Special Workshop in IT

Week 1: Installing Frappe

Sirasit Lochanachit, PhD

Today's Objectives

1) Installation of Prerequisites
    a) VirtualBox
    b) Ubuntu 20.04 LTS
2) Frappe Framework Installation
3) Starting and testing the Bench server environment

Installation of Prerequisites

VirtualBox

- Free and open source virtualization software
- Allows creation and management of "guest" virtual machines running Windows, Linux and etc.
    - Enables users and administrators to easily run multiple guest operating systems on a single host
- "Guest Additions" package of device drivers and system applications is available, which typically improves performance, especially that of graphics

Ubuntu

- Free (mostly) and open source operating system on Linux
- Generally used in an enterprise server, desktop, cloud, and IoT
- As of October 2020, the most recent long-term support (LTS) version is 20.04, which is supported until 2025.

Setting up a VM

1. Download and install VirtualBox
2. Download Ubuntu ISO
3. Install Ubuntu using VirtualBox

Step 3: Install Ubuntu using VirtualBox

1. Select Minimal install
2. After Installation has completed, select Device > Insert Guest Additions CD image..., then click run
3. This will open a terminal
4. To install a required kernel modules, run

   `sudo apt install build-essential dkms linux-headers-generic`

   `sudo rcvboxadd setup`
5. Restart Ubuntu
6. VM's screen resolution can be changed, including drag and drop setting

Prerequisites

```
Python 3.6+
Node.js 12
Redis 5                                 (caching and realtime updates)
MariaDB 10.3.x / Postgres 9.5.x         (to run database driven apps)
yarn 1.12+                              (js dependency manager)
pip 20+                                 (py dependency manager)
wkhtmltopdf (version 0.12.5 with patched qt)  (for pdf generation)
cron                                    (bench's scheduled jobs: automated certificate renewal, scheduled
backups)
NGINX                                   (proxying multitenant sites in production)
```

Steps

1. Update System

   `sudo apt update && sudo apt upgrade`

   `sudo apt -y install software-properties-common`
2. Install git, python, and redis

   `sudo apt install git python-dev redis-server`
3. Install MariaDB
4. Install Node.js
5. Install other utilities

Step 3: Install MariaDB

3.1 Import MariaDB gpg key

`sudo apt-key adv --fetch-keys 'https://mariadb.org/mariadb_release_signing_key.asc'`

- This will add repository key to the system

3.2 Add MariaDB APT repository

`sudo add-apt-repository 'deb [arch=amd64] http://mariadb.mirror.globo.tech/repo/10.5/ubuntu focal main'`

Step 3: Install MariaDB

3.3 Install MariaDB server on Ubuntu

`sudo apt update`

`sudo apt install mariadb-server mariadb-client`

- Hit the ‘y’ key to accept installation of MariaDB 10.5 on Ubuntu

3.4 Secure MariaDB server

`sudo mysql_secure_installation`

- Enter root password
- Can change password of MariaDB to a new one (123) - need to remember!
- Continue to hit the ‘y’ key

Step 3: Install MariaDB

3.5 Check if database service is started

`systemctl status mysql`

Press Enter to show terminal console

3.6 Try login to MariaDB

`sudo mysql -u root -p`

3.7 On MariaDB shell, check version by entering

`SELECT VERSION();`

Press Ctrl-C to exit MariaDB shell

3.8 Install MySQL DB development files

`sudo apt install libmysqlclient-dev`

Step 3: Install MariaDB

3.9 In order to enable MariaDB to work with Frappe development framework

`sudo nano /etc/mysql/my.cnf`

Then add this configuration

```
[mysqld]
character-set-client-handshake = FALSE
character-set-server = utf8mb4
collation-server = utf8mb4_unicode_ci

[mysql]
default-character-set = utf8mb4
```

3.10 Hit Ctrl-X followed by Y and Enter, and then Restart the MySQL service

`service mysql restart`

Step 4: Install Node.js

4.1 Install curl

`sudo apt install curl`

4.2 Install nvm (Node Version Manager)

`curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.33.11/install.sh | bash`

4.3 Open another terminal, verify nvm installation

`command -v nvm`

4.4 install Node version 12

`nvm install 12`

Step 4: Install Node.js

4.5 Verify the node installation

`node -v`

4.6 Install yarn (JS package manager) using npm (JS package manager)

- Yarn caches every package it has downloaded, allow faster installs
- Yarn guarantees that any installation that works on one system will work on another system

`npm install -g yarn`

Step 5: Install other utilities

`sudo apt install xvfb libfontconfig wkhtmltopdf`

Bench Installation

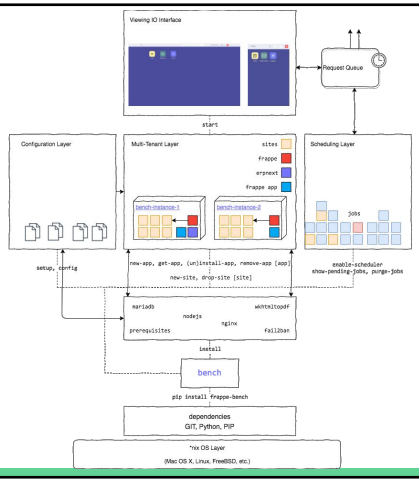
Steps

1. Bench can be installed via pip3, so need to install pip3 first

   `sudo apt install python3-pip`
2. Install bench

   `sudo pip3 install frappe-bench`

Overview



Frappe Framework Installation (Manual)

Frappe-Bench

- Bench is the command line tool to manage Frappe apps and sites.
- To create a project folder which contain apps and sites, run

  `bench init frappe-bench`

  - This will create a directory named “frappe-bench” in current working directory
  - Then, it creates a python virtual environment directory (env)
  - Fetch and install “frappe” app as a python package
  - Install node modules of “frappe”
  - Build static assets.

Frappe-Bench

```
.
├── Procfile
├── apps
│   └── frappe
├── config
│   ├── pids
│   ├── redis_cache.conf
│   ├── redis_queue.conf
│   └── redis_socketio.conf
├── env
│   ├── bin
│   ├── include
│   ├── lib
│   └── share
├── logs
│   ├── backup.log
│   └── bench.log
└── sites
    ├── apps.txt
    ├── assets
    └── common_site_config.json
```

- **env**: Python virtual environment
- **config**: Config files for Redis and Nginx
- **logs**: Log files for every process (web, worker)
- **sites:** Sites directory
  - **assets:** Static assets that served via Nginx in production
  - **apps.txt:** List of installed frappe apps
  - **common*site*config.json:** Site config that is available in all sites
- **apps:** Apps directory
  - **frappe:** The Frappe app directory
- **Procfile:** List of processes that run in development

Start the Bench Server

- To start the Frappe web server, go to frappe-bench directory

  `cd frappe-bench`

  `bench start`

  - This will create several processes:
    - Python web server based on Gunicorn
    - Redis servers for caching, job queuing and socketio pub-sub
    - Background workers
    - Node server for socketio
    - Node server for compiling JS/CSS files

Bench Server

- The web server will start listening on the port 8000
  - However, we don’t have any sites yet
  - Next step is to create an app and a site that will have this app installed (Next week!)
- When bench server is running, do not close the terminal!