KUBERNETES

SECTION-11

-> To learn Kubernetes visit “kubernetes.io”
-> Kubernetes is an open-source system for automating the deployment,scaling, and management of containerized applications.

-> why Kubernetes?
The problems:

1. Manual deployment of containers is hard to maintain, error-prone and annoying.
2. Containers might crash/go down and need to be replaced. When manually deploying on the ec2 instance, we have to manually monitor and replace them with new containers when they crash.
3. We need more container instances when there is more traffic.
4. Incoming traffic should distributed equally.

The solution :

1. Using ECS checks and automatically re-deploy when container crashes.
2. Autoscaling
4. Load balancer.

But disadvantage is that when we use AWS ECS we lock in that particular service.when we want to switch service provider we cannot use same configuration file and need to learn that service.
So we can use kubernetes.

-> What is kubernetes exactly:
we have a way of defining our deployments, our scaling of containers and how containers should be monitored and replaced if they fail. We have a way of defining all of that independent from the cloud service we're using.It can help us with automatic deployment,scaling ,loadbalancing and management.

We write configuration (i.e desired architecture number of running containers etc) we can actually pass that configuration with certain tools to any cloud provider or actually any machine owned by us which is configured correctly.If cloud provider does not support config file then we can install some kubernetes software.some providers require some configurations then that particular configurations can be written in same configuration file. When using different cloud provider that particular configuration need to be removed.

-> kubernetes is not alternative to AWS or azure.it is not a cloud service provider.It is an open source project.it is collection of software and collection of concepts.which can be together used with any cloud provider.It can be used with any provider.not restricted to only one provider.Its not alternativeto docker.It works with docker container.Its not a paid service.So kubernetes is a collection of concepts and tools which helps us with deploying containers anywhere.

-> pod is a smallest unit that holds container.which is responsible for running this container.
Now this pod with the container inside of it, then itself, runs on a so-called worker node. So a worker node is the thing in the Kubernetes world which runs your containers in the end. And you can think of worker nodes as your machines,each worker node can have more than one pod.Kubernetes also needs a proxy which in the end just is another tool Kubernetes sets up for you in the end on such a worker node to control the network traffic of the pods on that worker node. So basically to control whether these pods can reach the internet.kubernetes provides autoscaling by distributing pods to available worker nodes.master node interacts with worker nde and controls them.master node controls your deployment(i.e all worker nodes).

-> worker nodes and master node form a cluster and therefore one network,which forms a network.worker node then communicates with cloud provider API,to replicate this design in cloud provider.

So if we think about AWS, we might interact with AWS
such that AWS creates all the required EC2 instances
a load balancer which might be needed
and everything else which is required to have this network
and to then have Kubernetes and some Kubernetes tools
running on the master node, EC2 instance,
which then in turn controls the other EC2 instances,
which belong to this network
to then run containers in these pods on them.

-> What we need to do /setup:
* Create cluster and node instances(worker+master nodes)
* Setup API server,kubelet, and other kubernetes services on nodes.
* Create other provider resources that might be needed.

-> what kubernetes will do:
* create our objects(pods) and manage them.
* monitor pods and recreate them,scale pods.
* kebernetes utilizes the provided resources to apply our configuration.

-> Worker node:
Worker node is a machine/computer.It is managed by master node.Inside of worker node we have pods(hosts one or more application containers).pods are created and managed by

kubrenetes(master node).kubernetes can delete pods.each pod can have more than one container.each worker node can have more than one pod.pods in worker node may be same for scaling purpose or completely different.worker node also need software along with pods.It requires docker to create and run containers.another software called kubelet, which is communication device between master node and worker node.one more is kube-proxy which is responsible for incoming and outgoing to ensure everything is working and only allowed traffic is able to reach the nodes and allowed traffic is able leave the worker node.

> And the great thing is that with Kubernetes,
> you just need to define the desired end state,
> and if you're then using a cloud provider like AWS,
> they have services which allow you to provide
> this Kubernetes definition, and then AWS will
> set up all the instances and install all the
> required software for you.
> So you don't have to deal with that.
> You just have to know what's happening there,
> because as a developer, you should always know
> what your code, and what your configuration is doing.

-> 15 layers
command to know storage occupied by each ayer
alpine,bullsys
-> host.docker.internal :
  communication from inside of dockerized application to host machine.It is possible with this domain.
->why deploy on aws ec2, can't we run our local server after deploying a project?
why virtual machines are needed? can't we use our local system?
why dockerize an application?is it necessary?
what happens if we deploy the project on AWS without dockerizing?
->sudo su postgres
psql

*to access particular db
ubuntu@ip-172-31-5-32:~$ sudo -u postgres psql demo_db

*to know data inside the table
SELECT * FROM Demo_table;

*to know the list of db
sudo -u postgres psql
\l

sudo vim /etc/postgresql/15/main/postgresql.conf
sudo vim /etc/postgresql/15/main/pg_hba.conf

->->5 types
bridge
host
overlay
none
Macvlan

14 layers
FROM
MAINTAINER(READABLE ONLY)
COPY
ADD(UNZIP,DOWNLOAD THIRD PARTY PACKAGES)
RUN (MULTIPLE,EXECUTE IMAGE CREATION)
CMD (EXECUTE ONLY ONE CMD,EXECUTE CONTAINER CREATION)
ENTRYPOINT(OVERCOME CMD)
ENV
USER
WORKDIR
EXPOSE(READABLE)
VOLUME
ARGUEMENTS
->