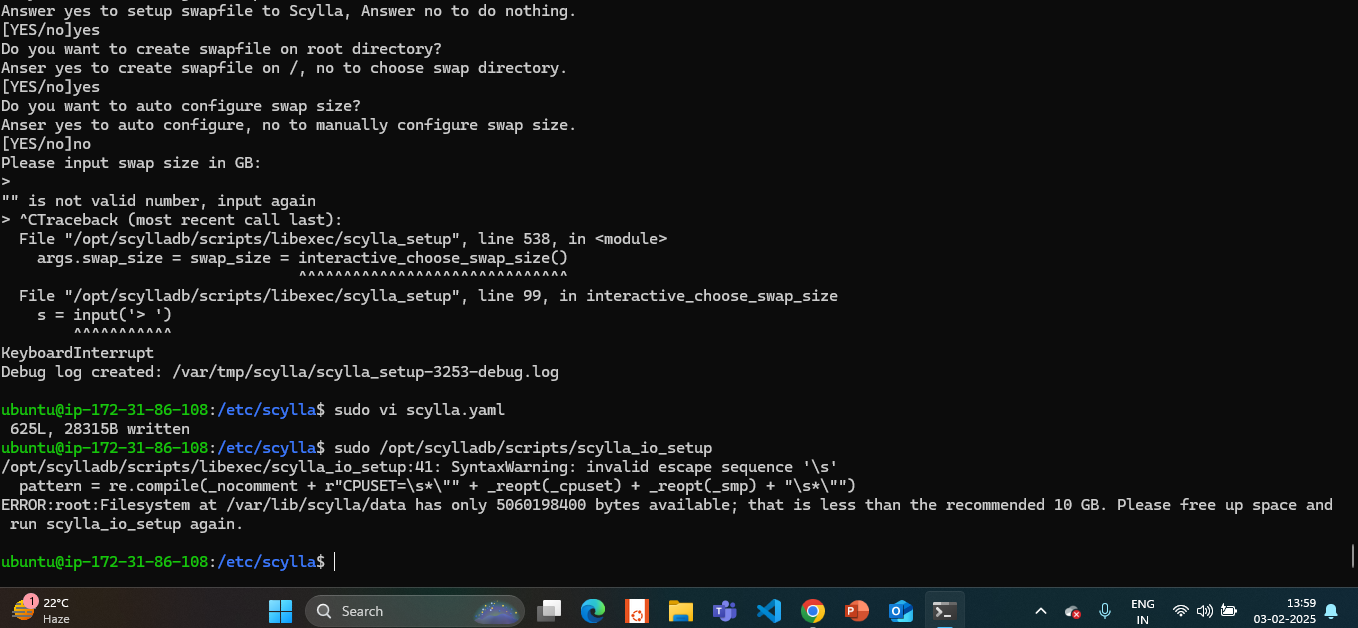
Employee-Api server setup

**Step 1**: Prepare AWS environment for your setup.

1. **Log in to your AWS account**
2. **Create an EC2 instance:**

* Go to EC2 Dashboard.
* Clink on Launch Instance and configure the following:  
   **~AMI:** Use Linux based AMI like **Ubuntu 22.04**.

~ **Instance type:** Select at least t2.medium {or larger type} otherwise it will throw the following error while setting up the server.



**~ Key pair:** Create or select the existing key pair.

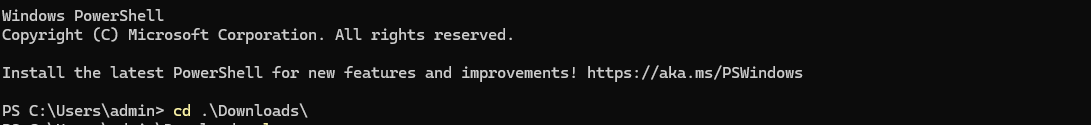
**~ Security group:** Open the following ports:

9042 for Scylla DB and for HTTPS.

**Step 2:** Go to Windows terminal to access the ec2 instance or you can directly access the aws terminal.

**Ensure Prerequisites Are Met:**

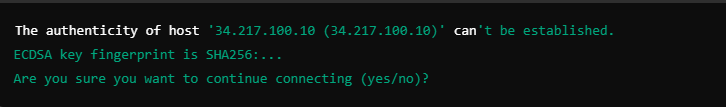
* **Amazon EC2 instance**: You need to have an EC2 instance running on AWS.
* **SSH Key Pair**: When you create an EC2 instance, you must either create or use an existing SSH key pair (your-key.pem).
* **Windows Environment**: Use a terminal like **Command Prompt** or **Windows PowerShell**, or ideally **Windows Subsystem for Linux (WSL)** or **Git Bash** (which is recommended for ease of use with SSH).
* **Security Groups**: The EC2 instance's security group should allow SSH (port 22) access from your IP address.

****

Check your pem key location accordingly.

And write **ssh -i <your\_key.pem> ec2-user@instance\_ip>**.

* Accept the host key:



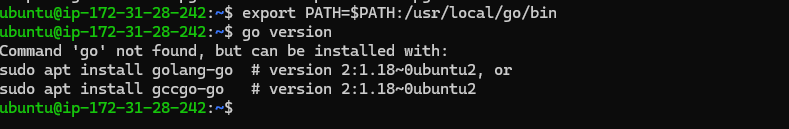
* Verify the connection:



Now you’re good to go.

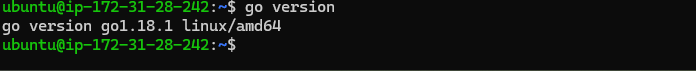
* Now use the following commands for Go installation:

1. **sudo apt update** : To update the packages.
2. **export PATH=$PATH:/usr/local/go/bin:** It allows you to run Go (go) commands from any terminal without specifying the full path to the Go binary.

****

**3. sudo apt install golang-go and** check the version by using **go version**

Installs the Go programming language package (golang-go) from the system’s package repository. This is used to install Go if it's not already installed.

****

**Scylla Installation**

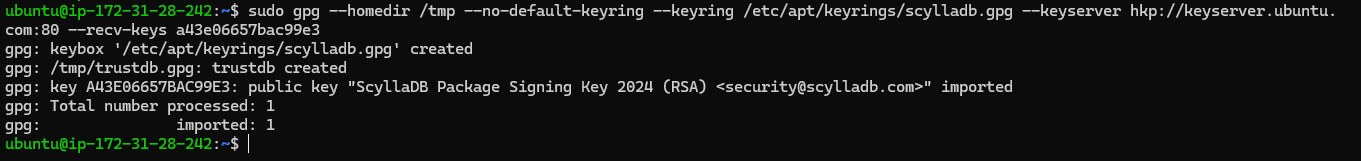
**4. sudo mkdir -p /etc/apt/keyrings**

* This creates a new directory (/etc/apt/keyrings) if it doesn't exist, using -p to ensure no errors if the directory already exists.



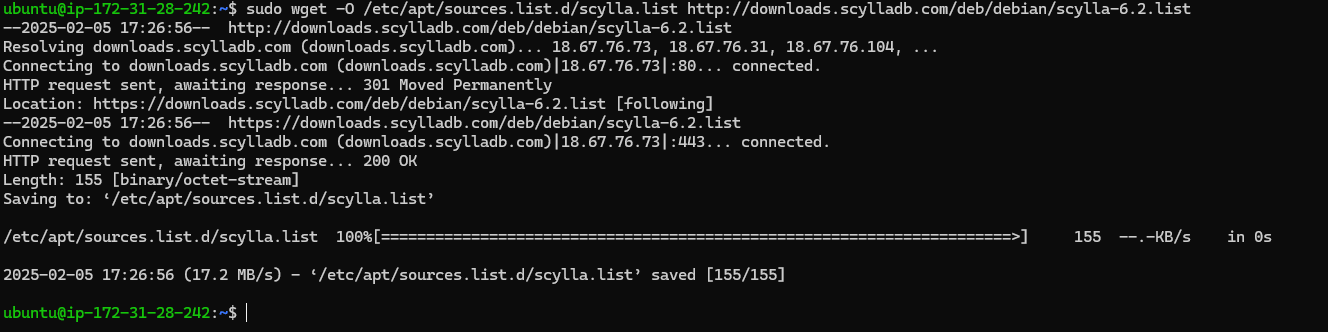
**5. sudo gpg --homedir /tmp --no-default-keyring --keyring /etc/apt/keyrings/scylladb.gpg --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys a43e06657bac99e3**

* This imports the ScyllaDB GPG public key into the system’s keyring. The key is used to verify the authenticity of the packages from the ScyllaDB repository. The key a43e06657bac99e3 is fetched from the key server.



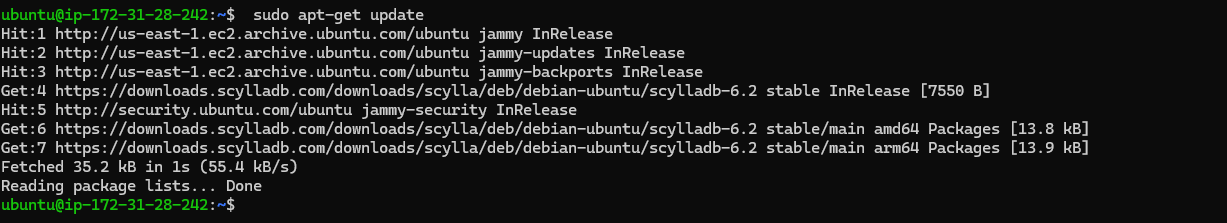
**6. sudo wget -O /etc/apt/sources.list.d/scylla.list http://downloads.scylladb.com/deb/debian/scylla-6.2.list**

Downloads the ScyllaDB repository list (scylla.list) and places it in the correct directory (/etc/apt/sources.list.d/). This list contains the URLs of repositories that provide ScyllaDB packages



**7. sudo apt-get update**

* Updates the package list again, this time including the newly added ScyllaDB repository, so that the system is aware of the available packages from ScyllaDB.



**8. sudo apt-get install -y scylla**

* Installs ScyllaDB using apt-get. The -y flag automatically confirms any prompts during installation.

**9. scylla --version**

* This command outputs the version of ScyllaDB currently installed on the system.



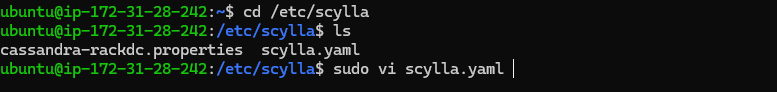
**Add authorization for CQL - in configuration file of Scylla**

**10. cd /etc/scylla**

* Changes the working directory to /etc/scylla, which is where the Scylla configuration files are stored.

**11. ls**

* Lists the contents of the /etc/scylla directory, so you can see the configuration files (like scylla.yaml).

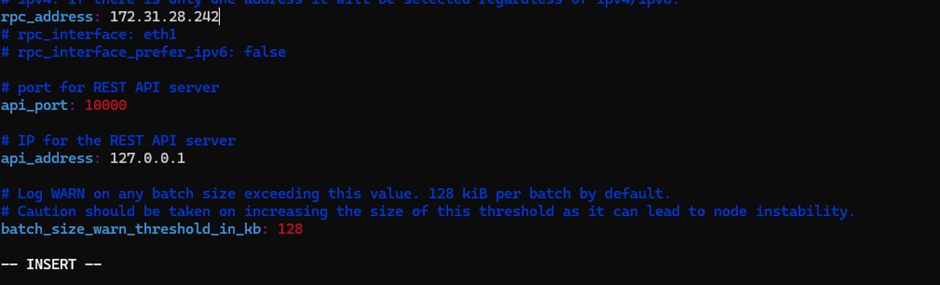


**12. sudo vi/nano scylla.yaml**

* Opens the scylla.yaml configuration file in the vi/nano text editor.

**Changes to make:**

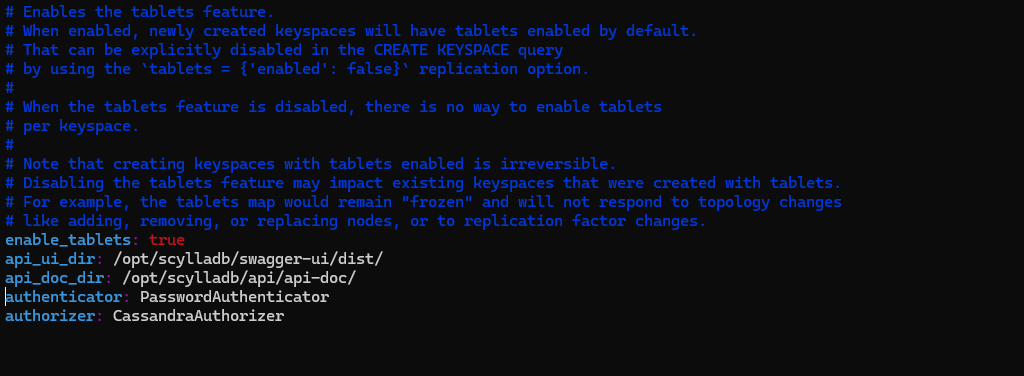
rpc\_address: [instance private IP address]



And add

authenticator: PasswordAuthenticator

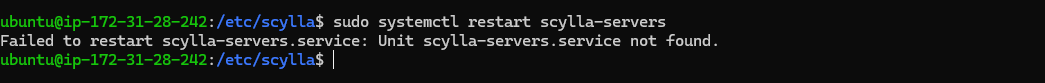
authorizer: CassandraAuthorizer



**Start Scylla service**

**13. sudo systemctl restart scylla**

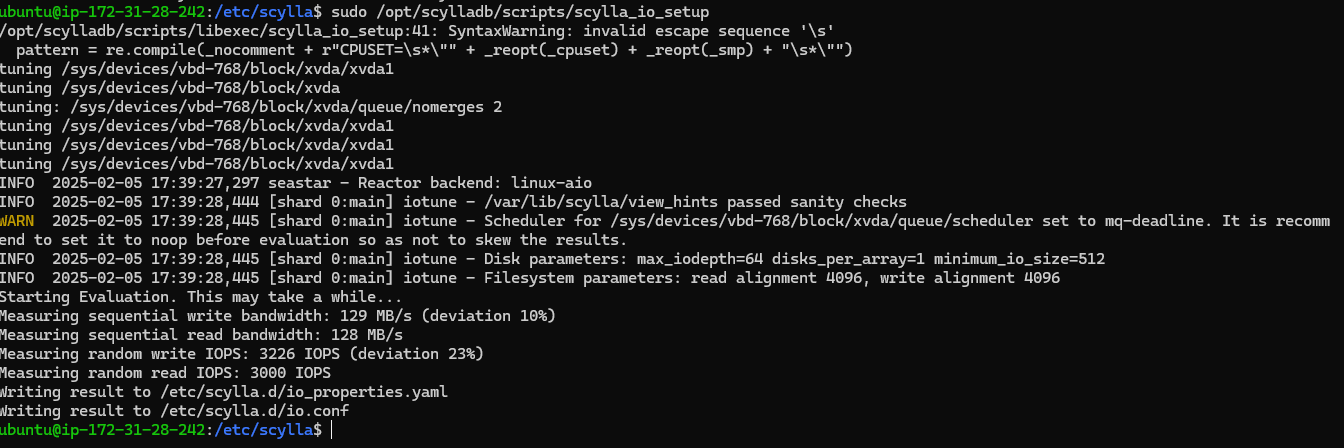
* Restarts the ScyllaDB service using systemctl, which is a tool for managing services in Linux. This reloads any configuration changes.
* It won’t start because we haven’t installed the dependencies yet.



**Scylla Dependencies installation:**

**14. sudo /opt/scylladb/scripts/scylla\_io\_setup**

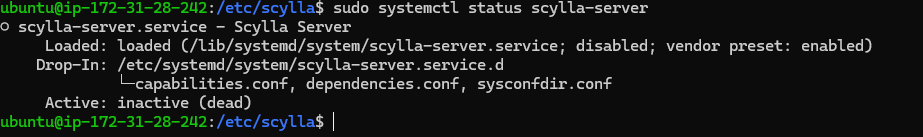
* Runs the Scylla I/O setup script, which configures the system for optimal disk performance for ScyllaDB.



**Again try to Start Scylla service**

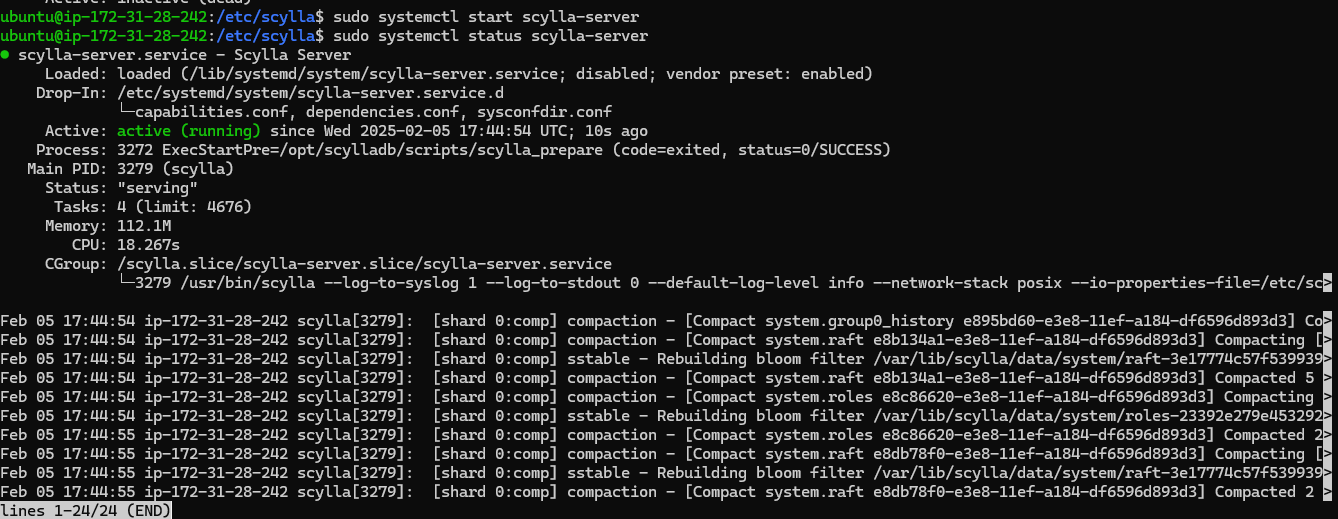
**15. sudo systemctl status scylla-server**

* Checks the status of the ScyllaDB server to see if it is running or if there are any issues.



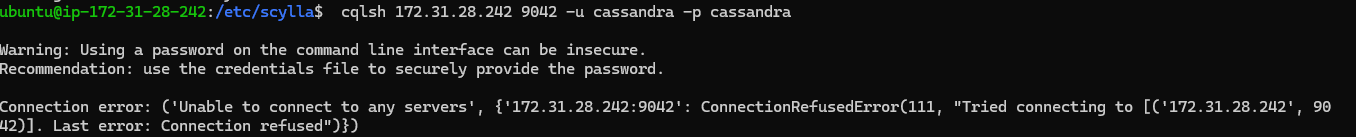
**16. sudo systemctl start scylla-server**

* Starts the ScyllaDB server service.



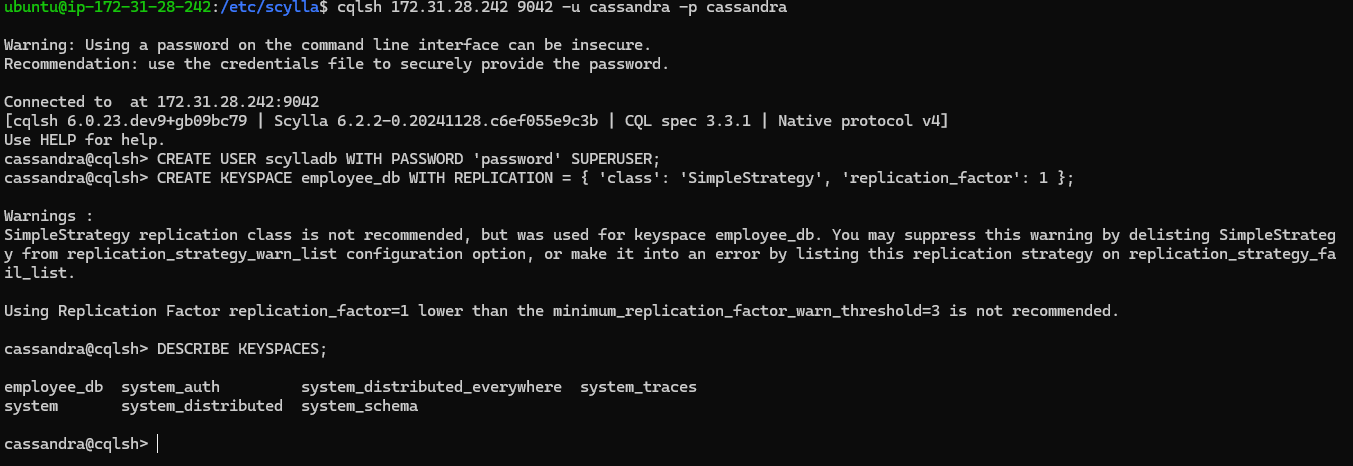
**17. cqlsh <instance-private-Ip> 9042 -u cassandra -p cassandra**

* Connects to ScyllaDB on the private IP with the cassandra credentials. If the scylla service is not active it will throw the following error.



So make sure that the scylla server is in active state. If it is active then use following Commands in CQL

* **CREATE USER scylladb WITH PASSWORD 'password' SUPERUSER;**
  + Creates a user scylladb with superuser privileges and the password password.
* **CREATE KEYSPACE employee\_db WITH REPLICATION = { 'class': 'SimpleStrategy', 'replication\_factor': 1 };**
  + Creates a new keyspace (database) named employee\_db with a replication factor of 1.
* **DESCRIBE KEYSPACES;**
  + Lists all available keyspaces in the database.



**18. cd**

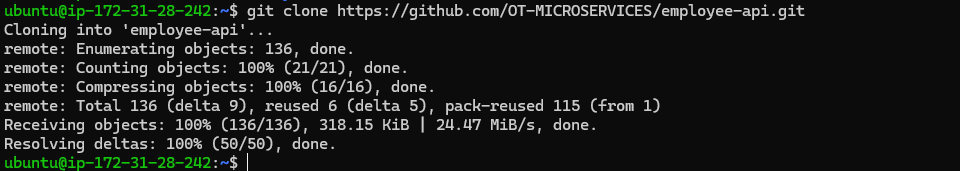
* Returns to the home directory.



**Now clone the Git repository**

**19. git clone https://github.com/OT-MICROSERVICES/employee-api.git**

* Clones the employee-Api repository using HTTPS (this typically works even without SSH keys).

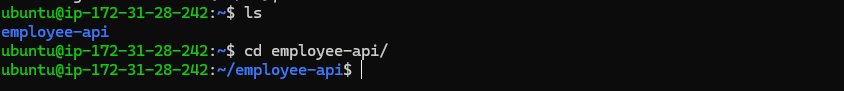


**20. ls**

* Lists the files and directories in the current working directory.

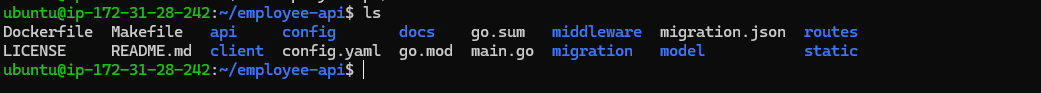
**21. cd employee-Api/**

* Changes the directory to employee-Api, where the cloned repository is located

.

**22. ls**

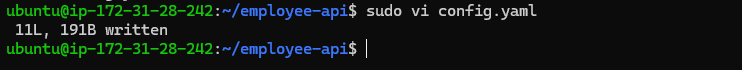
* Lists the files in the employee-Api directory.



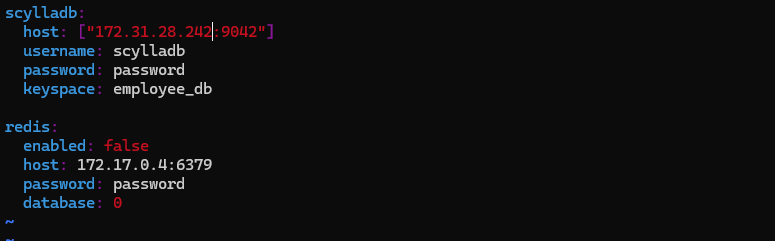
**Change Config File**

**23. sudo vi/nano config.yaml**

* Opens the config.yaml file in the nano/vi text editor. You might edit this file to configure settings for the application and write **replace the ip with your actual private** Ip of the instance in **host**





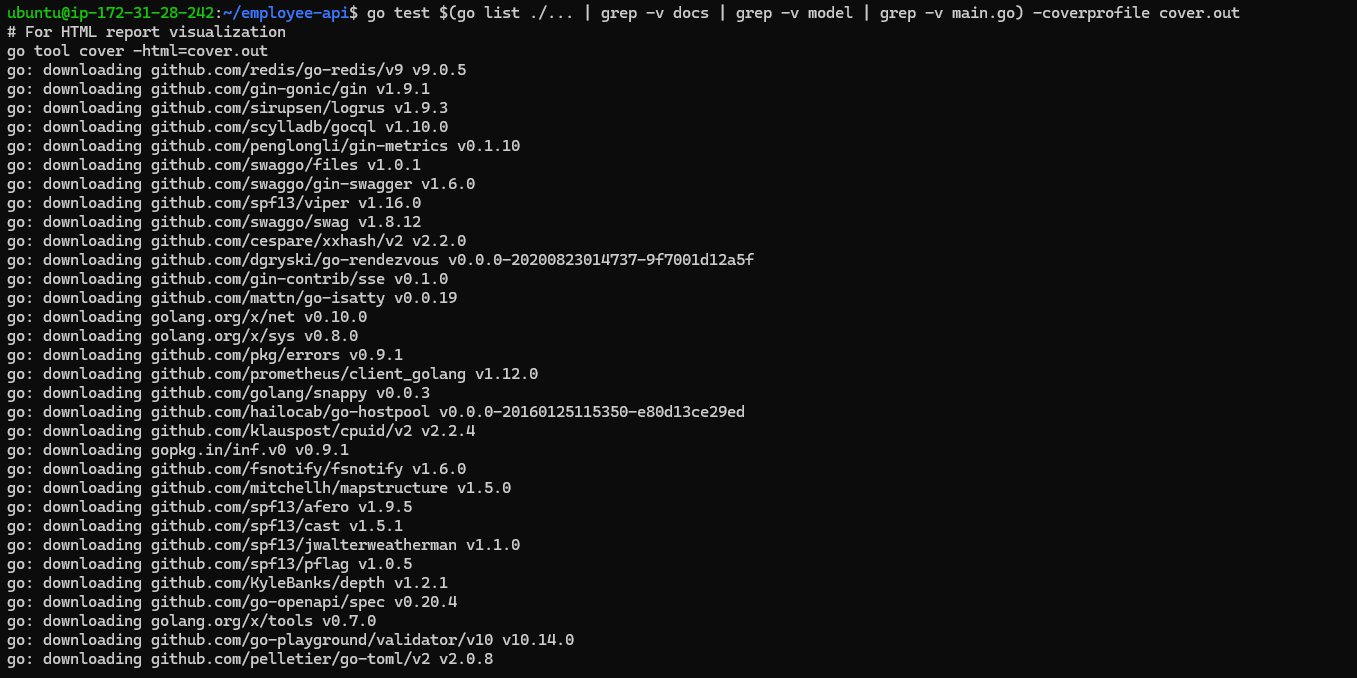


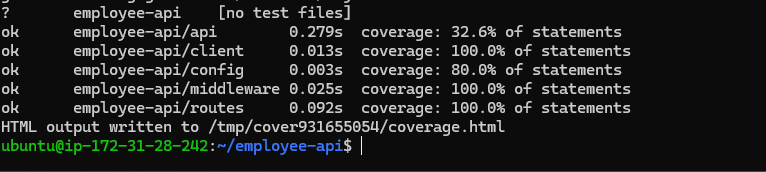


**24. go test $(go list ./... | grep -v docs | grep -v model | grep -v main.go) -coverprofile cover.out**

**# For HTML report visualization**

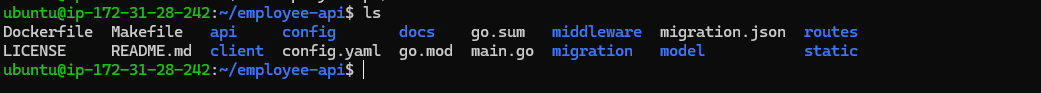
**go tool cover -html=cover.out**

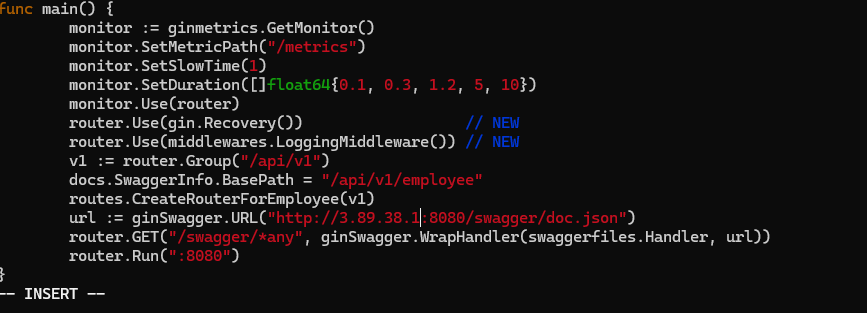
****

****

**25. sudo vi/nano main.go**

* Opens the main.go file in the nano editor for editing.

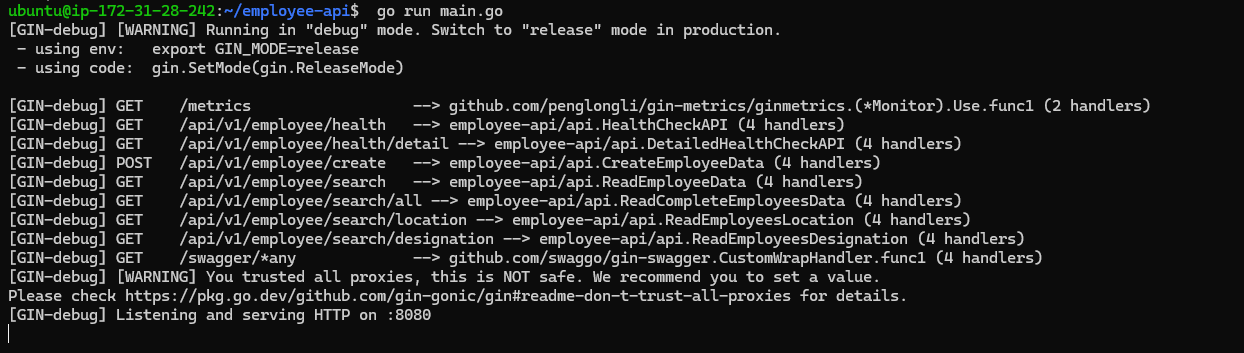




Replace localhost to <**public-ip:8080>**

**26. go run main.go**

* Runs the main.go file again after any changes.



**Now at last hit public-Ip:8080/<path> on your browser**

**The swagger page will be accessible on :** [**http://public-ip:8080/swagger/index.html**](http://public-ip:8080/swagger/index.html) **:D**

