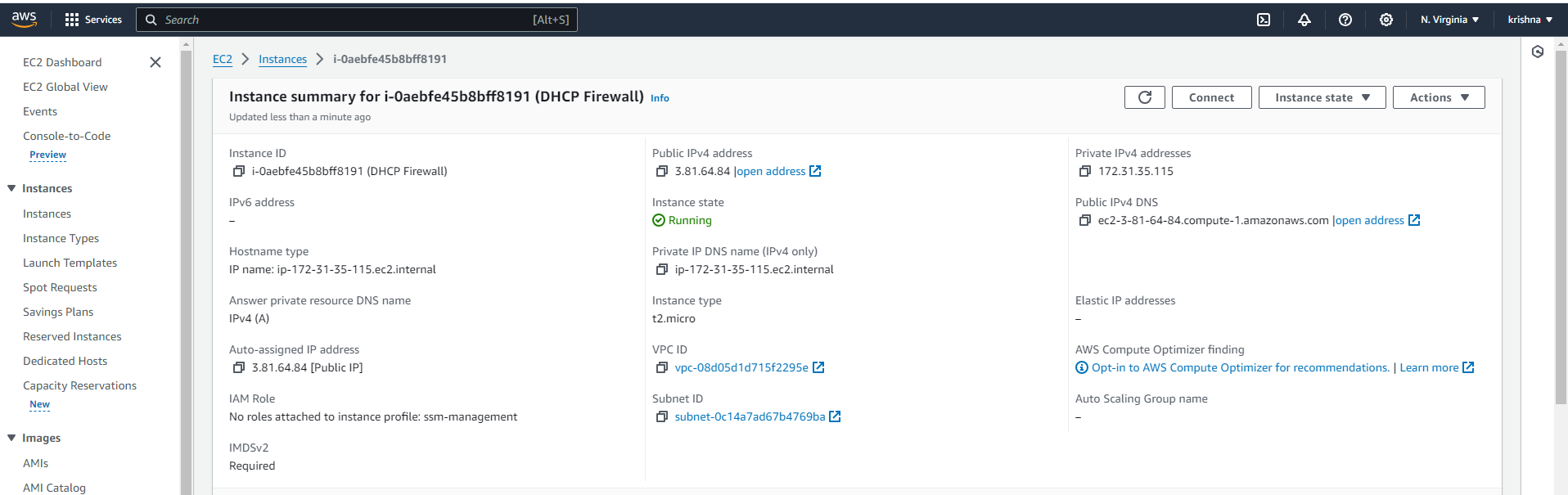
**Project Documentation: Firewall Creation Using Go Language**

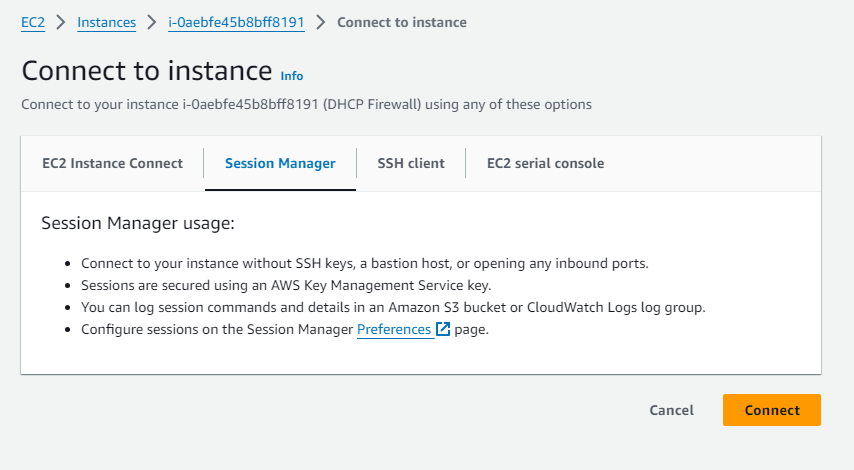
**Introduction**

This project focuses on the development of a firewall using the Go programming language. The primary objective is to utilize Go scripts to create and manage IP tables based on specific requirements. Our approach leverages Ubuntu 22 as the base image, taking advantage of its compatibility with the required packages and commands.

AWS EC2 dashboard



AWS Session Manager interface, which provides a secure browser-based session



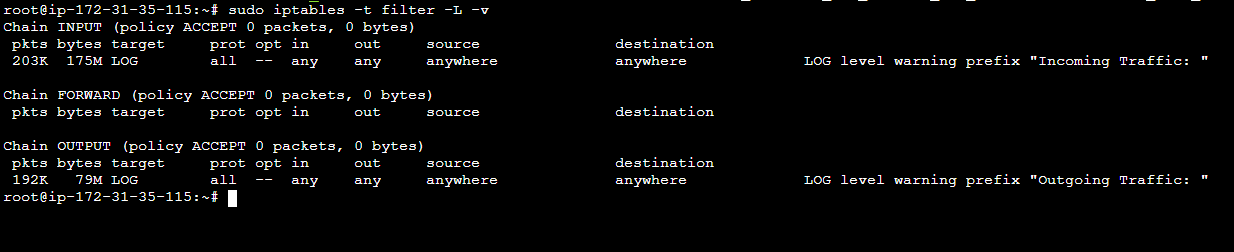
**Setting Up IP Tables in Ubuntu**

**Initial Configuration**

Ubuntu, by default, might not have IP tables installed. The first step involves checking for IP tables and installing them if necessary. The following commands are used to install IP tables in the Ubuntu machine:

1. Update package lists: **sudo apt-get update**
2. Upgrade packages: **sudo apt-get upgrade**
3. Install IP tables: **sudo apt-get install iptables**

To list the current IP tables, the command used is: **sudo iptables -t filter -L -v**. Initially, no IP tables list might be visible as the default rules are set to allow all INPUT and OUTPUT traffic.



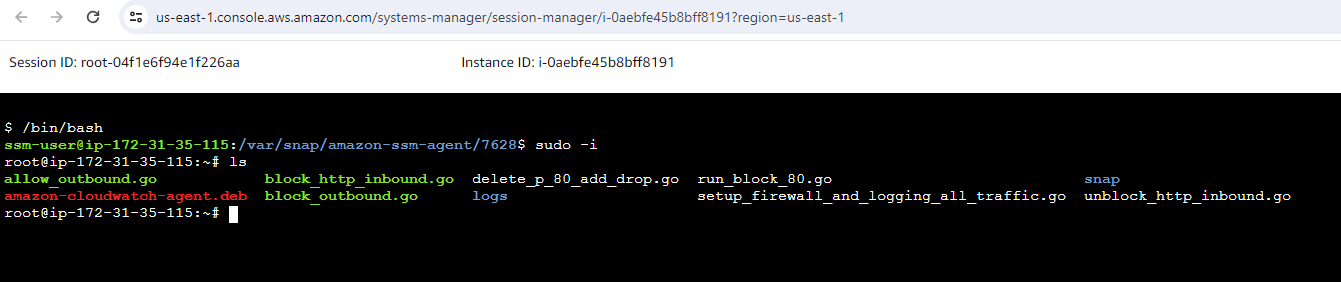
**Ensuring Persistence of IP Tables**

On restarting the operating system, IP tables may get deleted. To ensure persistence of the IP tables, the following commands are used:

1. Save current IP tables: **sudo service iptables save**
2. Install iptables-persistent package: **sudo apt-get install iptables-persistent**
3. Save the current netfilter rules: **sudo netfilter-persistent save**
4. Enable iptables to start on boot: **sudo systemctl enable iptables**

**Installing Go Language**

For the script development, Go language needs to be installed on the Ubuntu machine. This can be achieved through the command: **sudo apt install golang-go**.



**Firewall Script Development in Go**

**Blocking Inbound Traffic**

The **block\_http\_inbound.go** script is designed to block inbound HTTP traffic on port 80. The script uses the **exec** package to run the IP tables command.  
  
Github Script URL: <https://github.com/IndianGitG/golang_network_firewall/blob/main/block_http_inbound.go>

**Unblocking Inbound Traffic**

Similarly, the **unblock\_http\_inbound.go** script is used to delete the inbound block, allowing the inbound traffic on port 80.

Github Script URL: <https://github.com/IndianGitG/golang_network_firewall/blob/main/unblock_http_inbound.go>

**Implementing Rate Limiting for Inbound Traffic**

An essential feature of our firewall is the ability to limit the rate of inbound traffic to a specific number of requests per minute from a single IP address. The **limit\_http\_inbound.go** script is crafted to achieve this functionality.

Script URL: <https://github.com/IndianGitG/golang_network_firewall/blob/main/limit_http_inbound.go>

**Logging All Inbound and Outbound Traffic**

Another critical aspect of our firewall is the ability to log all inbound and outbound traffic, which aids in monitoring and security analysis. The **setup\_firewall\_and\_logging\_all\_traffic.go** script is developed for this purpose.

Script URL: <https://github.com/IndianGitG/golang_network_firewall/blob/main/setup_firewall_and_logging_all_traffic.go>

**Introduction to Log Visualization**

Having implemented logging for both inbound and outbound traffic with the logs stored at **/var/log/syslog**, the next step is to visualize these logs for better monitoring and analysis. As our Ubuntu machine is hosted on AWS, we will utilize Amazon CloudWatch for log visualization.

**Setting Up CloudWatch Agent**

Installing the CloudWatch Agent

To enable CloudWatch to access and visualize the logs, the CloudWatch agent must be installed on the Ubuntu machine. The installation and configuration process involves several steps:

1. Download and install the CloudWatch agent package:

wget https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb sudo dpkg -i -E ./amazon-cloudwatch-agent.deb

1. Start the CloudWatch agent configuration wizard:

sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-config-wizard

1. Fetch and start the agent configuration:

sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2 -c file:/opt/aws/amazon-cloudwatch-agent/bin/config.json -s

Configuring the Agent

1. Navigate to the agent configuration directory:

cd /opt/aws/amazon-cloudwatch-agent/etc/

Create and edit the **cloudwatch-config.json** file to specify log file paths and settings

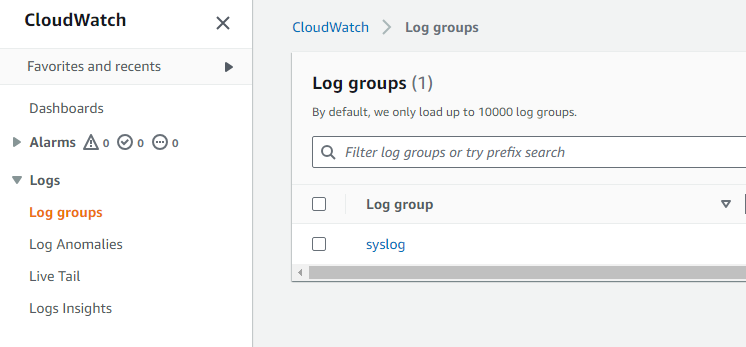
Script URL: <https://github.com/IndianGitG/golang_network_firewall/blob/main/cloudwatch-config.json>

Start the CloudWatch agent with the new configuration:

sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2 -c file:/opt/aws/amazon-cloudwatch-agent/etc/cloudwatch-config.json –s

Check the status of the CloudWatch agent:

sudo systemctl status amazon-cloudwatch-agent.service



**Visualizing Logs in CloudWatch**

Once the CloudWatch agent is correctly configured and running, it will start sending the specified logs to AWS CloudWatch. These logs can then be accessed in the CloudWatch console under the specified log group name (**syslog** in our case).

Creating Dashboards

In CloudWatch, you can create dashboards to visualize the log data. These dashboards can be customized to display various metrics and logs, providing a comprehensive view of network traffic and potential security incidents.

