

# Modul 2 - Automasi

## Dasar Terori

### Ansible

Ansible adalah sebuah provisioning tool yang dikembangkan oleh RedHat. Dimana kamu dapat mencatat setiap proses deployment ataupun konfigurasi yang biasa dilakukan berulang - ulang terhadap beberapa server. Misal saat pertama kali kita memasang Ubuntu Server di 10 mesin, maka kita akan melakukan apt-get update serta memasang beberapa komponen seperti PHP5 dan Apache2. Sebenarnya tidak akan menjadi masalah, bila kita hanya melakukan sedikit hal. Tapi bayangkan bila harus melakukan konfigurasi yang cukup kompleks dan dilakukan secara berulang - ulang ke 10 mesin tersebut.

### Pra - Instalasi

- Setup semua lxc menjadi autostart ketika vm dinyalakan

```
sysadmin@sas02:~$ sudo lxc-ls -f
NAME           STATE   AUTOSTART GROUPS IPV4        IPV6 UNPRIVILEGED
debian_php5.6  RUNNING 1       -      10.0.3.102 -      false
ubuntu_landing RUNNING 1       -      10.0.3.103 -      false
ubuntu_php7.4  RUNNING 1       -      10.0.3.101 -      false
sysadmin@sas02:~$ _
```

- Masuk ke LXC

```
sudo lxc-attach -n debian_php5.6
```

- Install openssh-server & python

```
sudo apt install openssh-server python
```

- Config ssh untuk enable root user

```
cd /etc/ssh
nano sshd_config

# setting config menjadi
PermitRootLogin yes
RSAAuthentication yes
```

```
GNU nano 2.7.4 File: sshd_config

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin yes
RSAAuthentication yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes
```

- Restart ssh service

```
sudo service sshd restart
```

- Setting password root

```
passwd
```

- Akses LXC melalui SSH

```
# jika masih di dalam lxc, silahkan di exit dulu
```

```
ssh root@lxc_php5.dev
```

```
sysadmin@sas02:~$ ssh root@lxc_php5.dev
The authenticity of host 'lxc_php5.dev (10.0.3.102)' can't be established.
ECDSA key fingerprint is SHA256:1F7e3y03+afSGbsKtq1uVx14hgdz9i0Y0kIEnHvFDRo.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'lxc_php5.dev,10.0.3.102' (ECDSA) to the list of known hosts.
root@lxc_php5.dev's password:
Linux debian_php5.6 5.4.0-89-generic #100-Ubuntu SMP Fri Sep 24 14:50:10 UTC 2021 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@debian_php5:~# _
```

- Keluar dari ssh

```
# bisa dengan menekan ctrl + D
# atau menulis command

exit
```

- Lakukan konfigurasi ini pada semua lxc

## Instalasi

- Pada VM ubuntu server 20.04, install ansible

```
sudo apt install ansible sshpass
```

## Getting started with Infrastructur as Code

### Cek Koneksi Semua LXC menggunakan Ansible

1. Buat folder simple-playbook pada ~/ansible/

```
mkdir -p ~/ansible/simple-playbook
```

2. Pindah ke folder simple-playbook

```
cd ~/ansible/simple-playbook
```

3. Buat file `hosts` sebagai vagrant inventory, dengan isi:

```
ubuntu_landing ansible_host=1xc_landing.dev ansible_ssh_user=root
ansible_become_pass=a
debian_php5 ansible_host=1xc_php5.dev ansible_ssh_user=root
ansible_become_pass=a
ubuntu_php7 ansible_host=1xc_php7.dev ansible_ssh_user=root
ansible_become_pass=a
```

Keterangan syntax:

```
nama_host ansible_host=[IP/DOMAIN] ansible_ssh_user=[UserName]
ansible_become_pass=[password user]
```

4. Berikutnya jalankan perintah:

```
ansible -i hosts -m ping all -k
```

```
sysadmin@sas02:~/ansible/simple-playbook$ ansible -i ./hosts -m ping all -k
SSH password:
[WARNING]: Platform linux on host debian_php5 is using the discovered Python interpreter at
/usr/bin/python, but future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more
information.
debian_php5 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
ubuntu_php7 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
ubuntu_landing | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
sysadmin@sas02:~/ansible/simple-playbook$ _
```

Keterangan:

1. Parameter -i digunakan untuk digunakan untuk mendeclare ansible inventory.

2. Parameter -m digunakan untuk declare module command
3. Parameter all digunakan untuk penanda ansible dijalankan di host mana. Parameter all bisa diganti dengan nama host.
4. Parameter -k digunakan untuk menanyakan password login ssh

## Menjalankan shell command menggunakan ansible

1. Jalankan perintah

```
ansible -i ./hosts -m shell -a 'uname -a' all -k
```

```
sysadmin@sas02:~/ansible/simple-playbook$ ansible -i ./hosts -m shell -a 'uname -a' all -k
SSH password:
[WARNING]: Platform linux on host debian_php5 is using the discovered Python interpreter at
/usr/bin/python, but future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more
information.
debian_php5 | CHANGED | rc=0 >>
Linux debian_php5.6 5.4.0-89-generic #100-Ubuntu SMP Fri Sep 24 14:50:10 UTC 2021 x86_64 GNU/Linux
ubuntu_landing | CHANGED | rc=0 >>
Linux ubuntu_landing 5.4.0-89-generic #100-Ubuntu SMP Fri Sep 24 14:50:10 UTC 2021 x86_64 x86_64 x86_
_64 GNU/Linux
ubuntu_php7 | CHANGED | rc=0 >>
Linux ubuntu_php7.4 5.4.0-89-generic #100-Ubuntu SMP Fri Sep 24 14:50:10 UTC 2021 x86_64 x86_64 x86_
64 GNU/Linux
sysadmin@sas02:~/ansible/simple-playbook$ _
```

## Grouping Host

1. Buka file `hosts` dan tambahkan nama group. Contoh nama groupnya adalah php5 dan php7

```
[php5]
ubuntu_landing ansible_host=1xc_landing.dev ansible_ssh_user=root
ansible_become_pass=a
debian_php5 ansible_host=1xc_php5.dev ansible_ssh_user=root
ansible_become_pass=a

[php7]
ubuntu_php7 ansible_host=1xc_php7.dev ansible_ssh_user=root
ansible_become_pass=a
```

2. Buat file `install-lynx.yml` dengan isi sebagai berikut

```
- hosts: php5
  tasks:
    - name: Install lynx
      become: yes #untuk menjadi superuser
      apt: name={{ item }} state=latest update_cache=true
      with_items:
        - lynx
```

3. Jalankan Perintah

```
ansible-playbook -i hosts install-lynx.yml -k
```

4. Hasil

```

sysadmin@sas02:~/ansible/simple-playbook$ ansible-playbook -i hosts install-lynx.yml -k
SSH password:

PLAY [php5] *************************************************************************************************************************************

TASK [Gathering Facts] *************************************************************************************************************************************
[WARNING]: Platform linux on host debian_php5 is using the discovered Python interpreter at
/usr/bin/python, but future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more
information.
ok: [debian_php5]
ok: [ubuntu_landing]

TASK [Install lynx] *************************************************************************************************************************************
[DEPRECATION WARNING]: Invoking "apt" only once while using a loop via squash_actions is
deprecated. Instead of using a loop to supply multiple items and specifying `name: "{{ item }}"`,
please use `name: ['lynx']` and remove the loop. This feature will be removed in version 2.11.
Deprecation warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.
[DEPRECATION WARNING]: Invoking "apt" only once while using a loop via squash_actions is
deprecated. Instead of using a loop to supply multiple items and specifying `name: "{{ item }}"`,
please use `name: ['lynx']` and remove the loop. This feature will be removed in version 2.11.
Deprecation warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.
changed: [debian_php5] => (item=['lynx'])
changed: [ubuntu_landing] => (item=['lynx'])

PLAY RECAP *********************************************************************
debian_php5      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=
0               ignored=0
ubuntu_landing   : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=
0               ignored=0

sysadmin@sas02:~/ansible/simple-playbook$

```

5. Cek ke lxc group php5, apakah sudah terinstall

```

sysadmin@sas02:~/ansible/simple-playbook$ ssh root@lxc_landing.dev
root@lxc_landing.dev's password:
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 5.4.0-89-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage
Last login: Sat Oct 23 14:51:43 2021 from 10.0.3.1
root@ubuntu_landing:~# dpkg -l | grep lynx
ii  lynx                2.8.9dev8-4ubuntu1          amd64        classic non-g
raphical (text-mode) web browser
ii  lynx-common         2.8.9dev8-4ubuntu1          all          shared files
for lynx package
root@ubuntu_landing:~# _

```

```

sysadmin@sas02:~/ansible/simple-playbook$ ssh root@lxc_php5.dev
root@lxc_php5.dev's password:
Linux debian_php5.6 5.4.0-89-generic #100-Ubuntu SMP Fri Sep 24 14:50:10 UTC 2021 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Oct 23 14:57:23 2021 from 10.0.3.1
root@debian_php5:~# dpkg -l | grep lynx
ii  lynx                2.8.9dev11-1+deb9u1         amd64
   classic non-graphical (text-mode) web browser
ii  lynx-common         2.8.9dev11-1+deb9u1         all
   shared files for lynx package
root@debian_php5:~#

```

## Copy File

1. Buat file copy.yml, dengan isi

```

- hosts: php5
  tasks:
    - name: Buat folder /tmp/dari-host
      command: mkdir -p /tmp/dari-host
    - name: Copy file install-lynx.yml
      copy: src=./install-lynx.yml dest=/tmp/dari-host/install-lynx.yml

```

## 2. Jalankan dengan perintah

```
ansible-playbook -i hosts copy.yml -k
```

```
sysadmin@sas02:~/ansible/simple-playbook$ ansible-playbook -i hosts copy.yml -k
SSH password:

PLAY [php5] *********************************************************************

TASK [Gathering Facts] *********************************************************
[WARNING]: Platform linux on host debian_php5 is using the discovered Python interpreter at
/usr/bin/python, but future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more
information.
ok: [debian_php5]
ok: [ubuntu_landing]

TASK [Buat folder /tmp/dari-host] *********************************************
[WARNING]: Consider using the file module with state=directory rather than running 'mkdir'. If you
need to use command because file is insufficient you can add 'warn: false' to this command task or
set 'command_warnings=False' in ansible.cfg to get rid of this message.
changed: [debian_php5]
changed: [ubuntu_landing]

TASK [Copy File install-lynx.yml] *********************************************
changed: [debian_php5]
changed: [ubuntu_landing]

PLAY RECAP *********************************************************************
debian_php5                : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=
0    ignored=0
ubuntu_landing             : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=
0    ignored=0

sysadmin@sas02:~/ansible/simple-playbook$
```

## 3. Cek ke lxc group php5, apakah sudah terinstall

```
sysadmin@sas02:~/ansible/simple-playbook$ ssh root@lxc_landing.dev
root@lxc_landing.dev's password:
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 5.4.0-89-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage
Last login: Sat Oct 23 15:13:51 2021 from 10.0.3.1
root@ubuntu_landing:~# ls -la /tmp/dari-host/
total 12
drwxr-xr-x 2 root root 4096 Oct 23 15:13 .
drwxrwxrwt 8 root root 4096 Oct 23 15:13 ..
-rw-r--r-- 1 root root 164 Oct 23 15:13 install-lynx.yml
root@ubuntu_landing:~# _
```

```
sysadmin@sas02:~/ansible/simple-playbook$ ssh root@lxc_php5.dev
root@lxc_php5.dev's password:
Linux debian_php5.6 5.4.0-89-generic #100-Ubuntu SMP Fri Sep 24 14:50:10 UTC 2021 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Oct 23 15:13:51 2021 from 10.0.3.1
root@debian_php5:~# ls -la /tmp/dari-host/
total 12
drwxr-xr-x 2 root root 4096 Oct 23 15:13 .
drwxrwxrwt 8 root root 4096 Oct 23 15:13 ..
-rw-r--r-- 1 root root 164 Oct 23 15:13 install-lynx.yml
root@debian_php5:~# _
```

## Referensi

1. <https://github.com/leucos/ansible-tuto>
2. <https://github.com/fathoniadi/cloud-2018/tree/master/ansible>
3. <https://www.youtube.com/watch?v=f6cKT0aylDo>

## Warning !!!

Untuk mengantisipasi hal yang tidak diinginkan (Ingat LXC digunakan sampai modul 5) silahkan backup semua lxc menggunakan command:

```
lxc-stop -n ubuntu_landing
lxc-copy -n ubuntu_landing -N ubuntu_landing_backup -SKD

# jangan lupa untuk mematikan auto start di config lxc backup
```

```
sysadmin@sas02:~$ sudo lxc-ls -f
[sudo] password for sysadmin:
NAME                STATE    AUTOSTART GROUPS IPV4        IPV6 UNPRIVILEGED
debian_php5.6        RUNNING  1         -      10.0.3.102 -      false
debian_php5.6_backup STOPPED  0         -      -          -      false
ubuntu_landing        RUNNING  1         -      10.0.3.103 -      false
ubuntu_landing_backup STOPPED  0         -      -          -      false
ubuntu_php7.4         RUNNING  1         -      10.0.3.101 -      false
ubuntu_php7.4_backup STOPPED  0         -      -          -      false
sysadmin@sas02:~$
```

## Soal Latihan Praktikum (*DIWYOR*)

Do It With Your Own Risk xixixi.

1. Buat SQL Server dengan lxc baru bernama lxc\_mariadb, gunakan ansible untuk instalasi dan konfigurasi mariadb serta phpmyadmin.

- Setup LXC lxc\_mariadb

```
sudo lxc-create -n lxc_mariadb -t download -- --dist ubuntu --release
bionic --arch amd64 --force-cache --no-validate --server
images.linuxcontainers.org

sudo lxc-start -n lxc_mariadb
```

- Set IP static lxc\_mariadb 10.0.3.200

```
sudo lxc-attach -n lxc_mariadb
apt update; apt upgrade -y; apt install -y nano
```

```
GNU nano 2.9.3 /etc/netplan/10-lxc.yaml
network:
  version: 2
  ethernet:
    eth0:
      dhcp4: false
      addresses: [10.0.3.200/24]
      gateway4: 10.0.3.1
      nameservers:
        addresses: [8.8.8.8, 1.1.1.1]
```

```
sudo netplan apply
```

- Konfigurasi ssh server untuk lxc\_mariadb

```
apt install openssh-server
nano /etc/ssh/sshd_config

# setting config menjadi
PermitRootLogin yes
RSAAuthentication yes

# end of config
service sshd restart
```

- Set password lxc\_mariadb

```
passwd
```

- Setting autostart lxc\_mariadb

```
exit
sudo su
cd /var/lib/lxc/lxc_mariadb
nano config
```

- Jangan lupa dibackup dulu :)
- Daftar kan domain lxc\_mariadb ke /etc/hosts

```
sudo nano /etc/hosts
```

```
10.0.3.101    lxc_php7.dev
10.0.3.103    lxc_landing.dev
10.0.3.102    lxc_php5.dev
10.0.3.200    lxc_mariadb_dev
```

- Buat ansible untuk install dan konfigurasi mariadb
  - Buat folder baru untuk mengerjakan ansible soal -latihan

```
cd ~/ansible
mkdir modul2-ansible
cd modul2-ansible
```

- Buat file hosts yang berisi:



```
[php5]
ubuntu_landing ansible_host=1xc_landing.dev ansible_ssh_user=root
ansible_become_pass=a
debian_php5 ansible_host=1xc_php5.dev ansible_ssh_user=root
ansible_become_pass=a

[php7]
ubuntu_php7 ansible_host=1xc_php7.dev ansible_ssh_user=root
ansible_become_pass=a

[database]
1xc_mariadb ansible_host=1xc_mariadb.dev ansible_ssh_user=root
ansible_become_pass=a
```

- Membuat install-mariadb.yml, yang berisi

```
- hosts: database
  vars:
    username: 'admin'
    password: 'SysAdminSas0102'
  roles:
    - { role: db }
```

- Disini kita akan membuat roles yang bernama `db`. roles ini berisi kumpulan task instalasi dan konfigurasi mariadb

```
mkdir -p roles/db
```

- Didalam folder `roles/db` terdapat beberapa folder, yakni `handlers`, `tasks` dan `templates`. Folder `handlers` berisi perintah perintah untuk menjalankan mariadb seperti restart, sedangkan folder `tasks` berisi script instalasi mariadb dan folder `templates` berisi template konfigurasi untuk mariadb.

```
mkdir -p roles/db/handlers
mkdir -p roles/db/tasks
mkdir -p roles/db/templates
```

- Membuat script instalasi pada folder `tasks`.

```
cd roles/db/tasks
nano main.yml
```

- roles/db/tasks/main.yml akan berisi:

```
---
- name: delete apt cache
  become: yes
  become_user: root
  become_method: su
  command: rm -vf /var/lib/apt/lists/*

- name: install mariadb
  become: yes
```

```

become_user: root
become_method: su
apt: name={{ item }} state=latest update_cache=true
with_items:
  - python
  - mariadb-server
  - python-mysqldb
  - python-pymysql
when: ansible_os_family == "Ubuntu"

- name: Stop MySQL
  service: name=mysql state=stopped

- name: set environment variables
  shell: systemctl set-environment MYSQLD_OPTS="--skip-grant-tables"

- name: Start MySQL
  service: name=mysql state=started

- name: sql query
  command: mysql -u root --execute="UPDATE mysql.user SET
authentication_string = PASSWORD('{{ password }}') WHERE User =
'root' AND Host = 'localhost';"

- name: sql query flush
  command: mysql -u root --execute="FLUSH PRIVILEGES"

- name: Stop MySQL
  service: name=mysql state=stopped

- name: unset environment variables
  shell: systemctl unset-environment MYSQLD_OPTS

- name: Start MySQL
  service: name=mysql state=started

- name: Create user for mysql
  command: mysql -u root --execute="CREATE USER IF NOT EXISTS '{{
username }}'@'localhost' IDENTIFIED BY '{{ password }}';"

- name: GRANT ALL PRIVILEGES to user {{username}}
  command: mysql -u root --execute="GRANT ALL PRIVILEGES ON * . *
TO '{{ username }}'@'localhost';"

- name: sql query flush
  command: mysql -u root --execute="FLUSH PRIVILEGES"

- name: Create DB Landing
  command: mysql -u root --execute="CREATE DATABASE IF NOT EXISTS
`landing`;"

- name: Create DB blog
  command: mysql -u root --execute="CREATE DATABASE IF NOT EXISTS
`blog`;"

- name: Create DB app
  command: mysql -u root --execute="CREATE DATABASE IF NOT EXISTS
`app`;"

```

```
- name: Copy .my.cnf file with root password credentials
  template:
    src=templates/my.cnf
    dest=/etc/mysql/mariadb.conf.d/50-server.cnf
  notify: restart mysql
```

- roles/db/templates/my.cnf akan berisi:

```
#
# These groups are read by MariaDB server.
# Use it for options that only the server (but not clients) should
see
#
# See the examples of server my.cnf files in /usr/share/mysql/
#

# this is read by the standalone daemon and embedded servers
[server]

# this is only for the mysqld standalone daemon
[mysqld]

#
# * Basic Settings
#
user          = mysql
pid-file      = /var/run/mysqld/mysqld.pid
socket        = /var/run/mysqld/mysqld.sock
port         = 3306
basedir       = /usr
datadir       = /var/lib/mysql
tmpdir        = /tmp
lc-messages-dir = /usr/share/mysql
skip-external-locking

# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address  = 0.0.0.0

#
# * Fine Tuning
#
key_buffer_size    = 16M
max_allowed_packet = 16M
thread_stack       = 192K
thread_cache_size  = 8
# This replaces the startup script and checks MyISAM tables if
needed
# the first time they are touched
myisam_recover_options = BACKUP
#max_connections    = 100
#table_cache        = 64
#thread_concurrency = 10

#
# * Query Cache Configuration
```

```

#
query_cache_limit    = 1M
query_cache_size     = 16M

#
# * Logging and Replication
#
# Both location gets rotated by the cronjob.
# Be aware that this log type is a performance killer.
# As of 5.1 you can enable the log at runtime!
#general_log_file     = /var/log/mysql/mysql.log
#general_log          = 1
#
# Error log - should be very few entries.
#
log_error = /var/log/mysql/error.log
#
# Enable the slow query log to see queries with especially long
duration
#slow_query_log_file   = /var/log/mysql/mariadb-slow.log
#long_query_time = 10
#log_slow_rate_limit   = 1000
#log_slow_verbosity = query_plan
#log-queries-not-using-indexes
#
# The following can be used as easy to replay backup logs or for
replication.
# note: if you are setting up a replication slave, see README.Debian
about
#      other settings you may need to change.
#server-id            = 1
#log_bin              = /var/log/mysql/mysql-bin.log
expire_logs_days      = 10
max_binlog_size       = 100M
#binlog_do_db         = include_database_name
#binlog_ignore_db     = exclude_database_name

#
# * InnoDB
#
# InnoDB is enabled by default with a 10MB datafile in
/var/lib/mysql/.
# Read the manual for more InnoDB related options. There are many!

#
# * Security Features
#
# Read the manual, too, if you want chroot!
# chroot = /var/lib/mysql/
#
# For generating SSL certificates you can use for example the GUI
tool "tinyca".
#
# ssl-ca=/etc/mysql/cacert.pem
# ssl-cert=/etc/mysql/server-cert.pem
# ssl-key=/etc/mysql/server-key.pem
#

```

```

# Accept only connections using the latest and most secure TLS
protocol version.
# ..when MariaDB is compiled with OpenSSL:
# ssl-cipher=TLSv1.2
# ..when MariaDB is compiled with YaSSL (default in Debian):
# ssl=on

#
# * Character sets
#
# MySQL/MariaDB default is Latin1, but in Debian we rather default
to the full
# utf8 4-byte character set. See also client.cnf
#
character-set-server = utf8mb4
collation-server     = utf8mb4_general_ci

#
# * Unix socket authentication plugin is built-in since 10.0.22-6
#
# Needed so the root database user can authenticate without a
password but
# only when running as the unix root user.
#
# Also available for other users if required.
# See https://mariadb.com/kb/en/unix\_socket-authentication-plugin/

# this is only for embedded server
[embedded]

# This group is only read by MariaDB servers, not by MySQL.
# If you use the same .cnf file for MySQL and MariaDB,
# you can put MariaDB-only options here
[mariadb]

# This group is only read by MariaDB-10.1 servers.
# If you use the same .cnf file for MariaDB of different versions,
# use this group for options that older servers don't understand
[mariadb-10.1]

```

- roles/db/handlers/main.yml akan berisi:

```

---
- name: restart mysql
  become: yes
  become_user: root
  become_method: su
  action: service name=mysql state=restarted

```

- Jalankan perintah

```

cd ~/ansible/modul2-ansible
ansible-playbook -i hosts install-mariadb.yml -k

```

```

changed: [lxc_mariadb]

TASK [db : Stop MySQL] *****
changed: [lxc_mariadb]

TASK [db : unset environment variables] *****
changed: [lxc_mariadb]

TASK [db : Start MySQL] *****
changed: [lxc_mariadb]

TASK [db : Create user for mysql] *****
changed: [lxc_mariadb]

TASK [db : GRANT ALL PRIVILEGES to user admin] *****
changed: [lxc_mariadb]

TASK [db : sql query flush] *****
changed: [lxc_mariadb]

TASK [db : Create DB Landing] *****
changed: [lxc_mariadb]

TASK [db : Create DB blog] *****
changed: [lxc_mariadb]

TASK [db : Create DB app] *****
changed: [lxc_mariadb]

TASK [db : Copy .my.cnf file with root password credentials] *****
ok: [lxc_mariadb]

PLAY RECAP *****
lxc_mariadb      : ok=18  changed=15  unreachable=0  failed=0  skipped=0  rescued=
0  ignored=0

sysadmin@sas02:~/ansible/module2-ansible$

```

- Check apakah db sudah terinstall

```
ssh root@lxc_mariadb.dev
```

```
mysql -u admin -p
show databases;
```

```

sysadmin@sas02:~/ansible/module2-ansible$ ssh root@lxc_mariadb.dev
root@lxc_mariadb.dev's password:
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-89-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage
Last login: Sun Oct 24 04:22:08 2021 from 10.0.3.1
root@lxc_mariadb:~# mysql -u admin -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 36
Server version: 10.1.48-MariaDB-0ubuntu0.18.04.1 Ubuntu 18.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| app      |
| blog     |
| information_schema |
| landing  |
| mysql    |
| performance_schema |
+-----+
6 rows in set (0.00 sec)

MariaDB [(none)]> _

```

- Setup ansible nginx untuk mengakses phpmyadmin
  - Edit install-mariadb.yml menjadi :

```
- hosts: database
  vars:
    username: 'admin'
    password: 'SysAdminSas0102'
    domain: 'lxc_mariadb.dev'
  roles:
    - db
    - pma
```

- Disini kita akan membuat roles yang bernama `pma` . roles ini berisi kumpulan task instalasi dan konfigurasi phpmyadmin

```
mkdir -p roles/pma
```

- Didalam folder `roles/pma` terdapat beberapa folder, yakni `handlers` , `tasks` dan `templates` . Folder `handlers` berisi perintah perintah untuk menjalankan nginx dan php seperti restart, sedangkan folder `tasks` berisi script instalasi phpmyadmin dan folder `templates` berisi template konfigurasi untuk nginx.

```
mkdir -p roles/pma/tasks
mkdir -p roles/pma/handlers
mkdir -p roles/pma/templates
```

- `roles/pma/tasks/main.yml` akan berisi:

```
---
- name: delete apt chache
  become: yes
  become_user: root
  become_method: su
  command: rm -vf /var/lib/apt/lists/*

- name: install nginx phpmyadmin
  become: yes
  become_user: root
  become_method: su
  apt: name={{ item }} state=latest update_cache=true
  with_items:
    - curl
    - nginx
    - nginx-extras
    - php7.2-fpm
    - php-mbstring
    - php-zip
    - php-gd
    - php-json
    - php-curl
    - phpmyadmin

- name: enable module php mbstring
  command: phpenmod mbstring
  notify:
    - restart php
```

```

- name: symlink phpmyadmin to nginx
  command: ln -sf /usr/share/phpmyadmin /var/www/html

- name: Copy pma.local
  template:
    src=templates/pma.local
    dest=/etc/nginx/sites-available/{{ domain }}
  vars:
    servername: '{{ domain }}'

- name: Symlink pma.local
  command: ln -sf /etc/nginx/sites-available/{{ domain }}
    /etc/nginx/sites-enabled/{{ domain }}
  notify:
    - stop apache2
    - restart nginx

- name: Write pma.local to /etc/hosts
  lineinfile:
    dest: /etc/hosts
    regexp: '.*{{ domain }}$'
    line: "127.0.0.1 {{ domain }}"
    state: present

```

- roles/pma/templates/pma.local akan berisi:

```

server {
    listen 80;

    server_name {{servername}};

    root /usr/share/phpmyadmin;

    index index.php;

    location / {

        try_files $uri $uri/ @phpmyadmin;

    }
    location @phpmyadmin {
        fastcgi_pass unix:/run/php/php7.2-fpm.sock; #Sesuaikan
dengan versi PHP

        fastcgi_param SCRIPT_FILENAME
/usr/share/phpmyadmin/index.php;

        include /etc/nginx/fastcgi_params;

        fastcgi_param SCRIPT_NAME /index.php;
    }
    location ~ /\.php$ {

        fastcgi_pass unix:/run/php/php7.2-fpm.sock; #Sesuaikan
dengan versi PHP

        fastcgi_index index.php;

```



```

        fastcgi_param SCRIPT_FILENAME
/usr/share/phpmyadmin$fastcgi_script_name;

        include fastcgi_params;

    }
}

```

- roles/pma/handlers/main.yml akan berisi:

```

---
- name: stop apache2
  become: yes
  become_user: root
  become_method: su
  action: service name=apache2 state=stopped

- name: restart nginx
  become: yes
  become_user: root
  become_method: su
  action: service name=nginx state=restarted

- name: restart php
  become: yes
  become_user: root
  become_method: su
  action: service name=php7.2-fpm state=restarted

```

- Jalankan perintah:

```

cd ~/ansible/modul2-ansible
ansible-playbook -i hosts install-mariadb.yml -k

```

- edit /etc/nginx/sites-available/vm.local menjadi:

```

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

    server_name vm.local;

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

    location /php5 {
        rewrite /php5/(.*)$ /$1 break;
        proxy_pass http://lxc_php5.dev;
    }

    location /php7 {
        rewrite /php7/(.*)$ /$1 break;
        proxy_pass http://lxc_php7.dev;
    }
}

```

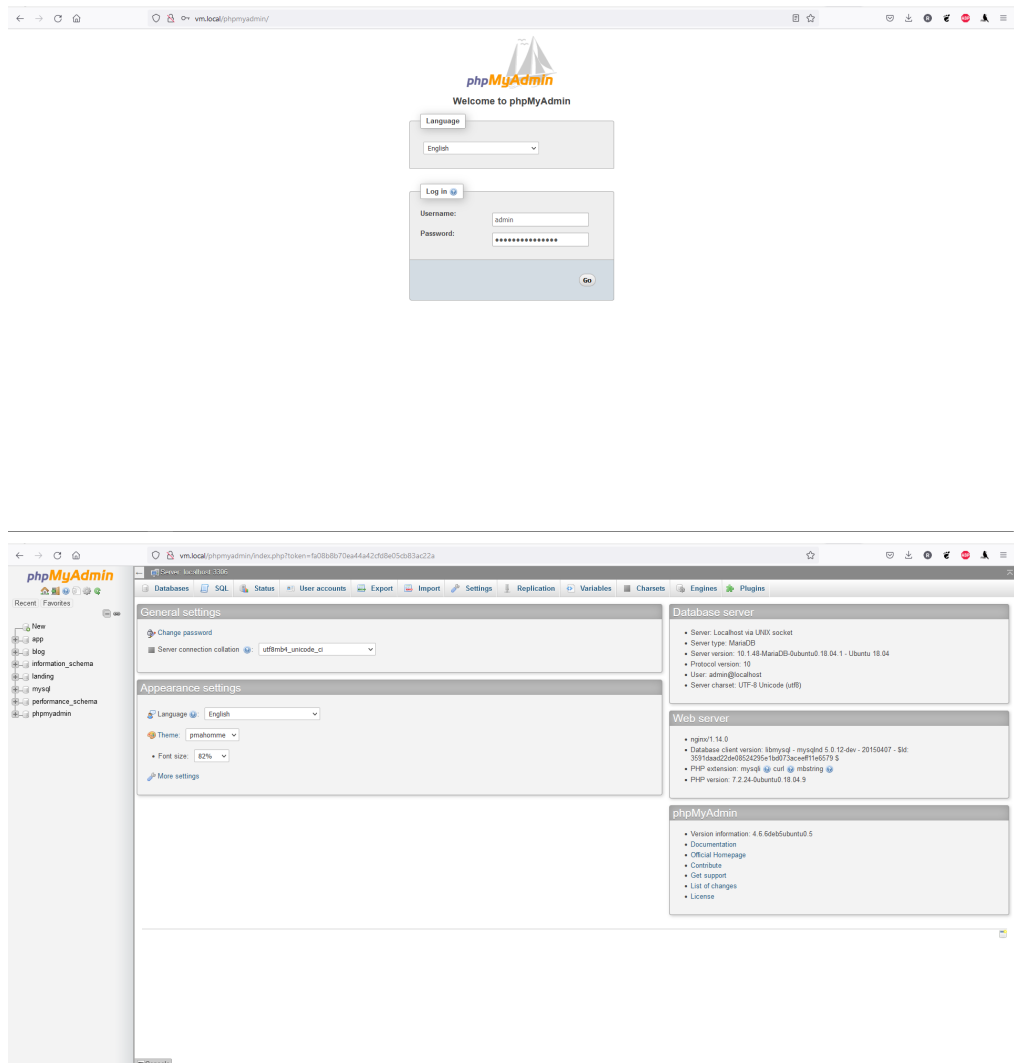
```

location /phpmyadmin {
    rewrite /phpmyadmin/?(.*)$ /$1 break;
    proxy_pass http://lxc_mariadb.dev;
}

location / {
    #rewrite /php7/?(.*)$ /$1 break;
    proxy_pass http://lxc_landing.dev;
}

```

- coba akses `vm.local/phpmyadmin/`:



2. Install codeigniter pada `lxc_landing` `ubuntu_landing` dengan beberapa requirement, yakni:

- Menggunakan PHP5, dikonfigurasi dengan ansible
- Clone Codeigniter dari <https://github.com/alldonesia/sas-ci>
- Menggunakan database landing yang terdapat pada `lxc_mariadb`