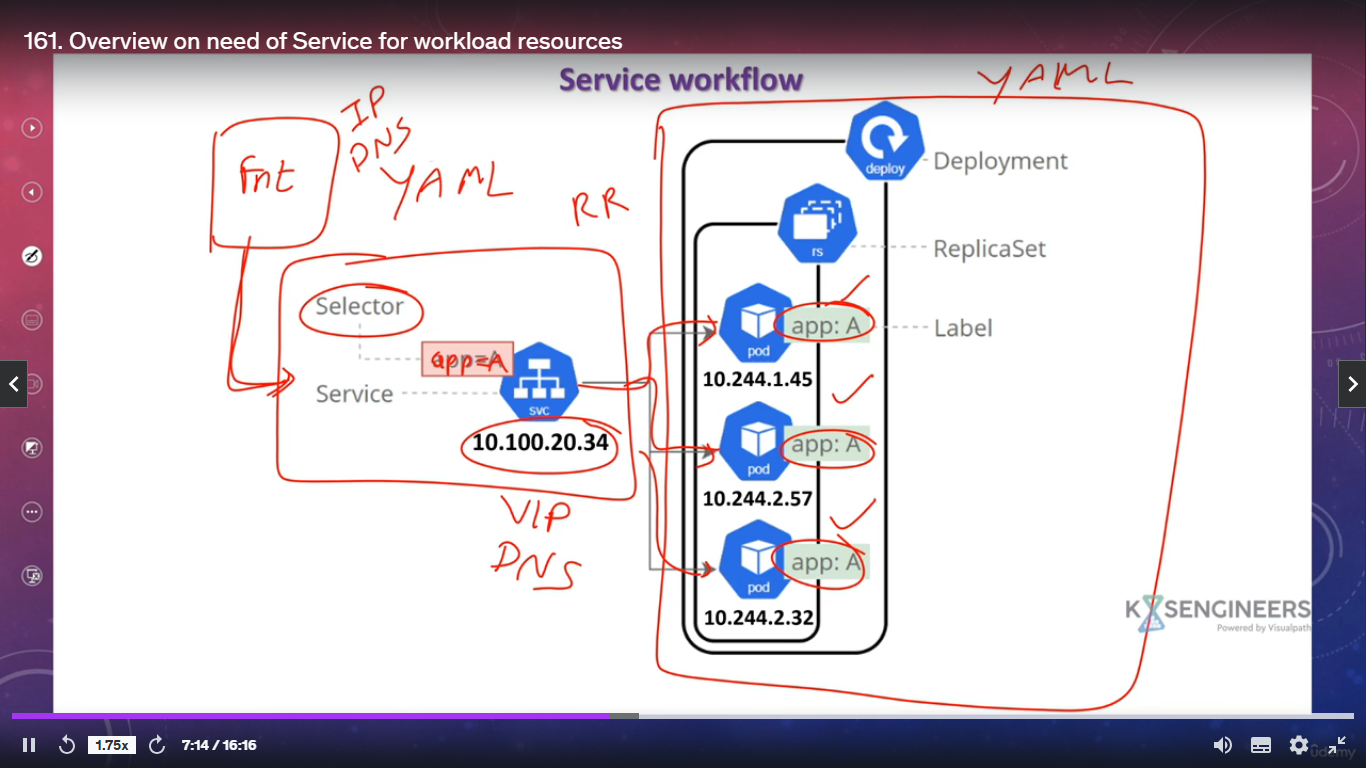
DOCKER

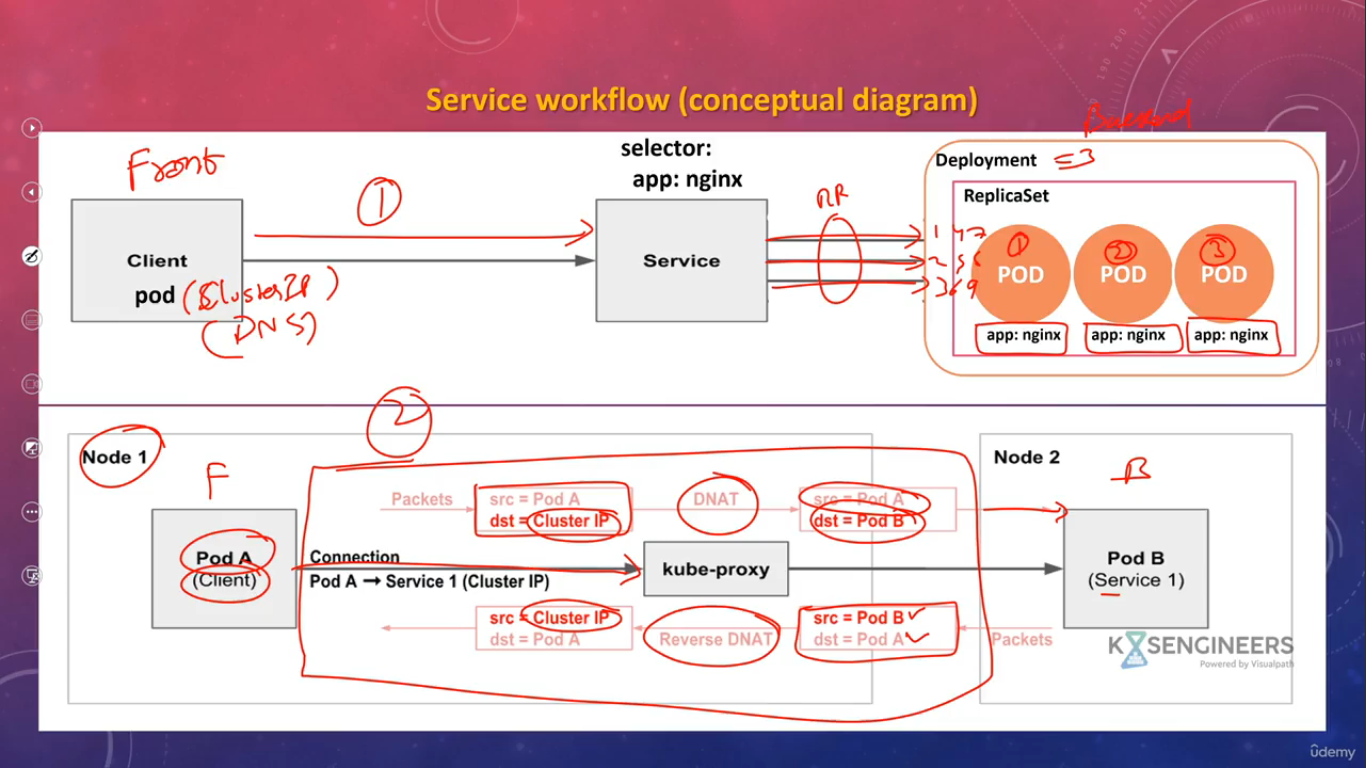
1. What is virtualization?
2. What are the limitations of virtualization?
3. What is containerization?
4. What is docker?
5. What are the underlying technologies of docker?
6. Difference between docker engine and docker desktop.
7. What are the components of docker engine?
8. ---
9. Explain in detail the docker container creation workflow.
10. How to access docker application from outside docker host machine?
    1. port forwarding (docker\_host\_machine\_ip:port\_no -> docker\_container\_ip:application\_port\_no)
11. What are different container restart policies?
    1. https://stackoverflow.com/questions/61725195/difference-in-docker-restart-policy-between-on-failure-and-unless-stopped
12. What are some important docker container commands?
    1. docker container run / create / start / stop / rm / exec / inspect
13. ---
14. What is a registry? What is docker hub?
15. What is a docker image?
16. Explain the architecture of docker image and container.
17. What are the different types of docker images?
    1. based on privacy - private, public
    2. based on trust - docker official, verified publisher, sponsored oss
18. Explain how docker pull works in detail.
19. Explain how docker push works in detail.
    1. docker login / logout
20. How will you share images in docker hub?
    1. https://stackoverflow.com/questions/49976188/copy-docker-image-between-repositories
21. What are some important docker image commands?
    1. docker image pull / push / ls / history / build / inspect
22. Why we need to build custom images?
    1. custom requirements
23. What is the process of building a custom image?
    1. using dockerfile
24. What are the different dockerfile parameters? Explain the functionality of each one of them.
    1. FROM, ENV …
25. Which parameters can be present in the first line of dockerfile?
    1. FROM, ARG
26. What is the difference between ENV and ARG?
27. How to choose right base image for custom image?
    1. size
    2. requirements
28. ---
29. What are the different storage types in docker engine? Explain the properties and uses of each one of them.
    1. volume mount
    2. bind mount
    3. tmpfs
    4. external storage linked to volume mount
30. What are some important docker volume commands?
    1. docker volume ls / inspect
31. ---
32. What are different supported network types in docker engine?
33. Explain the architecture of default bridge + user defined bridge network.
34. Can a default bridge network talk to a different user defined bridge network?
    1. by default, no
    2. to enable communication, use either docker connect command or modify iptables
35. Explain the architecture of host network.
36. Explain the architecture of none network.
37. Explain the architecture of overlay network and how two docker containers present in two different docker host machines can talk to each other?
38. What are some important docker network commands?
39. ---
40. How will you change the default docker root directory?
    1. first create a new folder to which you want to change. copy existing root directory contents to this new folder created. give this new folder path in /etc/docker/daemon.json with key as ‘data-root’. do all this after stopping docker service.
41. How to change default bridge network cidr?
    1. need to insert the new cidr range in /etc/docker/daemon.json file with key as ‘bip’. docker service needs to be stopped and then started after doing this change. all containers need to be deleted and created again, so backup of all containers need to be taken.
42. How to make custom network as default bridge network?
    1. first create a custom network using ip commands (and not using docker network create). once new custom network is created, add that network info in /etc/docker/daemon.json file with ‘bridge’ as key.
43. What are links and why we use them?
    1. used for container to container communication using names (which is still available in default bridge network).
    2. while creating a container, use --link container\_name:alias
44. How will you limit resources to your docker container?
    1. by default, a docker container can use full cpu, memory available in the host machine. to share resources among multiple containers, we need to limit resources that a particular container can use. we can use docker container update command with proper options to set the limits.
45. ---
46. What are the advantages of having an UI to handle docker environments?
47. ---
48. What are the challenges in using plain docker engine?
49. What is docker compose? Why we use it?
50. What are docker compose file parameters?
51. ---

KUBERNETES

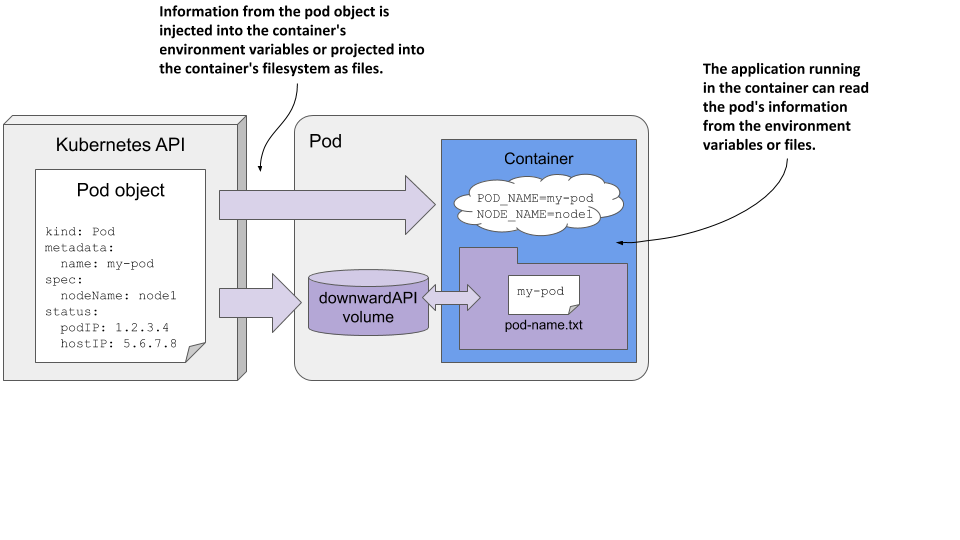
1. Important components in k8s.
   1. architecture
   2. installation, upgrade and backups
   3. pods
   4. replication controller
   5. replicaset
   6. deployment
   7. daemonset
   8. jobs
   9. cronjobs
   10. statefulset
   11. services
   12. volumes
   13. configmap
   14. secrets
   15. scheduling
   16. identities & authentication & authorization
   17. namespaces
   18. ingress
   19. network policy
   20. UI
   21. monitoring
2. Explain kubernetes architecture in detail. Explain what are the components and what each of them does.
3. Explain the flow of instructions using above k8s architecture.
4. What is etcd and why we use it?
5. ---
6. How will you spin up a kubernetes cluster?
   1. types - single node, multi node, HA setup
   2. basic installation flow
      1. prepare the virtual machines
      2. install containerd
      3. do some configuration changes for containerd
      4. install kubernetes packages
      5. do some configuration changes for those kubernetes packages
      6. bootstrap one control plane node, and then make other control plane nodes join it
      7. make compute plane nodes join the cluster
7. ---
8. How will you upgrade your k8s cluster?
   1. first unhold kubeadm.
   2. install required version of kubeadm.
   3. then hold kubeadm.
   4. then upgrade the node components with the help of ‘kubectl upgrade’ command.
   5. then drain the particular node
   6. unhold kubectl, kubelet etc packages.
   7. install required versions of kubectl, kubelet etc.
   8. then hold kubectl, kubelet etc.
   9. restart kubelet and restart daemon
   10. cordon the particular node
9. ---
10. What are objects in k8s?
11. What are different approaches to create objects in k8s?
12. ---
13. What are pods in k8s?
14. Explain the architecture of how pod creation work.
15. What are different approaches of creating pods?
    1. declarative - using yaml file
    2. imperative - using kubectl command
16. What is the advantage of using shared volume in a pod?
    1. shared volume will not be deleted if anything happens to the individual containers. shared volume will be deleted once entire pod is deleted.
17. How will you access your application running inside pod from outside of k8s cluster?
    1. a temporary solution is to do port forwarding. but its temporary solution because it is not guaranteed that the pod will be created in one particular node always using which we are doing port forwarding.
18. What are initContainers in pods and why we use them?
19. What are the different restart policies in pods?
20. What is a static pod?
21. What are the challenges of using standalone pods?
22. ---
23. Why we need replication of pods?
24. What is replication controller?
25. What is the use of selector in replication controller yaml file?
26. How to scale in/out in a replication controller?
    1. kubectl scale --replica=<number\_of\_replicas> rc/<name\_of\_rc>
27. Is it possible to delete rc without deleting the pods under it?
    1. kubectl delete –cascade=orphan rc <name\_of\_rc>
28. What will happen if I delete the rc, then change the yaml file, and recreate the rc? Will the rc reuse the pods? Will it update the pods?
    1. rc will reuse the pods, but might not update them. we might need to delete the pods (or scale to 0 and then scale back to required number), so that rc creates new pods with new config.
29. ---
30. What is replicaset?
31. Explain the use of matchLabels and matchExpressions in replicaset yaml.
32. How to scale in/out in a replicaset?
    1. kubectl scale --replica=<number\_of\_replicas> rs/<name\_of\_rs>
33. Is it possible to delete rs without deleting pods under it?
    1. kubectl delete –cascade=orphan rs <name\_of\_rs>
34. ---
35. What is deployment in k8s?
36. Why we use deployment in production?
37. How to scale in/out in a deployment?
    1. kubectl scale --replica=<number\_of\_replicas> deploy/<name\_of\_deployment>
38. What is rollout and rollback feature in deployment in k8s?
    1. we can rollout new updates as well as rollback to previous versions in a deployment.
    2. a newer update will create a newer version and a new replica set which in turn creates new pods having newer update. old replicaset will still be present but with 0 pods in it, it is not deleted to maintain the previous versions.
    3. rolling back to a previous version will delete that previous version and create a new version which will be exactly similar to the deleted previous version.
    4. replicaset ~ version
39. What is rollingupdate strategy?
    1. all the pods are not deleted at one time and recreated with new updates. based on the values of max unavailable and max surge, deletion and recreation of new pods will happen phase by phase with few pods taken at a time.
40. What is max unavailable and max surge in rollingupdate strategy?
41. What is recreate strategy?
    1. all the pods will be deleted and recreated with new updates at a time.
42. When to use rollingupdate and recreate?
    1. rollingupdate - u don’t want any downtime of u application
    2. recreate - u r ok with downtime and at a time update
43. What is revision history limit?
44. ---
45. What is DaemonSet object in k8s and when to use it?
46. How does the rollout and rollback feature work in daemonset in k8s?
47. What is rollingupdate strategy in daemonset?
    1. same as deployment
48. What is ondelete strategy in daemonset?
    1. updates will be rolled out to pods once we delete them manually in each node. they won’t update themselves.
49. How revision history limit works in daemonset?
50. ---
51. What are jobs in k8s?
52. When to use jobs?
53. What are different restart policies for jobs in k8s?
    1. never - will not restart the errored pod but will spawn a new pod for a few times (defined by backoffLimit value)
    2. onfailure - will try to restart the errored pod
    3. always is not allowed
54. What is the use of suspend parameter?
    1. suspend true will terminate any non-completed pods. once suspend is made false, pods will start again.
55. How will you cleanup your jobs?
    1. ttlSecondsAfterFinished parameter - will remove the job and corresponding pods details after this much seconds
56. ---
57. What are cronjobs in k8s?
58. What are different restart policies in cronjobs?
    1. same as jobs
59. Can you suspend a cronjob?
    1. yes, using suspend parameter
60. What are different concurrency policies in cronjobs?
    1. allow
    2. replace
    3. forbid
61. How to configure history limits in cronjobs?
    1. successful history limit, failed history limit parameters
62. How to cleanup ur cronjobs?
    1. ttlSecondsAfterFinished parameter - will remove jobs and corresponding pods details after this time. shd be mentioned under .spec.jobtemplate.spec section.
63. ---
64. What is a statefulset object in k8s?
65. How to scale in/out a statefulset object?
    1. the pv creation should be dynamic and not manual which will help in scale in/out operations. statefulset object uses volumeClaimTemplates to create dynamic pvc object.
66. Which service is recommended to use with statefulset object and why?
    1. headless
67. What are different update strategies in stateful set?
    1. rollingupdate
    2. ondelete
68. ---
69. What is a service in k8s?
70. Why service is needed?



1. What is endpoint object?
   1. if a service is created with selection mentioned, an endpoint object will be created which will figure out the pods based on required labels. among those pods, service will load balance the traffic.
2. What is clusterip service?
3. How does the traffic flow between source and destination pods in case of clusterip service?



1. What is the limitation of clusterip service?
   1. since the clusterip is private and can be used only inside the cluster (using ip or dns), we cant use clusterip for traffic originating external to the cluster.
2. What is nodeport?
3. Explain architecture of nodeport.
4. How will you change the port range for nodeport?
   1. go to any control plane node. at /etc/kubernetes/manifests/, open kube-apiserver.yaml file and add the entry –service-node-port-range=<port-range>.
5. How does the traffic flow from external source to an application running inside pods using nodeport?
6. What is load balancer service and what is the difference between service and load balancer?
7. How will you setup a load balancer in k8s cluster?
   1. do some initial configurations
   2. install metallb load balancer using its yaml file. it will deploy speaker pod in each node and a controller pod in its own namespace.
   3. do some post configurations
   4. now try to create a service of load balancer type.
8. How does traffic flow from external source to application pods when using load balancer service?
9. What is externalip service?
10. What is externalname service?
11. What is headless service and when should we use it?
12. ---
13. What are volumes?
    1. At its core, a volume is a directory, possibly with some data in it, which is accessible to the containers in a pod. How that directory comes to be, the medium that backs it, and the contents of it are determined by the particular volume type used.
    2. https://kubernetes.io/docs/concepts/storage/volumes/
14. What are different types of volumes?
    1. volume plugins can be ephemeral or persistent.
    2. emptyDir, hostPath, nfs, downwardapi, persistent volume claim, configmaps, secrets etc…
    3. don’t confuse. under volume section, we can write different plugins like emptyDir, hostPath, nfs etc and also we can write pvc. a pvc points to a pv which in-turn can use plugins like hostPath, nfs etc. the difference between plugins used as it is and plugins used in pv is that the first one is somehow attached to pod lifecycle, but pv in its own is a separate object.
15. Explain emptyDir volume type?
    1. its like volumemount in docker.
16. Where the emptyDir volume is mounted?
    1. disk - /var/lib/kubelet/pods/poduid/volumes/kubernetes.io~empty-dir/<volume-name>/
    2. memory - tmpfs
17. Is emptyDir persistent or ephemeral?
    1. ephemeral to pod, if pod is deleted the emptyDir volume will be deleted.
    2. persistent to containers inside pod
18. Explain hostPath volume type?
    1. its like bindmount in docker.
19. Where is hostPath volume mounted?
    1. disk - path u have given
20. Is hostPath ephemeral or persistent?
    1. persistent to pod, provided that pod has not changed the host node. if pod is deleted and created back in the same host node, then it can use the same hostPath dir. if pod changes host node, then in the new node, new hostPath dir will be created/needed.
21. What is Directory and DirectoryOrCreate in hostPath?
    1. Directory - pod will expect the hostPath dir to be present and pod creation will fail if its not present.
    2. DirectoryOrCreate - pod will create the hostPath dir if not present already.
22. What is nfs volume mount?
    1. its like nfs on volume mount in docker.
23. How will you make nfs work in k8s cluster?
    1. go to any server which you want to make as nfs server. Install nfs server package in it.
    2. create one folder which you want to expose as part of nfs and give proper permissions to it.
    3. open /etc/exports file and make an entry for the folder you want to expose.
    4. in all the k8s cluster nodes, install nfs common package so that they can talk to nfs server.
24. Is nfs ephemeral or persistent?
    1. multiple pods on diff nodes ~~-> multiple /var/lib/kubelet/pods/podid/volumes/…….. folders on diff nodes~~ -> mounted to same /var/nfs/<folder\_name> given nfs path (this way multiple pods running on diff nodes can share same data).
    2. this way if a pod gets deleted and recreated in same or other node, it will not lose data.
25. What is downwardapi volume?



1. What are fieldRef and resourceFieldRef?
2. What is persistent volume and persistent volume claim?
   1. pv is a separate volume object
   2. pv lifecycle is independent of pod workload which is using it.
   3. pv is taken care by an admin either manually or dynamically with the help of storageclasses.
   4. pv also supports plugins like hostPath, nfs etc.
   5. pv is cluster level object.
   6. pv cannot be accessed by workloads directly. they need pvc to access pv.
   7. pvc is kind of a request to use a pv.
   8. pvc will be created by developer.
   9. pvc is namespace level object.
3. What are the different types of reclaim policies in pv?
4. What are the different types of access modes in pv?
5. What are the different phases of a pv?
6. ---
7. What is configmap?
8. What are the different ways to create a configmap?
   1. imperative - kubectl create configmap <cm\_name> --from-env-file=<file\_path> this is just one method
   2. declarative
9. What are the different ways of utilizing configmaps?
   1. environment variables - static
   2. as volume plugin - dynamic
10. What is the use of immutable parameter in configmap?
    1. so that nobody can modify configmap data using edit or patch
11. What are secrets in k8s?
12. What are the different ways to create a secret?
    1. imperative
    2. declarative
13. What are the different ways of using secrets?
    1. volumes
    2. environment variables
    3. imagepullsecrets
14. What are the different types of secrets?
    1. generic
    2. dockerimage
    3. certs etc.
15. ---
16. How scheduler schedules pods in k8s cluster?
17. How will you tell the scheduler where you want your pods to be scheduled?
    1. nodeName
    2. nodeSelector where you can provide only one key-value label selector
    3. nodeAffinity
       1. requiredDuringSchedulingIgnoredDuringExecution
       2. preferredDuringSchedulingIgnoredDuringExecution
    4. nodeAffinity with NotIn operator (kinda nodeAntiAffinity)
    5. podAffinity
       1. requiredDuringSchedulingIgnoredDuringExecution
       2. preferredDuringSchedulingIgnoredDuringExecution
    6. podAntiAffinity
       1. requiredDuringSchedulingIgnoredDuringExecution
       2. preferredDuringSchedulingIgnoredDuringExecution
    7. taints and tolerations
       1. NoSchedule
       2. NoExecute
18. ---
19. Explain how authentication, authorization takes place in on-prem cluster.
20. Explain how authentication, authorization takes place in cloud (aws, azure) cluster.
21. Explain what is authentication, and what are various types of authentications.
    1. normal users
       1. certs - x509 using openssl, cfssl, easyrsa
       2. password - ldap with keystone
       3. token
    2. SA
       1. token
22. Explain what is authorization, and what are various types of authorizations.
    1. ABAC
    2. RBAC
       1. Role, ClusterRole
          * resources and verbs
       2. RoleBinding, ClusterRoleBinding
          * subjects and roleRef
    3. node
    4. webhook
23. What is .kube/config file and why is it important?
24. How will you make changes to .kube/config file?
    1. imperative - kubectl config command
25. Let’s say we have a user. We need to create proper authentication and authorization for it. How can you do that?
    1. authentication - certs via openssl
    2. authorization - role, rolebinding OR clusterrole, clusterrolebinding
    3. process
       1. generate a rsa private key using openssl
       2. generate csr for private key for a user/group
       3. generate client certificate for that user/group signed by k8s ca
       4. update kubeconfig file with details of key and crt file, context and other info
       5. create a role/clusterrole with proper permissions on proper resources
       6. create a rolebinding/clusterrolebinding binding it to the user
       7. check access now after running commands
26. What is service account?
    1. service account can be used like a normal user like above or by applications running inside pod.
27. Let’s say we have a SA. We need to provide proper authentication & authorization to it and use it like a normal user. How can we do it?
    1. authentication - token
    2. authorization - role/clusterrole with corresponding binding
    3. process
       1. create a service account inside a particular namespace.
       2. create a secret to store permanent serviceaccount token. the secret will contain the token as well as ca certificate.
       3. once u get ca certificate and token info, update them in .kube/config file. inside that file, also put a context section with proper details.
       4. create a role/clusterrole and corresponding bindings for that SA.
       5. now we are ready to use that SA.
28. How to make pods use user defined service account?
    1. while creating a pod, u need to mention the service account name and its corresponding namespace. along with that, u need to mention projected volume to mount serviceaccount token, crt etc. after that only applications running inside pod can communicate with api server.
29. ---
30. What are namespaces in k8s?
31. ---
32. What is ingress controller and why we need it?
33. What is nginx ingress controller?
34. Explain the architecture of nginx ingress controller as to how it works.
35. What are different nginx ingress controller components?
36. Explain the architecture of a simple host-based routing app which uses nginx ingress controller.
37. Explain the architecture of a simple path-based routing app which uses nginx ingress controller.
38. Explain tls termination for ingress controller.
    1. u need to have proper secrets which contains key and crt details. give those secrets details along with uri details in ingress yaml file while creating an ingress.
39. ---
40. Explain k8s network policy.
    1. ingress - from, egress - to
    2. podSelector, namespaceSelector, ipBlock, and-ing any two or three of them
    3. default policies
41. ---
42. What is k8s dashboard? Why we need it?
43. How will you install a dashboard?
    1. go to official k8s repo. there you will find the yaml file to create a dashboard. apply that yaml.
    2. in official docs, you will find yaml to create admin sa as well a clusterrolebinding. using that create admin sa with proper role & bindings.
    3. create a token for that sa and provide it in dashboard for login purposes.
44. ---
45. Why we need logging and monitoring for k8s cluster?
46. What is Prometheus and why we need it?
47. Explain the architecture of Prometheus.
48. What is Grafana and why we need it?
49. ---