Lab 2

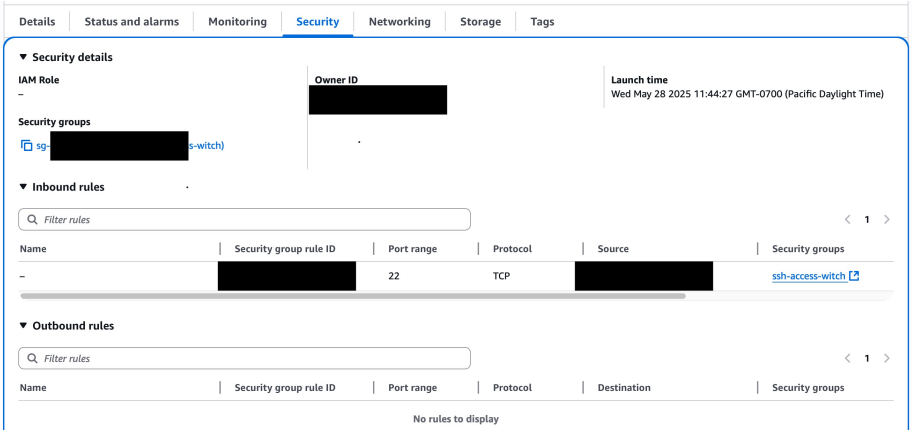
Tasks

- ☒ 1. Define Security Groups From here. Security groups control inbound & outbound networking traffic to your resource (in this case AWS EC2). The properties you can restrict by for each security group rule are destination, port range, & protocol. Security-wise it's best if a resource is completely locked down (no inbound/outbound traffic allowed at all), but since we live in the real world there are a lot of network rules to consider in order to allow your projects to do real work.
- ☒ 2. Discover your public ip
  - Recorded
- ☒ 3. Create a security group
- ☒ 4. Attach to ec2
- ☒ 5. Verify access
- ☒ 6. Terraform import

```
**main.tf**
import {
  id = /instance-id/
  to = aws_instances.my_first_linux
}
```

```
terraform init
terraform plan -generate-config-out=generated.tf
# manually fixed the generated.tf file
terraform apply
```

Reflection



I built a security group for my newly created ec2 instance (my-first-linux) and updated the ec2 so that it only used the newly created security group. This security group's only networking rule is to allow SSH connections coming from my home's IP address.

Challenges: I glanced over the AWS CLI – I've used it maybe a handful of times in my life (as a dev), and it's always made me a wee bit nervous.

I didn't attempt the CLI commands and instead used terraform to import the whole setup. This way I can store the current state of the resources on a git repo. This is helpful to me to remember what I just did. (At this point I've already had some experience with terraform so I'm somewhat confident about using it to deploy/tear down resources.)

Meta

- ~/Downloads/labs/aws.txt