DAYS

Containe

Containers run apps natively on the host machine's kernel. They have better performance characteristics than virtual machines that only get virtual access to host resources through a hypervisor. Containers can get native access, each one running in a discrete process, taking no more memory than any other executable.

Docke

as we know,is an open platform for developers and sysadmins to build, ship, and run distributed applications, whether on laptops, data center VMs, or the cloud.Docker provides an API for interacting with the Docker daemon (called the Docker Engine API).The Docker APIhas allowed limitless options for interfacing with Docker engine, containers, and images to emerge from CLIs to desktop applications and web-based management tools. Everything the Docker client can do can be done with the API.Below GUI tools extensively uses API to interface with Docker engine.

Tools for managing Docker environments

1.Kitematic
Kitematic is a simple application for managing Docker containers on Mac, Linux and Windows.It is an open source project built to simplify and streamline using Docker on a Mac or Windows PC.

2.Portainer
Portainer is a simple management solution for Docker,it consists of a web UI that allows you to easily manage your Docker containers, images, networks and volumes.

Tools for managing Docker environments

3.DockStation
DockStation is a developer-centric application for managing projects based on Docker. Instead of lots of CLI commands you can monitor, configure, and manage services and containers while using just a GUI.

4.Shipyard
Built on Docker Swarm, Shipyard gives you the ability to manage Docker resources including containers, images, private registries and more.

5.Docker Compose UI
Docker Compose UI is a web interface for Docker Compose.Its minimal HTTP API on top of Docker Compose while maintaining full interoperability with Docker Compose CLI.5.Docker Compose UI

Install and Configure Portainer

How to Manage Docker Containers using Portainer on Ubuntu

Portainer can be installed as a docker container and standalone without docker container.

```
docker pull portainer/portainer
```

```
mariam@mariam-VirtualBox:~$ sudo docker pull portainer/portainer
Using default tag: latest
latest: Pulling from portainer/portainer
d1e017099d17: Pull complete
fac26901c311: Pull complete
Digest: sha256:cc226d8a06b6d5e24b44a4f10d0d1fd701741e84a852adc6d40bef9424a000ec
Status: Downloaded newer image for portainer/portainer:latest
```

To run Portainer using the simple docker command below.

```
docker run -d -p 9000:9000 -v /var/run/docker.sock:/var/run/docker.sock portainer/portainer
```

Portainer is now running as a container, check it using the docker ps command.

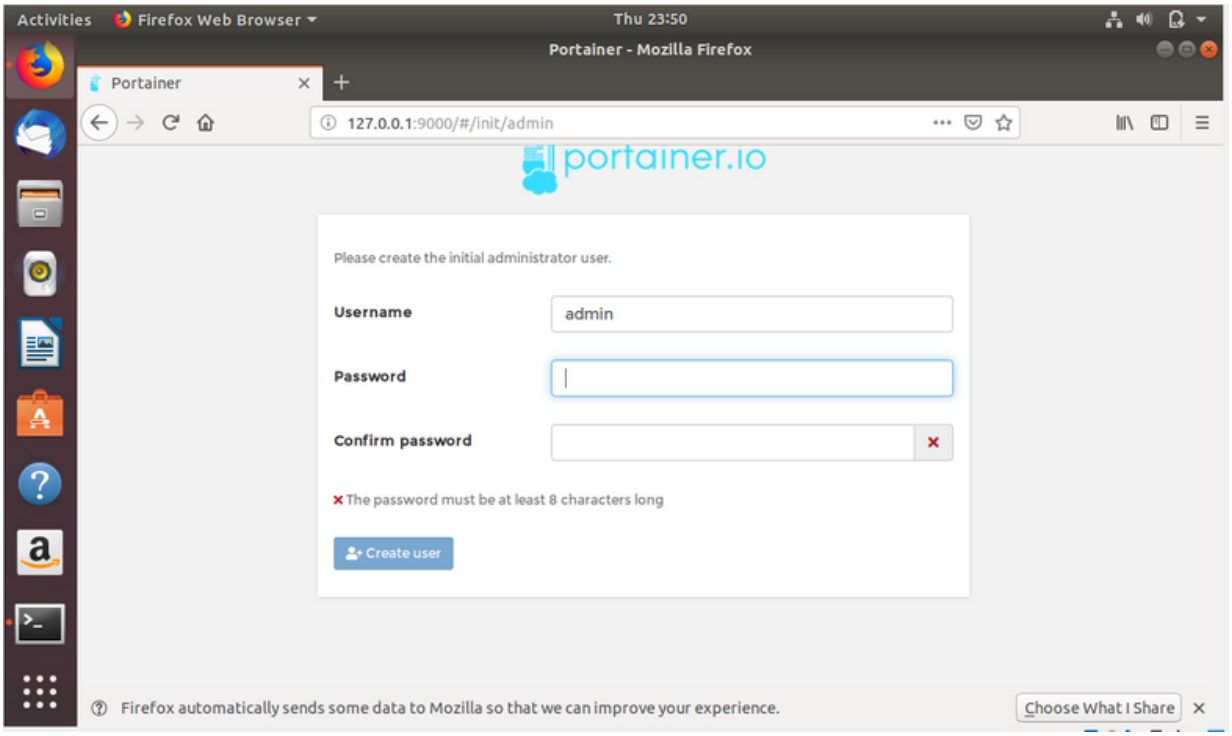
```
docker ps
```

```
mariam@mariam-VirtualBox:~$ sudo docker run -d -p 9000:9000 -v /var/run/docker.s
ock:/var/run/docker.sock portainer/portainer
3a48d187a0ff935895429842dab248c0647c96853f203728f97e2a23155b9279
mariam@mariam-VirtualBox:~$ sudo docker ps
CONTAINER ID        IMAGE                  COMMAND             CREATED
  STATUS              PORTS                    NAMES
3a48d187a0ff        portainer/portainer    "/portainer"        44 seconds ago
  Up 19 seconds       0.0.0.0:9000->9000/tcp   musing_robinson
```

How to Manage Docker Containers using Portainer on Ubuntu

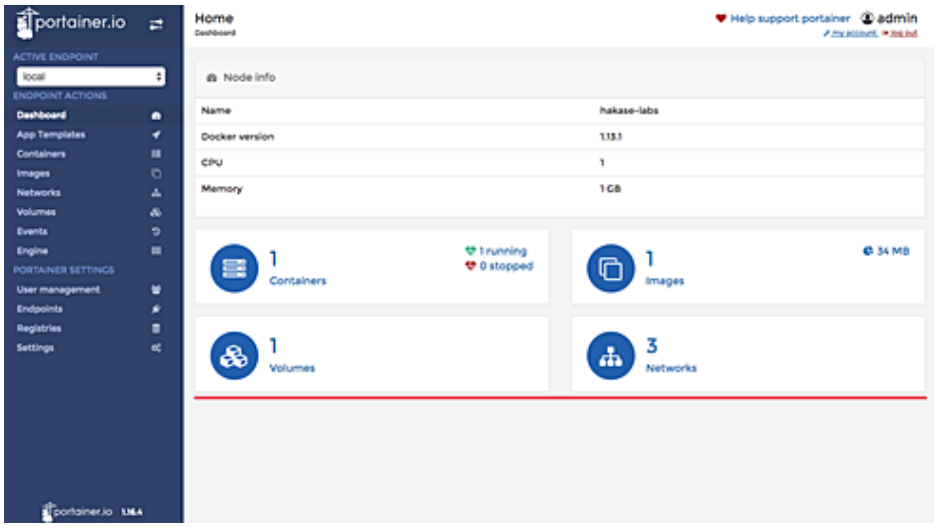
Portainer is now running as a Docker container under port 9000.
Next, we will configure the Admin password for the Portainer.
Open your web browser and type the server IP address with port 9000.
YOU can check server ip add by : ip add show

http://127.0.0.1:9000/



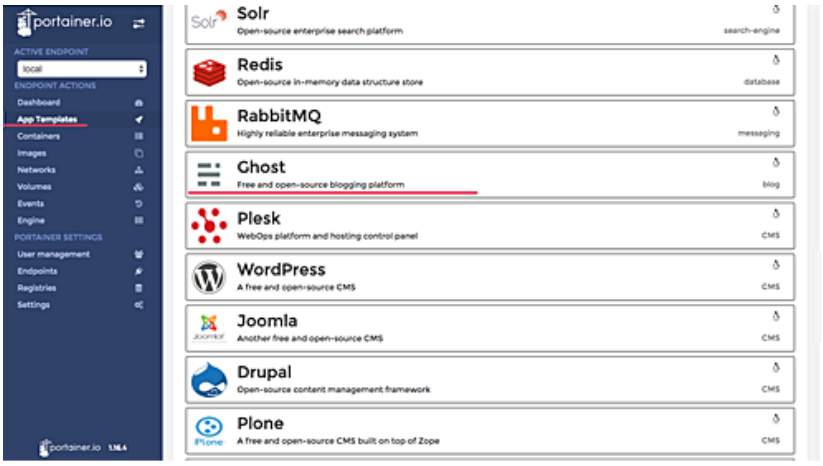
You will get the page about the admin user and password configuration.

Choose the 'Local' environment and click 'Connect' button.
And now you will see the Portainer Admin Dashboard.



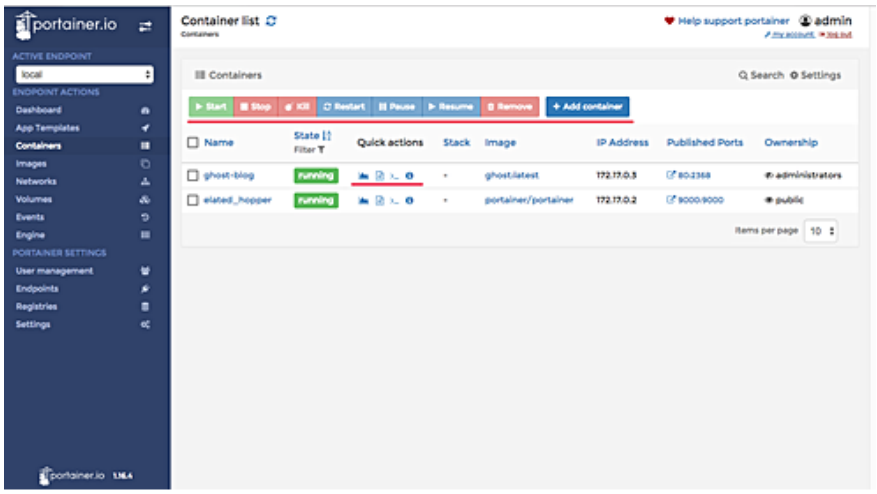
Deploy New App Container

After the Portianer installation, we will run the Application Container using Portainer.
Click the 'App Template' menu.
Now choose the application that you want to install.



Manage Docker Environment Using Portainer

In this step, we will configure Docker Environments such as Docker images, Container, Volumes, and Networks.



We can start, stop, restart, create a new container, access the shell of the container, see the container log, and stats of the container from this Portainer container management.