**Tugas Day 9**

Instalasi & Konfigurasi Nginx & PostgreSQL

pada sistem operasi CentOS

**Instalasi Nginx**

Untuk memulai proses instalasi web server nginx, terlebih dahulu dilakukan aktivasi terhadap repository **epel**. Barulah kemudian kita dapat melakukan instalasi dengan perintah sebagai berikut

|  |
| --- |
| $ sudo yum install epel-release -y  $ sudo yum install nginx -y |

**Konfigurasi Nginx**

Secara default, pengaturan file konfigurasi nginx dapat dilakukan pada file **/etc/nginx/nginx.conf**. Akan tetapi, di sini kita akan menerapkan konvensi **sites-available** dan **sites-enabled** kembali sebagaimana yang telah kita lakukan pada konfigurasi **apache** sebelumnya. Hal ini ditujukan agar pengelolaan *site* dapat lebih rapi dan terorganisir.

|  |
| --- |
| $ cd /etc/nginx  $ sudo mkdir sites-available  $ sudo mkdir sites-enabled |

Selanjutnya sebagaimana ketentuan yang telah disediakan, kita diminta untuk menjalankan layanan **nginx** pada TCP port 8080 mengingat layanan **apache** masih berjalan pada TCP port 80. Dengan demikian, akan diperlukan sedikit perubahan pada **nginx.conf** sedemikian sehingga kedua layanan ini tidak bentrok satu sama lain. Adapun berikut ini perubahan yang dilakukan

|  |
| --- |
| $ sudo vim nginx.conf |
| user nginx;  worker\_processes auto;  error\_log /var/log/nginx/error.log;  pid /run/nginx.pid;  # Load dynamic modules. See /usr/share/doc/nginx/README.dynamic.  include /usr/share/nginx/modules/\*.conf;  events {  worker\_connections 1024;  }  http {  log\_format main '$remote\_addr - $remote\_user [$time\_local] "$request" '  '$status $body\_bytes\_sent "$http\_referer" '  '"$http\_user\_agent" "$http\_x\_forwarded\_for"';  access\_log /var/log/nginx/access.log main;  sendfile on;  tcp\_nopush on;  tcp\_nodelay on;  keepalive\_timeout 65;  types\_hash\_max\_size 2048;  include /etc/nginx/mime.types;  default\_type application/octet-stream;  include /etc/nginx/conf.d/\*.conf;  **include /etc/nginx/sites-enabled/\*;**  server {  listen **8080** default\_server;  listen [::]:**8080** default\_server;  server\_name \_;  root **/var/www/html**;  # Load configuration files for the default server block.  include /etc/nginx/default.d/\*.conf;  location / {  }  error\_page 404 /404.html;  location = /40x.html {  }  error\_page 500 502 503 504 /50x.html;  location = /50x.html {  }  }  } |

Kali ini, kita akan fokus pada konfigurasi **localhost (default site)** sehingga dapat berjalan sebagaimana mestinya. Di sini dapat dilihat beberapa perubahan yang dilakukan, di antaranya:

1. Penambahan line **include /etc/nginx/sites-enabled/\*;**

Penambahan ini dimaksudkan agar nginx dapat memuat konfigurasi *virtual host* yang nantinya akan kita masukkan ke dalam *directory* ***sites-enabled*** dan ***sites-available***

1. Perubahan line **listen 80** menjadi **listen 8080**

Perubahan ini dilakukan agar layanan **nginx** akan menjalankan web server **localhost** pada port 8080sehingga tidak terjadi konflik dengan localhost yang dijalankan oleh layanan **apache**.

1. Perubahan line root **/usr/share/nginx/html** menjadi **/var/www/html**

Perubahan ini dilakukan untuk memindahkan *root* *directory* dari server\_name **localhost** sehingga kita tidak lagi melihat *default homepage* milik **nginx**

**Inisialisasi VirtualHost**

Sama halnya pada arsitektur apache, nginx juga mengenal model *virtual host* untuk mengatur sebuah *site*. Kendati demikian terdapat perbedaan *syntax* dalam pendeklarasiannya. Berikut kita akan membuat *virtual host* untuk site dengan domain **mfaqih.com**

1. **Skenario standar**

Pada skenario ini, kita hanya akan mengarahkan site **mfaqih.com** dengan port 8080 agar memuat HTML file yang terdapat di dalam *root directory* **/var/www/html/mfaqih.com/public\_html.** Berikut implementasinya

|  |
| --- |
| $ sudo vim sites-available/mfaqih.com |
| server {  listen 8080;  server\_name mfaqih.com;  root /var/www/html/mfaqih.com/public\_html;  index index.html;    location / {  try\_files $uri $uri/ =404;  }  } |

Sebagaimana bisa kita lihat, di sini kita mendeklarasikan sebuah **server block** yang memuat beberapa opsi pengaturan yang dapat diuraikan sebagai berikut:

1. Listen 8080

Sama halnya dengan penjelasan sebelumnya, argumen ini ditambahkan agar site berjalan pada port 8080

1. Server\_name mfaqih.com

Argumen server\_name berfungsi sebagai identifier bagi web server agar client dapat menjalankan *site* dengan nama domain yang diberikan

1. Index index.html

Argumen ini diberikan agar web server dapat mengenali file sehingga dapat diproses oleh HTTP server, pada kasus ini web server hanya akan memproses file dengan ekstensi **html**, selebihnya file tidak akan diproses dan akan dikembalikan sebagai raw file

1. Location / { }

Blok ini merepresentasikan endpoint dari site

1. Try\_files = 404

Argumen ini merupakan argumen handler yang diberikan untuk mengembalikan response 404 ketika endpoint dari site tidak ditemukan

1. **Skenario Reverse Proxy**

Pada skenario ini, sebut saja kita memiliki sebuah site yang berjalan pada suatu port. Namun kita ingin mengarahkan domain **mfaqih.com** dengan port 8080 agar dapat menggunakan layanan site tersebut. Berikut implementasinya

|  |
| --- |
| $ sudo vim sites-available/mfaqih.com |
| server {  listen 8080;  server\_name mfaqih.com;  location / {  proxy\_pass http://mfaqih.com;  proxy\_set\_header Host $host;  proxy\_set\_header X-Real-IP $remote\_addr;  proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;  proxy\_set\_header X-Forwarded-Proto $scheme;  }  } |

Berbeda halnya dengan sebelumnya, di sini kita tidak mendefinisikan argumen ***root*** dan ***index*** dikarenakan kita hanya meminjam layanan yang telah tersedia. Sebagai gantinya kita definisikan argumen **proxy\_pass** ***target\_url*** untuk men-forward traffic dari **mfaqih.com:8080 ke mfaqih.com**

Setelah semua konfigurasi telah diatur dilakukan perintah sebagai berikut untuk mengaktifkan *virtual host* dan layanan *nginx*

|  |
| --- |
| $ sudo ln -s /etc/nginx/sites-available/mfaqih.com ../sites-enabled  $ sudo systemctl enable nginx  $ sudo systemctl restart nginx |

**Inisialisasi site directory**

Apabila kita belum menginisialisasi site seperti yang sebelumnya kita lakukan saat instalasi **apache**, kita dapat membuatnya dengan perintah sebagai berikut

|  |
| --- |
| $ sudo mkdir -p /var/www/html/**mfaqih.com**/public\_html  $ sudo chown -R $USER:$USER /var/www/html/**mfaqih.com**/public\_html  $ sudo chmod -R 755 /var/www/html/**mfaqih.com**/public\_html |

Kemudian kita buat sebuah HTML File untuk menguji site

|  |
| --- |
| $ cd /var/www/html/**mfaqih.com**/public\_html  $ vim index.html |

|  |
| --- |
| <!DOCTYPE html> <**html**> <**head**>  <**title**>Testing</**title**> </**head**> <**body**>  <**h1**>M Faqih Jihan Insani</**h1**> </**body**> </**html**> |

Sebelum melakukan pengujian pada client, akan ada baiknya untuk mengatur *firewall rule* untuk mengizinkan akses port 8080 dengan perintah sebagai berikut

|  |
| --- |
| $ firewall-cmd --permanent --add-port=8080/tcp  $ firewall-cmd --reload |

Barulah saatnya melakukan pengujian web server. Hasilnya adalah sebagai berikut



**Instalasi PostgreSQL**

Sebelum melakukan instalasi **postgresql**, terlebih dahulu kita lakukan modifikasi pada file **/etc/yum.repos.d/CentOS-Base.repo** dengan menambahkan line exclude=postgresql\*pada section **base** dan **updates** sebagai berikut

|  |
| --- |
| $ sudo vim /etc/yum.repos.d/CentOS-Base.repo |
| ...  [base]  name=CentOS-$releasever - Base  mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=os&infra=$infra  #baseurl=http://mirror.centos.org/centos/$releasever/os/$basearch/  gpgcheck=1  gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7  exclude=postgresql\*  #released updates  [updates]  name=CentOS-$releasever - Updates  mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=updates&infra=$infra  #baseurl=http://mirror.centos.org/centos/$releasever/updates/$basearch/  gpgcheck=1  gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7  exclude=postgresql\*  ... |

Barulah kemudian dilakukan penambahan repositorydan instalasi **postgresql**. Dalam hal ini kita akan menginstall **postgresql-12** yang merupakan *latest release* dari ***postgresql***

|  |
| --- |
| $ sudo yum install https://download.postgresql.org/pub/repos/yum/reporpms/EL-7-x86\_64/pgdg-redhat-repo-latest.noarch.rpm  $ sudo yum install postgresql12-server |

**Konfigurasi PostgreSQL**

Untuk dapat menggunakan layanan database dari **postgreSQL**, pertama-tama kita lakukan inisialisasi database dengan perintah sebagai berikut

|  |
| --- |
| $ sudo /usr/pgsql-12/bin/postgresql-12-setup initdb |

Kemudian lakukan aktivasi layanan **postgresql** sebagai berikut

|  |
| --- |
| $ sudo systemctl enable postgresql-12  $ sudo systemctl start postgresql-12 |

Hasilnya kita dapat melihat bahwa layanan **postgresql** akan melakukan listening pada port **5432**

|  |
| --- |
| $ netstat -tulpn |
|  |

Dari sini, kita bisa juga melakukan pengujian terhadap **psql shell** untuk menampilkan list database sebagai berikut

|  |
| --- |
| $ sudo -u postgres psql  # \l |
|  |

Selain itu, apabila kita membutuhkan manajemen database postgresql berbasis web, dapat dilakukan instalasi **pgadmin** sebagai berikut

|  |
| --- |
| $ sudo rpm -i https://ftp.postgresql.org/pub/pgadmin/pgadmin4/yum/pgadmin4-redhat-repo-1-1.noarch.rpm  $ sudo yum install pgadmin4-web |

Perlu diketahui bahwa **pgadmin** dijalankan dengan web server apache. Untuk itu, terlebih dahulu lakukan aktivasi layanan **apache** apabila belum aktif

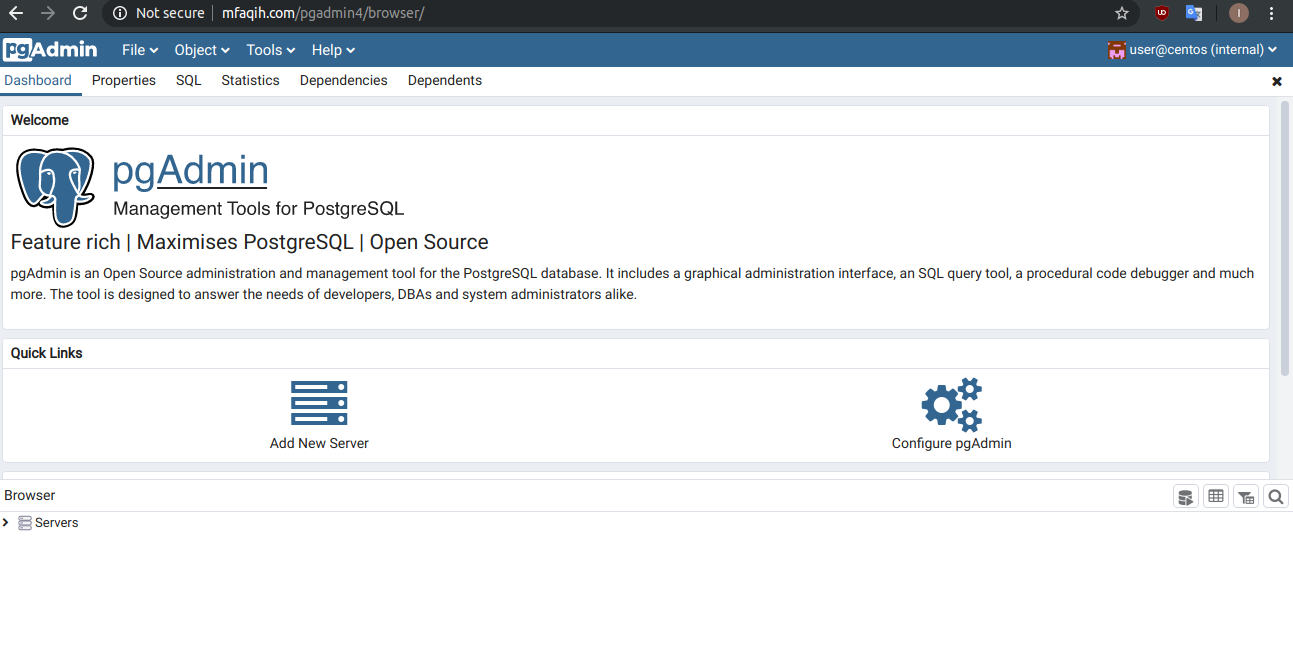
|  |
| --- |
| $ sudo systemctl start httpd |

Selanjutnya aktifkan panel web admin dengan perintah sebagai berikut

|  |
| --- |
| $ sudo /usr/pgadmin4/bin/setup-web.sh |

Di sini, kita akan disuguhkan dengan prompt yang meminta email dan password untuk mengakses Admin Panel. Masukkan data sesuai dengan kebutuhan dan web admin dapat diakses pada **http://nama-domain/pgadmin4**

****

****