**Kubernetes Clusters on local Machine**

**Introduction**

Before going deep into kubernetes, to better appreciate kubernetes it is essential to understand dockers and how dockers work. This will give you better insights into kubernetes.

Dockers serves as a ship which can house numerous containers. in deploying a highly scalable application over a cluster of VM machines or availabilty zones (it must be noted that Kubernets cannot spinng accross regions.This process is complex using dockers, this is where kubernetes comes into play. please in understanding Dockers technology please refer to the link I attach to this page (......). This is very essetial so as to understand the procedures I would be showing on this page.

Later on on this Tutorials, we would run complex clusters, and show various ways to prove test your application before pushing it production. I would start with a simple example. Then move on to a very complex example of applications, where if you follow me you would learn a great deal for your self or for your company.

In simple terms, kubernetes coordinates a highly available cluster of computers, VMs, Systems, applications, that are connected together to work as a single unit. Hence u can have many computer clusters connected to work as one unit using Kubernetes. Salient points about Kubernetes:

* Deploying a containerized application over a cluster of different machines.
* Highly scalable
* Automates distribution and scheduling of application containers efficiently.
* The Master controls/monitors the clusters, the nodes run applications.

You can deploy Kubernetes on AWS and GCE easily. Look into the pages

* AWS - (......)
* GCE - (.......)

To deploy Kubernetes on your local Machine: You will need to configure the machine.

* Install a hypervisor (This makes to create a virtualization environments In my case I am using Ubuntu, and I install VirtualBox) there are many option to use:
  + For Linux install VirtualBox or KVM
  + For windows install VirtualBOX or Hyper-V
  + For OS X, install xhype driver, VirtualBox, or VmwareFusion
* Install Kubectl: This is simply the command line tool used handle all operations of kubernetes on clusters in the cloud or in the local machine There are various ways to do it: I would only refer to the means that is convinient for me and I hope for redears as well.
  + If you are using Linux:
    - this is the link to the latest release with the command: curl -LO https://storage.googleapis.com/kubernetes-release/release/$(curl -s <https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/linux/amd64/kubectl>
    - Enable the kubernete release to be executable: chmod +x ./kubectl
    - Move the binary into your PATH: sudo mv ./kubectl /usr/local/bin/kubectl
  + If you are using MacOS:
    - this is the link to the latest release with the command: curl -LO https://storage.googleapis.com/kubernetes-release/release/`curl -s <https://storage.googleapis.com/kubernetes-release/release/stable.txt>`/bin/darwin/amd64/kubectl
    - Enable the kubernete release to be executable: chmod +x ./kubectl
    - Move the binary into your PATH: sudo mv ./kubectl /usr/local/bin/kubectl
  + If you are using windows:
    - and you have *curl* installed: use this command to install: curl -LO <https://storage.googleapis.com/kubernetes-release/release/v1.8.0/bin/windows/amd64/kubectl.exe>
    - Then add binary into your PATH:
* *Install Minikube:* This enable you to easily deploy and test your kubernetes clusters housing your applications on your local machine before pushing it to the cloud. It should be noted that Minikube only enables you to run a single node Kubernetes cluster in a VM, givinig the ability to test and develop your Applications. it should be noted that Minikube supports kubernetes features such as: Dashboards, DNS, ConfigMAps, Container Runtime etc. (Which I would show, if needed....) Since Minikube is still in its development stage, I would refer you to this link to download the version you want for your system (<https://github.com/kubernetes/minikube/blob/v0.23.0/CHANGELOG.md>). In my case I am using Linux hence this is the command I use to install minicube: curl -Lo minikube https://storage.googleapis.com/minikube/releases/v0.23.0/minikube-linux-amd64 && chmod +x minikube && sudo mv minikube /usr/local/bin/
  + If you are using OSX: use this link: curl -Lo minikube https://storage.googleapis.com/minikube/releases/v0.23.0/minikube-darwin-amd64 && chmod +x minikube && sudo mv minikube /usr/local/bin
* *Confirm installation* ( $ minikube version : ↵ minikube version: v0.23.0)
* Confirm IP: (in most situation it returns the local ip of the system) : ( $ minikube IP : ↵ 192.168.1.100)

* *Quick Start:* $ minikube start ........↵

Starting local Kubernetes v1.8.0 cluster...

Starting VM...

Getting VM IP address...

Moving files into cluster...

Setting up certs...

Connecting to cluster...

Setting up kubeconfig...

Starting cluster components...

Kubectl is now configured to use the cluster.

* Now just as you can deploy a simple container image for applications on docker machine (.....) you can do the same for minikube, In this case running cluster of kubernetes for applications.
* Simple Start: (Run-Hello World minikube image) use this link:

( $ kubectl run hello-minikube --image=gcr.io/google\_containers/echoserver:1.4 –port=8080

↵ deployment "hello-minikube" created)

* Now the cluster is configured in minikube: with container housing echoserver on port 8080, the next is to expose the port, for it to be assesible on your local machine browser: ( $ kubectl expose deployment hello-minikube –type=NodePort ↵service "hello-minikube" exposed
* Now similar to how you see if an application in a conatiner image is up and running on Dockers, In minikube, this will show all containers and clusters created in minkube type this on minikube ($ kubectl get pod: NAME READY STATUS RESTARTS AGE

frontend-685922934-7ydhf 1/1 Running 2 3h

frontend-685defefe- 9frgr 1/1 Running 2 3h

frontend-685d7ff496-rnnpv 1/1 Running 2 3h

**hello-minikube-5bc754d4cd-qsqlq 1/1 Running 0 9m**

* Note: if the staus is ContainerCreating, be patient enter the command again, until the status changes to ContainerCreated. Then you can curl the application on terminal ($ curl $(minikube service hello-minikube –url CLIENT VALUES:

client\_address=172.17.0.1

command=GET

real path=/

query=nil

request\_version=1.1

request\_uri=http://192.168.1.100:8080/

SERVER VALUES:

server\_version=nginx: 1.10.0 - lua: 103471

HEADERS RECEIVED:

accept=\*/\*

host=192.168.1.100:3182

user-agent=curl/7.85.1

BODY:

-no body in request) since no http server is deployed we leave that to future projects:

* To delete the hello-project: $ kubectl delete deployment hello-minikube ↵ deployment "hello-minikube" deleted
* if you want to see a GUI of your minikube

$ minikube dashboard this will show you a open in your browser a page where you can view and manage your pods.

Hence I have been able to show how to create your cluster Up and running deploy a simple application on it and delete the application on it.

**Projects on MiniKube:**

Project 1:

* Configure Kubectl