

# LAPORAN *HANDS-ON* MEMBUAT VM DI CLOUDMATIKA

## OLEH CLOUD SORE KELOMPOK 6

Nama Kelompok:

1. Rachmad Rinaldie
2. Rahmat Muliadi
3. Rahman Sujatman
4. Reisa Noor Maulani
5. Reza

1. Hal pertama yang dilakukan adalah masuk ke dalam akun Cloudmatika melalui link <https://cloudmatika.com/servlet/Turbine/frm/single/> dengan menggunakan username dan password yang telah ditentukan :  
username : cloudsore | password : Digitalent:2018
2. Terdapat 3 buah tab "Home", "Cloud", dan "Help & Support"
3. Untuk membuat server baru, pindah ke Tab "Cloud" lalu klik "New Server"
4. Setelah itu dilakukan konfigurasi untuk VM yang akan kita buat.
5. Untuk nama menggunakan "dbcloudsore6", OS-nya dipilih "Free Linux", jumlah IP yang kita miliki di set ke "1" dan preset diubah ke "Minimal", dan mencentang opsi "I will manage the OS kernel" setelah itu klik "Next" pada bagian bawah.

6. Lalu di bagian selanjutnya memilih OS Linux mana yang akan digunakan. Kali ini kita pilih "Ubuntu 16.04", kemudian klik "Next" lagi dan kembali ke tab "Cloud", nanti akan terlihat status pembuatan VM kita hingga "running"
7. Setelah selesai, klik nama server yang telah dibuat lalu ubah passwordnya.
8. Masuk ke server tersebut via SSH.  
**ssh 103.101.224.204**
9. login sebagai root, dengan menggunakan password yang sudah kita ubah

10. Lalu kita update repositorynya dengan menggunakan perintah :  
**apt-get update**
11. Lanjutkan dengan melakukan instalasi webserver Nginx menggunakan perintah :  
**apt-get install nginx**
12. Untuk memeriksa apakah webserver kita sudah berjalan atau tidak pada VM yang telah kita buat, cukup langsung buka IP Public VM kita di **103.101.224.204** melalui web browser.
13. Ketika membuka halaman IP Public akan langsung disambut dengan tulisan "Welcome to Nginx!" ini artinya instalasi webserver Nginx di VM telah berhasil dan dapat dijalankan.
14. Langkah selanjutnya adalah menginstall MySQL dengan menggunakan perintah :  
**apt-get install mysql-server**  
kemudian kita akan diminta untuk membuat password yang nantinya akan digunakan untuk membuka MySQL.
15. Setelah MySQL berhasil diinstall, selanjutnya adalah menginstall PHP untuk processing dengan menggunakan perintah :  
**apt-get install php-fpm php-mysql**
16. selanjutnya edit file php.ini. Untuk membukanya, gunakan perintah :  
**nano /etc/php/7.0/fpm/php.ini**

ubah baris dengan isi  
**;cgi.fix\_pathinfo=1**

menjadi

**cgi.fix\_pathinfo=0**  
**systemctl restart php7.0-fpm**

17. Kemudian, kita mengkonfigurasi Nginx untuk menggunakan PHP Processor dengan menggunakan perintah:  
**nano /etc/nginx/sites-available/default**

Isi awal:

```
server {  
    listen 80 default_server;  
    listen [::]:80 default_server;  
  
    root /var/www/html;  
    index index.html index.htm index.nginx-debian.html;  
  
    server_name _;  
  
    location / {
```

```

    try_files $uri $uri/ =404;
  }
}

```

ubah menjadi:

```

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

    root /var/www/html;
    index index.php index.html index.htm index.nginx-debian.html;

    server_name server_domain_or_IP;

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

    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/run/php/php7.0-fpm.sock;
    }

    location ~ /\.ht {
        deny all;
    }
}

nginx -t
systemctl reload nginx

```

18. Untuk tes pengkonfigurasiannya tersebut, kita membuat file PHP dengan perintah sebagai berikut:

```

nano /var/www/html/info.php
<?php
phpinfo();
akses http://public ip address/info.php
rm /var/www/html/info.php

```

19. Terakhir, kita menginstall phpMYAdmin yang berfungsi untuk mengelola database MySQL, yaitu dengan membuat perintah sebagai berikut:

```

apt-get install phpmyadmin
ln -s /usr/share/phpmyadmin /var/www/html
phpenmod mcrypt
systemctl restart php7.0-fpm

```

**akses [http://public ip adress/phpmyadmin](http://public_ip_address/phpmyadmin)**