

Hiten Pratap Slngh <u>https://github.com/hitenpratap/</u> <u>https://hprog99.wordpress.com/</u> <u>hiten@fintechlabs.in</u> <u>https://hitenpratap.github.io</u>











66

Docker is an open platform for developing, shipping, and running applications. Docker is designed to deliver your applications faster. With Docker you can separate your applications from your infrastructure and treat your infrastructure like a managed application.





2. Features of Docker!



Features of Docker!

- Extremely fast and elegant isolation framework
- Inexpensive
- Low CPU/memory overhead
- Fast boot/shutdown
- Cross cloud infrastructure



6



3. Virtual Machine V/S Docker



Virtual Machines V/S Docker







Docker Architecture



Docker Architecture







Docker Components



Docker Components!

- Docker Client
- Docker Daemon
- Docker Index
- Docker Containers
- Docker Images
- Dockerfile



12

Docker Components!

- Namespaces serve as the first level of isolation. Makes sure a process running in a container cannot see or affect processes running outside the container.
- Control Groups, the key component of LXC, have resource accounting and limiting as their key functionality.
- UnionFS (FileSystem) serves as a building blocks of containers. It creates layers, and, thereby, accounts for Docker's lightweight and fast features.





Run Applications

6.



Run Applications

- **Step 1:** Build an image.
- **Step 2:** Run the container.



Build an Image

Docker Image is a read-only template to build containers. An image holds all the information needed to bootstrap a container, including what processes to run and the configuration data. Every image starts from a base image, and a template is created by using the instructions that are stored in the DockerFile. For each instruction, a new layer is created on the image.



Run The Container

Running the container originates from the image we created in the previous step. When a container is launched, a read-write layer is added to the top of the image. After appropriate network and IP address allocation, the desired application can now be run inside the container.



Host

image name: myapp:0.9 image id: e72ac664f4f0 Ubuntu 14.04 + Django app

image name: pgsql:9.2 image id: bc840bd687e3 Ubuntu 14.04 + postgresql

container name: myapp1 container id: 44a87fdaf870

port: 80 port: 443 port: 8080 port: 4430

container name: myapp_db container id: 35005d564268

port: 80 port: 8081

container name: app-dev container id: 9433b2b904de

port: 80 port: 8000

Docker Demo





7.



What's Kubernetes?

- Kubernetes(K8S) is an open source tool for managing containerised workloads.
- It operated at the container(not hardware)
 level to automate the deployment, scaling
 and management of applications.
- K8S works alongside a containerisation tool, like Docker. So if containers are the 'Ingredients' of an application, then K8S would be the 'Chef'.



What's Kubernetes?

- As well as managing individual containers, K8S can also manage clusters:
 - A cluster is a series of servers connected to run containers.
 - K8S can scale upto 5000 server and 150,000 pods in a single cluster.
 - A pod is a group of containers that share resources, a network and can communicate with one another.





8. Why should it be used?



Why should I use it?

As an orchestration platform, K8S provides features to make the management, maintenance and life-cycle of containers easier than using a container-engine alone.

- Horizontal Scaling
- Self Healing
- Automated Rollouts
- Various other features like Service
 Discovery and load balancing etc.







Kubernetes Clusters



Kubernetes Clusters

Containerised applications are deployed with K8S into highly available clusters

- Clusters run over several computers called Worker Nodes, that are connected to work as a single unit.
- Containerised apps are automatically distributed among the Worker Nodes at deploy time.
- A Master Node manages the cluster coordinating scheduling, scaling and rolling updates.



Kubernetes Clusters



10. How does Kubernetes work?



How does K8S Work?

Worker Node(Slave)

This is where containers are deployed. These nodes contains:

- Multiple Pods
- Docker Engine
- Any add-ons e.g. DNS
 - Kubelet This is the most important component as it carries out the instructions from the master node.

Master Node(Master)

This controls the deployment. This node contains:

- API Server
- Controller
- Scheduler
- Etcd handles config management, service discovery etc









Kubernetes Deployments



Kubernetes Deployments

To run a containerised application in a cluster, a Deployment configuration is used:

- The Deployment describes how K8S should create and update application instances.
- The K8S Master uses this to schedule the instances onto the Worker Nodes.















Kubernetes Pods

A pod is an abstraction that represents a group of one or more application containers and shared resources.

- Pods are the atomic unit on the K8S platform.
- Deployments create pods, with containers running inside these pods.
- The pods are scheduled on the Worker Nodes and remain on their assigned Nodes till their termination.
- Worker Nodes can run several Pods.



34









Kubernetes Services



Kubernetes Services

A service is a logical set of Pods and a policy to access them - they are used to expose your pods to access from outside the cluster.

- Cluster IP
- NodePort
- LoadBalancer











K8S architecture can be viewed in two parts: Components that run on a Master Node and those that run on the Worker Nodes.





- Kube Controller Manager: Runs common controllers for Nodes, Services etc
- Cloud Controller Manager: Run separate processes for KCM controllers that contains cloud vendor specific implementations
- Kube-apiserver: Hub of communication between Masters and Workers.
- etcd: A key:value database that stores all the information of current and desired state of clusters.



- kube-scheduler: Based on workload, scheduled upon which Worker Nodes that Pod should run on.
- kubelet: Effectively the K8S agent running on each node, for communication with the apiserver.
- kube-proxy: A network proxy that is used to implement service backends in K8S, providing appropriating traffic routing to virtual IPs via iptables config.
- Docker: Out of box, docker engine is running as the container engine implementation.



THANKS!

Any questions?

https://github.com/hitenpratap/ https://hprog99.wordpress.com/ hiten@fintechlabs.in https://hitenpratap.github.io



References

- https://www.slideshare.net/MichaelOSullivan41/kubernete s-an-introduction-to-the-open-source-container-orchestr ation-platform
- <u>https://www.slideshare.net/BMPresentations/kubernetes-</u> <u>for-beginners-an-introductory-guide</u>
- https://kubernetes.io/docs/
- https://sway.office.com/QewzAReCr1Y34XTE?ref=Link&lo
 <u>c=mysways</u>
- <u>https://kubernetes.io/docs/reference/kubectl/cheatsheet</u>

