Virtual Machine Infrastructure

Overall Goal

The overall goal of this project is to be able to maintain virtual machines easier and provide a secure and stable environment for docker-compose stacks and other binaries to run. One requirement is virtual machines should be generated on demand to allow for scalability. This could be achieved by netbooting a Linux operating system, configuring the machine with cloud-init and then deploying a docker-compose stack on it.

> Note that OPNSense and the WIndows Machine are not discussed in this document. While they do exist, they will not be a part of this infrastructure

Final Solution

Generating VM's on Demand

Using the power of the Debian Live project, we are able to customize a live netboot image to provide a base operating system for all VMs. This will also allow us to generate "live cds" on demand and then push the update to a central boot server where the VMs can then boot the latest copy with the latest kernel and bug fixes. See Gitea repo here

Using Cloud-Init

The Xen Orchestra appliance has built in support for using a cloud-init template. Users will be able to specify whichever packages they want installed and other configuration items. Docker compose will be deployed using cloud-init

Ideal Implementation

The Genesis Machine

Function

This machine will provide all other Virtual Machines with the latest image to boot up via PXE. Additionally, this machine will house its own version of Gitea and Jenkins. This machine will also be home to LDAP and Keycloak for identity management since they have both been identified as critical services.

VM Resources

- 2 CPU Cores
- 2 GB of RAM

Networking

- Internal Hostname: genesis.heestand.local
- IP: 192.168.2.10

Containers Used

- Gitea
- Jenkins
- Netbootxyz
- Traefik
- SmallStep
- LDAP
- FusionDirectory
- Keycloak
- Portainer

Other Configurations

- Docker Engine v2
- Encrypted overlay network

Edge Machine

Function

This machine will serve as the edge for incoming traffic. All incoming traffic to all services must go through here

VM Resources

- 1 CPU Cores
- 2 GB of RAM

Networking

- Internal Hostname: edge.heestand.local
- IP: 192.168.2.11

Containers Used

- Traefik
- Fail2ban?

Other Configurations

- Docker Engine v2
- smallstep client (system)
- portainer agent (auto startup docker container)
- Custom SSH shell to portal to different machines?
- Encrypted overlay network

Database Machine

Function

This machine will provide all of the databases for the services machine.

VM Resources

- 8 CPU Cores
- 8 GB of RAM

Networking

- Internal Hostname: database.heestand.local
- IP: 192.168.2.12

Containers Used

- MariaDB
- MongoDB
- Postgres
- Redis
- MySQL
- InfluxDB
- mindmax
- elasticsearch
- Adminer

Other Configurations

- Docker Engine v2
- smallstep client (system)
- portainer agent (auto startup docker container)
- Encrypted overlay network

Prod Service Machine

Function

This machine will provide all of the services

VM Resources

- 2 CPU Cores
- 8 GB of RAM

Networking

- Internal Hostname: prod.heestand.local
- IP: 192.168.2.13

Containers Used

- Traefik
- <Insert other services here>

Other Configurations

- Docker Engine v2
- smallstep client
- Encrypted overlay network

Dev Service Machine

Function

This machine will provide development versions of services

VM Resources

- 1 CPU Cores
- 8 GB of RAM

Networking

- Internal Hostname: dev.heestand.local
- IP: 192.168.2.14

Containers Used

- Traefik
- <Insert other services here>

Other Configurations

- Docker Engine v2
- smallstep client (system)
- portainer agent (auto startup docker container)
- Encrypted overlay network

Logging and Metrics Machine

Function

This machine will be used for logging and managing the virtual machines

VM Resources

- 1 CPU Cores
- 4 GB of RAM

Networking

- Internal Hostname: dev.heestand.local
- IP: 192.168.2.15

Containers Used

- traefik
- grafana
- graylog

Other Configurations

- Docker Engine v2
- smallstep client (system)
- portainer agent (auto startup docker container)

Build Machine

Function

This machine will replace Linuxbox and will be used for development activities. It will also host the big Jenkins build cluster.

VM Resources

- 20 CPU Cores
- 64 GB of RAM

Networking

- Internal Hostname: linuxbox.heestand.local
- IP: 192.168.2.16

Containers Used

Other Configurations

- Jenkins installed directly

Implementation Notes

Operating System Security

Using a live CD with a readonly file system can be tricky, below are the current security precautions that are being taken:

User and Permission Changes

- User "live" (default live user) will have it's password locked to prevent ssh login
- User "live" will be assigned the shell /usr/sbin/nologin to prevent anyone from logging in as that user
- A user will not be allowed to ssh into the system as user "root"
- Console will be disabled to prevent rogue admins and unauthorized access

> Due to the above security measures, you MUST provide a SSH key to the Virtual Machine for the user debian (or whatever user you made via the cloud-init process). If you do not, you

will not be able to remote into the machine!

File System

The file system for the operating system is not encrypted and will be lost on reboot, it is not recommended you store non-critical information on it. Use your dedicated data disk to store information instead.

Revision #7
Created 2023-03-07 23:57:46 UTC by Kyle Heestand
Updated 2024-01-14 01:58:32 UTC by Kyle Heestand