Web Runbook

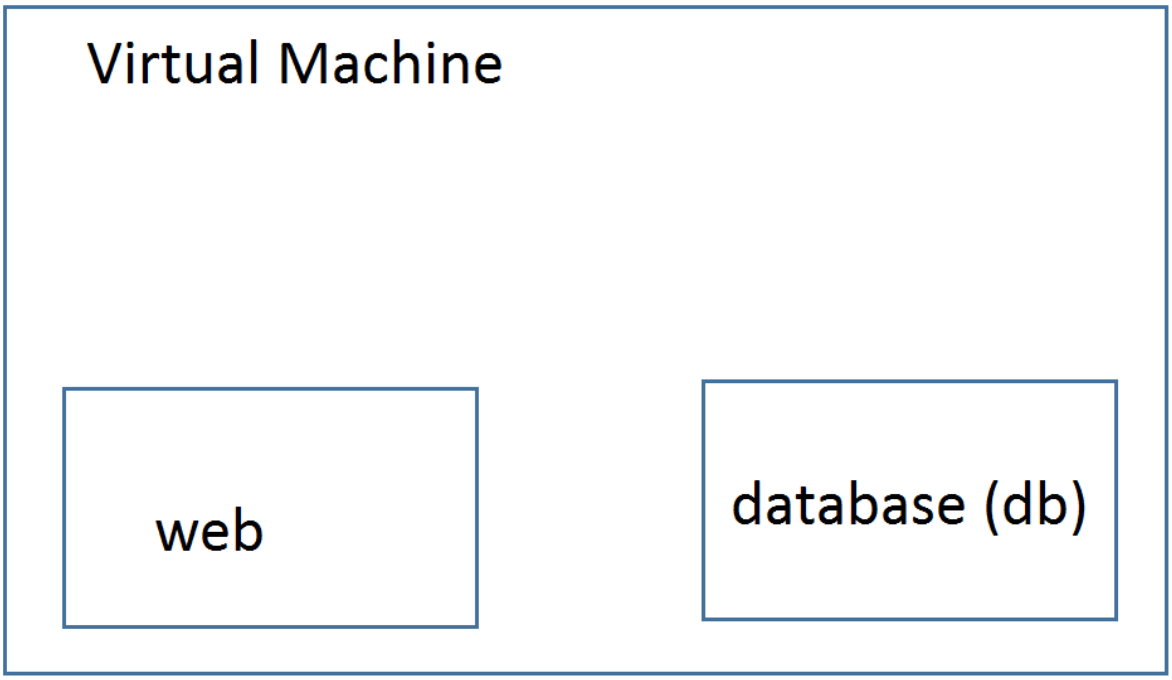
Short Description

The web playbook configures, installs, and updates all of the necessary files need to set up all the necessary web services.

Required Software

The software required to run web.yml includes SElinux, epel, nginx including modules, nginx,php,php-fpm,php-ldap,php-mbstring,php-mcrypt,php-mysql,php-phpunit-PHPUnit, LDAP, composer, vhost, fastCGI process manager (when nginx is changed),

Architecture Diagram



Deployment

To deploy web.yml the following tools and commands will need to be used:

First task disables SELinux and Becomes Root
Accepted Output: ok: [localhost]

Put SELinux in permissive mode and becomes Root
Accepted Output: ok: [localhost]

Install Epel-Release and update cache
Accepted Output: changed: [localhost]

Task installs latest nginx modules, framework PHP, LDAP, one way hashing, MYSQL database, phpUNIT programming framework
Accepted Output: ok: [localhost]

- The original command needed to be changed which initially used curl to download and install composer, to make the task run, I deconstructed it into two steps

Download Composer
Accepted Output: ok: [localhost]

Install Composer
Accepted Output: changed: [localhost]

Copy configuration file of Nginx to destination file and become root
Accepted Output: ok: [localhost]

Create Vhost for site and become root
Accepted Output: ok: [localhost]

Template PHP initiation file, use mode 0644
Accepted Output: ok: [localhost]

Copy configuration file into destination file and become root
Accepted Output: ok: [localhost]

Run idempotent action that starts nginx on initial boot
output: skipping: [localhost] -

Restart FastCGI process manager when nginx is changed
output: skipping: [localhost]

Execute service_dir: /usr/share/nginx/{{ service_name }}
Accepted Output: ok: [localhost]

unpack service_name using mode 0755
Accepted Output: ok: [localhost]

Become root and issue composer command then update service_dir
Accepted Output: changed: [localhost]

Run Shell of service_dir with root privileges
Accepted Output: changed: [localhost]

> *For the next two commands, which initially threw out an error,*
> *I realized I had to break up the command into separate tasks.*

Create Nginx Group
Accepted Output: ok: [localhost]

Create Nginx User
Accepted Output: ok: [localhost]

Create Nginx data directory
Accepted Output: changed: [localhost]

Template word press env.j2 file into destination file service_dir with mode 0644 and owner nginx
Accepted Output: ok: [localhost]

Issues

Title: System Overload
Description: A server overloads can be caused by a large bottleneck influx of traffic which clogs the network. Typically, this happens during peak traffic hours.
Remediation Steps: Do a deep capacity analysis of the network traffic, to ensure there is a balanced ratio of virtual servers, examining the loads on the CPU and memory to the amount of “work” needed on each server. Sometimes this means improving the physical hardware to support a higher capacity.

Title: DDos attack

Description: Distributed Denial of Service attack basically bombards a server with an overflow of packets that the server is not able to handle create a bottleneck state, which makes the service (server) become unavailable.
Remediation Steps: Several technical things can be done, 1) rate limit your router to prevent the server from being overwhelmed, add filters to tell the router to drop packets from “suspicious sources” 3) timeout half-open connects more aggressively, 4) drop spoofed and malformed packets. If the problem persists, call a DDoS specialist to analyze your network setup.

Title: Natural Disaster
Description: Natural Disaster wipes out all your electricity thereby taking your severs down.
Remediation Steps: Possibly look into investing in cloud technology some kind of SaaS type of technology. For example, with Amazon there are server farms, which in case “your server” goes down there is redundancy so that it is backed up on another server which takes command.