

A dark blue vertical bar runs along the left edge of the slide. A blue arrow-shaped banner points to the right from this bar, containing the date. In the bottom-left corner, several thin, curved lines in dark blue and light gray sweep upwards and to the right.

8/15/2023

The first SRE boot camp task

Linux and database

Amin Ziaei

Steps to install PostgreSQL 13 on Ubuntu 22.04 LTS

```
sudo apt update && sudo apt upgrade
```

```
curl https://www.postgresql.org/media/keys/ACCC4CF8.asc | gpg --dearmor | sudo tee  
/etc/apt/trusted.gpg.d/apt.postgresql.org.gpg >/dev/null
```

```
echo "deb [arch=amd64] http://apt.postgresql.org/pub/repos/apt/ jammy-pgdg main" | sudo tee  
/etc/apt/sources.list.d/postgresql.list
```

```
sudo apt update
```

```
sudo apt install postgresql-13
```

Set Password for Admin User

```
sudo passwd postgres
```

```
sudo su - postgres
```

```
psql
```

```
CREATE DATABASE mtr;
```

```
CREATE USER amin WITH ENCRYPTED PASSWORD 'sina1384';
```

```
GRANT ALL PRIVILEGES ON DATABASE mtr to amin;
```

now database is ready

BashScript:

```
#!/bin/bash
```

```
# List of domains
```

```
domains=("google.com" "arvancloud.ir" "zoomit.ir" "kavenegar.com")
```

```

# Database connection details

db_host="127.0.0.1"

db_port="5432"

db_name="mtr"

db_user="amin"

db_password="sina1384"


# Create table if it doesn't exist

psql -h "$db_host" -p "$db_port" -d "$db_name" -U "$db_user" -c "
CREATE TABLE IF NOT EXISTS network (
    id SERIAL PRIMARY KEY,
    domain VARCHAR(255),
    jitter FLOAT,
    rtt FLOAT,
    loss FLOAT
);"


# Process each domain

for domain_name in "${domains[@]}"
do

    # Run the modified script and capture the output in a variable

    command=($(mtr -c 2 --csv -o MAL "$domain_name" | tail -1 | awk -F ',' '{print $4, $7, $8, $9,
$3}'))


    # Insert the output into the database

    sudo PGPASSWORD="sina1384" psql -h "$db_host" -p "$db_port" -d "$db_name" -U
"$db_user" -c "INSERT INTO network (domain, jitter, rtt, loss) VALUES ('${command[0]}',
${command[1]}, ${command[2]}, ${command[3]});" > /dev/null 2>&1

done

```

در این اسکریپت، از چندین کامند استفاده شده است. این کامندها عبارتند از:

1. `psql`: این کامند برای اتصال به پایگاه داده PostgreSQL استفاده می‌شود. با استفاده از پارامترهای `-h` (میزبان)، `-p` (پورت)، `-d` (نام دیتابیس)، `-U` (نام کاربری) و `-c` دستور SQL، به پایگاه داده متصل شده و دستور SQL را اجرا می‌کند.
- 2.
3. `mtr`: این کامند برای اندازه‌گیری و نمایش اطلاعات مسیریابی شبکه استفاده می‌شود. با استفاده از پارامترهای `-c` تعداد بسته‌های ارسالی، `--csv` فرمت خروجی به صورت CSV، `-o` (نوع خروجی) و نام دامنه، اطلاعات شبکه را برای دامنه مورد نظر دریافت می‌کند.
4. `tail`: این کامند برای خواندن خطوط آخر یک فایل استفاده می‌شود. با استفاده از پارامتر `-1`، آخرین خط را خوانده و نمایش می‌دهد.
5. `awk`: این کامند برای پردازش و استخراج داده‌ها از یک فایل یا خروجی دستورات دیگر استفاده می‌شود. با استفاده از پارامترهای `-F` (جداکننده) و نام دامنه، داده‌های مورد نظر را استخراج می‌کند.
6. `sudo`: این کامند برای اجرای دستورات با دسترسی ادمین (superuser) استفاده می‌شود.
7. `PGPASSWORD`: این متغیر محیطی برای تعیین رمز عبور پایگاه داده PostgreSQL استفاده می‌شود. با استفاده از این متغیر محیطی، می‌توان رمز عبور را به صورت اتوماتیک در دستور `psql` قرار داد.
8. `>/dev/null 2>&1`: این بخش از کد برای تغییر مسیر خروجی دستورات `psql` به فایل `/dev/null` استفاده می‌شود. این کار باعث می‌شود خروجی دستورات به صفر تغییر یابد و در خروجی نمایش داده نشود.

با استفاده از این کامندها، اسکریپت می‌تواند اطلاعات شبکه برای هر دامنه را دریافت کرده و در پایگاه داده ذخیره کند.

برای اجرای اسکریپت تحت مدیریت **systemd** نیاز به ایجاد فایل کانفیگ به شرح زیر می‌باشیم.

```
sudo vi /etc/systemd/system/network-mtr.service
```

```
[Unit]
```

```
Description=Script Service
```

```
After=network.target
```

```
[Service]
```

```
ExecStart=/bin/bash /home/amin/Downloads/test/rc2-2.sh
```

```
WorkingDirectory=/home/amin/Downloads/test/
```

```
Restart=always
```

```
RestartSec=10
```

```
StandardOutput=null
```

```
[Install]
```

```
WantedBy=default.target
```

این بخش از یونیت فایل سرویس (Unit file) مربوط به تنظیمات سرویس است. در این بخش، تنظیمات مربوط به سرویس اجرای اسکریپت تعیین می‌شود. بیایید به توضیح هر بخش بپردازیم:

```
[Unit]
```

Description: -توضیحی درباره سرویس و عملکرد آن را تعیین می‌کند. در اینجا، متن "Script Service" به عنوان توضیحی برای سرویس اسکریپت استفاده شده است.

After: -تعیین می‌کند که سرویس باید بعد از سرویس "network.target" اجرا شود. این بدان معناست که قبل از اجرای اسکریپت، سرویس شبکه باید در حالت آماده‌باش قرار گیرد.

```
[Service]
```

ExecStart: -دستوری که برای اجرای سرویس استفاده می‌شود. در اینجا، دستور `/bin/bash` `/home/amin/Downloads/test/rc2-2.sh` برای اجرای اسکریپت تعیین شده است.

WorkingDirectory: -مسیری که سرویس در آن اجرا می‌شود. در اینجا، مسیر `/home/amin/Downloads/test/` برای اجرای سرویس تعیین شده است.

Restart: -تعیین می‌کند که سرویس در صورت خطا یا توقف ناخواسته، باید مجدداً راه‌اندازی شود. در اینجا، مقدار `always` برای تعیین اینکه سرویس همیشه مجدداً راه‌اندازی شود، استفاده شده است.

RestartSec: -زمان تأخیر بین دو بار راه‌اندازی مجدد سرویس را تعیین می‌کند. در اینجا، مقدار `10` برای تعیین زمان تأخیر ۱۰ ثانیه بین دو بار راه‌اندازی مجدد استفاده شده است.

StandardOutput: -تعیین می‌کند که خروجی سرویس به کجا هدایت شود. در اینجا، مقدار `null` برای تعیین اینکه خروجی سرویس به جایی هدایت نشود، استفاده شده است.

[Install]

WantedBy: -تعیین می‌کند که سرویس باید در چه مرحله‌ای از بارگذاری سیستم عامل (`boot process`) فعال شود. در اینجا، مقدار `default.target` برای تعیین اینکه سرویس در مرحله پیش‌فرض بارگذاری سیستم عامل فعال شود، استفاده شده است.

```
sudo systemctl daemon-reload
```

```
sudo systemctl enable network-mtr.service
```

```
sudo systemctl start network-mtr.service
```

برای اعمال تغییرات در دسترسی به دیتابیس

1. ویرایش فایل کانفیگ PostgreSQL. در مسیر `/etc/postgresql//main/pg_hba.conf` قرار دارد. با استفاده از ویرایشگر متنی آن را باز می‌کنیم

```
sudo nano /etc/postgresql/13/main/pg_hba.conf
```

2. پیدا کردن بخشی از فایل که شبیه به زیر است:

#	TYPE	DATABASE	USER	ADDRESS	METHOD
---	------	----------	------	---------	--------

3. در خطوط زیر، دسترسی مورد نظر را اضافه کنید:

host	mtr	amin	192.168.100.1/32	md5
------	-----	------	------------------	-----

در اینجا:

`mtr` - نام دیتابیس است که می‌خواهید دسترسی را به آن اضافه کنید.

`amin` - نام کاربری است که می‌خواهید دسترسی را به آن اضافه کنید.

`192.168.100.1/32` - آی‌پی مورد نظر است.

`md5` - نوع روش احراز هویت است. در این مثال، احراز هویت بر اساس رمز عبور (md5) استفاده می‌شود. شما می‌توانید نوع دیگری را بر اساس نیاز خود انتخاب کنید.

برای اعمال تغییرات نیازمند reload سرویس پستگرس می‌باشیم.

```
sudo systemctl reload postgresql
```

یا

```
su - postgres
```

```
/usr/bin/pg_ctl reload
```