How to: Docker

What is Docker?

Introduction

Docker is a platform that allows you to easily package, deploy, and run applications in a lightweight, portable, and consistent way. It uses a technology called containers to isolate applications and their dependencies (such as libraries and other software they need) from the host system, making it easier to run the application anywhere, on any machine. ლ(□ _□ ლ)

Capabilities of Docker

- **Containers**: Think of containers as small, lightweight "boxes" that hold everything an application needs to run—like the application code, system libraries, and configurations—so it can run anywhere without problems. A container is isolated, meaning it doesn’t interfere with other applications on the same system.
- **Portability**: Once you package an application in a Docker container, you can move it across different environments (like your local computer, a developer's laptop, or a server in a data center). As long as Docker is installed, the application will run the same way, no matter where it is.
- **Consistency**: Docker ensures that the application behaves the same way in every environment. It eliminates the problem of “it works on my machine,” where an app runs perfectly on one computer but not on another because of differences in setup or installed software.
- **Efficient Use of Resources**: Containers are lightweight compared to traditional virtual machines because they share the same operating system kernel. This makes them faster to start and use fewer resources, making them more efficient for running multiple applications.
- **Easy Deployment**: With Docker, you can define all the settings, libraries, and services an application needs in a file called a Dockerfile. This file can be shared with others or used to automatically build containers in any environment.

Last Updated: 2/19/2025
BY: Lilian

1.0 Installation and Swarm Initialization

Materials / Pre-Setup

1. Create new Ubuntu Server VM on the Proxmox server (Refer to Create a VM in How to: Proxmox)

*****TO BE ADDED*

2.0 Adding Users to Docker Group

**Must be done on each docker node if there's more than one*

> ***Any confusion on command syntax/structure can be clarified in Legend**

1. Login to the docker node
2. Add the user

> ***Set the password to something easy i.e. password // can also be changed using the 'passwd' command**

```
sudo adduser [USER]
```

3. Give user permission to the docker and sudoers group

```
sudo usermod -aG docker [USER]
sudo usermod -aG sudo [USER]
```

4. Connect to user’s account and import user’s Github keys

```
su [USER]
ssh-import-id-gh [GITHUB USERNAME]
```

5. Test whether or not the user can connect to each docker swarm:

```
ssh [USER]@[IP ADDRESS]
```

6. Have user change their password NOT ON ROOT on each docker node created:

```
passwd
```

Last Updated: 2/22/2025
Contributors: Lilian, Vivian