|  |  |
| --- | --- |
|  |  |
| KUBERNETES : openssrc conatinaer orchestr f/w . Manages containers  DevELOPED by google | |
| Helps manage appl made up of 100/1000s of containers. == containerized apps in diff deplomnt environ == VM, phy m/c, cloud env , een hybrid depl environment | |
| Tasks does ? Containers offer perfct host for small indep app like MicSrvc.. rise of containers n MicS resulted in magemet require in mulitple env – using script is complex n impossible. So the Need for conatiner orchestration technical | |
| K guarantee-  1. high availa, 2 scalability (scale app fast wen more load/more usrs ) scale down == flexible to load,  3. disaster recovery :: backup and restores .. if data lost/infrast has problems/srvr explode/srvc center issues . Infra should have a m/cnsm to pick up dat an restore to latest state. | |
| Archi:: K cluster made of atleast one **Master node** , conneted to 1/> worker nodes  Node = virtual / phsM/c **workerNodes**= just refered as nodes.. each node has a **kubelet** a k process makes easy for cluster to communicate with each other execu takss on nodes/run applica .  Each node has diff app container say docker containers .. wN- all app runing  MstrNd: imp several K8 process r running –  imp Components:: manageing clus proper like **API Server (also** a container) = entry point to K8s cluster. Like K proces like UI, dashboard,api usign scripts, clients all talk to Api srvr  **Controller manager** :: has overview of happenign in cluster, some needs repar, if container needs restart….  **Scehduler**: scheduling containers on diff nodes, based on workload n availabil srvr rsrcs on each node. Which WN nxt container should be scheduled  **etcd**- keyvalue storage:: any time holds current state of K8 cluster. Has all confi data inside n status dat aof each node n container inside of node. Backup & restore is actualy made from this etcd snapshots ;; can recover while cluster state from this.  **Virtual N/w**:: Enables WN,MN talk to each other . Turns all nodes inside cluster into one powerful machine that has sum of all resrcs of indiv nodes. | |
| WN : has most load as app runign inside- bigger & more resoruces  MasterN:: only handful of master processes. Not need much rsrcs.. but must to be active. So backup MUST  So **prod** has atleast two masters and more inside K8 cluster. | |
| **K8 COMPON : pod, service, ingress, configmap, secret, deploynet, statefulset, daemonSet** | |
| Node= virtual / phsM/c.. simple srvr  POD: smallest unit of K8. Abstraction over conatiner… docker contain/imgs POD creates running env or laer on top of container.. since K8 wants to abst away container techno .. so can replace them if want to n also no directly work with docker tecnoli  App pod – own appl  usualy 1 appp per POD. :: pod meant to run one appcontainer inside of it . Can run Multiple containers inside a pod-> but only if 1 main app container requried helper side srvcs that has to run inside of pod. |  |
| A Pod is a self-sufficient higher-level construct. All pod's containers run on the same machine (cluster node), their lifecycle is synchronized, and mutual isolation is weakened to simplify the inter-container communication. This makes Pods much closer to traditional Vms  K offers virtual netwok.. pod not container gets IP address . Internal not public one.  Each pod can communi each other using ip adrs. However pod componen in K are weak.. they can die easily – newone will create in place with new IP. Adjust it every time pod restarts ==???? SO **service** is used. | |
| Service , Ingress Service is permanent adrs attaced to each pod. My-app has service1, db has service2.  Even if pod dies – service wil stay.  App to be access through browser == External Service. This opens a communication to browser  **Internal service**: container like DB , we dont want to expose or public req so create **internalService , specified while creating one**  Extsrvc url is not practical. : ex- http://<ipadrs of **NODE**>:port of **SERVC =** htt[“//124.89.101.2:8080  god for test purpose to fast. Would want something like with domian name.hashCode  So request is first forward to **Ingress (http://my-app.com)**→ then to servc : → this looks like a single server with 2 containers with a service.  Here Kuberntes adv wont be efficient | |
| **ConfigMap and secret** my-app should comm to db (say mongo-db-servc) wer is db creds configured? Usually appli, properied, ext env variable .i.e inside of built image of application  if any change – say db/ ip/port changes > adj url in app , rebuild app , push it to repo < pull new img in pod  So **ConfigMap =** ext config to app - contains config data like urls of db or other srvcs  In K8s – CM is connected to pod to read data from it. So , any changes to name/url of srvc goes to **config map** only .. no other changes.  **CREDS :: pswd, certificates , all those dont want access to others only read by pod.** ?? even in ext config dont put plain creds.. so use **Secret just alone base64** encode wont make it secure  secret compo are ment to be encrypted using thirdparty tools in K8 . K8 doesnot have that feature to automa’ do.  Cloud providers or toher 3rd party tools - that can deploy on K8 to encrypt secrets. | |
| **VOLUME : datastorage**  DB pod that replication uses n generates or has some data(Log data/ DB data ). If pod gets restarted data would be gone.. inconvenient.  Log data/ DB data persisted reliably long term .  Using Volumes : attaching a phystorage to pod – can be on same server node or on remote storage – outside of K8 lcuster (cloud/own premise)  Storage /// as an ext harddrive in k8 cluster.. since k8 cluster doesnot explicitly manage any data persistent. K8 user/adm respo for backing up, replicating , managing .. data | |
| DEPLOYMENT :  if app pod dies/creashes ? Restart pod ;;can have down time so  so relying on multiple servers  replicating the app on another Node .connected also to same service. Srvc = persistent static IP adrs with dns name. .  srvc is also like loadbalancer . It wil catch req n forwards to less busy pod.  On other Node – would not create second pod but just define a bluerint and psecify how many replicas of that pod want ot run.  Blueprint = deployment.    In practical we wont directly creting pods. We only create **deployments**  here specify replicas , sclaeup/down replicas . Pod is layer of abbs on top of container. .  deployment is another abstraction on top of pods > makes it convenie to interact with pods , replicate and configrue.  SO .. only work with deployment.s  DB cant be replicate using deployment – since it has a **state = its data ,** clone of replicas of DB means share same datastorgage – n maintain all writing n readnin g. n leads to dat ainconsistencies..  k\* COMPONENT so **STATEFULSET :** FOR APPS LIKE DB , MYSQL, MONGODB, ELSASTIC SEARCH – SHOULD BE CREATED USING StFl sets and not deployments    stfs : takes care of replicating pods, scaling up/down also making sure DB reads/writes ares synchronized.  But deploying DB app using stflsts in k8 cluster is TEDIOUS: difficlut.  So , common practice to host DB app outside of K8 cluster n have stless app that can be relicated with sacale with ease inside and communi with ext db.  With two replicas of app pods, db both load balanced – set up is rbust. Second node full set and app is usable if one down. Until replicas get recreated | |
| powerful K8 cluster can be build with 4 above components | |
| Free Kubernetes Backup and Migration - Download Kasten's K10 and Get 10 nodes free forever: https://www.kasten.io/nana | |
| In action configuring 8 cluster | |
|  | |