**How to setup Postgress on VM machine**

**Installation on ubuntu**

apt update

apt install postgresql -y

sudo apt install postgresql-client -y

sudo systemctl status postgresql.service

sudo systemctl restart postgresql.service

**Configure PostgreSQL**

sudo -su postgres

psql

alter user postgres with password 'admin@123';

show hba\_file;

\q

vi /etc/postgresql/14/main/postgresql.conf

listen\_addresses = '\*'

**Now edit the PostgreSQL access policy configuration file.**

\q

vi /etc/postgresql/14/main/pg\_hba.conf

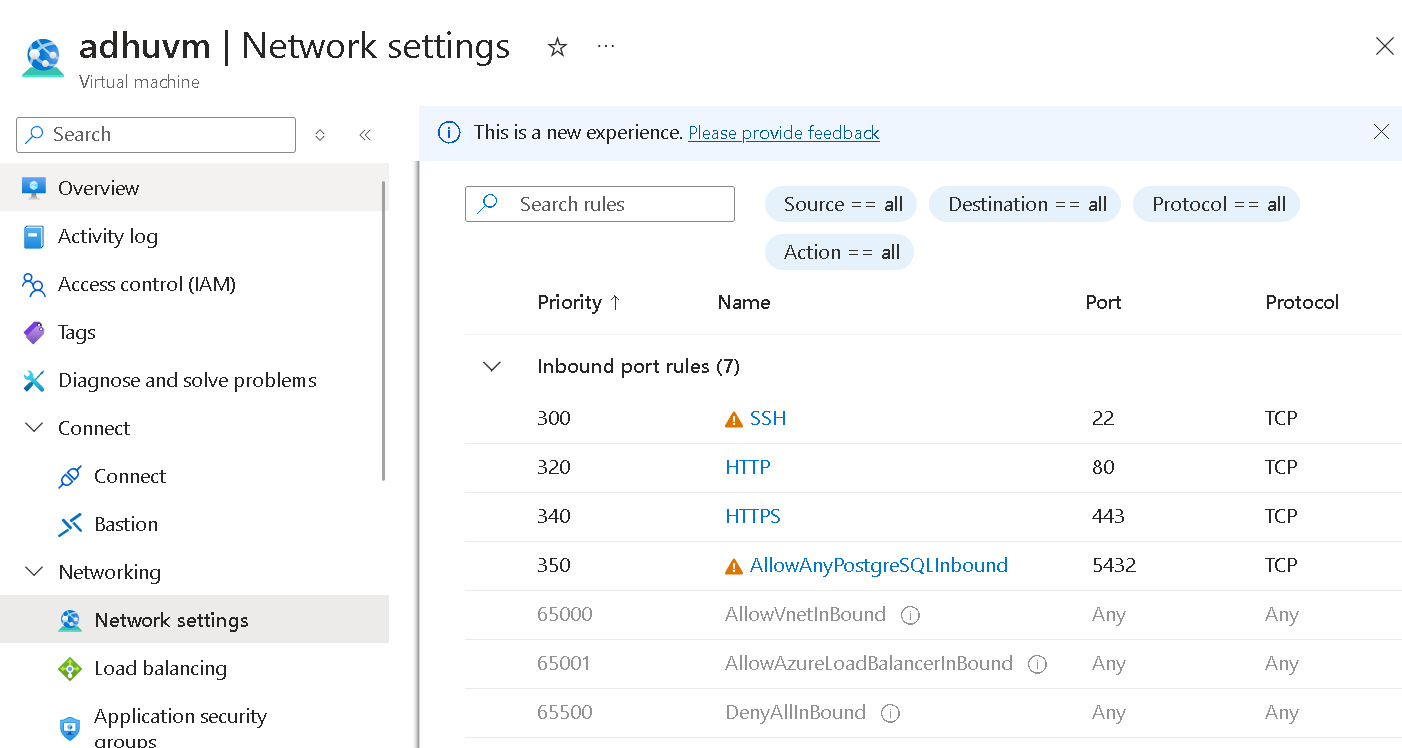
**host all all 0.0.0.0/0 md5**

psql --host adhuvm.eastasia.cloudapp.azure.com --username postgres --password --dbname template1

**How to connect the psql:**

psql --host adhuvm.eastasia.cloudapp.azure.com --username postgres --password --dbname springbootdb

**How to allow Traffic from Network Interface on PostgreSQL 5432**



**How to install pgadmin and connect to DB server from remotely.**

**A screenshot of a computer

Description automatically generated**

**Once connected, please create a Database**

**sudo -u postgres psql**

**CREATE DATABASE springbootdb;**

**\conninfo**

Check if the database was created:

**\l**

Exit psql:

**\q**

**Ingest the Data into Postgresql using stored procedure**

-- Create a function to insert employees

CREATE OR REPLACE FUNCTION insert\_employees()

RETURNS VOID AS $$

DECLARE

i INT := 1;

BEGIN

WHILE i <= 4000000 LOOP

INSERT INTO employees (first\_name, last\_name, email)

VALUES (

CONCAT('sumanth', i),

CONCAT('krishna', i),

CONCAT('email', i, '@example.com')

);

i := i + 1;

END LOOP;

END;

$$ LANGUAGE plpgsql;

-- Execute the function

SELECT insert\_employees();

-- Verify the number of records inserted

SELECT COUNT(\*) FROM employees;

**Run the Procedure in PSQL from Query Editor**

**CREATE TABLE employees (**

**id SERIAL PRIMARY KEY,**

**first\_name VARCHAR(50),**

**last\_name VARCHAR(50),**

**email VARCHAR(100)**

**);**

**CREATE OR REPLACE FUNCTION InsertEmployees() RETURNS VOID AS $$**

**DECLARE**

**i INT := 1000001;**

**BEGIN**

**WHILE i <= 2000000 LOOP**

**INSERT INTO employees (first\_name, last\_name, email)**

**VALUES (CONCAT('FirstName', i), CONCAT('LastName', i), CONCAT('email', i, '@example.com'));**

**i := i + 1;**

**END LOOP;**

**END;**

**$$ LANGUAGE plpgsql;**

**SELECT InsertEmployees();**

SELECT count(1) FROM employees;

SELECT \* FROM employees LIMIT 100;

**SIMULATE PERFORMANCE PROBLEMS**

**-- Switch to the desired database**

**\c springbootdb;**

**-- Create an index on the first\_name column**

**CREATE INDEX idx\_first\_name ON employees(first\_name);**

**-- Verify the index creation**

**SELECT indexname, indexdef FROM pg\_indexes WHERE tablename = 'employees';**

**-- Switch to the desired database**

**\c springbootdb;**

**-- Drop the index on the first\_name column**

**DROP INDEX idx\_first\_name;**

**-- Verify the index has been dropped**

**SELECT indexname, indexdef FROM pg\_indexes WHERE tablename = 'employees';**

**Connect to the Database:**

**psql -U username -d springbootdb**

**CREATE INDEX idx\_first\_name ON employees(first\_name);**

**SELECT indexname, indexdef FROM pg\_indexes WHERE tablename = 'employees';**

**DROP INDEX idx\_first\_name;**

**SELECT indexname, indexdef FROM pg\_indexes WHERE tablename = 'employees';**

**====Connection exploit testing=====**

**sudo vi /etc/postgresql/14/main/postgresql.conf**

**# Maximum number of concurrent connections**

**max\_connections = 20 # Example: Set to a low value to easily simulate saturation**

**# Minimum reserved connections for superusers**

**superuser\_reserved\_connections = 3**

**sudo systemctl restart postgresql.service**

**sudo systemctl reload postgresql**

**pip install psycopg2-binary**

**sudo apt update**

**sudo apt install libpq-dev python3-dev build-essential**

**pip install psycopg2**

**Check Active Connections**

**SELECT state, COUNT(\*) FROM pg\_stat\_activity GROUP BY state;**

**SELECT COUNT(\*) AS active\_connections FROM pg\_stat\_activity;**

**SELECT name, setting AS max\_connections FROM pg\_settings**

**WHERE name = 'max\_connections';**