**Frappe Framework**

Frappe, pronounced fra-pay, is a full stack, batteries-included, web framework written in Python and Javascript with MariaDB as the database. It is the framework which powers ERPNext, is pretty generic and can be used to build database driven apps.

The key difference in Frappe compared to other frameworks is that meta-data is also treated as data. This enables you to build front-ends very easily. We believe in a monolithic architecture, so Frappe comes with almost everything you need to build a modern web application. It has a full featured Admin UI called the Desk that handles forms, navigation, lists, menus, permissions, file attachment and much more out of the box.

**Pre-requisites:**

Python 3.7+ (version 13) / Python 3.8+ (develop branch)

Node.js 14

Redis 6 (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)

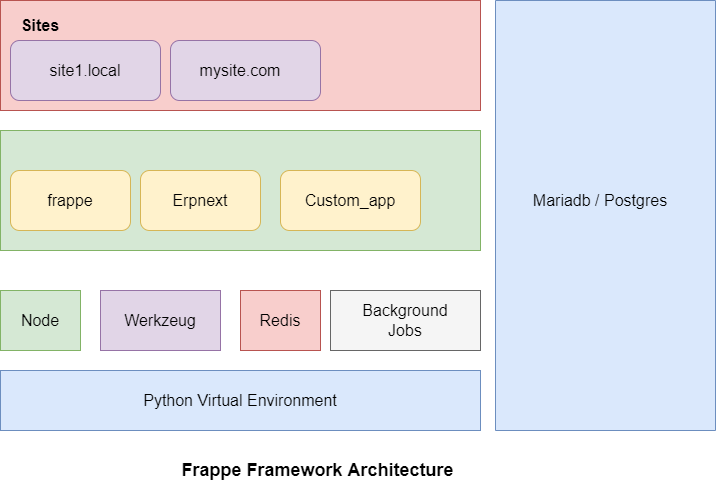
Bench CLI Installation

Sudo – H pip3 install frappe-bench

Bench is a CLI tool to manage Frappe Deployments. It provides an easy interface to help you setup and manage multiple sites and apps based on Frappe Framework.

It handles app updates, database migrations, generating configs for nginx and supervisor, scaffolding new apps and much more.

**Architecture**



As frappe framework is managed through bench CLI.

There were few security related commands comes under bench setup CLI.

* **manager**: Setup bench-manager.local site with the [Bench Manager](https://github.com/frappe/bench_manager) app, a GUI for bench installed on it.
* **production**: Setup a Frappe production environment for the specified user. This installs ansible, NGINX, supervisor, fail2ban and generates the respective configuration files.
* **nginx**: Generate configuration files for NGINX
* **fail2ban**: Setup fail2ban, an intrusion prevention software framework that protects computer servers from brute-force attacks
* **firewall**: Setup firewall for system
* **ssh-port**: Set SSH Port for system
* **reload-nginx**: Checks NGINX config file and reloads service
* **supervisor**: Generate configuration for supervisor
* **lets-encrypt**: Setup lets-encrypt SSL for site
* **wildcard-ssl**: Setup wildcard SSL certificate for multi-tenant bench.

As Bench CLI plays a vital role for managing frappe framework, Here is the link to bench commands and site commands, which will be useful in setting up and managing the frappe apps.

[Bench Commands (frappeframework.com)](https://frappeframework.com/docs/v13/user/en/bench/bench-commands)

Site manager is GUI resource, which will be useful, if we’ve multi-tenancy sites.

As multi-tenancy sites works on port based and DNS based routing.

This Site manager will provide graphical user interface to manage the multiple sites using web interface.

By using above parameters we can secure ERPNext.

As frappe framework will be supports \*nix like operating systems only, we can take granular control of everything that needs to protect the application from any security breaches.

Mostly we could consider security on the Network and Application side.

By changing the defaults ports, configuring the fail2ban, setting up firewall, etc. we can achieve networking side security.

On the Other hand, Application side security can be provided by using proper Authentication, Authorization, Encryption, logging, and application security testing.

As security is one the important component of an application. Each layer of network and application side will be secured to an extent.