DEVOPS INTERVIEW QUESTIONS & ANSWERS

VERSION CONTROL (GIT):

1. Difference b/w Git Pull & Git Fetch

Git Fetch is the command that tells the local repository that there are changes available in the remote repository without bringing the changes into the local repository.

Git Pull on the other hand brings the copy of the remote directory changes into the local repository.

1. What is amend in git?

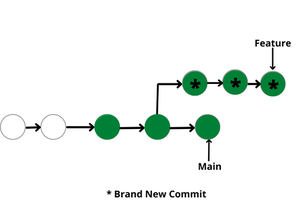
The git commit --amend command is a convenient way to modify the most recent commit. It lets you combine staged changes with the previous commit instead of creating an entirely new commit. It can also be used to simply edit the previous commit message without changing its snapshot.

1. Uses of Version control tool?

Version control systems are a category of software tools that helps in recording changes made to files by keeping a track of modifications done in the code

**Use of Version Control System:**

* A repository: It can be thought of as a database of changes. It contains all the edits and historical versions (snapshots) of the project.
* Copy of Work (sometimes calle as checkout): It is the personal copy of all the files in a project. You can edit to this copy, without affecting the work of others and you can finally commit your changes to a repository when you are done making your changes.
* Working in a group: Consider yourself working in a company where you are asked to work on some live project. You can’t change the main code as it is in production, and any change may cause inconvenience to the user, also you are working in a team so you need to collaborate with your team to and adapt their changes. Version control helps you with the, merging different requests to main repository without making any undesirable changes. You may test the functionalities without putting it live, and you don’t need to download and set up each time, just pull the changes and do the changes, test it and merge it back. It may be visualized as.



Types of Version Control Systems:

* Local Version Control Systems
* Centralized Version Control Systems
* Distributed Version Control Systems

1. Difference b/w Git revert and Git commit?

**git** **revert** as a tool for undoing committed changes, while git reset HEAD is for undoing uncommitted changes. Like git checkout , git revert has the potential to overwrite files in the working directory, so it will ask you to commit or stash changes that would be lost during the revert operation

**git** **commit** is a snapshot of your repo at a specific point in time. To help further understand what a Git commit is, we need to review your Working Directory vs your Staging Directory and how files changes are reflected in your Git repository.

1. What is mean by pull request in GIT?

Pull requests **let you tell others about changes you've pushed to a branch in a repository on GitHub**. Once a pull request is opened, you can discuss and review the potential changes with collaborators and add follow-up commits before your changes are merged into the base branch.

1. How do you make sure that people directly cannot merge into higher branches?

**To protect the branch:**

Navigate to the main page of the project.

In the upper right corner, click the settings wheel and select protected branches.

From the Branch dropdown menu, select the branch you want to protect and click Protect.

Once done, the protected branch will appear in the "Already protected" list.

<https://stackoverflow.com/questions/62759666/how-to-restrict-who-can-merge-to-master-on-a-github-repo>

1. What is status check’s in Pull requests?

Status checks are based on external processes, such as continuous integration builds, which run for each push you make to a repository. You can see the *pending*, *passing*, or *failing* state of status checks next to individual commits in your pull request.

## Checks

When checks are set up in a repository, pull requests have a **Checks** tab where you can view detailed build output from status checks and rerun failed checks.

<https://docs.github.com/en/github-ae@latest/pull-requests/collaborating-with-pull-requests/collaborating-on-repositories-with-code-quality-features/about-status-checks>

1. What is difference between branch and tag in git?

A branch is an active line of development whereas a tag is a an immutable reference to a specific commit on a branch.

<https://medium.com/@vogeleah/tags-vs-branches-in-git-16c4a3dc0a1f>

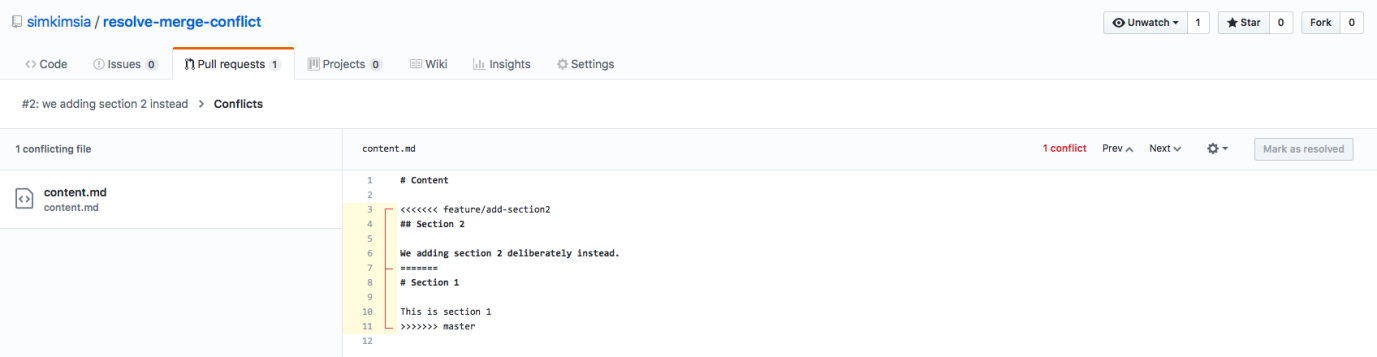
1. How to resolve the merge conflicts in remote repo directly?

<https://www.cloudbees.com/blog/resolve-github-merge-conflicts>

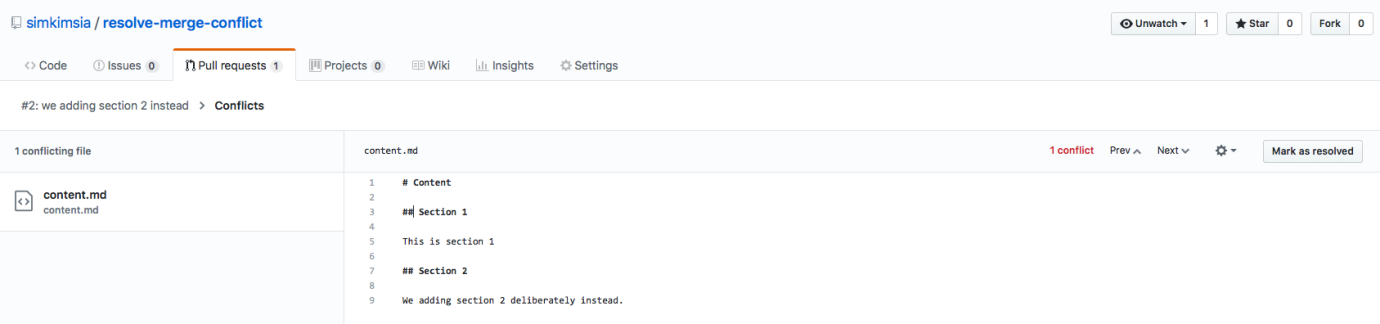
You don't—indeed, cannot—work on a "remote repository", with git. Everything is local. When you fetch or push, your local git contacts a remote—think of it as calling up the other guys on the Internet-phone—and your git and their git exchange information, after which either you give them stuff (git push) or they give you stuff (git fetch).

### Resolve within GitHub's Web Editor

1. Click on **Resolve conflicts** and you should see the entire display of the changed files in the pull request. Notice that GitHub has disabled the **Mark as resolved** button.



1. Resolve the conflicts in the first file you see.
2. Ensure that all traces of <<<<<<, >>>>>>, and ====== are removed.
3. If you do this correctly, you should see the button **Mark as resolved** become available for that particular file.



1. If you have multiple files with conflicts, select the next file to resolve. Repeat steps two through four until you've resolved all of your pull requests' merge conflicts.
2. Now the **Commit merge** button is available.

MAVEN:

Maven life cycle goals:

the default lifecycle comprises of the following phases (for a complete list of the lifecycle phases, refer to the [Lifecycle Reference](https://maven.apache.org/guides/introduction/introduction-to-the-lifecycle.html#Lifecycle_Reference)):

* validate - validate the project is correct and all necessary information is available
* compile - compile the source code of the project
* test - test the compiled source code using a suitable unit testing framework. These tests should not require the code be packaged or deployed
* package - take the compiled code and package it in its distributable format, such as a JAR.
* verify - run any checks on results of integration tests to ensure quality criteria are met
* install - install the package into the local repository, for use as a dependency in other projects locally
* deploy - done in the build environment, copies the final package to the remote repository for sharing with other developers and projects.

JENKINS:

1. How to trigger Jenkins pipeline automatically?

<https://blog.knoldus.com/jenkins-build-triggers/>

1. What is logger in Jenkins?

To view logs from a specific action exposed by Jenkins, or a plugin not typically monitored, you can add a logger under **Manage** **Jenkins** -> **System** **Log** -> Add **new** **log** **recorder**. To add a new logger you will be asked to provide it a name and choose a function to monitor. You'll also have the ability to change the verbosity.

<https://docs.cloudbees.com/docs/cloudbees-ci-kb/latest/client-and-managed-masters/how-do-i-create-a-logger-in-jenkins-for-troubleshooting-and-diagnostic-information>

1. How to provide email notifications to developers in Jenkins if job gets failed?

Solution: Email Extension plugin

<https://www.edureka.co/blog/email-notification-in-jenkins/>

1. What are plugins you installed in Jenkins?  
   Job Import plugin

Email extension plugin

Git plugin

Dockers build & Publish plugin

Think Backup plugin

Remote SSH plugin

Version Number plugin

Ansible plugin

Kubernetes plugin

Terraform plugin

Sonar scanner plugin

Maven release plugin

1. What is Master /Slave configuration in Jenkins?

**The Jenkins master acts to schedule the jobs, assign slaves, and send builds to slaves to execute the jobs.** It will also monitor the slave state (offline or online) and get back the build result responses from slaves and the display build results on the console output.

In some cases, files may have different code and require multiple builds and the Jenkins server cannot handle multiple builds simultaneously for this, **the Master distributes the workload and allows us to run different builds on different environments each called a Slave.**

1. How to schedule a Jenkins job each 15 minutes?

To run the job at a regular interval of 15 minutes you have to write it like below:

\*/15 \* \* \* \* - Will run at every 15 minutes (may be at XX:01,XX:16,XX:31 ..)

Where \*/15 specifies no matter whatever is Hour (H) run it at 15 every minutes.

Currently i'm using the Build periodically feature but that enables me to schedule the job once in 1 hour maximum.

What i got now is:

15 \* \* \* \* is running the job HOURLY (each XX:15)

15 0 \* \* \* is running the job DAILY on 00:15

1. how to make Jenkins backup for your work?

Go to Manage Jenkins — > ThinBackup.

Click settings option.

Enter the backup options as shown below and save them. ...

Now, you can test if the backup is working by clicking the Backup Now option.

<https://devopscube.com/jenkins-backup-data-configurations/>

1. Have to written Jenkins pipeline from scratch or any enhancements or just do support to the tasks? what is role on that?

To create a simple pipeline from the Jenkins interface, perform the following steps: Click **New** **Item** on your Jenkins home page, enter a name for your (**pipeline**) job, select Pipeline, and click OK. In the Script text area of the configuration screen, enter your pipeline syntax.

1. What are the stages in scripted pipeline in Jenkins?

Your pipeline will consist of several steps that can be grouped in stages.  Among these stages you might have:

* Pull code from repository
* Build your project
* Deploy your application
* Perform functional tests
* Perform performance tests

<https://www.blazemeter.com/blog/jenkins-scripted-pipeline>

1. Difference b/w scripted and declarative pipeline in Jenkins?

Declarative pipeline is a relatively new feature that supports the pipeline as code concept. It makes the pipeline code easier to read and write. This code is written in a Jenkinsfile which can be checked into a source control management system such as Git.

Whereas, the scripted pipeline is a traditional way of writing the code. In this pipeline, the Jenkinsfile is written on the Jenkins UI instance.

Though both these pipelines are based on the groovy DSL, the scripted pipeline uses stricter groovy based syntaxes because it was the first pipeline to be built on the groovy foundation.

<https://www.edureka.co/community/54705/difference-between-declarative-pipeline-scripted-pipeliine>

**DOCKER:**

* 1. **Difference b/w Docker swarm and kubernetes?**Docker Swarm is a lightweight, easy-to-use orchestration tool with limited offerings compared to Kubernetes. In contrast, Kubernetes is complex but powerful and provides self-healing, auto-scaling capabilities out of the box.
  2. Docker Commands list?

<https://medium.com/edureka/docker-commands-29f7551498a8>

* docker –version
* docker pull
* docker run
* docker ps
* docker ps -a
* docker exec
* docker stop
* docker kill
* docker commit
* docker login
* docker push
* docker images
* docker rm
* docker rmi
* docker build
  1. Difference between Docker COPY and ADD?

**COPY** is a docker file command that copies files from a local source location to a destination in the Docker container.

**ADD** command is used to copy files/directories into a Docker image. It only has only one assigned function. It can also copy files from a URL.

* 1. Difference b/w Docker CMD and Entrypoint?

The main purpose of a **CMD** is to provide defaults for an executing container.

An **ENTRYPOINT** helps you to configure a container that you can run as an executable.

* 1. Docker networking issues and troubleshooting

[ht tps://birthday.play-with-docker.com/troubleshooting-network-issues/](https://birthday.play-with-docker.com/troubleshooting-network-issues/)

* 1. Docker file issues and troubleshoot:

<https://docs.docker.com/desktop/windows/troubleshoot/>

* 1. **how to backup of your container if it is failes**

## backup docker images

To backup docker images, use the [docker save](https://docs.docker.com/engine/reference/commandline/save) command that will produce a tar archive that can be used later on to create a new docker image with the [docker load](https://docs.docker.com/engine/reference/commandline/load/) command.

## backup docker containers

You can backup a docker container by different means

* by committing a new docker image based on the docker container current state using the [docker commit](https://docs.docker.com/engine/reference/commandline/commit/) command
* by exporting the docker container file system as a tar archive using the [docker export](https://docs.docker.com/engine/reference/commandline/export/) command. You can later on create a new docker image from that tar archive with the [docker import](https://docs.docker.com/engine/reference/commandline/import/) command.

Be aware that those commands will only backup the docker container layered file system. **This excludes the data volumes**.

## backup docker data volumes

To backup a data volume you can run a new conta iner using the volume you want to backup and executing the tar command to produce an archive of the volume content as described in the [docker user guide](https://docs.docker.com/storage/volumes/" \l "backup-restore-or-migrate-data-volumes).

In your particular case, the data volume is used to store the data for a MySQL server. So if you want to export a tar archive for this volume, you will need to stop the MySQL server first. To do so you will have to stop the wordpress container.

**Docker Volumes:**

Volumes are the preferred mechanism for persisting data generated by and used by Docker containers. While [bind mounts](https://docs.docker.com/storage/bind-mounts/) are dependent on the directory structure and OS of the host machine, volumes are completely managed by Docker. Volumes have several advantages over bind mounts:

* Volumes are easier to back up or migrate than bind mounts.
* You can manage volumes using Docker CLI commands or the Docker API.
* Volumes work on both Linux and Windows containers.
* Volumes can be more safely shared among multiple containers.
* Volume drivers let you store volumes on remote hosts or cloud providers, to encrypt the contents of volumes, or to add other functionality.
* Create a volume
* $ docker volume create [OPTIONS] [VOLUME]
* docker volume create hello
* hello
* $ docker run -d -v hello:/world busybox ls /world

# docker manifest:

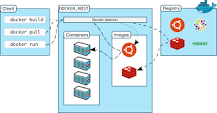
The docker manifest command by itself performs no action. In order to operate on a manifest or manifest list, one of the subcommands must be used.

A single manifest is information about an image, such as layers, size, and digest. The docker manifest command also gives users additional information such as the os and architecture an image was built for.

A manifest list is a list of image layers that is created by specifying one or more (ideally more than one) image names. It can then be used in the same way as an image name in docker pull and docker run commands, for example.

### manifest inspect , manifest create

**What is a Docker namespace?**

[[](https://www.google.com/search?rlz=1C1GCEJ_enIN974IN974&sxsrf=ALiCzsapA-QcSeGJDMfF93C-tqV18oQaUQ:1655868500139&q=What+is+a+Docker+namespace?&tbm=isch&source=iu&ictx=1&vet=1&fir=ozZsDDX9ZCaf4M,3Rlh9Be8h2ItMM,_&usg=AI4_-kR4jS6uiaP5Zll7eGDpZkrO6PDLxw&sa=X&ved=2ahUKEwjA9frtjsD4AhVf1jgGHcSdBKMQ9QF6BAgcEAE#imgrc=ozZsDDX9ZCaf4M)](https://www.google.com/search?rlz=1C1GCEJ_enIN974IN974&sxsrf=ALiCzsapA-QcSeGJDMfF93C-tqV18oQaUQ:1655868500139&q=What+is+a+Docker+namespace?&tbm=isch&source=iu&ictx=1&vet=1&fir=ozZsDDX9ZCaf4M%252C3Rlh9Be8h2ItMM%252C_&usg=AI4_-kR4jS6uiaP5Zll7eGDpZkrO6PDLxw&sa=X&ved=2ahUKEwjA9frtjsD4AhVf1jgGHcSdBKMQ9QF6BAgcEAE" \l "imgrc=ozZsDDX9ZCaf4M)

Docker uses a technology called namespaces **to provide the isolated workspace called the container**. When you run a container, Docker creates a set of namespaces for that container. These namespaces provide a layer of isolation.

**Namespace Types:**

1. Process ID
2. Mount
3. IPC (Interprocess communication)
4. User (currently experimental support for)
5. Network

8. Write a Docker file to push Jfrog image?

9. How to create container in Docker?

Create a new container

## Usage

$ docker container create [OPTIONS] IMAGE [COMMAND] [ARG...]

### Create and start a container

The following example creates an interactive container with a pseudo-TTY attached, then starts the container and attaches to it:

$ docker container create -i -t --name mycontainer alpine

6d8af538ec541dd581ebc2a24153a28329acb5268abe5ef868c1f1a261221752

$ docker container start --attach -i mycontainer

/ # echo hello world

9. How to update running container?

Docker update

Update configuration of one or more containers

## Usage

$ docker update [OPTIONS] CONTAINER [CONTAINER...]

## Examples

The following sections illustrate ways to use this command.

### Update a container’s cpu-shares

To limit a container’s cpu-shares to 512, first identify the container name or ID. You can use docker ps to find these values. You can also use the ID returned from the docker run command. Then, do the following:

$ docker update --cpu-shares 512 abebf7571666

KUBERNETES:

1.What is Kubernetes Auto Scaling?

In Kubernetes, **a HorizontalPodAutoscaler automatically updates a workload resource (such as a Deployment or StatefulSet), with the aim of automatically scaling the workload to match demand**. Horizontal scaling means that the response to increased load is to deploy more Pods

#### Starting Horizontal Pod Autoscaler

Now that the deployment is running, we will create a Horizontal Pod Autoscaler for it. To create it, we will use kubectl autoscale command, which looks like this:

$ kubectl autoscale deployment php-apache --cpu-percent=50 --min=1 --max=10

We may check the current status of autoscaler by running:

$ kubectl get hpa

NAME REFERENCE TARGET CURRENT MINPODS MAXPODS AGE

php-apache Deployment/php-apache/scale 50% 0% 1 20 14s

## 2.How to Create a Pod in Kubernetes

To begin, you need to [launch a Kubernetes cluster](https://www.fairwinds.com/developer-hub/security-basics-for-your-first-kubernetes-cluster). Once you’re in the Kubernetes sandbox environment, make sure you’re connected to the Kubernetes cluster by executing kubectl get nodes in the command line to see the cluster's nodes in the terminal. If that worked, you’re ready to create and run a pod.

To create a pod using the nginx image, run the commandkubectl run nginx --image=nginx --restart=Never. This will create a pod named **nginx**, running with the [nginx image](https://hub.docker.com/_/nginx/) on Docker Hub. And by setting the flag --restart=Neverwe tell Kubernetes to create a single pod rather than a Deployment.

<https://www.fairwinds.com/blog/how-to-create-view-and-destroy-a-pod-in-kubernetes#:~:text=To%20create%20a%20pod%20using,pod%20rather%20than%20a%20Deployment>.

2.Kubernets yaml script for pod creation?

<https://eskala.io/tutorial/how-to-write-yaml-files-for-kubernetes/#:~:text=How%20to%20create%20a%20kubernetes,the%20containers%20within%20the%20pod>.

To create a Kubernetes pod with YAML, you first create an empty file, assign it the necessary access permissions, and then define the necessary key-value pairs. The important ones are the apiVersion, the kind (pod), name, and the containers within the pod.

For instance, below is the YAML code to create a pod named mywebapp1 that has 2 containers: One is a web server and the other is a database server. It is also assigned to a specific volume named websvr-storage:

Filename: /k8s/pods/pod1.YAML

apiVersion: v1

kind: Pod

metadata:

name: mywebapp1

labels:

role: webserver-role

app: nginx

spec:

containers:

- name: webserver1

image: nginx:1.6

ports:

- containerPort:80

- name: database-server

image: mysql-3.2

ports:

- containerPort:3306

volumes:

- name: websvr-storage

emptyDir: {}

1. How to recovery if pod gets deleted?

When the pod is recreating automatically even after the deletion of the pod manually, then those pods have been created using the Deployment. When you create a deployment, it automatically creates ReplicaSet and Pods. Depending upon how many replicas of your pod you mentioned in the deployment script, it will create those number of pods initially. When you try to delete any pod manually, it will automatically create those pod again.

Yes, sometimes you need to delete the pods with force. But in this case force command doesn’t work.

I have started pods with command

$ kubectl run busybox \

--image=busybox \

--restart=Never \

--tty \

-i \

--generator=run-pod/v1

Something went wrong, and now I can't delete this Pod.

I tried using the methods described below but the Pod keeps being recreated.

$ kubectl delete pods busybox-na3tm

pod "busybox-na3tm" deleted

$ kubectl get pods

NAME READY STATUS RESTARTS AGE

busybox-vlzh3 0/1 ContainerCreating 0 14s

$ kubectl delete pod busybox-vlzh3 --grace-period=0

$ kubectl delete pods --all

pod "busybox-131cq" deleted

pod "busybox-136x9" deleted

pod "busybox-13f8a" deleted

pod "busybox-13svg" deleted

pod "busybox-1465m" deleted

pod "busybox-14uz1" deleted

pod "busybox-15raj" deleted

pod "busybox-160to" deleted

pod "busybox-16191" deleted

$ kubectl get pods --all-namespaces

NAMESPACE NAME READY STATUS RESTARTS AGE

default busybox-c9rnx 0/1 RunContainerError 0 23s

3.kubernetes replicaset and scalability?

**A ReplicaSet (RS) is a Kubernetes object that ensures there is always a stable set of running pods for a specific workload**. The ReplicaSet configuration defines a number of identical pods required, and if a pod is evicted or fails, creates more pods to compensate for the loss.

**Scalability** :In a Kubernetes cluster, scalability refers to **the ability of the cluster to grow while staying within its service-level objectives (SLOs)**. Kubernetes also has its own set of SLOs. Kubernetes is a complex system, and its ability to scale is determined by multiple factors.

## 4. What happens when one of your Kubernetes nodes fails?

This section details what happens during a node failure and what is expected during the recovery.

1. Post node failure, in about **1 minute**, kubectl get nodes will report NotReady state.
2. In about **5 minutes**, the states of all the pods runni ng on the NotReady node will change to either Unknown or NodeLost.This is based on [pod eviction timeout](https://kubernetes.io/docs/concepts/architecture/nodes/#condition) settings, the default duration is **five minutes**.
3. Irrespective of deployments (StatefuleSet or Deployment), Kubernetes will automatically evict the pod on the failed node and then try to recreate a new one with old volumes.
4. If the node is back online within 5 – 6 minutes of the failure, Kubernetes will restart pods, unmount, and re-mount volumes.
5. If incase if evicted pod gets stuck in Terminating state and the attached volumes cannot be released/reused, the newly created pod(s) will get stuck in ContainerCreating state. There are 2 options now:
   1. Either to forcefully delete the stuck pods manually (or)
   2. Kubernetes will take about another [**6 minutes**](https://github.com/kubernetes/kubernetes/blob/5e31799701123c50025567b8534e1a62dbc0e9f6/pkg/controller/volume/attachdetach/attach_detach_controller.go#L95) to delete the VolumeAttachment objects associated with the Pod and then finally detach the volume from the lost Node and allow it to be used by the new pod(s).

In summary, if the failed node is recovered later, Kubernetes will restart those terminating pods, detach the volumes, wait for the old VolumeAttachm e

1. Configmap and secretes in kubernetes?

Both ConfigMaps and secrets store the data the same way, with key/value pairs, but **ConfigMaps are meant for plain text data, and secrets are meant for data that you don't want anything or anyone to know about except the application**.

<https://www.cloudtruth.com/blog/whats-the-difference-between-configmaps-and-secrets#:~:text=Secrets%20in%20Kubernetes&text=Both%20ConfigMaps%20and%20secrets%20store,know%20about%20except%20the%20application>.

# What is Kubernetes Security?

**Kubernetes Security** is based on the 4C’s of cloud native security: Cloud, Cluster, Container, and Code:

* **Cloud** (or Corporate Datacenter/Colocation facility): The underlying physical infrastructure is the basis of Kubernetes security. Whether the cluster is built on one’s own datacenter or a cloud provider, basic cloud provider (or physical security) best practices must be observed.
* **Cluster:** Securing a Kubernetes cluster involves both the configurable components such as the Kubernetes API and security of all the applications that are part of the cluster. Since most cloud-native applications are designed around microservices and APIs, applications are only as secure as the weakest link in the chain of services that comprise the entire application.
* **Container**: Container design best practices consist of: starting with the smallest code base possible (excluding unnecessary libraries or functions), avoiding granting unnecessary privileges to users in the container, and ensuring that containers are scanned for vulnerabilities at build time.
* **Code**: Code presents a major attack surface for any Kubernetes environment. Simple policies such as encrypting TCP using TLS handshakes, not exposing unused ports, scanning, and testing regularly can help prevent security issues from arising in a production environment.

# What are Kubernetes Services?

A **Kubernetes service** is a logical abstraction for a deployed group of pods in a cluster (which all perform the same function).

## What are the types of Kubernetes services?

* **ClusterIP**. Exposes a service which is only accessible from within the cluster.
* **NodePort**. Exposes a service via a static port on each node’s IP.
* **LoadBalancer**. Exposes the service via the cloud provider’s load balancer.
* **ExternalName**. Maps a service to a predefined externalName field by returning a value for the CNAME record

# [Scale down Kubernetes pods](https://stackoverflow.com/questions/47572597/scale-down-kubernetes-pods)

I am using

**kubectl scale --replicas=0 -f de ployment.yaml**

to stop all my running pods. Please let me know if there are better ways to bring down all running pods to Zero keeping configuration, deployments etc.. intact, so that I can scale up later as required.

# Scale up and down manually with the kubectl scale command

Assume that today we'd like to scale our nginx Pods from two to four:

// kubectl scale --replicas=<expected\_replica\_num> deployment <deployment\_name># kubectl scale --replicas=4 deployment my-nginxdeployment "my-nginx" scaled

Let's check how many pods we have now:

# kubectl get podsNAME READY STATUS RESTARTS AGEmy-nginx-6484b5fc4c-9v7dc 1/1 Running 0 1mmy-nginx-6484b5fc4c-krd7p 1/1 Running 0 1mmy-nginx-6484b5fc4c-nsvzt 0/1 ContainerCreating 0 2smy-nginx-6484b5fc4c-v68dr 1/1 Running 0 2s

We could find two more Pods are scheduled. One is already running and another one is creating. Eventually, we will have four Pods up and running if we have enough compute resources.

**How to run a pod in particular node?**

[**https://kubernetes.io/docs/tasks/configure-pod-container/assign-pods-nodes/#:~:text=Create%20a%20pod%20that%20gets%20scheduled%20to%20specific%20node,specific%20node%20via%20setting%20nodeName%20.&text=nginx%20imagePullPolicy%3A%20IfNotPresent-,Use%20the%20configuration%20file%20to%20create%20a%20pod%20that,scheduled%20on%20foo%2Dnode%20only**](https://kubernetes.io/docs/tasks/configure-pod-container/assign-pods-nodes/#:~:text=Create%20a%20pod%20that%20gets%20scheduled%20to%20specific%20node,specific%20node%20via%20setting%20nodeName%20.&text=nginx%20imagePullPolicy%3A%20IfNotPresent-,Use%20the%20configuration%20file%20to%20create%20a%20pod%20that,scheduled%20on%20foo%2)**.**

## Create a pod that gets scheduled to specific node

You can also schedule a pod to one specific node via setting nodeName.

[pods/pod-nginx-specific-node.yaml](https://raw.githubusercontent.com/kubernetes/website/main/content/en/examples/pods/pod-nginx-specific-node.yaml)

**apiVersion**: v1

**kind**: Pod

**metadata**:

**name**: nginx

**spec**:

**nodeName**: foo-node *# schedule pod to specific node*

**containers**:

- **name**: nginx

**image**: nginx

**imagePullPolicy**: IfNotPresent

Use the configuration file to create a pod that will get scheduled on foo-node only.

How to setup kubernets cluster using kubeadm?

The **[kubeadm](https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/create-cluster-kubeadm/" \t "_blank) tool** is used to bootstrap smaller Kubernetes clusters so that you can experience all the kubernetes features. The cluster spin-up using kubeadm is eligible to pass the [**Kubernetes Conformance Program**](https://kubernetes.io/blog/2017/10/software-conformance-certification/). The cluster life-cycle functions and cluster upgrades are also supported by kubeadm

### Prerequisites

1. One or more machines running a [Linux](https://www.educative.io/edpresso/what-is-kali-linux) operating system like deb/rmp.
2. 2 GiB+ of RAM/machine (works with less RAM, but you cannot run heavy resource applications later on).
3. Minimum 2 vCPUs for the master node (control-plane node).
4. Full network connectivity between machines (machines can either be in a public or private network).

**what is kube controller manager?**

The Kubernetes controller manager is **a daemon that embeds the core control loops shipped with Kubernetes**. In applications of robotics and automation, a control loop is a non-terminating loop that regulates the state of the system.

**What is Kube proxy?**

kube-proxy is **a network proxy that runs on each node in your cluster, implementing part of the Kubernetes Service concept**. kube-proxy maintains network rules on nodes. These network rules allow network communication to your Pods from network sessions inside or outside of your cluster.

**Are you working on managed kubernets cluster(ex:AKS) or self-managed kubernetes cluster**?

### [Managed Kubernetes vs self-managed Kubernetes](https://www.x-cellent.com/posts/managed-kubernetes-vs-self-managed-kubernetes" \l ":~:text=The%20provider%2Dmanaged%20Kubernetes%20service,well%20as%20on%2Dsite%20infrastructure.)

[**https://www.weave.works/blog/managed-kubernetes**](https://www.weave.works/blog/managed-kubernetes)

The provider-managed Kubernetes service reduces the time and effort required to administer and maintain a cluster by taking care of the master node. **One has more control over the cluster using self-managed Kubernetes (flexibility)**. Also, one can use various cloud computing services as well as on-site infrastructure

TERRAFORM:

Terraform commands:

Terraform Providing providers to build infrastructure in any cloud platform like AWS, Azure, GCP

**Terraform state file**:

Terraform have feature to maintain the state file with help of state file we will reuse the resources which we have build already.

It contain details about infrastructure which we have build, we can store state file for backup and we can use the same for infracture creation

Terraform having 2 files:

Main.tf -- resources

Variable.tf – aws accesskeys

**Terraform init** – It will download the required plugins for your resource

**Terraform plan** – It will be check what are resources your going to be create (or) destroy. While running terraform plan, it will create automatically terraform state file.

**Terraform apply** – It will create the resources.

**Terraform destroy** – It will be deleting the resources from your account.

Stages of terraform

<https://developer.hashicorp.com/terraform/cloud-docs/run/states>

**The Pending Stage   
The Fetching Stage The Pre-Plan Stage The Plan Stage The Post-Plan Stage The OPA Policy Check Stage**

terraform resources

Resources are the most important element in the Terraform language. Each resource block describes one or more infrastructure objects, such as virtual networks, compute instances, or higher-level components such as DNS records.

<https://developer.hashicorp.com/terraform/language/resources/syntax>

### Providers, Resource Arguments

Terraform Modules:

# Modules

Modules are self-contained packages of Terraform configurations that are managed as a group.

A module is a container for multiple resources that are used together. You can use modules to create lightweight abstractions, so that you can describe your infrastructure in terms of its architecture, rather than directly in terms of physical objects.

The .tf files in your working directory when you run [terraform plan](https://developer.hashicorp.com/terraform/cli/commands/plan) or [terraform apply](https://developer.hashicorp.com/terraform/cli/commands/apply) together form the root module. That module may [call other modules](https://developer.hashicorp.com/terraform/language/modules/syntax#calling-a-child-module) and connect them together by passing output values from one to input values of another.

To learn how to use modules, see [the Modules configuration section](https://developer.hashicorp.com/terraform/language/modules). This section is about creating re-usable modules that other configurations can include using module blocks.

1. Root Module:

The root module is **the directory that holds the Terraform configuration files that are applied to build your desired infrastructure**. These files provide an entry point into any nested modules you might utilize. Any module should include, at minimum, a main.tf , a variables.tf , and an outputs.tf file.

1. Child Module

A Terraform module (usually the root module of a configuration) can call other modules to include their resources into the configuration. **A module that has been called by another module** is often referred to as a child module.

Creating modules:

<https://www.terraform.io/language/modules/develop>

**Terraform data sources:**

Data sources **allow Terraform to use information defined outside of Terraform, defined by another separate Terraform configuration, or modified by functions**.

<https://developer.hashicorp.com/terraform/language/data-sources>

How to do you automate your process using terraform?

How do you manage terraform state file in your project, where exactly state file will store?

State file stores the current state of infrastructure, If you make any changes to code it will compare to your state file what is currently there and what is currently required and it will show you the changes in your plan,

Ex: You have run the plan and you made some changes to infrastructure and don't have state file that will stored in your local If i run the plan, how would i know the changes?

# Remote State

By default, Terraform stores state locally in a file named terraform.tfstate. When working with Terraform in a team, use of a local file makes Terraform usage complicated because each user must make sure they always have the latest state data before running Terraform and make sure that nobody else runs Terraform at the same time.

With remote state, Terraform writes the state data to a remote data store, which can then be shared between all members of a team. Terraform supports storing state in [Terraform Cloud](https://www.hashicorp.com/products/terraform/), [HashiCorp Consul](https://developer.hashicorp.com/consul), Amazon S3, Azure Blob Storage, Google Cloud Storage, Alibaba Cloud OSS, and more.

Remote state is implemented by a [backend](https://developer.hashicorp.com/terraform/language/settings/backends/configuration) or by Terraform Cloud, both of which you can configure in your configuration's root module.

## Locking and Teamwork

For fully-featured remote backends, Terraform can also use [state locking](https://developer.hashicorp.com/terraform/language/state/locking) to prevent concurrent runs of Terraform against the same state.

<https://developer.hashicorp.com/terraform/language/state/remote>

PROD ISSUES:

Production support:

A production support person/team is responsible for monitoring the production servers, scheduled jobs, incident management and receiving incidents and requests from end-users, analyzing these and either responding to the end user with a solution or escalating it to the other IT teams

Production Support Steps:

Recording Production Error

Notification of Production Error

Investigation or Analysis of Production Error

Resolution of Production Error

Production job/program code correction

Production Process correction

Infrastructure Issue correction

Performence of your application is too low,it is not accessing very good, you got a coustmer feedback, what components you will check if performing not well?

<https://vaadin.com/blog/how-to-improve-the-performance-of-your-web-application>

## Using load balancing

## Using network cache

### HTTP caching

### Reverse proxy server caching

## Using database cache

## Optimizing the application

### Storing less data in the session

### Avoiding running out of memory

AWS:

1. What is ECS and how it work?

<https://devops4solutions.com/how-to-deploy-containerized-applications-with-amazon-ecs/>

**Amazon Elastic Container Service** (ECS) is a highly scalable, high performance container management service that supports Docker containers and allows you to easily run applications on a managed cluster of Amazon Elastic Compute Cloud (Amazon EC2) instances.

**Steps :**

1. Store docker file in git hub

2. configure aws cli into your instance

3. create ECR reposiory --> Push your image to ECR(using push commands for deploy

4. create a cluster(AWS fargate & AWS ec2)

5. task definations--> containers- image provide

6. ecs tasks

7. ecs services

Youtube: <https://www.youtube.com/watch?v=vyznPmkVC9k>

AWS Cloud watch, AWS Lambda, AWS VPC, Auto scaling, Cloud Formation

2.AWS S3 and Bucket types?

Amazon S3 offers a range of storage classes that you can choose from based on the data access, resiliency, and cost requirements of your workloads. S3 storage classes are purpose-built to provide the lowest cost storage for different access patterns. S3 storage classes are ideal for virtually any use case, including those with demanding performance needs, data residency requirements, unknown or changing access patterns, or archival storage.

The S3 storage classes include **S3 Intelligent-Tiering** for automatic cost savings for data with unknown or changing access patterns, S3 Standard for frequently accessed data, **S3 Standard-Infrequent Access (S3 Standard-IA)** and **S3 One Zone-Infrequent Access (S3 One Zone-IA)** for less frequently accessed data, **S3 Glacier Instant Retrieval** for archive data that needs immediate access, **S3 Glacier Flexible Retrieval (formerly S3 Glacier)** for rarely accessed long-term data that does not require immediate access, and **Amazon S3 Glacier Deep Archive (S3 Glacier Deep Archive**) for long-term archive and digital preservation with retrieval in hours at the lowest cost storage in the cloud. If you have data residency requirements that can’t be met by an existing AWS Region, you can use the **S3 Outposts** storage class to store your S3 data on premises. Amazon S3 also offers capabilities to manage your data throughout its lifecycle. Once an S3 Lifecycle policy is set, your data will automatically transfer to a different storage class without any changes to your application.

What is AWS VPC Wizard?

Amazon Virtual Private Cloud (Amazon VPC) **enables you to launch AWS resources into a virtual network that you've defined**. This virtual network closely resembles a traditional network that you'd operate in your own data center, with the benefits of using the scalable infrastructure of AWS.

Is IT possible to run legacy applications on a cloud?

**There are multiple benefits to migrating your legacy applications to the cloud**. Migration advocates for the modernization of legacy applications, while greatly reducing infrastructure costs and improving scalability and efficiency.

1. Step 1: Initialize App2Container. ...
2. Step 2: Analyze Using App2Container. ...
3. Step 3: Containerize Using App2Container. ...
4. Step 4: Generate Pipeline in App2Container.

# AWS Lambda Features

* Extend other AWS services with custom logic. ...
* Build custom backend services. ...
* Bring your own code. ...
* Completely automated administration. ...
* Built-in fault tolerance. ...
* Package and deploy functions as container images. ...
* Automatic scaling. ...
* Connect to relational databases.

parameters in lambda:

* A Python lambda function behaves like a normal function in regard to arguments. Therefore, a lambda parameter can be initialized with a default value: the parameter n takes the outer n as a default value. The Python lambda function could have been written as lambda x=n: print(x) and have the same result.

How to provide access to cross acccount using IAM?

<https://aws.amazon.com/premiumsupport/knowledge-center/cross-account-access-iam/>

Diff b/w IAM policies and S3 bucket policies?

S3 bucket policies (as the name would imply) only control access to S3 resources, whereas IAM policies can specify nearly any AWS action. One of the neat things about AWS is that you can actually apply both IAM policies and S3 bucket policies simultaneously, with the ultimate authorization being the least-privilege union of all the permissions (more on this in the section below titled *“How does authorization work with multiple access control mechanisms?”*).

9. How to control the traffic of only one EC2 instance?

To allow or block specific IP addresses for your EC2 instances, use a network Access Control List (ACL) or security group rules in your VPC. Network ACLs and security group rules act as firewalls allowing or blocking IP addresses from accessing your resources. Network ACLs control inbound and outbound traffic at the subnet level. Because network ACLs function at the subnet level, rules apply to all instances in associated subnets. Security group rules act as a firewall for associated Amazon EC2 instances, controlling both inbound and outbound traffic at the instance level.