

Project Sesi 4 Bootcamp DevOps Engineer Digital Skola

A. Lakukan langkah berikut untuk manajemen user dan group

- Buat tiga pengguna baru bernama **user1**, **user2**, dan **user3**.
- Buat grup baru bernama **devteam**.
- Masukkan **user1** & **user2** ke **devteam**
- Buat direktori baru **/home/shared** dengan nama **sesi_3**
- Set kepemilikan folder **sesi_3** ke grup **devteam**.
- Set *permission* akses sehingga hanya grup **devteam** yang dapat membaca, menulis, dan eksekusi di dalam folder **project**, sedangkan pengguna lain tidak memiliki akses.

Jawaban berupa script yang kalian gunakan untuk step diatas serta screenshot untuk :

- Masuklah sebagai user 1 apakah bisa membuat file txt di dalam **sesi_3**
- Masuklah sebagai user 2 apakah bisa membuat file txt di dalam **sesi_3**
- Masuklah sebagai user 3 apakah bisa membuat file txt di dalam **sesi_3**

1. Membuat Tiga Pengguna Baru:

- Gunakan perintah **useradd** untuk membuat pengguna baru.

```
[adminboss@MyLinuxCentOS ~]$ sudo useradd user1
[sudo] password for adminboss:
[adminboss@MyLinuxCentOS ~]$ sudo useradd user2
[adminboss@MyLinuxCentOS ~]$ sudo useradd user3
```

- Perintah ini akan membuat tiga pengguna baru dengan nama **user1**, **user2**, dan **user3**.

2. Membuat Grup Baru:

- Gunakan perintah **groupadd** untuk membuat grup baru.

```
[adminboss@MyLinuxCentOS ~]$ sudo groupadd devteam
```

- Perintah ini akan membuat grup baru dengan nama **devteam**.

3. Menambahkan Pengguna ke Grup:

- Gunakan perintah **usermod** untuk menambahkan **user1** dan **user2** ke grup **devteam**.

```
[adminboss@MyLinuxCentOS ~]$ sudo usermod -aG devteam user1
[adminboss@MyLinuxCentOS ~]$ sudo usermod -aG devteam user2
```

- Opsi **-aG** menambahkan pengguna ke grup tanpa menghapus keanggotaan grup lainnya.

4. Membuat Direktori Baru:

- Gunakan perintah **mkdir** untuk membuat direktori baru

```
[adminboss@MyLinuxCentOS ~]$ sudo mkdir -p /home/shared/sesi_3
```

- Perintah ini akan membuat direktori **sesi_3** di dalam **/home/shared**. Opsi **-p** pada perintah **mkdir** berfungsi untuk sekaligus memastikan parent direktori dari direktori **sesi_3** yaitu **/home/shared/** tersedia. Dalam case ini, sebelumnya direktori **home** sudah ada namun direktori **shared** belum ada. Sehingga perintah **mkdir -p** memastikan direktori **sesi_3** yang dibuat beserta semua parent direktori-nya tersedia.

5. Mengubah Kepemilikan Direktori:

- Gunakan perintah **chown** untuk mengubah kepemilikan direktori ke grup **devteam**

```
[adminboss@MyLinuxCentOS ~]$ sudo chown :devteam /home/shared/sesi_3
```

- Perintah ini mengubah grup pemilik direktori **sesi_3** menjadi **devteam**.

6. Mengatur Permission Akses:

- Gunakan perintah **chmod** untuk mengatur permission sehingga hanya grup **devteam** yang dapat membaca, menulis, dan mengeksekusi.

```
[adminboss@MyLinuxCentOS ~]$ sudo chmod 770 /home/shared/sesi_3
```

- Permission 770 berarti pemilik dan grup memiliki hak baca (r), tulis (w), dan eksekusi (x), sedangkan pengguna lain tidak memiliki akses.

7. Verifikasi Akses:

- Masuk sebagai **user1** dan coba buat file di dalam **sesi_3**.

```
[adminboss@MyLinuxCentOS ~]$ sudo su - user1
[user1@MyLinuxCentOS ~]$ touch /home/shared/sesi_3/test_user1.txt
```

- **user1** berhasil membuat file **test_user1.txt** di dalam **sesi_3** dengan perintah **touch** yang ditunjukkan tidak adanya pesan *permission denied*.

- Masuk sebagai **user2** dan coba buat file di dalam **sesi_3**.

```
[user1@MyLinuxCentOS ~]$ sudo su - user2
[sudo] password for user1:
user1 is not in the sudoers file. This incident will be reported.
[user1@MyLinuxCentOS ~]$ exit
logout
```

- Saat mencoba masuk langsung ke **user2** dari **user1** ternyata tidak bisa dilakukan karena **user1** yang baru dibuat belum dimasukkan sebagai sudoer. Kita masuk melalui **adminboss** saja yang sudah menjadi sudoer.

```
[adminboss@MyLinuxCentOS ~]$ sudo su - user2
[user2@MyLinuxCentOS ~]$ touch /home/shared/sesi_3/test_user2.txt
[user2@MyLinuxCentOS ~]$ ls /home/shared/sesi_3
test_user1.txt  test_user2.txt
```

- **user2** berhasil membuat file test_user2.txt dengan perintah **touch**. Lalu dengan perintah **ls**, **user1** dan **user2** dapat membaca apa saja yang ada di dalam direktori **sesi_3**.
- Masuk sebagai **user3** dan coba buat file di dalam **sesi_3**

```
[adminboss@MyLinuxCentOS ~]$ sudo su - user3
```

```
[user3@MyLinuxCentOS ~]$ touch /home/shared/sesi_3/test_user3.txt
touch: cannot touch '/home/shared/sesi_3/test_user3.txt': Permission denied
```

```
[user3@MyLinuxCentOS ~]$ ls /home/shared/sesi_3
ls: cannot open directory '/home/shared/sesi_3': Permission denied
```

- **user3** tidak berhasil membuat file test_user3.txt dan tidak bisa membaca apa yang ada dalam direktori **sesi_3** karena **user3** termasuk dalam *other user* yang tidak diberi *permission* apapun (0).

B. Lakukan konfigurasi dan instalasi NGINX berikut ini, lakukan verifikasi menggunakan curl untuk memastikan servis berjalan. Lalu lakukan verifikasi service nginx dengan : netstat -tuln dan ps aux | grep nginx

Jawaban berupa screenshot hasil **curl**, **netstat**, dan **ps aux**, lalu jelaskan apa yang kamu tahu dari hasil

1. Instalasi Nginx

```
meno@MyLinuxUbuntu:~$ sudo su
root@MyLinuxUbuntu:/home/meno# apt update
```

sudo su: Masuk ke mode superuser untuk menjalankan perintah dengan hak akses root.

apt update: Memperbarui daftar paket yang tersedia di repositori.

```
root@MyLinuxUbuntu:/home/meno# apt install -y nginx
```

apt install -y nginx: Menginstal Nginx. Opsi -y secara otomatis menjawab "yes" untuk semua pertanyaan selama instalasi.

```
root@MyLinuxUbuntu:/home/meno# systemctl start nginx
root@MyLinuxUbuntu:/home/meno# systemctl enable nginx
```

systemctl start nginx: Memulai layanan Nginx.

systemctl enable nginx: Mengaktifkan Nginx untuk memulai secara otomatis saat sistem boot.

```
root@MyLinuxUbuntu:/home/meno# systemctl status nginx
```

systemctl status nginx: Memeriksa status layanan Nginx untuk memastikan bahwa layanan berjalan dengan baik.

Output dari command tersebut adalah sebagai berikut

```
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2025-02-17 17:03:43 WIB; 1min 39s ago
     Docs: man:nginx(8)
    Main PID: 6359 (nginx)
      Tasks: 3 (limit: 2264)
     Memory: 6.2M
        CPU: 44ms
    CGroup: /system.slice/nginx.service
            └─6359 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
              └─6361 "nginx: worker process"
                └─6362 "nginx: worker process"

Feb 17 17:03:43 MyLinuxUbuntu systemd[1]: Starting A high performance web server and a reverse proxy server...
Feb 17 17:03:43 MyLinuxUbuntu systemd[1]: Started A high performance web server and a reverse proxy server.
```

2. Membuat File HTML

Setelah Nginx terinstal, kita perlu membuat file HTML yang akan disajikan oleh Nginx:

```
root@MyLinuxUbuntu:/home/meno# sh -c 'echo "<!DOCTYPE html> \
<html> \
<head> \
<title>Welcome to Nginx!</title> \
</head> \
<body> \
<h1>Success! Nginx is serving this HTML page.</h1> \
</body> \
</html>" > /var/www/html/index.html'
```

sh -c: Ini memberi tahu shell untuk menjalankan string perintah yang mengikutinya.

echo : Perintah ini mengeluarkan teks yang mengikutinya.

\: Backslash digunakan untuk melanjutkan perintah ke baris berikutnya, membuatnya lebih mudah dibaca dan diketik.

> **/var/www/html/index.html**: Ini mengarahkan output dari perintah echo ke file /var/www/html/index.html, membuat atau menimpa file dengan konten yang ditentukan.

Untuk memverifikasi file html yang telah dibuat dapat menggunakan perintah **cat /var/www/html/index.html**

```
root@MyLinuxUbuntu:/home/meno# cat /var/www/html/index.html
<!DOCTYPE html> <html> <head> <title>Welcome to Nginx!</title> </head> <body> <h1>Success! Nginx is serving this HTML page.</h1> </body> </html>
```

3. Memberi Akses Permission pada File

Setelah membuat file HTML, kita perlu mengatur izin file tersebut:

```
root@MyLinuxUbuntu:/home/meno# chmod 644 /var/www/html/index.html
```

chmod 644: Mengatur izin file sehingga pemilik file dapat membaca dan menulis, sedangkan pengguna lain hanya dapat membaca.

4. File Konfigurasi Nginx

Selanjutnya, kita perlu mengkonfigurasi Nginx untuk melayani file HTML yang baru saja kita buat:

```
root@MyLinuxUbuntu:/home/meno# echo 'server {
    listen 80 default_server;
    listen [::]:80 default_server;

    root /var/www/html;
    index index.html;

    server_name _;

    location / {
        try_files $uri $uri/ =404;
    }
}' | sudo tee /etc/nginx/sites-available/default
```

```
server {
    listen 80 default_server;
    listen [::]:80 default_server;

    root /var/www/html;
    index index.html;

    server_name _;

    location / {
        try_files $uri $uri/ =404;
    }
}
```

Konfigurasi ini mendefinisikan server Nginx yang mendengarkan (*listen*) pada port 80, dengan root direktori di `/var/www/html` dan file indeks `index.html`.

server_name _; Menentukan bahwa server ini akan merespons permintaan untuk semua nama domain.

location /; Blok ini menentukan bagaimana Nginx harus menangani permintaan ke root URL.

5. Reload Nginx

Setelah mengubah konfigurasi, kita perlu memeriksa sintaks konfigurasi dan memuat ulang Nginx:

```
root@MyLinuxUbuntu:/home/meno# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful

root@MyLinuxUbuntu:/home/meno# systemctl reload nginx
```

nginx -t: Memeriksa sintaks konfigurasi Nginx untuk memastikan tidak ada kesalahan.

systemctl reload nginx: Memuat ulang konfigurasi Nginx tanpa menghentikan layanan.

6. Verifikasi Nginx

Untuk memverifikasi bahwa Nginx berjalan dengan benar dan melayani halaman HTML yang dibuat, kita dapat menggunakan **curl**:

```
root@MyLinuxUbuntu:/home/meno# curl -v http://localhost
Command 'curl' not found, but can be installed with:
snap install curl # version 8.12.1, or
apt install curl # version 7.81.0-1ubuntu1.18
See 'snap info curl' for additional versions.
root@MyLinuxUbuntu:/home/meno# apt install curl
```

```
root@MyLinuxUbuntu:/home/meno# curl -v http://localhost
* Trying 127.0.0.1:80...
* Connected to localhost (127.0.0.1) port 80 (#0)
> GET / HTTP/1.1
> Host: localhost
> User-Agent: curl/7.81.0
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Server: nginx/1.18.0 (Ubuntu)
< Date: Mon, 17 Feb 2025 10:53:58 GMT
< Content-Type: text/html
< Content-Length: 145
< Last-Modified: Mon, 17 Feb 2025 10:36:01 GMT
< Connection: keep-alive
< ETag: "67b31111-91"
< Accept-Ranges: bytes
<
<!DOCTYPE html> <html> <head> <title>Welcome to Nginx!</title> </head> <body> <h1>Success! Nginx is serving this HTML page.</h1> </body> </html>
* Connection #0 to host localhost left intact
```

Perintah ini akan mengirim permintaan HTTP ke server lokal atau *localhost* dan menampilkan respons, termasuk header dan konten halaman:

- Pastikan respons HTTP adalah 200 OK.
- Konten HTML yang ditampilkan sesuai dengan yang dibuat di file *index.html*.

7. Install netstat

Untuk memeriksa port yang sedang mendengarkan dan proses yang berjalan, kita perlu menginstal *net-tools* yang mencakup *netstat*:

```
root@MyLinuxUbuntu:/home/meno# apt-get install net-tools
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  net-tools
0 upgraded, 1 newly installed, 0 to remove and 520 not upgraded.
Need to get 204 kB of archives.
After this operation, 819 kB of additional disk space will be used.
Get:1 http://id.archive.ubuntu.com/ubuntu jammy/main amd64 net-tools amd64 1.60+git20181103.0eebece-1ubuntu5 [204 kB]
Fetched 204 kB in 4s (46,2 kB/s)
Selecting previously unselected package net-tools.
(Reading database ... 197957 files and directories currently installed.)
Preparing to unpack .../net-tools_1.60+git20181103.0eebece-1ubuntu5_amd64.deb ...
Unpacking net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Setting up net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Processing triggers for man-db (2.10.2-1) ...
```

```
root@MyLinuxUbuntu:/home/meno# netstat -tuln
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN
tcp6       0      0 :::80                   :::*                    LISTEN
tcp6       0      0 :::1:631                 :::*                    LISTEN
udp        0      0 127.0.0.53:53           0.0.0.0:*               *
udp        0      0 0.0.0.0:42168           0.0.0.0:*               *
udp        0      0 0.0.0.0:5353            0.0.0.0:*               *
udp        0      0 0.0.0.0:631             0.0.0.0:*               *
udp6       0      0 :::53222                :::*                    *
udp6       0      0 :::5353                  :::*                    *
```

netstat -tuln: Menampilkan port yang sedang mendengarkan dan statusnya. Memastikan Nginx mendengarkan pada port yang benar (80 atau 443).

Penjelasan umum dari setiap kolom dari output yang ditampilkan adalah:

- **Proto:** Protokol yang digunakan (TCP atau UDP).
- **Recv-Q** dan **Send-Q:** Jumlah data dalam antrian penerimaan dan pengiriman (biasanya 0 jika tidak ada masalah).

- **Local Address:** Alamat lokal (IP dan port) yang sedang mendengarkan.
- **Foreign Address:** Alamat remote (biasanya 0.0.0.0:* atau :::* untuk koneksi yang sedang mendengarkan).
- **State:** Status koneksi (misalnya, LISTEN berarti port sedang mendengarkan koneksi masuk).

Dapat dilihat bahwa terdapat dua koneksi pada port 80 yaitu **tcp 0 0 0.0.0.0:80 0.0.0.0:* LISTEN** dan **tcp6 0 0 :::80 :::* LISTEN**. Ini menunjukkan bahwa web server berjalan dan siap menerima koneksi HTTP baik pada koneksi jaringan IPv4 maupun IPv6.

```
root@MyLinuxUbuntu:/home/meno# ps aux | grep nginx
root        6359  0.0  0.5 55340 11956 ?        S    17:03   0:00 nginx: master process /usr/sbin/nginx
-g daemon on; master_process on;
www-data    7511  0.0  0.2 55968 5648 ?        S    17:51   0:00 nginx: worker process
www-data    7512  0.0  0.2 55968 5648 ?        S    17:51   0:00 nginx: worker process
root        8203  0.0  0.1 17868 2424 pts/1    S+   18:01   0:00 grep --color=auto nginx
```

- **ps aux | grep nginx:** Menampilkan proses Nginx yang sedang berjalan. Ini memastikan bahwa proses master dan worker Nginx aktif. Pastikan ada proses **nginx: master process** dan **nginx: worker process** yang sedang berjalan.

Penjelasan umum dari setiap kolom dari output yang ditampilkan yang mana tidak terlihat karena tabel output yang diambil adalah bagian dengan keyword nginx (**grep nginx**) ini adalah:

- **USER:** Pengguna yang menjalankan proses.
- **PID:** Process ID (ID proses).
- **%CPU dan %MEM:** Persentase penggunaan CPU dan memori oleh proses.
- **VSZ:** Virtual Memory Size (ukuran memori virtual yang digunakan).
- **RSS:** Resident Set Size (ukuran memori fisik yang digunakan).
- **TTY:** Terminal yang terkait dengan proses (? berarti tidak terkait dengan terminal).
- **STAT:** Status proses (misalnya, S untuk sleeping, + untuk proses foreground).
- **START:** Waktu proses dimulai.
- **TIME:** Waktu CPU yang digunakan oleh proses.
- **COMMAND:** Perintah atau proses yang dijalankan

Analisis Output:

1. **root 6359 0.0 0.5 55340 11956 ? S 17:03 0:00 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;**
 - **Proses Master Nginx:**
 - Dijalankan oleh pengguna root.
 - PID: 6359.
 - Menggunakan sekitar 0.5% memori (11956 KB RSS).
 - Proses ini adalah **master process** dari Nginx, yang bertanggung jawab untuk mengontrol proses worker.

- Opsi -g daemon on; master_process on; menunjukkan bahwa Nginx berjalan sebagai daemon (layanan latar belakang) dan menggunakan mode master-worker.
2. **www-data 7511 0.0 0.2 55968 5648 ? S 17:51 0:00 nginx: worker process**
 - **Proses Worker Nginx:**
 - Dijalankan oleh pengguna www-data.
 - PID: 7511.
 - Menggunakan sekitar 0.2% memori (5648 KB RSS).
 - Proses ini adalah **worker process** yang menangani permintaan klien. Biasanya, Nginx memiliki beberapa worker process untuk menangani banyak koneksi secara bersamaan.
 3. **www-data 7512 0.0 0.2 55968 5648 ? S 17:51 0:00 nginx: worker process**
 - **Proses Worker Nginx Lainnya:**
 - Dijalankan oleh pengguna www-data.
 - PID: 7512.
 - Menggunakan sekitar 0.2% memori (5648 KB RSS).
 - Ini adalah worker process kedua yang juga menangani permintaan klien.
 4. **root 8203 0.0 0.1 17868 2424 pts/1 S+ 18:01 0:00 grep --color=auto nginx**
 - **Proses grep:**
 - Dijalankan oleh pengguna root.
 - PID: 8203.
 - Proses ini adalah perintah grep yang dijalankan untuk mencari proses Nginx. Ini bukan bagian dari Nginx itu sendiri sehingga dapat diabaikan.

Notes: Tugas A menggunakan CentOS sedangkan tugas B menggunakan Ubuntu. Alasannya saya baru mau belajar CentOS yang mana sama dengan RedHat. Saat ingin mengerjakan tugas B ternyata ada perbedaan command. Ditambah lagi CentOS di VirtualBox saya belum berjalan dengan mulus, masih sering crashed. Tugas A sudah terlanjur dibuat di CentOS. Semoga tidak masalah ya. Terima kasih.