K8s

**Docker -🡪 to implement microservicess**

**Lets take my application having multiple microservicess**

**MS1 –80(HOSTORT)**

**MS2---90**

**MS3---81 -----EVERY MS RUNS ON THEIR OWN CONT LIKE 5 CONS ARE THERS**

**MS4---84**

**MS5---8080**

**EVERY MICROSERVICE RUN’S ON ONE PORT NUMBER IN CONTAINER HOW TO EXPOSE TO OUT-SIDE USING –P WE MAPPED TO THE HOST PORT TO THE CONTAINER PORT**

**I WANT MS3 MULTIPLE REPLICAS**

**MS3-1 --81**

**MS3-2--81**

**MS3-3—81**

**MS3-4--81**

**MS3—5-81**

**I WANT TO SEND A REQ TO 81 PORT WHICH MICROSERVICE REQ GOES**

**IT IS NOT POSIBLE BECAUSE SINGLE PORT NUMBER SINGLE SERVICE WILL RUNNS**

**MS3-1 --81**

**MS3-2--82**

**MS3-3—83**

**MS3-4--84**

**MS3—5-85**

**LIKE THIS I REPICATED**

**ALL THE SERVICESS I MAPPED TO HSBC.CO.IN**

**HOW APP KNOWS MICROSERVICESS ARE RUNNING WHICH PORT NUMBER**

**IT IS EASY TO SCALE-UP AND SCALE-DOWN MICROSERVICESS**

1. **THE END USER REQ NEEDS TO GOTO PARTICULAR MICROSERVICESS**
2. **SUDENLY MS-2 WAS FAILD HOW TO ACHIVE HIGH AVALABILITY FOR MICROSERVICESS MICROSERVICESS ARE FYAUMRUM SYSTEM IS RUNNING CONDITION BUT MICROSERVICE IS NOT RUNNING COND HOW IT IS ACHIVE HIGH AVALABILITY**
3. **ALL MICROSERVICESS ARE RUNNING IN ONE MACHINE THAT MACHINE IS FAILD**

**RESOURCESS ARE NOT ENOUGH SO IWANT TO MULTIPLE MICROSERVICESS**

1. **I WANT TO RUN MULTIPLE MICROSERVICESS ON MULTIPLE MACHINES**

**WHO IS DISTUBITED THE LOAD TO ALL MACHINES TO MICROSERVICESS**

**WE REQUIRED SOME SOFTWARE THAT SOFTWARE IS CALLED ORCHASTAION SOFTWARE .**

**TO MANAGE MICROSERVICES WE WANT SOFTWARE ORACHSTAION SOFTWARE**

**DOCKER IS HAVING IT OWN DOCKERSWARM BUT NOT USING HEAVILY.**

**OTHER TOOL IS K8S ALL ARE WORKING ON THAT**

1. **K8S IS TOOL FOR TO ORACHSTAION OF YOUR MICROSERVICESS**

**ORACHSTAION MEANS**

1. **HIGH AVALIBILITY ACHIVING FOR MICROSERVICESS**
2. **SCALLING ACTIVITY**
3. **DISTUBUTED THE LOAD TO MULTIPLE MACHINES**
4. **TO ENSURE THAT YOUR REQ WILL REACHES TO RIGHT MICROSERVICESS**
5. **DEPLOYMENT EACH AND APPLICATION RUN ON MICROSERVICESS THAT DEPLOYMENT IS QUITE TOUGHT IN K8S EASY.**

**K8S: DOING EASY OF HIGH AVALABILTY OF MICROSERVICESS AND SCALLING ACTIVITY AND DEPLOYMENT OF MICROSERVICESS OF APP AND LOAD BALANCING IT MIGHT BE RUNNING ON ONE MACHINE IT MIGHT BE RUNNING ON MULTIPLE MCAHINES. Re goes to mach1 or mac 2 or mac3 who distubute**

**REQ IS GOES TO RIGHT MICROSERVICESS EASY USING OF K8S.**

**K8S MANAGE MICROSERVICESS.**

**=================================================================**

**What is CLUSTER: cluster means combination of multiple machines they work together.**

**Cluster is created to fullfill one requirement.**

**Requirement means working together one machine fails then only providing high avalabiltiy**

**Cluster: combination of multiple machines and thal all machines are working for requirement and to achive high availability for that reqrment**

**Reqgoes to any machine of in cluster out put is smane**

**===================================**

**In k8s we are creating cluster cluster means managing something**

**In k8s cluster having two things**

1. **Master—(control plane)**
2. **workernodes**

**technically specing master also server and workernodes is servers.**

**Both is having ther own responsibilities**

**To primary work of master Is to distubute the loads means our app runs on worker nodes req does not comes to worker nodes directly**

**Means req comes to master .master shudle your req to worker nodes.**

**One reqerment is there our app deploy on workernodes so we allways comnicate with master**

**Master our app deploy into multiple worker nodes(machines)**

**End user req giving that req comes to master ,master that req redirect into workernode whwre microservice is runs**

**Master is managing entire k8s archcture means app whwre to run which machine to run req goes to where all things managed by the k8s.**

**Master having 4 procesers or service or software**

1. **api server process**
2. **shudular procer(service)**
3. **etcd processer(service)**
4. **controller processer(service)**
5. **kube proxy processer(service ) it is optional**

**workernode’s are maintaine how much you want**

**each and every worker node containg 2 processer’s**

1. **kubeproxy service**
2. **kubelet service**

**===============================================================**

**Each and every component work is diff**

**We allways intract with master in masterserver we comnicate with api server**

**In master server one processer is thre it is running we allways comicate with that processer only. We are not intract with shudular,etcd,controaller,kubeproxy.**

**Api process is service or processer it will run on 6443 port on that machine.**

**What ever req giving to api server .api server stored in etcd. What req we are giving**

**Once it is done api server inform to controaller for to create microservicess.**

**Controller work is to create your microservicess.**

**But**

**Contoaller don’t know which worker node want’s to create.**

**We have multiple workernodes he don’t know which workernode having how much resources**

**Controaller asking to shedular for which machine I want to deploy this microservice.**

**Here shuduler knows how much resources is there thatb machiner is working condition or not all information about worker nodes having shedular.**

**Controaller asking to shedular for which machine I want to deploy this microservice.**

**Shudelar tells workernode1 or workernode2 or all worker nodes you want to deploy that microservicess to controaller**

**Controaleer will provision on worker node youe microservicess.**

**After that con will inform to shudulear then shudular will give to etcd**

**This is archisture in k8s**

**===========================================================================we are giving req to api server on master api server req stored in etcd then api server give req to contoaller ans contoaller asking to where I want to provison microservicess shudular will tells to information which worker node then controaller taking information deploy our microservicess on particular worker nodes .then contoaller inform to shudular then shudulat inform to etcd.**

**===============================================**

**Each and every microservicess replicas having uniqu ip address.**

**In docker container having unique ip address in docker**

**Kube proxy job is assigning to unique ip address to every microservicess replicas .**

**We are having ms1 image from that ms1 we are creating 3 replicas**

**Microservice means image that having service of application**

**From that microservice we are creating replicas in k8s**

**=====================================**

**We are sending req to api server one microservice create 3 replicas**

**Api server stored that req in etcd then send req to controller then contraoller ask to shudular where I want to deploy that microservices replicas.**

**Here 3 worker nodes will be there . here 3 replicas having unque ip address .**

**When thare are in different workernodes also because there are working together.**

**Each and every microservices replicas containg unique ip address in cluster that job belong to kubeproxy.**

**Here one repica die then created new repica from that microservicess that replica asin new ip address . who is send user req goes to right microservice replica**

**Kubeproxy to ensure that end user req send to wright microservice repica.**

**Kube-proxy job is: 1) assign to unique ip address to replicas**

**2) to ensure user req send to correct microservice replica**

**=========================================================================**

**In 3 repicas one die controaller will create one more .**

**Scheduler job : it mainatine all worker nodes information and on that pods status pods are running or not on that particular machines .**

**How scheduler knows all information kubelet is giving information on worker nodes and pod information to schudular and api server.**

**Scheduler and api server running on master how knows all information on workernodes**

**Because of kubelet is working on worker node so kubelet having know information about that workernodes and pod information continuosly, so kublet give information to schedulat and api server.**

**Kublet job:it is working on workernode and kubelet is taking information about workernode resourecss and pod running or not all information giving to api server .**

**Api server giving to shedulat then api server giving information to controaller contoaleers creating and deploying our microservicess repicas on worker nodes.**

**This all those things working on a cluster only other wise vluster is not working.**

**When kublet is not working that machine is not working.**

**============================================**

**Instead of microservic and instead of container we are having new keyword in k8s.**

**Pod----> entire k8s is work is make running condition of pod.**

**In k8s how many services are their but main work is to make a pos id working condition**

**Pod : means run-time environment for a container it might be a docker con or openshift con etc..**

**Container having our applicating is running.**

**Pod containg containers continers containg application.**

**Each and every pod having unique ip address.**

**Unique ip address ensure kubeproxy.**

**Pod having multiple container we are technical specking but that is not recomnded.**

**In k8s pod have unique ip address not have ip address of container**

**All operations doing in pod level.**

**======================================================**

**Pod-🡪containers**

**Server1---67.90.90.90**

**Pod1(10.0.0.34)------>c1 (90)\_\_\_>8080**

**Pod2()10.0.0.134---->c1(90)----->9090**

**c1(80)----->8081**

**67.90.90.90:8080----------------------->pod1**

**67.90.90.90:9090----------------------->pod2-----c1**

**67.90.90.90:8081----------------------->pod2------c2**

**=======================================**

**Here pod is emprun it will gone pod isnot come.**

**Techinally pod we can create indepdentli but not recomnded.**

**In aws ec2 –machine terminated then we gave qg to desire 3 it will automaticaklly created ec2 instance**

**So in k8s we can give desire to deployment**

**The deployment job is to ensure desire no of pods**

**When it is die.**

**In k8s point of deployment diff service diff pod is diff but form req ther are working together.**

**Deployment:like ag**

**Service: service is ensure your req is send to pod who is implemented is kubproxy**

**Our app runs on pod so app is acess to outside so we create service object**

**In docker we create pot maping instead of portmapping we create service object**

**Service ensure enduser req to your pod**

**App is running inside pod you need to acess from out-side purpose we need to create service object.**

**Types of service**

1. **cluster ip**
2. **nodeport**
3. **loadbalancer**

**why we need to create a service object why weneed to give req to service object why we are not send to directly pod.**

**Ms1**

**Ms2**

**Ms3**

**Pod1---ms1-----80-------80**

**Pod2—ms2---80--90**

**Pod---ms3-------80-------8080**

**Worker node1 -80----pod1---8080**

**Why we req a service object**

**Service is act as a load balancer**

**Here I have 2 rep of same microservicess that is runs on 2 pods**

**Then end user give a req to hostip:port but inside that hostip:port giving req to only one pod not sending req to pod 2 in ther also same micro repica running**

**For high availability purpose we crea a service object when end user giving req to hostip:port that mapped to service object that service object we mapped to 2 pod’s when end user req send service object send’s re to 2 pods it act as a load balancing.**

**So we can create service objevt high avavalablity.**

**Pod1---ms1-----80-------80**

**-----------------service ------------------enduser**

**Pod3---ms1-------80-------8080**

**Enduser-----service-------pod-----con-----app**

**When new pod came that pod automatically connect to service then load is automatically send to all pods.**

**Undylaying infrasture is abstraction.**

**Pod:**

**Deployment act as ag for pod**

**Service act as a lb for pod’s**

**Pod is taking req from service giving to con**

**Containers running on pod’s**

**Pod having unique ip address and app running in con on port number.**

**Api server takes tahe req from enduser**

**Controaller created a deployment microservicess on workernodes**

**Scheduler having all information of workernodes resources and pod information**

**Etcd having req information**

**Kuberproxy assign a uniqu ip address to each and every pod and it attached to the servie**

**Kubelet: it is working on workernode then it will give a information about workernode and pods to apiserver and scheduler**

**Service taking req from end user then forwarded to multiple pods.**

**Pod is runtime env of pods.**

**Service from running 100 pods which pod is fails we want to identified**

**Service deployment pod replicas 4 things very imporatant.**

**K8s: using for orchastaion tool**

1. **high availability of your microservice app 2) deployment3) load balancing**

**===============================================**

**Cluster**

**1)master—contoalpalne every thing contoals from master**

**Apiserver:6443**

**Scheduler :runs one port**

**Con: runs one port**

**Etcd: runs one port**

**2)workernode**

**Kubeproxy**

**Kubelet**

**Kubeproxy runs on cluster on every machine**

**Req to api server then etcd storing then controller then ask to schdular then cont taking deployment done then kubproxy cni implentaion(con network implemtaion)**

**Kubelet give infor work node info to api server ans scholar.**

**Node running or not resource information pod status.**

**Kublwt work on each and every workernode.**

**Pod is runtime envirment inside con running pod having unique ip pod contain multiple containers**

**Deployment right no.of pod**

**Pod app acess service object creat.**

**Cluster ip**

**Nodeport**

**Loadbalancer.**