Docker:

**Docker Engine:**

1) **Container tools**

* lxc, Docker,containerd etc.,

2) **Docker Engine Installion**

* Manual deployment of Docker CE on Ubuntu and CentOS
* Automate deployment with help of Vagrant

3) **Basic introduction to Docker Engine**

* Datapath
* version check
* default locations
* commands execution

4) **Docker architecture**

* Docker daemon
* Docker client
* Docker registry
* Docker objects
* imgaes
* containers
* netwokring
* services

5) **Underlying** **technologies of Docker**

* namespaces like pid,net,ipc,mnt,uts
* cgroup

UFS (Union File System)

* container format like libcontainer,BSD jails etc.,

6) **Images and container management**

* docker image pull/push
* Image layers
* container creation
* Sample application deployment like web server
* Basic things that to be considered while working with images and containers

7) **Docker networking**

* Understanding netwokring like IP, Classes, subnetmask, CIDR, gateway, broadcast, public and provate network in each class etc.,

Types of network in Docker:

* host
* bridge
* null
* overlay
* macvlan

8) **Docker storage**

* Basic understanding of emphemeral and persistent storage
* Types of volume mounts
* Bind mount
* Volume mount
* tempfs mount

**9) Backup and restore of Images, containers and Docker/Application data**

10) **Building custom images**

* Develop custom image through Dockerfile
* Convert container into Image

11) **Registry management**

* Docker Hub
* Gitlab registry
* Local registry

12) **Securing Docker**

* Ensure Docker communication through TLS
* Firewall ports
* Communication to Docker through unix socket and tcp protocal

13) **Docker overlay network configuration(hardway) without using Docker Swarm**

14**) Docker compose, Docker machine and Docker swarm**

* Building compose file for orchestration of application deployment
* Docker machine management
* Deployment and understanding of Docker Swarm cluster
* Difference between docker swarm and kubernetes

15) **Task**

* Deployment of application like nginx, mariadb and jenkins
* Deploy wordpress as webserver and database for storage using Dockercompose
* Deploy Node.js application with custom Dockerfile
* Automate Image build with help of Jenkins CICD pipeline

**JSON and YAML core concepts:**

1) what is JSON?

2) how to write a JSON file and validate for testing?

3) what is YAML?

4) how to write YAML files and validate for testing?

**kubernetes Deep Dive:**

1) kubernetes components and responsibility

controller nodes:

* kube-apiserver
* kube-control-manager/cloud-control-manager
* kube-schduler
* calico,flannel(deprecated from 1.18),wavenet etc.,
* coreDNS
* etcd key-value pair storage
* kubelet
* kube-proxy
* Docker Engine/containerd/cri-o etc.,

**compute nodes:**

* kubelet
* kube-proxy

Docker Engine/containerd/cri-o etc.,

2**) Deployment of kubernets cluster using microk8s(ubuntu),minikube,kubeadm**

Three nodes (1 for controller and 2 for compute nodes)

3) **POD object creation using single container and multi-container**

* Understanding common networking and common storage in POD (shared namespaces)
* Sample application deployment using nginx for single container in POD
* Multi-container deployment, one for web application and second for data pull
* POD creation with initContainer advantages for pre-request
* static POD deployment
* Deploying application from Private repository
* resource allocation like CPU and MEMORY

4) **ReplicationController object (RC)**

* When to use replication controller?
* Disadvanatages of deploying application with single POD
* Understanding template and selector parameters

5) **ReplicaSet object (RS)**

* Difference between Rreplication Controller and ReplicaSet
* Advantages of adding matchLabels
* when to use replicaSet?

6) **Deployment object (deploy)**

* Advantages of Deployment over POD, RC and RS
* Sample application deployment like Jenkins, nginx and capture the advantages over other objects
* Scaleout and Scalein for Deployment
* Deployment Stratagies like RollingUpdate and ReCreate difference
* Rollout, Rollback and RolloutHistory of deployment

7) **Daemonset object (DS)**

* DaemonSet usecases over deployment object
* Advantages of using monitoring tools to deploy through DaemonSet obejct
* Deployment Stratagies like RollingUpdate and OnDelete difference
* Rollout, Rollback and RolloutHistory of DaemonSet

8) **StatefulSet object (sts)**

* Difference between stateful and stateless applications
* StatefulSet application deployment advantages
* Scaleout and Scalein for Deployment
* Deployment Stratagies like RollingUpdate and OnDelete difference
* Rollout, Rollback and RolloutHistory of StatefulSet

9) **volumes:**

* empheimeral and persistant storage differences
* Different types of volume plugins like emptyDir, hostPath, NFS etc.,
* Do's and Dont's of using volumes for application deployment

10) **PersistentVolume, PersistantVolumeClaim and StorageClass (PV and PVC)**

* Advanatages of using PV over volume plugins for data dependent applications
* Static and Dynamic PV creation
* PVC assignment in POD, Deployment, Daemonset, StatefulSet objects
* StargaeClass creation for Dynamic provision
* NFS, Ceph RBD storage as examples

11) **Services (svc)**

* Exposing application running on different object through service
* Understanding differenece between ClusterIP, NodePort, LoadBalancing, ExternalIP and ExternalName
* When to use these different types of services?

12) **secrets and configmap (secret, cm)**

* How to pass sensitive data through secret like certs, password, token etc.,
* How to pass paintext data through configMap like configuration files, scripts etc.,
* Using secret in POD object through env and volumes
* using configMap in POD object through env and volumes

13) **Ingress (ing)**

* Access application deployed with the cluster through IngressController
* Deploying Nginx IngressController
* Ingress rules creation secured and non-secure communication

14) **Horizontal POD AutoScalling (HPA):**

* HPA deployent to test auto scalling
* Deployment of metric server to achieve HPA

15) **Scheduling and probes**

* Different types of scheduling like nodename, nodeSelector, podAffinity, podAntiAffinity, nodeAffinity
* Advantages of using liveness, readiness and startup probes to verify application availability

16) **Jobs and CronJobs**

* How to achieve auto backup and restore using CronJobs for application like etcd key-value pair storage
* for kubernetes cluster
* Jobs responsibility

17) **calico networking**

* How to migrate from flannel to calico?
* How to use flannel for POD networking and Calico for network security
* calicoctl operations
* network policy and global network policy

18) **authetication and authorization**

* authetication:
* user, group and serviceaccount (token), certs
* kubeconfig entries and access multiple cluster through kubernetes config file
* autherization:
* Role Back Access Control (RBAC)
* role, rolebinding, clusterrole, clusterrolebinding

19) **Dashboard**

* deployment of kubernetes dashboard and creating kubernetes object through dashboard

How to autheticate to dashboard through token and kubeconfig

20) **Task**:

a. wordpress and database application deployment

b. Jenkins deployment with data persistent after upgrading/downgrade

c. Backup and restore of etcd service data using cronjob

d. kubernetes upgrade from one version to another (1.20.1 to 1.20.2)

e. maintaince task using taint and toleration

f. Automate application deployment with the help of GitOps and ArgoCD

g. Understanding CRD and CR for operators

**Helm charts:**

1) what is chart and maifest?

2) helm architecture and components

3) helm configuration and repo setup

4) helm application deployment

a. Deploy nginx web application with custom values

b. Rollout and Rollback application using helm cli

c. test applications deployed through Helm using #helm test

5) helm custom chart development

a. helm default directory structure

b. update \_helper.tpl file for custom methods

c. how to call data from Chart.yaml and values.yaml file to templates

6) Deploy monitoring tools using custom helm charts

a. EFK deployment

b. Prometheus and Grafana deployment