



CENTOS 8

Web Server with Nginx on Centos 8



IN THIS MODULE

Centos 8 Installation, SSH Configuration, Nginx & PHP FPM Configuration, MariaDB Configuration, Wordpress Installation.

Muhammad Ridwan Na'im

Dinas Komunikasi dan Informatika Kota Tangerang

1. INTRODUCTION

A. WHAT IS CENTOS?

CentOS (Community Enterprise Operating System) is a Linux distribution that is suitable for use on an Enterprise or enterprise scale that is free or free. CentOS is coded from the Red Hat Enterprise (RHEL) source code released under the General Public License (GPL) and later developed by a community called the CentOS Project. CentOS's .iso can be downloaded at <http://isoredirect.centos.org/centos/>.

For details, you can visit the website <http://www.centos.org>. Now the latest version of CentOS is 8.1.x.

BENEFITS :

1. Easy to maintain
2. Suitable for long-term use, especially for non-experimental and other production environments
3. Easy to use in software packages and users
4. There is a long-term support from the developers
5. An active development community
6. Community-based infrastructure
7. An open management system

DEFICIENCY :

1. All integration is very dependent on the Red Hat distribution. Because CentOS always comes out after Red Hat
2. Because Centos is built on Red hat, all the advantages and disadvantages of Red hat are also part of centos.

B. WHAT IS NGINX?

NGINX (read: engine-x) is an open source web server software. When it was first released, NGINX only functioned as HTTP web serving. But now, the software also acts as a reverse proxy, HTTP load balancer, and email proxy for IMAP, POP3, and SMTP.

NGINX was officially introduced in October 2004. The creator or creator of this software, Igor Sysoev, started his project in 2002 to answer the C10k problem. C10k itself is described as a challenge faced by servers when they have to manage ten thousand connections at the same time. Until now, the number of connections managed by the web server continues to grow. For this reason, NGINX offers event-driven and asynchronous architecture. This architecture makes NGINX one of the servers with reliable speed and scalability.

Because of its speed and ability to handle a large number of connections, NGINX services are often used by websites with high traffic. Some examples of these websites are Google, Netflix, Adobe, Cloudflare, WordPress.com, and many more. For detail information of nginx, please visit <https://www.nginx.com/>.

C. WHAT IS PHP?

At first PHP was short for Personal Home Page (personal site). PHP was first created by Rasmus Lerdorf in 1995. At that time PHP was still called Form Interpreted (FI), which was in the form of a set of scripts used to process form data from the web. Next Rasmus released the source code to the public and named it PHP / FI. With the release of this source code being open source, many programmers are interested in developing PHP.

In November 1997, PHP / FI 2.0 was released. In this release, the PHP interpreter has been implemented in the C program. This release also includes extension modules that significantly improve PHP / FI capabilities. In 1997, a company called Zend rewrote PHP interpreters to be cleaner, better, and faster. Then in June 1998, the company released a new interpreter for PHP

and inaugurated the release as PHP 3.0 and the abbreviation for PHP was changed to a PHP acronym: Hypertext Preprocessing.

And now, the latest version of PHP is 7.4.2. For more information, you can visit <https://php.net>.

D. WHAT IS MARIADB?

MariaDB is an open and independent development version of MySQL. Since the acquisition of MySQL by Oracle in September 2010, Monty Program as the initial author of the MySQL source code broke away from development and made a more independent version, MariaDB. MariaDB is an implementation of a relational database management system (RDBMS) that is distributed free of charge under the GPL (General Public License) license. Each user can freely use MariaDB, but with the limitation of the software it should not be made a commercial derivative product.

Benefits of Using MariaDB :

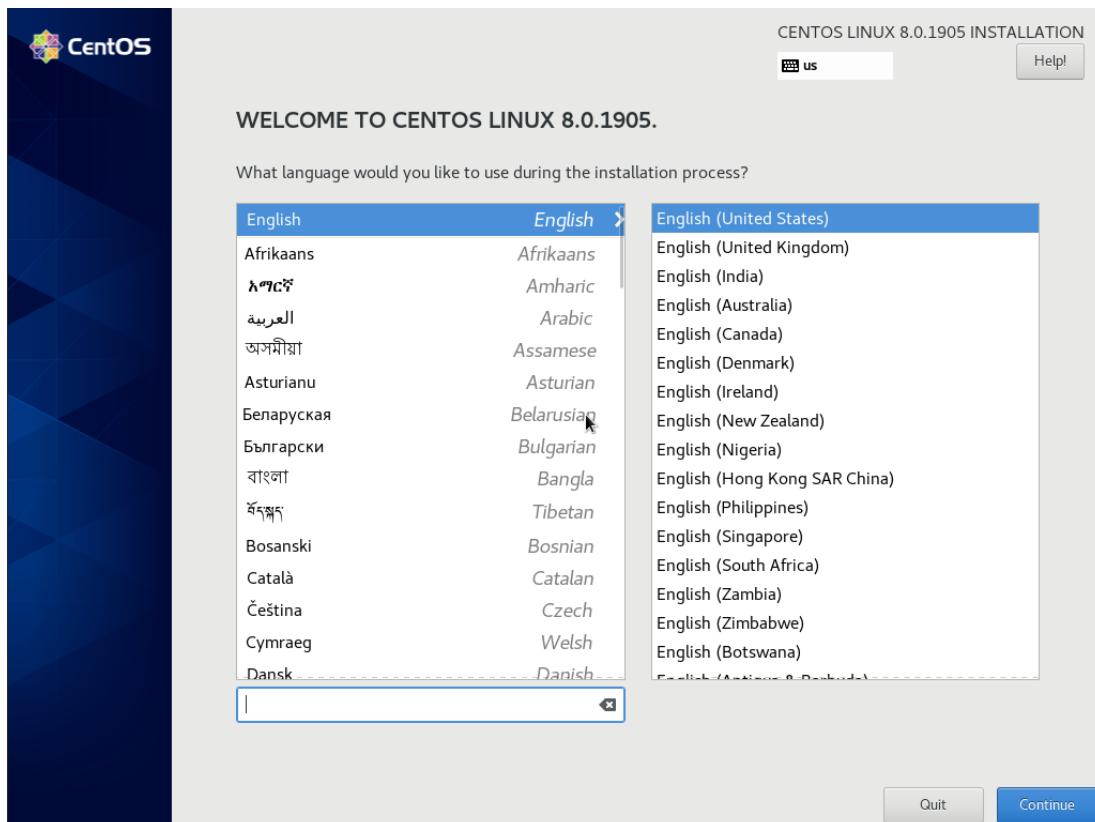
1. Portability. MariaDB can run stable on various operating systems such as Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, and many more.
2. Open source software. MariaDB is distributed as open source software, under the GPL license so that it can be used free of charge.
3. Multi-user. MariaDB can be used by several users at the same time without experiencing problems or conflicts.
4. Performance tuning, MariaDB has an amazing speed in handling simple queries, in other words it can process more SQL per unit time.
5. Variety of data types. MariaDB has a variety of very rich data types, such as signed / unsigned integers, float, double, char, text, date, timestamp, and others.
6. Commands and functions. MariaDB has full operators and functions that support the Select and Where commands in the query.
7. Security. MariaDB has several layers of security such as the subnet mask level, host name, and user access permissions with a detailed licensing system and encrypted password.
8. Scalability and Restrictions. MariaDB is able to handle databases on a large scale, with more than 50 million records and 60 thousand tables and 5 billion rows. In addition, the index limit that can be accommodated reaches 32 indexes in each table.
9. Connectivity. MariaDB can connect with clients using TCP / IP protocols, Unix sockets (UNIX), or Named Pipes (NT).
10. Language localization. MariaDB can detect error messages on clients by using more than twenty languages. Even so, Indonesian is not yet included.
11. The interface. MariaDB has an interface for various applications and programming languages using the API (Application Programming Interface) function.
12. Clients and equipment. MariaDB is equipped with a variety of tools (tools) that can be used for database administration, and on every piece of equipment available online instructions are included.
13. Table structure. MariaDB has a table structure that is more flexible in handling ALTER TABLE, compared to other databases such as PostgreSQL or Oracle.

E. WHAT IS WORDPRESS?

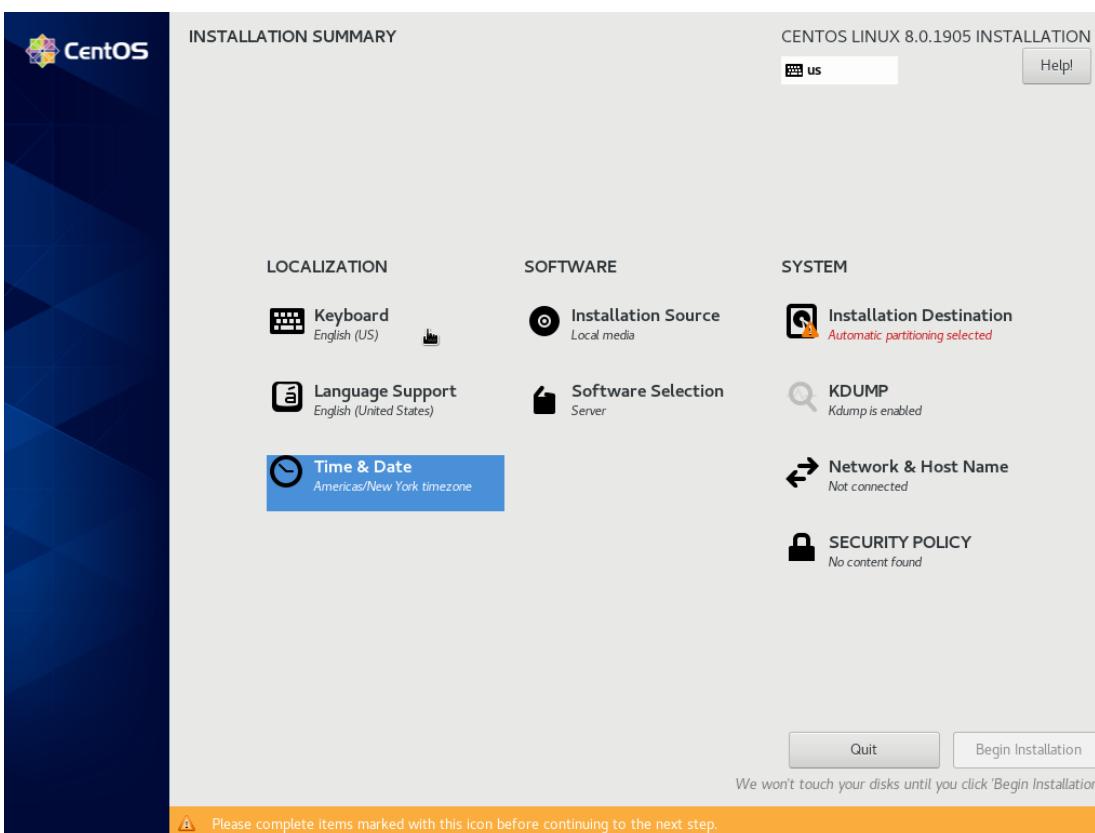
WordPress is an open source application that is very popular to be used as a blog engine. WordPress is built with PHP programming language and MySQL database. PHP and MySQL, both are open source software (open source software).] Apart from being a blog, WordPress also began to be used as a CMS (Content Management System) because of its ability to be modified and adapted to the needs of its users. WordPress is the official successor of b2 / cafelog which was developed by Michel Valdrighi. The name WordPress was proposed by Christine Selleck, friend of Matt Mullenweg. WordPress is now a content management system (CMS) platform for several well-known websites such as CNN, Reuters, The New York Times, TechCrunch, and others

2. CENTOS 8 INSTALLATION

Before we begin installation step. Please download the CentOS 8.0 ISO from isoredirect.centos.org/centos/8/isos/x86_64/. Burn the CentOS ISO into your installation media (Flashdisk or DVD) and setting your server's boot option to your installation media. You can also try installing CentOS on Virtualbox, VMWare, or Proxmox VE. In this module, i used Proxmox VE.



Choose installation language which you understand



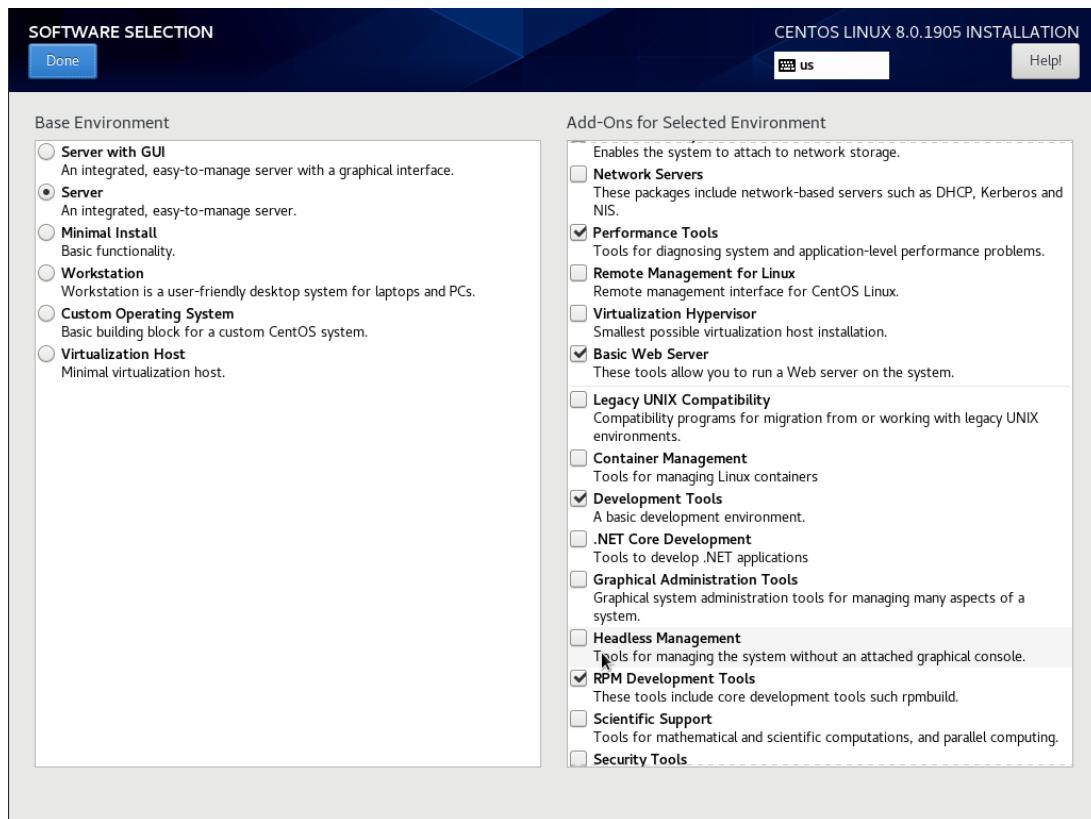
.Centos 8 Installation Summary Display

INSTALLATION SUMMARY MENUS DESCRIPTION

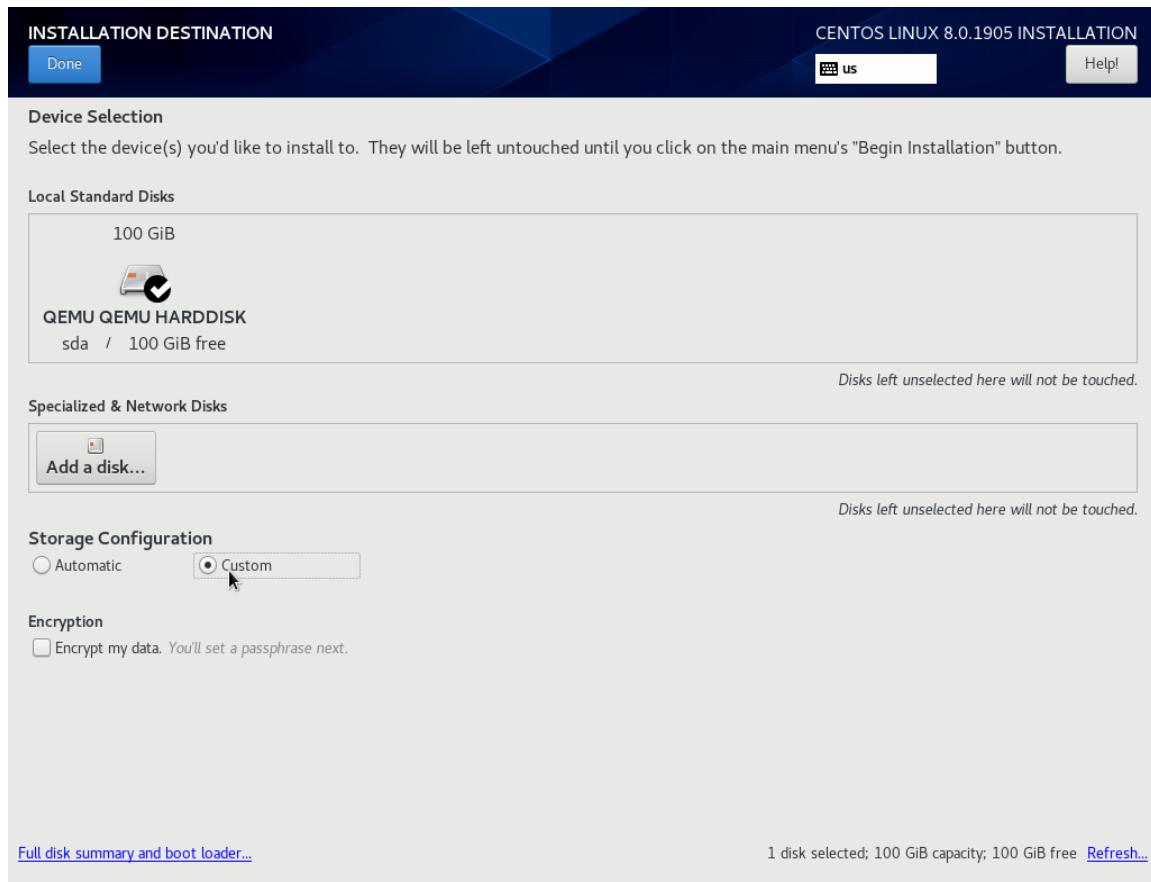
1. Keyboard : Set your keyboard layout. The default is English (US).
2. Language Support : Set your server language. The default is English (United States)
3. Time & Date : Set your date and timezone.
4. Installation Source : Centos Installation Source Setting
5. Software Selection : Select packages which you want to install
6. Installation Destination : Select where you want to install Centos and disk configuration
7. Network & Host Name : Set your Centos IP Address and host name.
8. Security Policy : Security Policy Settings.



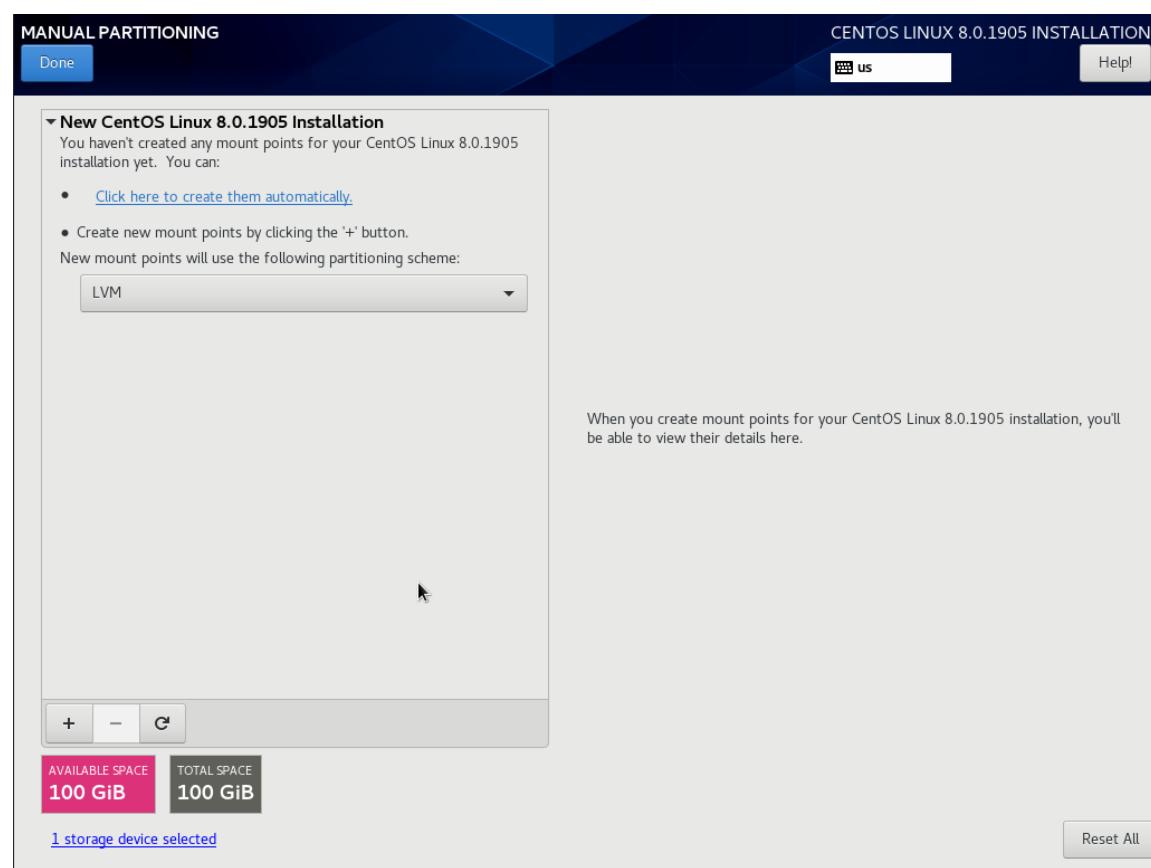
Date and Timezone setting. I choose Asia/Jakarta.



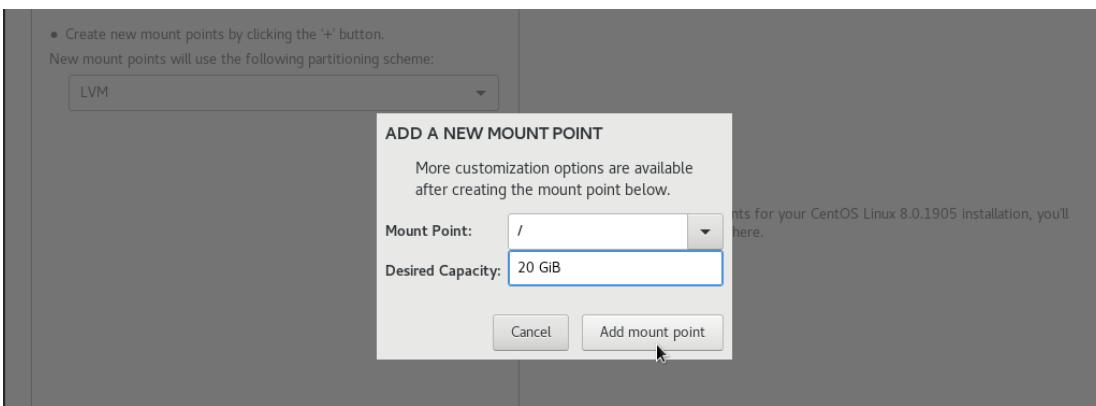
Centos 8 Software selection. Select software which you want to install. For standard webserver, select which software candidates to install like the image above.



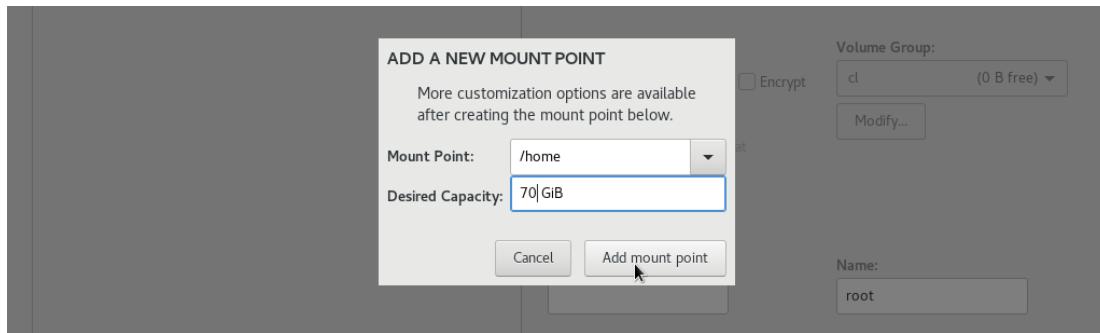
Installation Destination Configuration. You can select automatic or custom for storage configuration. In this module, i choose custom and i will configure my own Centos 8 Disk Partition.



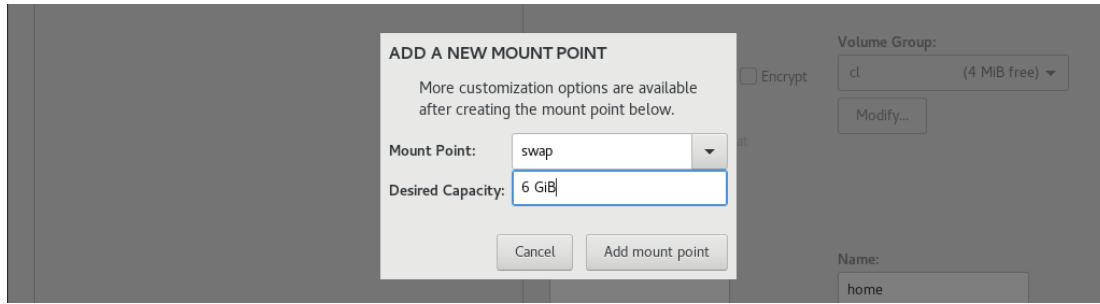
I choose custom and click done. Now, Manual Partitioning Configuration displayed. Add partition with clicking the + button.



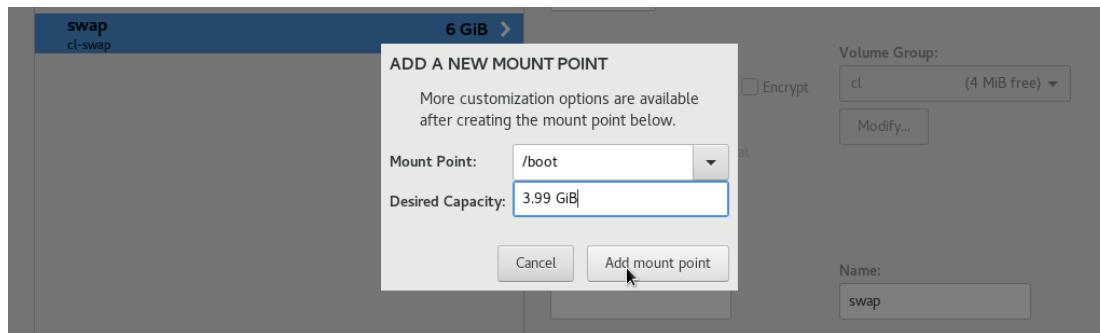
Add a root partition. Determine the capacity of your root partition according to your disk size. I have 100 GB disk capacity.



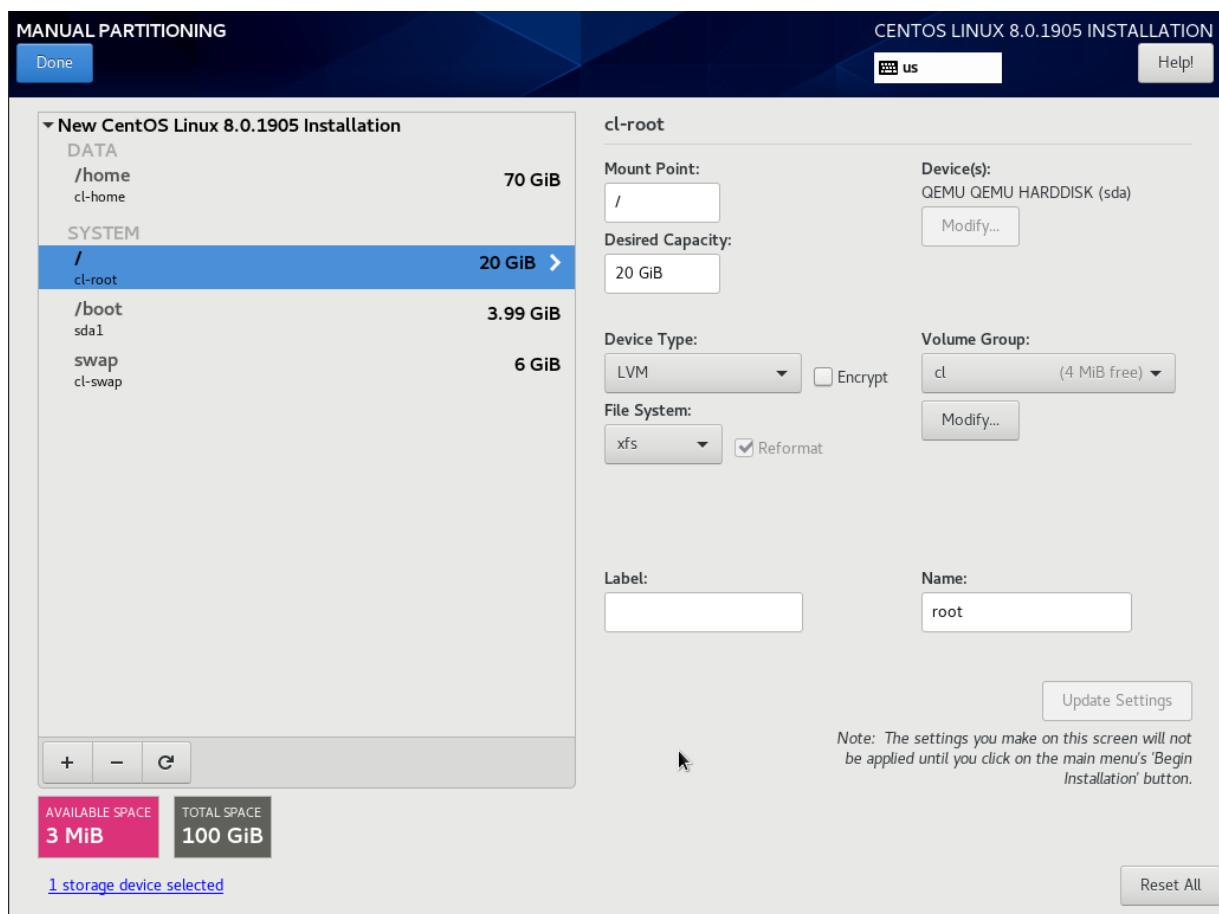
Add home partition.



Add swap partition. SWAP is a space on the hard disk that is used as virtual memory space that is used when the computer / laptop needs more memory. In the sense of partitions with this SWAP filesystem it works as a backup, if the RAM used is full. You can set swap partition of half form your memory or even twice of memory if needed.



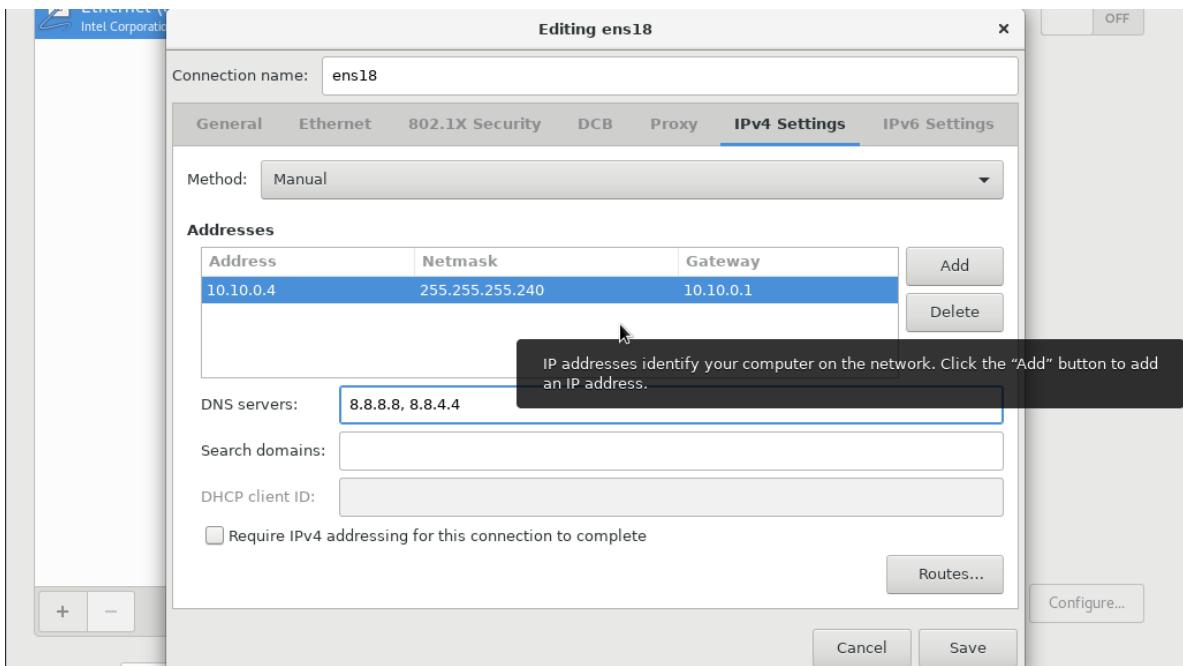
And the last, create boot partition.



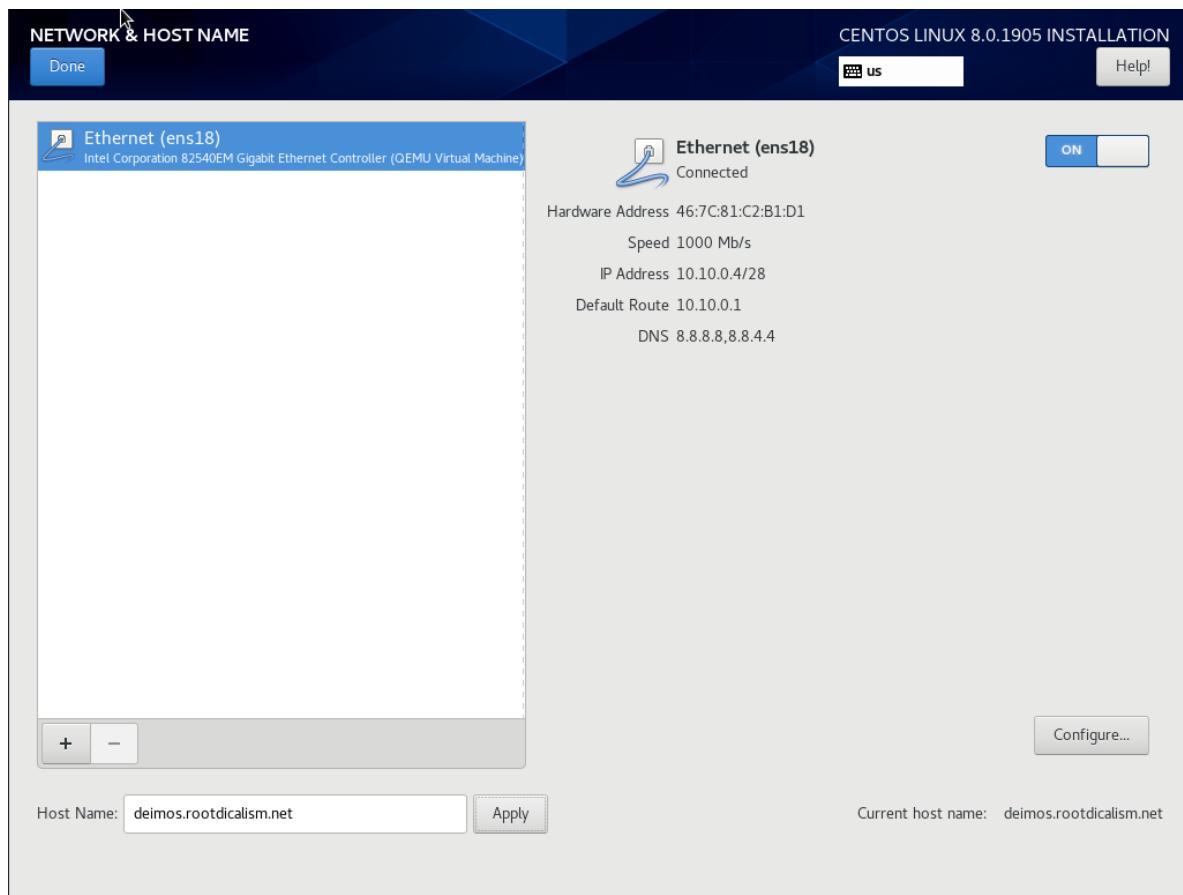
Here's the result. Click Done.

Order	Action	Type	Device	Mount point
1	Destroy Format	Unknown	QEMU QEMU HARDDISK (sda)	
2	Create Format	partition table (MSDOS)	QEMU QEMU HARDDISK (sda)	
3	Create Device	partition	sda1 on QEMU QEMU HARDDISK	
4	Create Format	ext4	sda1 on QEMU QEMU HARDDISK	/boot
5	Create Device	partition	sda2 on QEMU QEMU HARDDISK	
6	Create Format	physical volume (LVM)	sda2 on QEMU QEMU HARDDISK	
7	Create Device	lvmvg	cl	
8	Create Device	lvmlv	cl-swap	
9	Create Format	swap	cl-swap	
10	Create Device	lvmlv	cl-home	
11	Create Format	xfs	cl-home	/home
12	Create Device	lvmlv	cl-root	

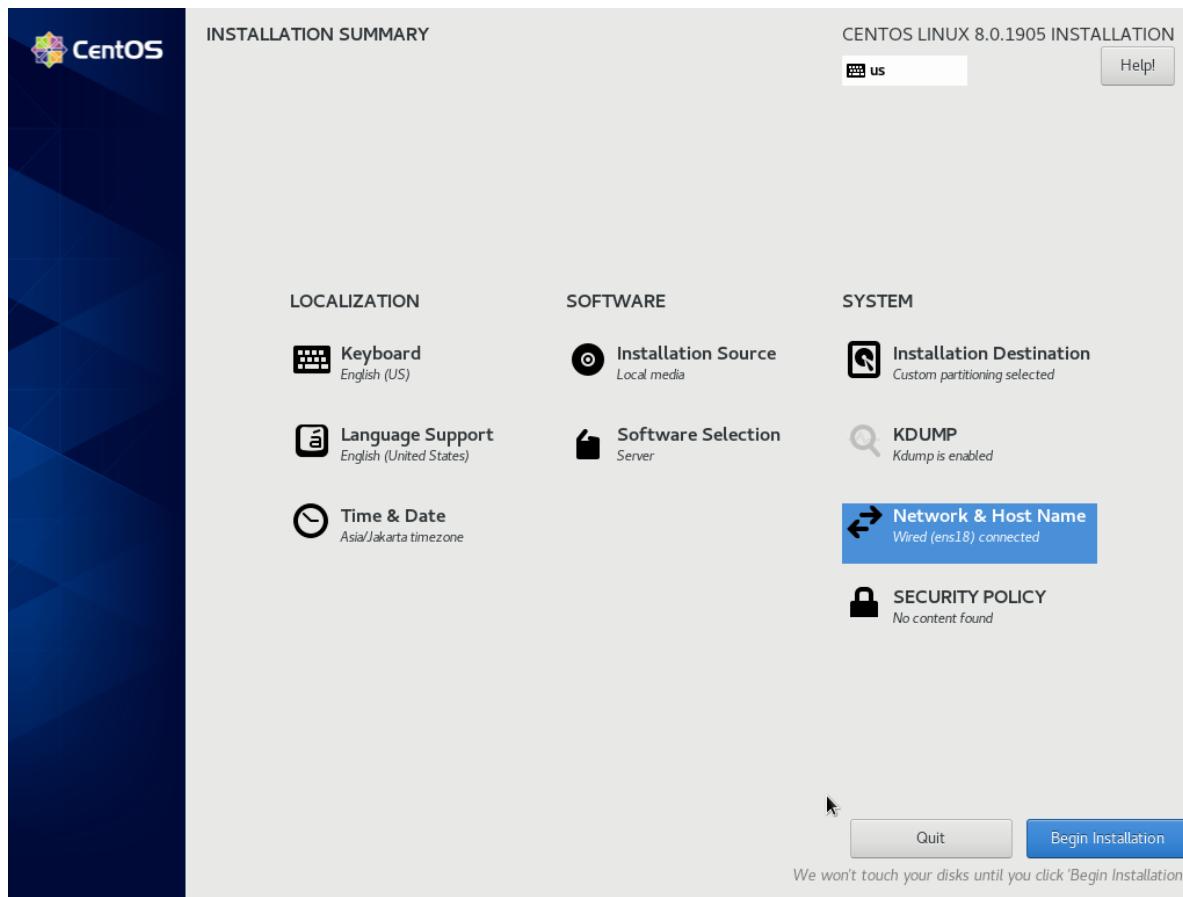
Confirmation required. If you sure with the configuration, click Accept Changes.



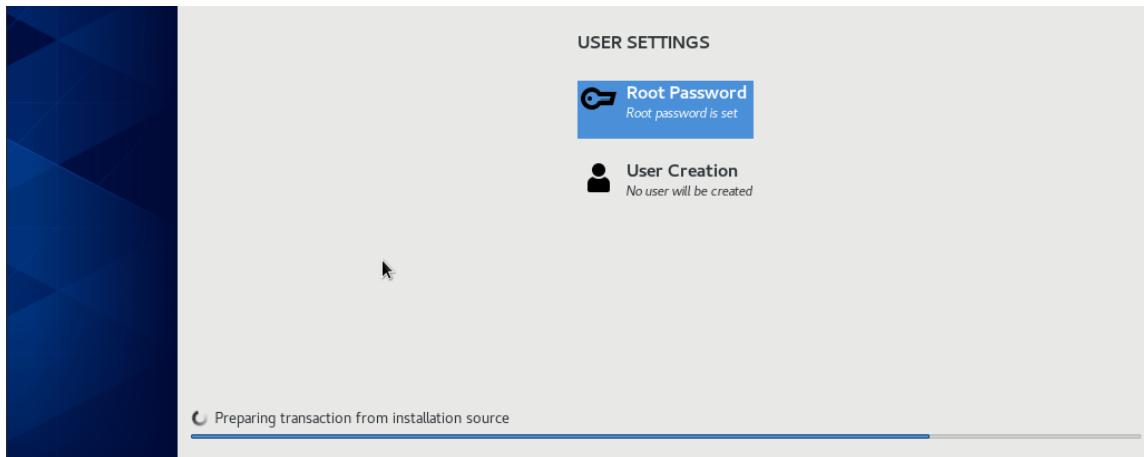
Configure your IP Address. You can do this now or after installation finished.



Turn on Network Service and Configure your host name.



Installation Summary Result. If you sure with all configurations, click Begin Installation.



Installation is on progress. During the installation process, set the root password and add another user.

The root account is used for administering the system. Enter a password for the root user.

Root Password: Strong

Confirm:

Root password setting

Full name: deimos

User name: deimos

Tip: Keep your user name shorter than 32 characters and do not use spaces.

Make this user administrator

Require a password to use this account

Password: Strong

Confirm password:

Advanced...

Create sudoers user following the image above. And then click Advanced.

Full name: deimos

User name: deimos

Tip: Keep your user name shorter than 32 characters and do not use spaces.

Make this user administrator

ADVANCED USER CONFIGURATION

Home directory: /home/deimos

User and Group IDs

Specify a user ID manually: 1000

Specify a group ID manually: 1000

Group Membership

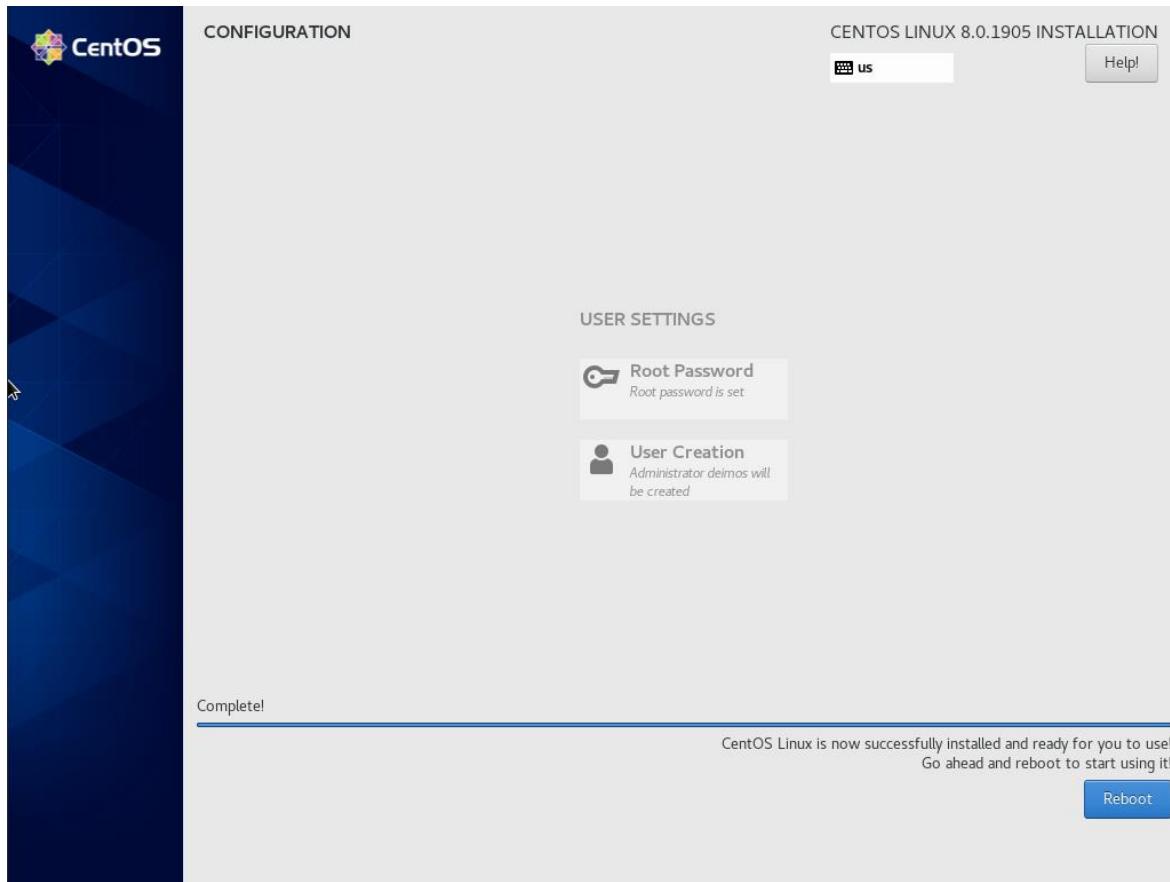
Add user to the following groups:

wheel

Tip:
You may input a comma-separated list of group names and group IDs here. Groups that do not already exist will be created; specify their GID in parentheses.

Cancel Save Changes

Add the user to wheel group. Save Changes.



Centos 8 Installation is finished. Reboot your server.

3. WEB SERVER CONFIGURATION

A. SELINUX AND SSH CONFIGURATION

```
GNU nano 2.9.8          /etc/selinux/config          Modified

# This file controls the state of SELinux on the system.
# SELINUXTYPE= can take one of these three values:
#       enforcing - SELinux security policy is enforced.
#       permissive - SELinux prints warnings instead of enforcing.
#       disabled - No SELinux policy is loaded.
SELINUXTYPE=disabled
# SELINUXTYPE= can take one of these three values:
#       targeted - Targeted processes are protected,
#       minimum - Modification of targeted policy. Only selected processes are protected.
#       mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

Disable selinux (optional).

```
# nano /etc/selinux/config
```

Change SELINUX=enforcing => SELINUX=disabled

SSH Configuration. For security reason, change your default SSH port to another port and disable root login from SSH Client.

```
# nano /etc/ssh/sshd_config
```

Change Port 22 => Port 1212 [Line 17]

Set PermitRootLogin no [Line 46]

Open the new SSH port.

```
# firewall-cmd --perma --add-port=1212/tcp
# firewall-cmd -reload
```

Update Centos and reboot after finished

```
Installed:
  kernel-4.18.0-147.3.1.el8_1.x86_64
  kernel-debug-devel-4.18.0-147.3.1.el8_1.x86_64
  kernel-modules-4.18.0-147.3.1.el8_1.x86_64
  dconf-0.28.0-3.el8.x86_64
  adwaita-cursor-theme-3.28.0-2.el8.noarch
  at-spi2-atk-2.26.2-1.el8.x86_64
  boost-filesystem-1.66.0-6.el8.x86_64
  colord-libs-1.4.2-1.el8.x86_64
  libXst-1.2.3-7.el8.x86_64
  librsqg2-tools-2.42.7-3.el8.x86_64
  libwayland-egl-1.15.0-1.el8.x86_64
  podman-manpages-1.4.2-5.module_e18.1.0+237+63e26edc.noarch
  tbb-2018.2-9.el8.x86_64
  centos-repos-8.1-1.1911.0.8.el8.x86_64
  libgusb-0.3.0-1.el8.x86_64
  mozjs60-60.9.0-3.el8.x86_64
  python3-pip-wheel-9.0.3-15.el8.noarch
  kernel-core-4.18.0-147.3.1.el8_1.x86_64
  kernel-devel-4.18.0-147.3.1.el8_1.x86_64
  abattis-cantarell-fonts-0.0.25-4.el8.noarch
  oddjob-mkhomedir-0.34.4-7.el8.x86_64
  adwaita-icon-theme-3.28.0-2.el8.noarch
  at-spi2-core-2.28.0-1.el8.x86_64
  boost-timer-1.66.0-6.el8.x86_64
  gtk3-3.22.30-4.el8.x86_64
  libepoxy-1.5.2-1.el8.x86_64
  libwayland-cursor-1.15.0-1.el8.x86_64
  oddjob-0.34.4-7.el8.x86_64
  rest-0.8.1-2.el8.x86_64
  centos-gpg-keys-8.1-1.1911.0.8.el8.noarch
  grub2-tools-efi-i-1:2.02-78.el8.x86_64
  libssh-config-0.9.0-4.el8.noarch
  psmisc-23.1-3.el8.x86_64
  python3-setuptools-wheel-39.2.0-5.el8.noarch

Complete!
[root@deimos ~]# reboot
```

```
# dnf update -y
```

```
# reboot
```

Now you can remote your Centos from SSH Client after reboot process finished.

```
root@deimos:/home/deimos
login as: deimos
deimos@10.10.0.4's password:
Activate the web console with: systemctl enable --now cockpit.socket

[deimos@deimos ~]$ sudo su

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

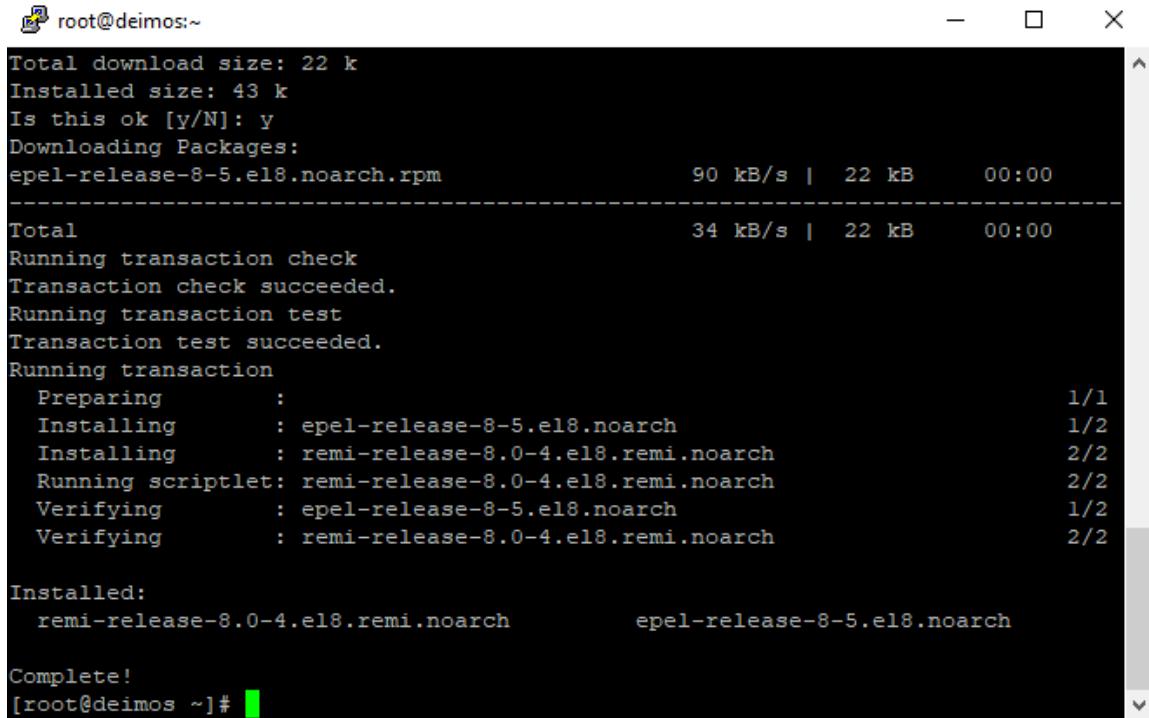
    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for deimos:
[root@deimos deimos]#
```

B. PHP 7.4 INSTALLATION

Install yum-utils and remi repo.

```
# yum -y install yum-utils
# yum install http://rpms.remirepo.net/enterprise/remi-release-8.rpm
```



A terminal window titled 'root@deimos:~' showing the output of the yum command. It shows the download of epel-release-8-5.el8.noarch.rpm (90 kB/s), transaction check, transaction test, and the installation of remi-release-8.0-4.el8.remi.noarch and epel-release-8-5.el8.noarch. The process is marked as complete.

```
Total download size: 22 k
Installed size: 43 k
Is this ok [y/N]: y
Downloading Packages:
epel-release-8-5.el8.noarch.rpm                                90 kB/s | 22 kB     00:00
-----
Total                                         34 kB/s | 22 kB     00:00

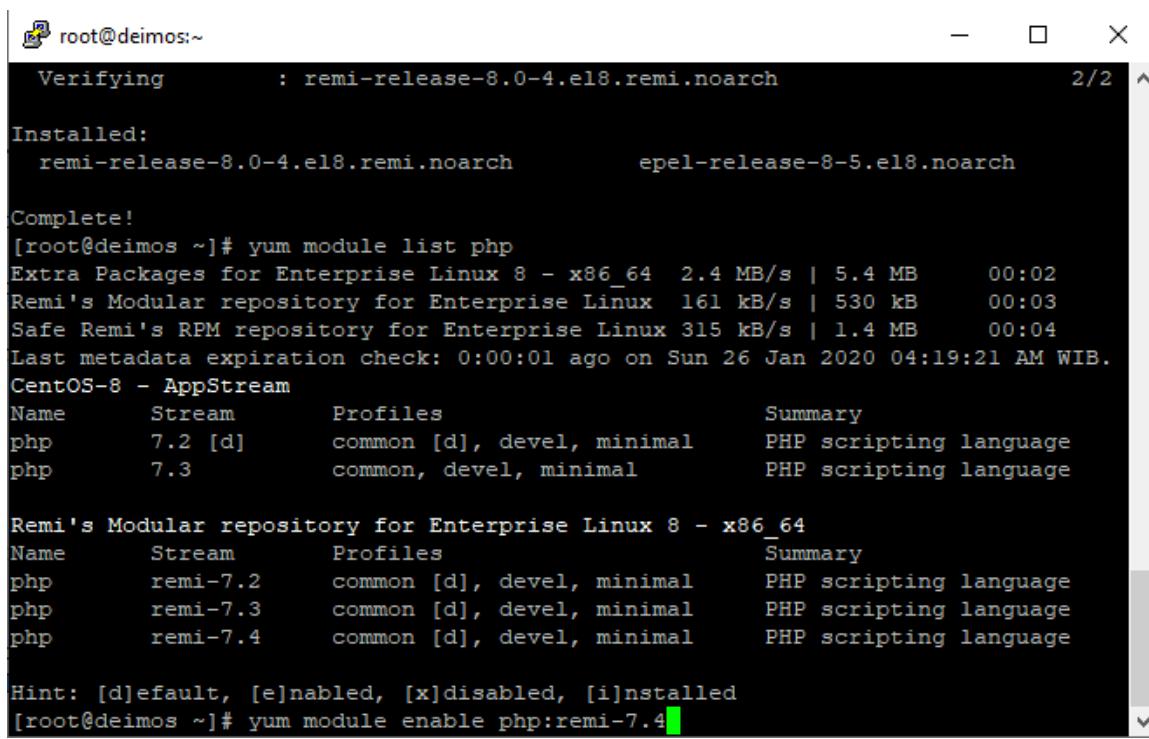
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing          :                                1/1
  Installing         : epel-release-8-5.el8.noarch      1/2
  Installing         : remi-release-8.0-4.el8.remi.noarch 2/2
  Running scriptlet: remi-release-8.0-4.el8.remi.noarch 2/2
  Verifying          : epel-release-8-5.el8.noarch      1/2
  Verifying          : remi-release-8.0-4.el8.remi.noarch 2/2

Installed:
  remi-release-8.0-4.el8.remi.noarch      epel-release-8-5.el8.noarch

Complete!
[root@deimos ~]#
```

Enable php:remi-7.4 for installing PHP 7.4.x.

```
# yum module list php
# yum module enable php:remi-7.4
```



A terminal window titled 'root@deimos:~' showing the output of the yum module commands. It shows the verification of remi-release-8.0-4.el8.remi.noarch, the installed packages remi-release-8.0-4.el8.remi.noarch and epel-release-8-5.el8.noarch, and the completion of the process. It also lists available PHP modules (7.2, 7.3, 7.4) and their profiles (common, devel, minimal). The Remi's Modular repository for Enterprise Linux 8 - x86_64 section is also shown.

```
Verifying      : remi-release-8.0-4.el8.remi.noarch          2/2

Installed:
  remi-release-8.0-4.el8.remi.noarch      epel-release-8-5.el8.noarch

Complete!
[root@deimos ~]# yum module list php
Extra Packages for Enterprise Linux 8 - x86_64   2.4 MB/s | 5.4 MB     00:02
Remi's Modular repository for Enterprise Linux 161 kB/s | 530 kB     00:03
Safe Remi's RPM repository for Enterprise Linux 315 kB/s | 1.4 MB     00:04
Last metadata expiration check: 0:00:01 ago on Sun 26 Jan 2020 04:19:21 AM WIB.
CentOS-8 - AppStream
Name      Stream      Profiles           Summary
php       7.2 [d]     common [d], devel, minimal    PHP scripting language
php       7.3         common, devel, minimal      PHP scripting language

Remi's Modular repository for Enterprise Linux 8 - x86_64
Name      Stream      Profiles           Summary
php       remi-7.2    common [d], devel, minimal    PHP scripting language
php       remi-7.3    common [d], devel, minimal    PHP scripting language
php       remi-7.4    common [d], devel, minimal    PHP scripting language

Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled
[root@deimos ~]# yum module enable php:remi-7.4
```

Install PHP 7.4.x and the modules and check the version after installation process finished.

```
# yum -y install php php-fpm php-common php-pdo php-cli \
  php-mysqlnd php-gd php-intl php-xml php-mbstring php-xmlrpc \
  php-zip
# php -v
```

```
root@deimos:~  
Installed:  
php-7.4.2-1.el8.remi.x86_64  
php-cli-7.4.2-1.el8.remi.x86_64  
php-common-7.4.2-1.el8.remi.x86_64  
php-fpm-7.4.2-1.el8.remi.x86_64  
php-gd-7.4.2-1.el8.remi.x86_64  
php-intl-7.4.2-1.el8.remi.x86_64  
php-mbstring-7.4.2-1.el8.remi.x86_64  
php-mysqlnd-7.4.2-1.el8.remi.x86_64  
php-pdo-7.4.2-1.el8.remi.x86_64  
php-pear-1:1.10.10-4.el8.remi.noarch  
php-pecl-zip-1.15.5-1.el8.remi.7.4.x86_64  
php-xml-7.4.2-1.el8.remi.x86_64  
php-xmlrpc-7.4.2-1.el8.remi.x86_64  
nginx-filesystem-1:1.14.1-9.module_el8.0.0+184+e34fea82.noarch  
php-opcache-7.4.2-1.el8.remi.x86_64  
php-sodium-7.4.2-1.el8.remi.x86_64  
oniguruma-6.8.2-1.el8.x86_64  
libsodium-1.0.18-2.el8.x86_64  
libzip-1.6.0-1.el8.remi.x86_64  
php-fedora-autoloader-1.0.0-5.el8.remi.noarch  
php-json-7.4.2-1.el8.remi.x86_64  
php-process-7.4.2-1.el8.remi.x86_64  
libicu62-62.1-3.el8.remi.x86_64  
  
Complete!  
[root@deimos ~]# php -v  
PHP 7.4.2 (cli) (built: Jan 21 2020 11:35:20) ( NTS )  
Copyright (c) The PHP Group  
Zend Engine v3.4.0, Copyright (c) Zend Technologies  
    with Zend OPcache v7.4.2, Copyright (c), by Zend Technologies  
[root@deimos ~]#
```

PHP installed successfully.

C. MARIADB 10.4.11 INSTALLATION

Download the stable latest version from MariaDB and Install.

```
# wget https://downloads.mariadb.com/MariaDB/mariadb_repo_setup  
# chmod +x mariadb_repo_setup  
# ./mariadb_repo_setup  
# yum -y install perl-DBI libaio libsepol lsof \  
boost-program-options  
# yum -y install --repo="mariadb-main" MariaDB-server
```

```
root@deimos:~  
Complete!  
[root@deimos ~]# wget https://downloads.mariadb.com/MariaDB/mariadb_repo_setup  
--2020-01-26 04:24:13-- https://downloads.mariadb.com/MariaDB/mariadb_repo_setup  
p  
Resolving downloads.mariadb.com (downloads.mariadb.com) ... 104.20.158.76, 104.20  
.159.76, 2606:4700:10::6814:9f4c, ...  
Connecting to downloads.mariadb.com (downloads.mariadb.com) |104.20.158.76|:443..  
. connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 19162 (19K) [application/octet-stream]  
Saving to: 'mariadb_repo_setup'  
  
mariadb_repo_setup 100%[=====>] 18.71K --.-KB/s in 0.001s  
  
2020-01-26 04:24:14 (29.8 MB/s) - 'mariadb_repo_setup' saved [19162/19162]  
[root@deimos ~]# chmod +x mariadb_repo_setup
```

```
[root@deimos ~]# ./mariadb_repo_setup
[info] Repository file successfully written to /etc/yum.repos.d/mariadb.repo
[info] Adding trusted package signing keys...
[info] Successfully added trusted package signing keys
[root@deimos ~]# yum install perl-DBI libaio libsepol lsof boost-program-options
MariaDB Server                                189 kB/s | 489 kB     00:02
MariaDB MaxScale                               4.0 kB/s | 6.5 kB     00:01
MariaDB Tools                                 1.6 kB/s | 1.7 kB     00:01
Package perl-DBI-1.641-3.module_el8.1.0+199+8f0a6bbd.x86_64 is already installed
.
Package libaio-0.3.112-1.el8.x86_64 is already installed.
Package libsepol-2.9-1.el8.x86_64 is already installed.
Package lsof-4.91-2.el8.x86_64 is already installed.
Dependencies resolved.

=====
          Package           Arch         Version      Repository      Size
=====
Installing:
  boost-program-options    x86_64       1.66.0-6.el8   AppStream      143 k

Transaction Summary
=====
Install 1 Package

Total download size: 143 k
Installed size: 686 k
Is this ok [y/N]:
```

Choose Y [Enter]

```
# systemctl enable mariadb
# systemctl start mariadb
# mysql_upgrade
```

```
Phase 4/7: Running 'mysql_fix_privilege_tables'
Phase 5/7: Fixing table and database names
Phase 6/7: Checking and upgrading tables
Processing databases
information_schema
performance_schema
test
Phase 7/7: Running 'FLUSH PRIVILEGES'
OK
```

Change MySQL root password and disallow root login remotely.

```
# mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL
MariaDB SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP
CAREFULLY!
```

In order to log into MariaDB to secure it, we'll need the current password for the root user. If you've just installed MariaDB, and haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

```
Switch to unix_socket authentication [Y/n] [Enter]
Enabled successfully!
Reloading privilege tables..
... Success!
```

```
You already have your root account protected, so you can safely answer 'n'.
```

```
Change the root password? [Y/n] [Enter]
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!
```

```
By default, a MariaDB installation has an anonymous user, allowing anyone to log into MariaDB without having to have a user account created for them. This is intended only for testing, and to make the installation go a bit smoother. You should remove them before moving into a production environment.
```

```
Remove anonymous users? [Y/n] [Enter]
... Success!
```

```
Normally, root should only be allowed to connect from 'localhost'. This ensures that someone cannot guess at the root password from the network.
```

```
Disallow root login remotely? [Y/n] [Enter]
... Success!
```

```
By default, MariaDB comes with a database named 'test' that anyone can access. This is also intended only for testing, and should be removed before moving into a production environment.
```

```
Remove test database and access to it? [Y/n] [Enter]
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!
```

```
Reloading the privilege tables will ensure that all changes made so far will take effect immediately.
```

```
Reload privilege tables now? [Y/n] [Enter]
... Success!
```

```
Cleaning up...
```

```
All done! If you've completed all of the above steps, your MariaDB installation should now be secure.
```

```
Thanks for using MariaDB!
```

D. NGINX INSTALLATION

```
# dnf install nginx
```

```

Installed:
  nginx-1:1.14.1-9.module_e18.0.0+184+e34fea82.x86_64
  nginx-all-modules-1:1.14.1-9.module_e18.0.0+184+e34fea82.noarch
  nginx-mod-http-image-filter-1:1.14.1-9.module_e18.0.0+184+e34fea82.x86_64
  nginx-mod-http-perl-1:1.14.1-9.module_e18.0.0+184+e34fea82.x86_64
  nginx-mod-http-xslt-filter-1:1.14.1-9.module_e18.0.0+184+e34fea82.x86_64
  nginx-mod-mail-1:1.14.1-9.module_e18.0.0+184+e34fea82.x86_64
  nginx-mod-stream-1:1.14.1-9.module_e18.0.0+184+e34fea82.x86_64

Complete!
[root@deimos ~]#

```

Start nginx and php-fpm service.

```

# systemctl enable php-fpm
# systemctl start php-fpm
# systemctl enable nginx
# systemctl start nginx

```

Open http port.

```

# firewall-cmd --perma --add-port=80/tcp
# firewall-cmd --reload

```

Check nginx and test PHP on nginx with create file info.php.

```
# nano /usr/share/nginx/html/info.php
```

```

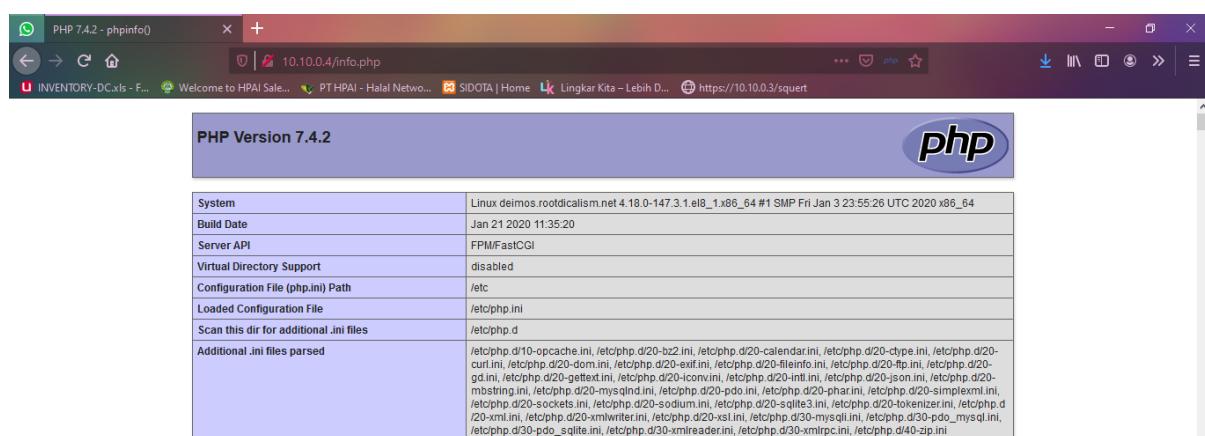
root@deimos:~#
GNU nano 2.9.8          /usr/share/nginx/html/info.php      Modified
<?php phpinfo();?>

```

Access it from browser. My IP Address is <http://10.10.0.4/>.



Check info.php. <http://10.10.0.4/info.php>.



Webserver running well.

Add non sudoer user for webadmin.

```
# adduser wpuser
# passwd wpuser
```

Create web directory in wpuser's home directory. Set permission and owner.

```
# mkdir -p /home/wpuser/rootdicalism.site/logs
# chown -R wpuser:wpuser /home/wpuser/rootdicalism.site/
# chmod -R 755 /home/wpuser/rootdicalism.site/
```

Configure php-fpm for user wpuser.

```
# cp /etc/php-fpm.d/www.conf /etc/php-fpm.d/wpuser.conf
# nano /etc/php-fpm.d/wpuser.conf
```

Edit these line below and follow the instruction.

Line 4 => [wpuser]

Line 24 => user = wpuser

Line 26 => group = wpuser

Line 38 => listen = /run/php-fpm/wpuser.sock

Line 48 - 50 => Comment this line

Line 55 => listen.acl_users = apache,nginx,wpuser

Line 325 => slowlog = /var/log/php-fpm/wpuser-slow.log

Line 423 => php_admin_value[error_log] =
/home/wpuser/rootdicalism.site/logs/php-error.log

```
# nano /etc/php.ini
```

Set PHP Timezone. Find this section.

```
; Defines the default timezone used by the date functions
; http://php.net/date.timezone
date.timezone = Asia/Jakarta
```

Disable CGI FIX PATHINFO. Find this line, uncomment, and change 1 to 0 like this.

```
cgi.fix_pathinfo=0
```

Restart php-fpm service.

```
# systemctl restart php-fpm
```

Create Nginx Server Block and copy the configuration below.

```
# nano /etc/nginx/conf.d/wordpress.conf

error_log /home/wpuser/rootdicalism.site/logs/error.log;
log_format wpuser '$remote_addr - $remote_user [$time_local] "$request"
$status $body_bytes_sent "$http_referer"
"$http_user_agent" "$http_x_forwarded_for"';
access_log /home/wpuser/rootdicalism.site/logs/access.log wpuser;
server {
    listen 80;
    server_name 10.10.0.4;
    root /home/wpuser/rootdicalism.site/wordpress;

    location / {
        index index.php index.html;
    }
}
```

```
location ~ \.php$ {
    include /etc/nginx/fastcgi_params;
    fastcgi_pass unix:///run/php-fpm/wpuser.sock;
    fastcgi_index index.php;
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
}
```

E. WORDPRESS INSTALLATION

Download the latest version of wordpress and unzip into user's web directory.

```
# cd /home/wpuser/rootdicalism.site/  
# wget https://wordpress.org/latest.zip  
# unzip latest.zip
```

Set wordpress directory owner and create wp-config.php file

```
# chown -R wpuser:wpuser wordpress
# cd wordpress
# cp wp-config-sample.php wp-config.php
# nano wp-config.php
```

Edit the wp-config.php file like this.

```
[root@deimos:~]# root@deimos:/home/wpuser/rootdicalism.site/wordpress
GNU nano 2.9.8 wp-config.php

// ** MySQL settings - You can get this info from your web host ***
/** The name of the database for WordPress */
define( 'DB_NAME', 'db_wordpress' );

/** MySQL database username */
define( 'DB_USER', 'userdb_wp' );

/** MySQL database password */
define( 'DB_PASSWORD', 'LoadIn9321!@#' );

/** MySQL hostname */
define( 'DB_HOST', 'localhost' );

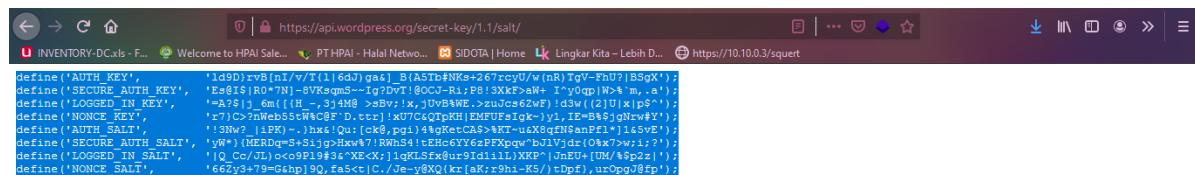
/** Database Charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );

/** The Database Collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );

/**#@+

```

Get wordpress salt from the wordpress API. Visit <https://api.wordpress.org/secret-key/1.1/salt/>.



And replace this wp-config.php file section with salt above.

```
define( 'AUTH_KEY', 'put your unique phrase here' );
define( 'SECURE_AUTH_KEY', 'put your unique phrase here' );
define( 'LOGGED_IN_KEY', 'put your unique phrase here' );
define( 'NONCE_KEY', 'put your unique phrase here' );
define( 'AUTH_SALT', 'put your unique phrase here' );
define( 'SECURE_AUTH_SALT', 'put your unique phrase here' );
define( 'LOGGED_IN_SALT', 'put your unique phrase here' );
define( 'NONCE_SALT', 'put your unique phrase here' );

/**#@-*/
```

Modify wp-config.php file owner.

```
# chown wpuser:wpuser wp-config.php
```

Set file permission

```
# find . -type f -exec chmod 0664 {} \;
# find . -type d -exec chmod 0775 {} \;
```

Create database for wordpress and the database user like you defined it in wp-config.php file.

```
[root@deimos wordpress]# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 31
Server version: 10.4.11-MariaDB MariaDB Server

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)]> create database db_wordpress;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> create user userdb_wp@localhost identified by 'Loadln9321!@#';
Query OK, 0 rows affected (0.006 sec)

MariaDB [(none)]> create user 'userdb_wp'@'%' identified by 'Loadln9321!@#';
Query OK, 0 rows affected (0.005 sec)

MariaDB [(none)]> grant all on db_wordpress.* to userdb_wp@localhost;
Query OK, 0 rows affected (0.004 sec)

MariaDB [(none)]> grant all on db_wordpress.* to 'userdb_wp'@'%';
Query OK, 0 rows affected (0.009 sec)

MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.001 sec)

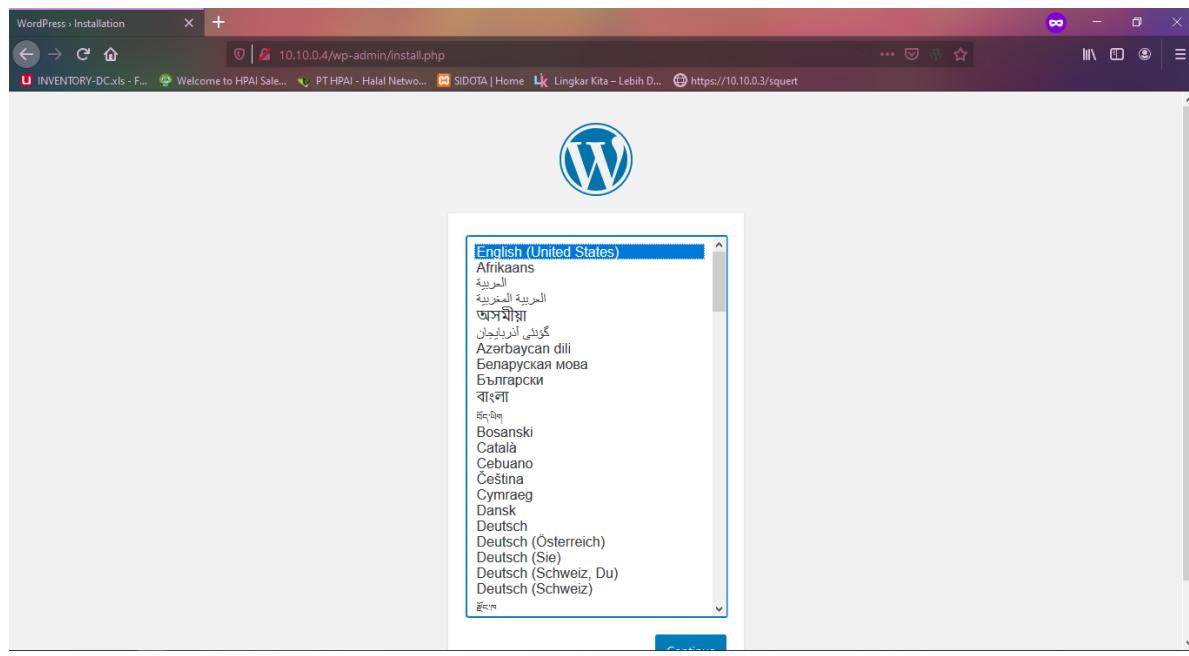
MariaDB [(none)]> show databases;
+-----+
| Database      |
+-----+
| db_wordpress  |
| information_schema |
| mysql          |
| performance_schema |
+-----+
4 rows in set (0.001 sec)

MariaDB [(none)]> 
```

Restart nginx service.

```
# systemctl restart nginx
```

Access your wordpress from browser. <http://10.10.0.4/>. Begin wordpress installation.



Chose Wordpress Language.

WordPress > Installation

10.10.0.4/wp-admin/install.php?step=1

INVENTORY-DC.xls - F... Welcome to HPAI Sale... PT HPAI - Halal Netwo... SIDOTA | Home Lingkar Kita - Lebih D... https://10.10.0.3/sequit

on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Don't worry, you can always change these settings later.

Site Title Root-dicalism

Username rootdicalism

Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password Show Strong

Important: You will need this password to log in. Please store it in a secure location.

Your Email rootdicalism@somewhere.com

Double-check your email address before continuing.

Search Engine Visibility Discourage search engines from indexing this site

It is up to search engines to honor this request.

Install WordPress

Site setting.

The screenshot shows a Microsoft Edge browser window. The title bar reads "WordPress > Installation". The address bar shows the URL "10.10.0.4/wp-admin/install.php?step=2". The main content area displays the WordPress logo and a success message: "Success! WordPress has been installed. Thank you, and enjoy!". Below this, it shows the configuration details: "Username" is "rootdicalism" and "Password" is "Your chosen password.". At the bottom is a blue "Log In" button.

Installation Success.

A screenshot of a Microsoft Edge browser window. The address bar shows the URL '10.10.0.4'. The page content is a WordPress blog post titled 'Hello world!'. The post is categorized as 'Uncategorized' and was published on January 25, 2020. The post content is a simple welcome message: 'Welcome to WordPress. This is your first post. Edit or delete it, then start writing!'. There is an 'Edit' button at the bottom of the post.