Public IP:

Here you can open the terminal:

- **type:** cd SimpliSafeHW (my current dir) then press “Enter”
- **type:** terraform state list, press “Enter” (this will show the services that are currently coded with terraform and provisioned into AWS.
- **Copy** the entire instance name (example: aws_instance.ss_instance)
- **type:** terraform state show + (paste instance name) then press “Enter” This will provide the information available for the service. For now, we’re going to look for the public IP (which changes every time the instance is booted up again)
- **Copy** that public ip and paste into your Browser URL to be redirected to a page which should show “Hello World from IPxxxxxxxx”

Using SSH to connect to the Public EC2 Instance:

Here you can open the terminal:

- **type:** cd SimpliSafeHW (my current dir) and press “Enter” this changes the current directory that you’re working in
- **type:** terraform state list, press “Enter” (this will show the services that are currently coded with terraform and provisioned into AWS).
- **Copy** the entire instance name (example: aws_instance.ss_instance)
- **type:** terraform state show (paste instance name) press “Enter” This will provide the information available for the service. For now, we’re going to look for the public IP (which changes every time the instance is booted up again)
- **type:** ssh -i ~/.ssh/sskey ec2-user@ec2-insert-the-public-ip-here.compute-1.amazonaws.com (example: ssh -i ~/.ssh/sskey ec2-user@ec2-54-91-14-14.compute-1.amazonaws.com)

Note: If you’re unable to find the code required:

- Log into the AWS Console and navigate to the EC2 service page and along the left
- Click “Instances” you’ll then be redirected to the instance dashboard.
- Select the Instance you’re wanting to connect to, then click the“connect” button to be redirected to different options to connect.
- Then click “SSH client”. There will be a section that provides you the public IP with the appropriate code

Another Note: EC2 Instance Connect, is very identical to SSH. It’s AWS CLI.

4. Now that you’re able to connect, via SSH and publicly via the Web Browser. Feel free to use the following command(s) to shut down what was built
- **type:** terraform destroy, press “Enter” (this command will destroy/shut down all services within the directory)
- **type:** terraform destroy —target “resource_type.resource_name”, press “Enter” (this command will ONLY destroy / shut down the specific service(s)