International Journal of Innovative Research in Computer Science & Technology (IJIRCST)
ISSN: 2347-5552, Volume-6, Issue-3, May- 2018
DOI: 10.21276/ijircst.2018.6.3.1

Deployment of Docker Services on Private Cloud

Prof. Rakesh Kadu, Shikha Agrawal, Swinal Kamble, Tejaswini Bhardwaj

Abstract **- The proposed paper presents the topic "Deployment of Docker Services on Private Cloud". It briefs how the system is implemented to provide Docker services to client, in order to reduce client's overall system load and to make the client enjoy the application layer facilities easily. The paper tells how this system uses Docker Hub which works on the concept of Container as a Service, thus making the task of creating, deploying and running applications simpler and light weighted. Docker Server in this system acts as a bridge between Docker client and Docker Hub. Client requests server for applications, server fetches the images from Docker Hub via Docker Engine which is responsible for packing these images along with its dependencies and binaries into a Container. Client can access these applications using web browser (URL,Tags). Traditionally, server could access Docker through Command Line Interface (CLI); hence to make it simpler, we present this paper illustrating creation of User Interface (UI) for the server side. Thus this paper summarizes how Docker services are deployed on private cloud**

Keyword – Cloud, Docker, Container.

I. INTRODUCTION

Docker is an open source tool that packs the images of applications into containers and serves these containers to users. Containers are images of applications stored along with all its binaries and libraries into a packet, making them platform independent. These containers run remotely on top of the operating system's kernel. Even though, container technologies have been around for over 10 years, docker is right now a buzz amongst the best innovations, since it accompanies new capacities that prior technologies lacked. It gives the facility to create, pull, push and control containers. Besides that, developers can pack applications into lightweight docker containers. The applications in these containers can easily be worked anywhere without any alteration. Moreover, docker can convey much more virtual situations than other innovations, on the same equipment. In short, docker can easily deploy and manage docker containers. Docker's Architecture includes three main components of docker:

Manuscript Received April 25, 2018.

Prof. Rakesh Kadu, Shri Ramdeobaba College of Engineering and Management, Nagpur,Maharashtra, India
Shikha Agrawal, Swinal Kamble, Tejaswini Bhardwaj, Shri Ramdeobaba College of Engineering and Management, Nagpur, Maharashtra, India
(email: kamblesm@rknec.edu)

A. Docker client and server

The docker server fulfills the docker client's request from docker host through API or a Command Line Interface. The docker server is also named as Docker daemon. Docker client and server can run on same machine or
Client can be a remote docker client. Docker server is responsible to pull, push and manage resources for client.

B. Docker Registry

Docker registry stores docker images and enables complete image management workflow. Users looking for zero-maintenance are encouraged to use Docker Hub, which provides a fee-to-use hosted Registry. While users looking for commercially supported version should look into Docker Trusted Registry.

C. Docker Host

Docker image is wrapped with container that holds the whole kit required to run the application in an isolated environment.

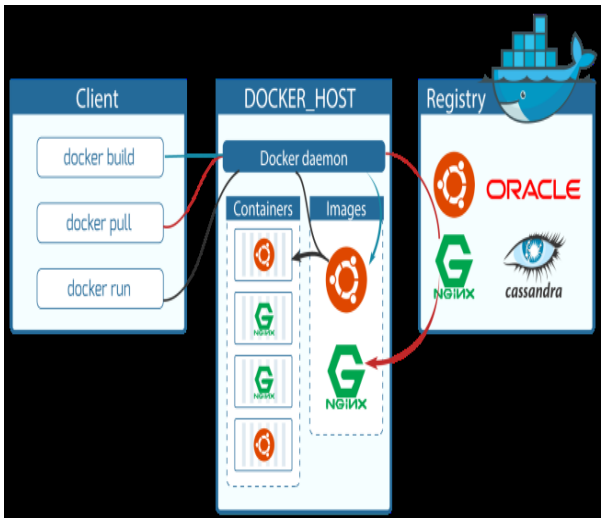

Figure 1: Docker Architecture

II. RELATED WORK

This section presents some of the existing Docker User Interfaces.
Kinematic [1] is a UI that ships with docker for Mac and Windows machines. It is an open source System. Kitematic automates the Docker installation and setup process and provides an intuitive graphical user interface (GUI) for running Docker containers. It integrates with Docker Machines to provision a Virtual Box VM and install the Docker Engine locally on your machine.
Shipyard [2] is another Docker web based user interface that starts with running command that handles multiple operating systems:
Curl –s https://shipyard-project.com/deploy|bash –s

The only feature that makes Shipyard standout is the container creation screen and the ability to add private repositories and it is also the easiest to start and deploy, working perfectly on Docker for Mac.

III. PROPOSED SYSTEM

Proposed model consists of Docker Client, Docker Server and Docker Registry.

A. Docker client

Any user who wants to run light weighted applications into the machine can be a client. It requests Docker server for the applications. It can access the applications using Docker server's IP address and a Tag. This tag is the unique ID of each Docker applications.

B. Docker server

The machine which serves the client's request is Docker server. Whenever a client makes any application request, the server pulls the application container form Docker host. Installs the application on his machine and then the client can use this application remotely through his browser without downloading the application on his own machine.

C. Docker hub

Docker Hub is a repository which has a pool of applications. Each application is registered with a unique tag ID in the Docker registry. Every outgoing application from the pool is wrapped in a container which contains all the dependencies required to run an application.

IV. IMPLEMENTATION

To develop this system we carried out following steps on client and server side.

A. Docker server

a) Installed Docker.
b) Configured static IP address for the server system.
c) Developed User Interface(UI) for Docker. For the development of UI we used Angular .js and html for front-end of UI and GO language for back-end of UI.
d) Hosted UI locally using Yarn Server.

B. Docker client

a) Open web browser.
b) Enter static IP of docker server and the tag of the application to be used.

V. CONCLUSION

In this paper we have presented implementation GUI for deploying application on Docker platform on private cloud. Now Docker Server can simply use web User Interface for fulfilling client's requests instead of using Command Line Interface hence making Docker user-friendly for client's who do not belong to IT background.

REFRENCES

[1] David Bernstein, "Containers and Cloud: From LXC to Docker to Kubernetes",IEEE Cloud Computing , Year: 2014, Volume: 1, Issue: 3

[2] Sachchidanand Singh, Nirmala Singh, Containers & Docker: Emerging Roles & Future of Cloud Technology , 2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), Year:2016, 2016 IEEE

[3] Babak Bashari Rad, Harrison John Bhatti

[4] Mohammad Ahmadi, "An Introduction to Docker and Analysis of its Performance", IJCSNS International Journal of Computer Science and Network Security, VOL.17 No.3, March 2017

[5] Harald Albers, Allen Sun, "Kitematic user guide ",2014.

[6] Dmitry Medvinsky,Luke Hutscal,Adrien Magnus", Managing Containers with Docker Shipyard", Russia,2015.

[7] E. Bacis et al. , "DockerPolicyModules:Mandatory Access Control for Docker Container", Proceeding of 2015 IEEE Conference Communications and Network Security (CNS),2015.

Author's Profile

Rakesh K Kadu received the Master's Degree in Computer Science and Engineering, from Nagpur University, Nagpur, India in 2009. He is currently pursuing the Ph.D degree from RTMNU, Nagpur, India. He is currently an Assistant Professor with the Department of Information Technology in Shri Ramdeobaba College of Engineering and Management, Nagpur. His current research interest includes Virtualization Technology, Cloud Computing, Cryptography and Networking.

Shikha Agrawal pursuing her Bachelor's degree in Information Technology from Shri Ramdeobaba College of Engineering and Management, Nagpur. Her research interest includes Cloud Computing, Container based technology and Networking.

Swinal Kamble pursuing her Bachelor's degree in Information Technology from Shri Ramdeobaba College of Engineering and Management, Nagpur. Her research interest includes Cloud Computing, Container based technology and Networking.

Tejaswini Bhardwaj pursuing her Bachelor's degree in Information Technology from Shri Ramdeobaba College of Engineering and Management, Nagpur. Her research interest includes Cloud Computing, and Networking.

