**Deploying Containerized Web App on Kubernetes Cluster**

(Hands on Practice from Udemy) Date: June 28, 2022

**Scenario**

Consider a Multitier Java Web Application Stack consists of tomcat application server, MySQL db, memcache, rabbitmq, nginx is running on VMs/ EC2, and it has containerized, also tested. Now it is required to host for production with High Availability, fault tolerance, easily scalable, platform independent, portable and flexible.

**Solution**

We can use container orchestration tool Kubernetes and run the containerized application on Kubernetes Cluster

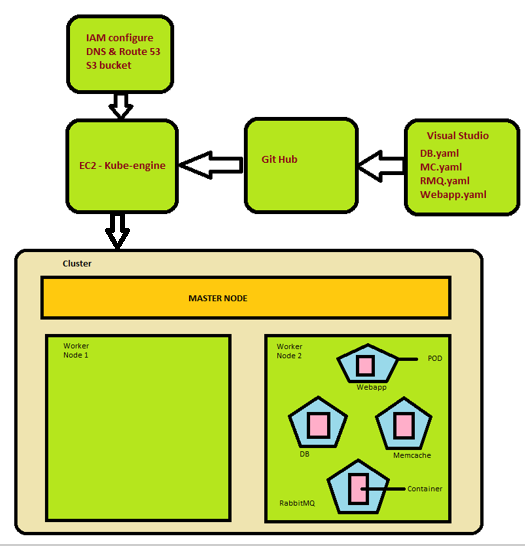
**Tools to be used**

* AWS, Kubernetes CLI, KOPS for cluster, EBS volume for DB Pod

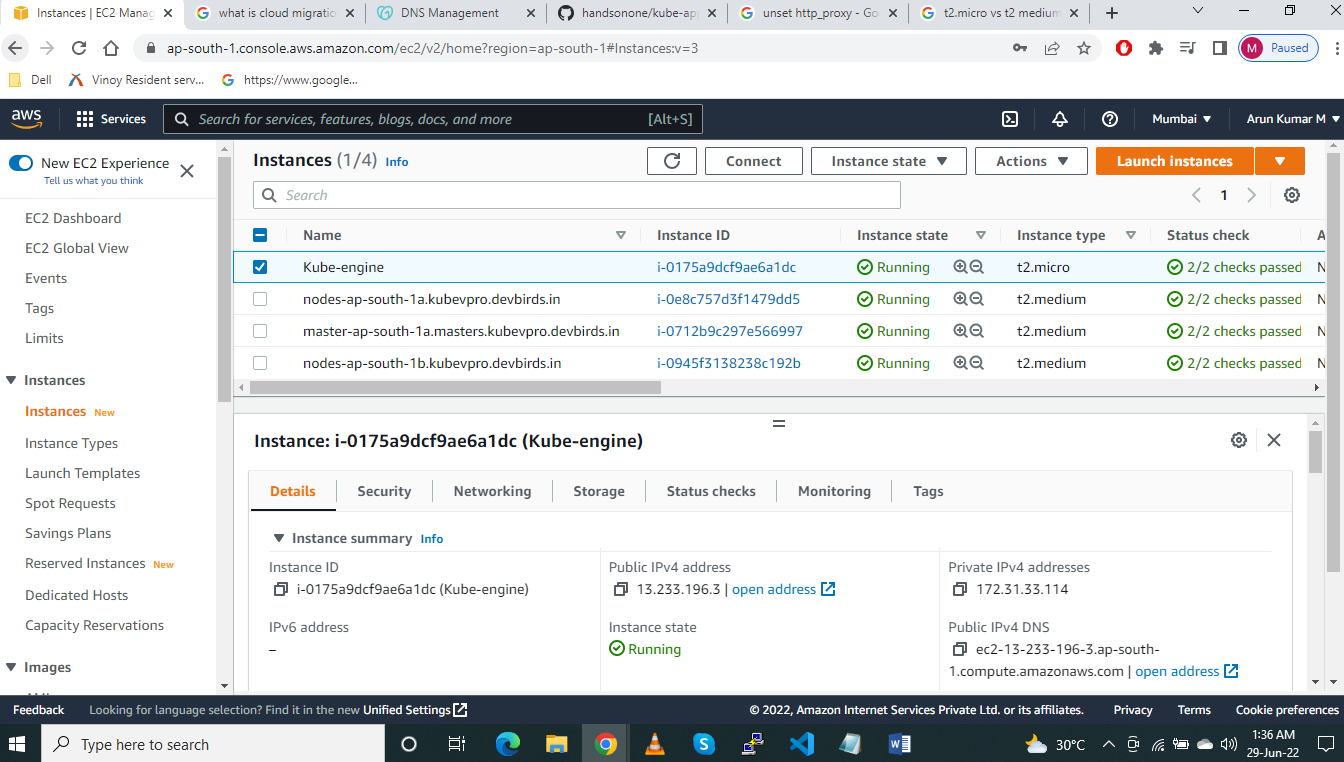
**Steps to be followed:**

* Launch one Ubuntu instance, and name it as Kube-engine.
* Create IAM user and configure with the instance.
* Create one hosted zone by using Route53 and sync with one DNS.
* Create a bucket in S3 to store the state files of Kubernetes cluster.
* Create the volume using EBS, give a name and tag then note down the volume id.
* Install Kubernetes CLI using Kubectl from official page.
* Download & install Kubernetes using KOPS. Validate the version of KOPS and Kubectl.
* Create Kubernetes Cluster on the same machine with the required specification of master and node components.
* Validate the cluster by checking the nodes whether it is up and the cluster should be returned as healthy to launch application.
* Label the worker nodes to different region names to choose the right node to create the pod.
* Write the definition file for Secret using yaml, where mention db password and memcached password in base 64 format.
* Write the yaml definition file to create the Deployment (POD+ Replicas) for db, web app, memcache, and rabbitMQ using visual studio code editor. The container images to be pulled from Docker hub where we stored images for Docker project.
* Write the yaml definition file to create the Service for above PODs for db, webapp, memcache, rabbitMQ. We mentioned the service type as load balancer for webapp.
* Commit the above yaml files and push it to git hub.
* Then git clone the yaml files from git hub repo to Kube-engine server.
* Run the yaml files using kubectl create –f filename command to launch all the services.
* Get the pod & services enabled. Copy the endpoint of the webapp services validate in browser for application is running.

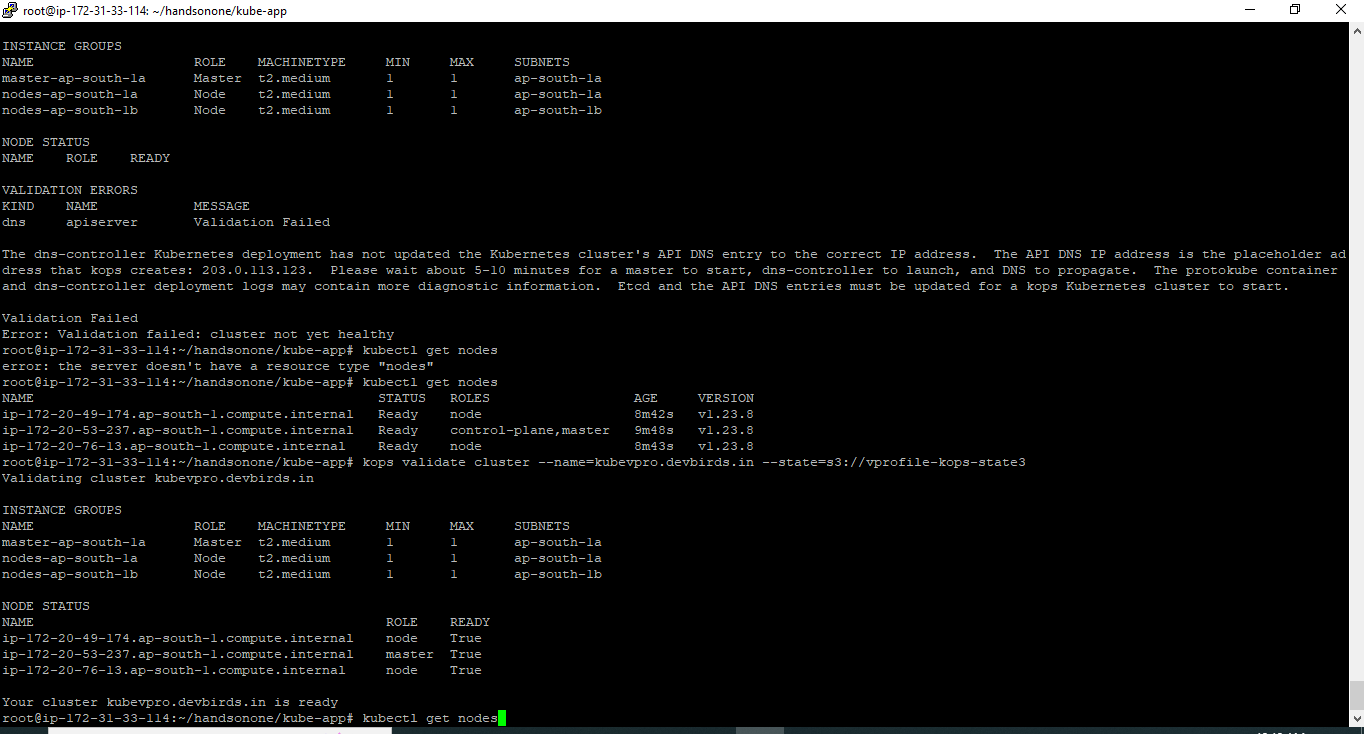
**Workflow Diagram**



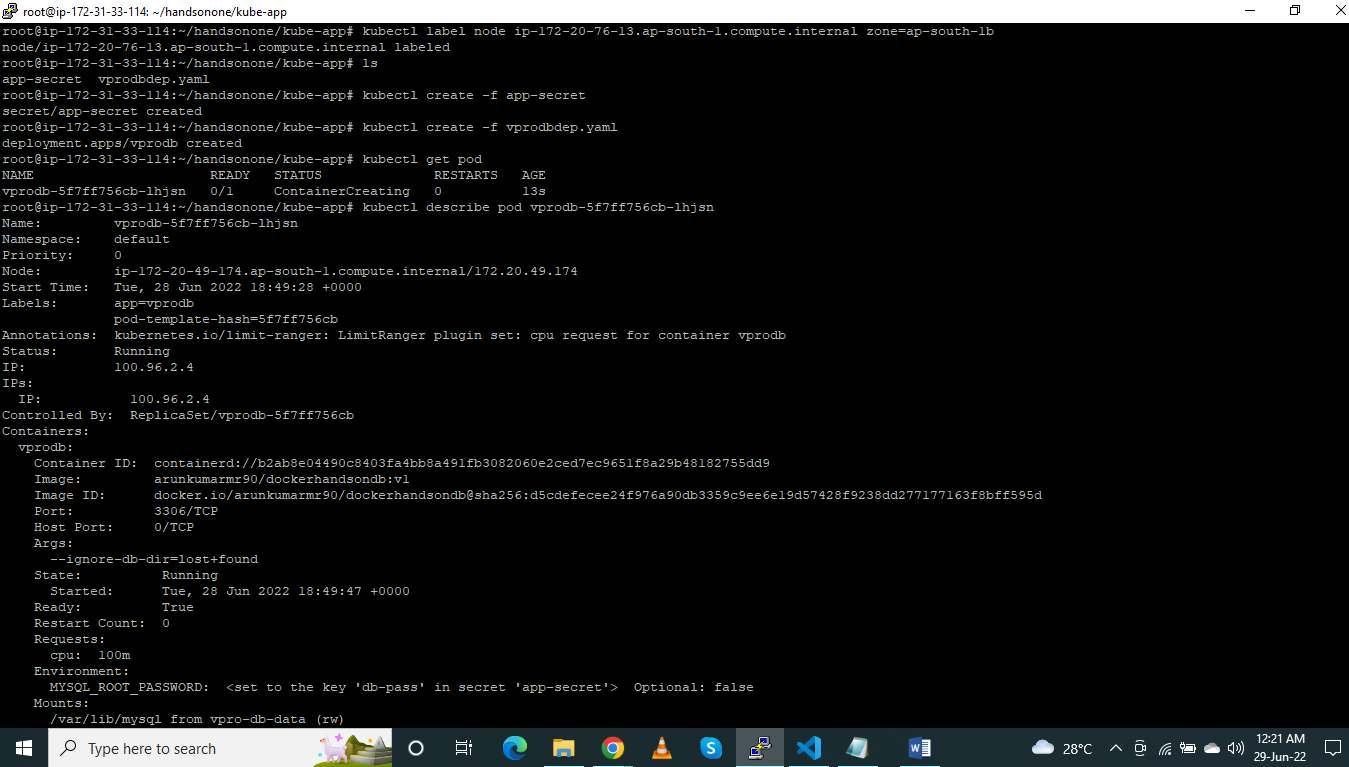
*Screenshots of Kube-engine server, Master and worker nodes*



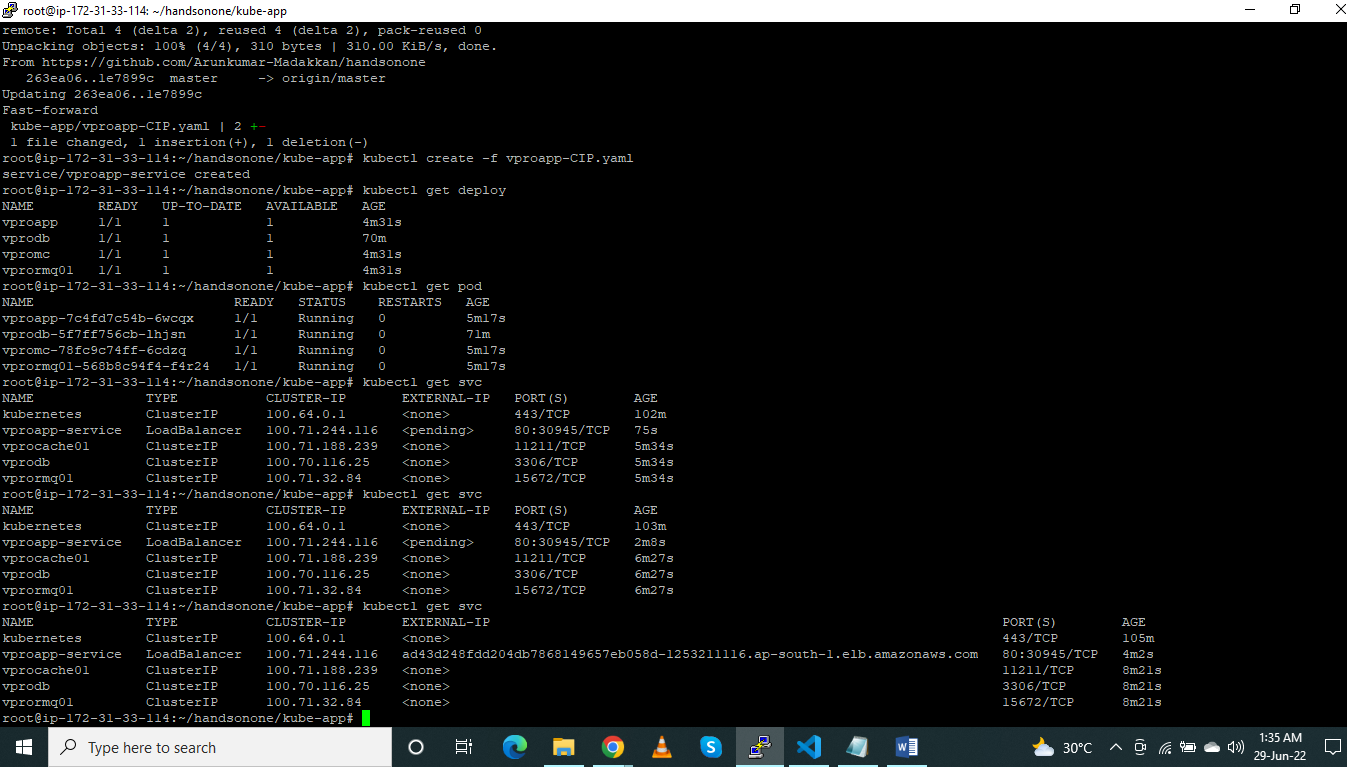
*Cluster creation screenshot*



*Description of POD screenshot*



*Screenshot of service and status up and endpoint enabled*



*Screenshot of validating the endpoint in browser, which ran successfully*

