SRE/DevOps-Troubleshooting

https://awstip.com/sre-devops-interview-questions-linux-troubleshooting-extended-c12cb5ded3b0

Linux Troubleshooting

2. What happens when you type ls on terminal

These type of questions are used to understand interviewee attention to details and depth of Linux Internals . Basically the interviewer wants to know if you under forks() and exec() system calls.

the shell reads what you typed using the getline() *function and function called strtok() which took the line to tokenize. Shell also check if the 1st token ls is a Shell alias or not. If it's not a built-in function, shell will find the PATH variable in the directory. Since it holds the absolute paths for all the executable binary files. Once it finds the binary for ls , the program is loaded in memory and a system call fork() is made. This creates a child process as ls and the shell will be the parent process. The fork() returns 0 to the child process so it knows it has to act as a child and returns PID of the child to the parent process(i.e. the shell).*

Next, the ls process executes the system call execve() that will give it a brand new address space with the program that it has to run. Now, the ls can start running its program. The ls utility uses a function to read the directories and files from the disk by consulting the underlying filesystem's inode entries.

Once ls process is done executing, it will call the _exit() system call with an integer 0 that denotes a normal execution and the kernel will free up its resources.

Note: you can use strace ls to dig deeper into the system calls

3. Explain Linux Inodes

An Inode number points to an Inode. An Inode is a data structure that stores the information about the file or folder

Detailed Answer is available here

4. Crash vs Panic

Crash usually happens when a trap occurs when the application trying to access memory incorrectly. Panic usually when the application kill/shutdown itself abruptly. Main difference between crash and panic is that crash is hardware or OS initiated and panic usually imitated by application by calling abort() function. Some applications use a special function called a signal handler to generate information about the trap other can use gdb to collection information about the same.

Most common bad programming signals are SIGSEGV , SIGBUS and SIGILL usually caused by ***bad memory management, a bad pointer, uninitialized values or memory corruption.***

5. Explain the /proc filesystem

/proc is very special in that it is also a virtual filesystem. It's sometimes referred to as a process information pseudo-file system. It doesn't contain 'real' files but

runtime system information. Lot of system utilities are simply calls to files in this directory
/proc file system has the pid for the process running. if you do cd /procs/ self you will see al lot file and there size is 0 however you will see that they do contain information
/maps provides information about the memory address space of the process
/cmdline contains the arguments for the ***commandline***
/environ provides information about the process' current environment
/fd contains symbolic link pointing to each file for which the process currently has file descriptor
/proc/locks shows all the locks on currently exist in the system
/proc/sys/fs contains some useful information like file-nr which tells you the number of open files and available on the system
/proc/sys/vm holds files and information to tune virtual memory

6. When I get a "filesystem is full" error, but "df" shows there is free space

Check if you see zero IFree by using df -i . If that is not the case then see if deleted files are still in use using lsof and restart those processes

7. What are the performance tools you would use on Linux Machine

uptime
dmesg | tail
vmstat 1
mpstat -P ALL 1
pidstat 1
iostat -xz 1
free -m
sar -n DEV 1
sar -n TCP,ETCP 1
top
Detailed Answer is available here

8. Explain Linux FileSystem

Interviewer wants to know how much you understand about linux filesystems. A specific type of data storage format, such as EXT3, EXT4, BTRFS, XFS, and so on. Linux supports almost 100 types of filesystems.
Detailed Answer is available here

9. Explain Kernel Space and User Space

This can be a rabbit hole question, Interviewer can go as deep as possible to see what are your limits. This is also the most interesting topic about Linux that how the control flows from User Space to Kernel Space and why that is important. Why can't we directly access the Kernel Space. What are use internal libraries like libc and why we need system call
Detailed Answer is available here

10. How would you troubleshoot a High I/O Issue

Detail Answer is available here

11. What are processes and threads?

Process are basically the programs which are dispatched from the ready state and are scheduled in the CPU for execution. PCB(Process Control Block) holds the concept of process. A process can create other processes which are known

as Child Processes. The process takes more time to terminate and it is isolated means it does not share the memory with any other process.
Detailed Answer is available here

12. Explain Kernel Memory Management

This is not a trivial question. It is very deep and convoluted. So I would hope that interviewer will only be trying to see if you understand the basics around the Kernel Memory Management
Detailed Answer is available here

13. Explain Processes and threads ?

Detailed Answer is available here

14. Explain different type of task status ?

Detailed Answer is available here

15. Explain Linux Concurrency and Race Conditions ?

Detailed Answer is available here

16. Explain STACK and HEAP in Operating System ?

Detailed Answer is available here

17. Explain Memory Leak ?

Naive definition: *Failure to release* unreachable *memory, which can no longer be allocated again by any process during execution of the allocating process. This can mostly be cured by using GC (Garbage Collection) techniques or detected by automated tools.*
Subtle definition: *Failure to release* reachable *memory which is no longer needed for your program to function correctly. This is nearly impossible to detect with automated tools or by programmers who are not familiar with the code. While technically it is not a leak, it has the same implications as the naive one. This is not my own idea only. You can come across projects that are written in a garbage collected language but still mention fixing memory leaks in their changelogs.*

18. How does Linux handles Interrupts ?

Detailed Answer is available here

19. Explain Load Average ?

The best definition and internals about load average can be is explained here. I would encourage everybody to go though this website for more deeper understanding about internals

20. What happens when you try to curl to website ?

This is very famous question and comes to life every now and then. However I would think that we all should be aware of the internal process flow when you do curl www.google.com . Once the best ***detailed*** *explanation I found is here. One can certainly argue that this way too much detail but hey no harm in knowing things completely, you may not say this whole thing when asked but one should certainly know about it*

Question: There is a service running on a system and you have been told that the service is not running properly. How would you troubleshoot?
Answer: After verifying the DNS and other things, I would try to SSH to the system and try to see what is going on
Follow Question: You get an error message like the below:

ssh: connect to host example.com port 22: Resource temporarily unavailable
however you did find out that you have **IPMI** access to the machine, so you can use that to login and you get something like this:
root@example.com#
Follow up Answer: So now I will try to run some basic commands to see what is going on like top , ps etc
Follow Question: You still get a similar error for any command you run:
fork: retry: Resource temporarily unavailable
***Note**: Now you might already know what is the issue, at least have some idea about it. It is related to **Resource being exhausted** may be files, processes etc etc and you figured that you can't run any command (specifically any external command, which needs forking).*
Actual Question: How would you troubleshoot a linux box, given none of the external commands executes?
Facts: What should we do ? Use commands which are **internal** or **built-in** to the shell .
So how to find what commands are built-in and can be useful to you for troubleshooting ? Also if external commands are not available then how to gather information about the system (top, ps, lsof....even cat is not available)..... 🤔
→ /proc filesystem and couple of **built-in** like read and for
*Type help in the shell and it will give you all the commands which are **in-build** to the shell*
Read about /proc filesystem in my previous blog SRE/DevOps Interview Questions — Linux Troubleshooting and also here
Apparently, all the information coming from commands like ps , lsof , vmstat etc etc can be found in /proc file system if you look at the right file.
For example cmdline file tells you about the last command ran on the system, fd directory contains all the file descriptors and all the folder with numbers are the PIDs running on the system.
So assuming system has too many files open and all resources are exhausted, how to see if which pid is using how many file without using external commands
for fd in /proc/[0-9]*/fd/*; do echo $fd ; done
/proc/1/fd/0
/proc/1/fd/1
/proc/1/fd/2
/proc/1/fd/255
/proc/1/fd/3
This way you can see the file descriptors any pid is using. If you want to see what is last command ran on the system without cat or see the content of the file with cat
read $(</proc/${PID}/cmdline)
So using read , for and other internal commands you can use to find out what is going on in the system.
Trick Question: How do you delete a file named -f or --file ?
Answer: Well explained here. Though the interviewer expects you to walk him/

her through the **command execution process** on the linux system
Question: Explain the output below and what command produces this?

❓

Question: Name the fields and what they represents ?
Question: Looking at the below output, please explain what is going on in the machine ? Please be as detail and descriptive as possible

❓

❓

Question: Compare above 2 vmstat outputs ? and explain the similarities and differences ?
Question: What do you understand from below vmstat output and what is the relationship between in and cs column ?

❓

Trick Question: Looking at vmstat can you tell how many CPU and Cores the system has ?
Question: How to find the config and other related files if only the process is known ?
Answer: check lsof
Question: Given there is a TCP Connection between 2 machine? How does packets move if they are IP Datagram packets? If Yes, what would be an advantage? Faster? What about latency?
Explanation: Good content to read about TCP and UDP
Question: When you do curl example.com what happens?
Explanation: I don't have to say what happens when you do curl example.com , What I am going to suggest is that this is basically a combination of multiple questions

- How does curl executes on the system? Basically, go over the whole process of command execution from **user** space to **kernel** space.
- Explain the fork() and exec() calls.
- Explain the well known system calls which might be involved
- DNS Resolution from example.com to an **IP Address**
- Explain the Request transmission from your terminal to the destination machine and then respond back to your terminal
- If possible go into detail about the packet transmission **Network layer 2–3** concepts

Follow Up Question: The output of curl example.com is this? Please explain?

```
>> curl example.com
<HTML>
<HEAD>
<TITLE>Document Has Moved</TITLE>
</HEAD>
<BODY BGCOLOR="white" FGCOLOR="black">
```

<H1>Document Has Moved</H1>
<HR>
<FONT FACE="Helvetica,Arial"><B>
Description: The document you requested has moved to a new location. The new location is "https://www.example.com/".
</B></FONT>
<HR>
</BODY>
Explanation: Talk about the **Response code** you have received (not visible in output) and why. What can you do to get 200 response code ?
Explain how **HTTPS** works and go into detail in a **3-Way handshake**. Explain the **Asymmetric** and **Symmetric encryption** in terms of **SSL.** Here is a nice writeup about it
Follow Up Question: How do you tell your system to not used /etc/hosts Does file override for DNS Resolution?
Answer: nsswitch.com read about it here
Question: What is nscd service and why it is used ?
Answer: Read about nscd here
Follow Up Question: How does the Linux system know that your application or any application will use which protocol (TCP or UDP) to communicate?
Answer: /etc/services nice write-up here on the same.
Question: How to simulate 50% packet drop for Testing purposes?
Explanation: Here
Question: Talk about gRPC and why would one use it ? Pros and cons ?
Explanation: Good Read
Question: What are the ways to count total **TCP Connections** on the system ?
Explanations: couple of commands can be used like netstat , ss and not to forget the /proc filesystem (specifically /proc/net/sockstat file). Good examples here
Question: How to troubleshoot **Short Lived Process** or **Processes** causing CPU Spikes ?
Explanation: Good Readings here and here
Hope this helps in your journey and provides more food for the thoughts

==========

How can containers within a pod communicate with each other?

Answer

Containers within a pod share networking space and can reach other on localhost. For instance, if you have two containers within a pod, a MySQL container running on port 3306, and a PHP container running on port 80, the PHP container could access the MySQL one through localhost:3306.

Explain what is a Master Node and what component does it consist of?

Answer

- The **master node** is the most vital component responsible for Kubernetes architecture
- It is the central controlling unit of Kubernetes and manages workload and communications across the clusters

The master node has various **components**, each having its process. They are:

- ETCD
- Controller Manager
- Scheduler
- API Server

ETCD (Cluster store)

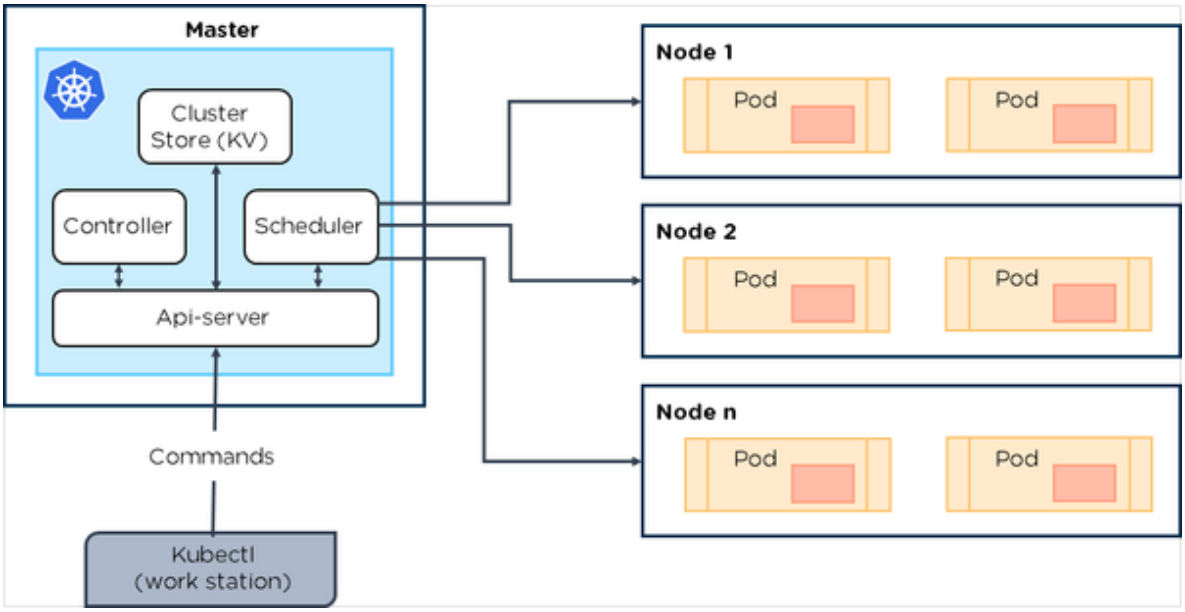
1. This component stores the configuration details and essential values
2. It communicates with all other components to receive the commands and work in order to perform an action
3. It also manages network rules and posts forwarding activity

Controller Manager

1. It is responsible for most of the controllers and performs a task
2. It is a daemon which runs in a continuous loop and is responsible for collecting and sending information to API server
3. The key controllers handle nodes, endpoints, etc.

Scheduler

1. It is one of the key components of the master node associated with the distribution of workload
2. The scheduler is responsible for workload utilization and allocating pod to a new node
3. The scheduler should have an idea of the total resources available as well as resources allocated to existing workloads on each node



What are *namespaces*? What is the problem with using one default namespace?

Answer

Namespaces allow you split your cluster into virtual clusters where you can group your applications in a way that makes sense and is completely separated from the other groups (so you can for example create an app with the same name in two different namespaces).

- When using the default namespace alone, it becomes hard over time to get an overview of all the applications you manage in your cluster. Namespaces make it easier to organize the applications into groups that makes sense, like a namespace of all the monitoring applications and a namespace for all the security applications, etc.
- Namespaces can also be useful for managing Blue/Green environments where each namespace can include a different version of an app and also share resources that are in other namespaces (namespaces like logging, monitoring, etc.).
- Another use case for namespaces is one cluster, multiple teams. When multiple teams use the same cluster, they might end up stepping on each others toes. For example if they end up creating an app with the same name it means one of the teams overriden the app of the other team because there can't be too apps in Kubernetes with the same name (in the same namespace).

What happens when a master fails? What happens when a worker fails?

Answer

Kubernetes is designed to be resilient to any individual node failure, master or worker. When a master fails the nodes of the cluster will keep operating, but there can be no changes including pod creation or service member changes until the master is available. When a worker fails, the master stops receiving messages from the worker. If the master does not receive status updates from the worker the node will be marked as NotReady. If a node is NotReady for 5 minutes, the master reschedules all pods that were running on the dead node to other available nodes.

What is a StatefulSet in Kubernetes?

Answer

When using Kubernetes, most of the time you don't care how your pods are scheduled, but sometimes you care that pods are deployed in order, that they have a persistent storage volume, or that they have a unique, stable network identifier across restarts and reschedules. In those cases, **StatefulSets** can help you accomplish your objective. It manages the deployment and scaling of a set of Pods, and provides guarantees about the **ordering** and **uniqueness** of these Pods.
StatefulSets are valuable for applications that require one or more of the following.

- Stable, unique network identifiers.
- Stable, persistent storage.
- Ordered, graceful deployment and scaling.
- Ordered, automated rolling updates.

What is a *DaemonSet*?

Answer

DaemonSets are used in Kubernetes when you need to run one or more pods on all (or a subset of) the nodes in a cluster. The typical use case for a DaemonSet is logging and monitoring for the hosts. For example, a node needs a service (daemon) that collects health or log data and pushes them to a central system or database.
As the name suggests you can use daemon sets for running daemons (and other tools) that need to run on all nodes of a cluster. These can be things like cluster storage daemons (e.g. Quobyte, glusterd, ceph, etc.), log collectors (e.g. fluentd or logstash), or monitoring daemons (e.g. Prometheus Node Exporter, collectd, New Relic agent, etc.)

Scenario-Based Interview Questions

This section of questions will consist of various scenario-based questions that you may face in your interviews.

Scenario 1: Suppose a company built on monolithic architecture handles numerous products. Now, as the company expands in today's scaling industry, their monolithic architecture started causing problems.

How do you think the company shifted from monolithic to microservices and deploy their services containers?

Solution:

As the company's goal is to shift from their monolithic application to microservices, they can end up building piece by piece, in parallel and just switch configurations in the background. Then they can put each of these built-in microservices on the Kubernetes platform. So, they can start by migrating their services once or twice and monitor them to make sure everything is running stable. Once they feel everything is going good, then they can migrate the rest of the application into their Kubernetes cluster.

Scenario 2: Consider a multinational company with a very much distributed system, with a large number of data centers, virtual machines, and many employees working on various tasks.

How do you think can such a company manage all the tasks in a consistent way with Kubernetes?

Solution:

As all of us know that I.T. departments launch thousands of containers, with tasks running across a numerous number of nodes across the world in a distributed system.

In such a situation the company can use something that offers them agility, scale-out capability, and DevOps practice to the cloud-based applications.

So, the company can, therefore, use Kubernetes to customize their scheduling architecture and support multiple container formats. This makes it possible for the affinity between container tasks that gives greater efficiency with an extensive support for various container networking solutions and container storage.

Scenario 3: Consider a situation, where a company wants to increase its efficiency and the speed of its technical operations by maintaining minimal costs.

DevOps Training

Kubernetes Certification Training Course: Administrator (CKA)*Reviews*
5(9297)

DOCKER CERTIFICATION TRAINING COURSE
Docker Certification Training Course*Reviews*
5(6556)

GIT CERTIFICATION TRAINING
Git Certification Training*Reviews*
5(4340)

ANSIBLE CERTIFICATION TRAINING COURSE
Ansible Certification Training Course*Reviews*
5(334)

JENKINS CERTIFICATION TRAINING COURSE
Jenkins Certification Training Course*Reviews*
5(8622)

Next

How do you think the company will try to achieve this?

Solution:

The company can implement the DevOps methodology, by building a CI/CD pipeline, but one problem that may occur here is the configurations may take time to go up and running. So, after implementing the CI/CD pipeline the company's next step should be to work in the cloud environment. Once they start working on the cloud environment, they can schedule containers on a cluster and can orchestrate with the help of Kubernetes. This kind of approach will help the company reduce their deployment time, and also get faster across various environments.

Scenario 4: Suppose a company wants to revise it's deployment methods and wants to build a platform which is much more scalable and responsive.

How do you think this company can achieve this to satisfy their customers?

Solution:

In order to give millions of clients the digital experience they would expect, the company needs a platform that is scalable, and responsive, so that they could quickly get data to the client website. Now, to do this the company should move from their private data centers (if they are using any) to any cloud environment such as AWS. Not only this, but they should also implement the microservice architecture so that they can start using Docker containers. Once they have the base framework ready, then they can start using the best orchestration platform available i.e. Kubernetes. This would enable the teams to be autonomous in building applications and delivering them very quickly.

Scenario 5: Consider a multinational company with a very much distributed system, looking forward to solving the monolithic code base problem.

How do you think the company can solve their problem?

Solution

Well, to solve the problem, they can shift their monolithic code base to a

microservice design and then each and every microservices can be considered as a container. So, all these containers can be deployed and orchestrated with the help of Kubernetes.

Kubernetes Interview Questions

Scenario 6: All of us know that the shift from monolithic to microservices solves the problem from the development side, but increases the problem at the deployment side.

How can the company solve the problem on the deployment side?

Solution

The team can experiment with container orchestration platforms, such as Kubernetes and run it in data centers. So, with this, the company can generate a templated application, deploy it within five minutes, and have actual instances containerized in the staging environment at that point. This kind of Kubernetes project will have dozens of microservices running in parallel to improve the production rate as even if a node goes down, then it can be rescheduled immediately without performance impact.

Scenario 7: Suppose a company wants to optimize the distribution of its workloads, by adopting new technologies.

How can the company achieve this distribution of resources efficiently?

Solution

The solution to this problem is none other than Kubernetes. Kubernetes makes sure that the resources are optimized efficiently, and only those resources are used which are needed by that particular application. So, with the usage of the best container orchestration tool, the company can achieve the distribution of resources efficiently.

Scenario 8: Consider a carpooling company wants to increase their number of servers by simultaneously scaling their platform.

How do you think will the company deal with the servers and their installation?

Solution

The company can adopt the concept of containerization. Once they deploy all their application into containers, they can use Kubernetes for orchestration and use container monitoring tools like Prometheus to monitor the actions in containers. So, with such usage of containers, giving them better capacity planning in the data center because they will now have fewer constraints due to this abstraction between the services and the hardware they run on.

Scenario 9: Consider a scenario where a company wants to provide all the required hand-outs to its customers having various environments.

How do you think they can achieve this critical target in a dynamic manner?

Solution

The company can use Docker environments, to put together a cross-sectional team to build a web application using Kubernetes. This kind of framework will help the company achieve the goal of getting the required things into production within the shortest time frame. So, with such a machine running, the company can give the hands-outs to all the customers having various environments.

Scenario 10: Suppose a company wants to run various workloads on different

cloud infrastructure from bare metal to a public cloud.
How will the company achieve this in the presence of different interfaces?
Solution
The company can decompose its infrastructure into microservices and then adopt Kubernetes. This will let the company run various workloads on different cloud infrastructures.

What is the difference between config map and secret? (Differentiate the answers as with examples)

Config maps ideally stores application configuration in a plain text format whereas Secrets store sensitive data like password in an encrypted format. Both config maps and secrets can be used as volume and mounted inside a pod through a pod definition file.

Config map:

```
kubectl create configmap myconfigmap
 --from-literal=env=dev
```

Secret:

```
echo -n 'admin' > ./username.txt
echo -n 'abcd1234' ./password.txt
kubectl create secret generic mysecret --from-file=./username.txt --from-file=./password.txt
```

If a node is tainted, is there a way to still schedule the pods to that node?

When a node is tainted, the pods don't get scheduled by default, however, if we have to still schedule a pod to a tainted node we can start applying tolerations to the pod spec.

Apply a taint to a node:

```
kubectl taint nodes node1 key=value:NoSchedule
```

Apply toleration to a pod:

```
spec:
tolerations:
- key: "key"
operator: "Equal"
value: "value"
effect: "NoSchedule"
```

Can we use many claims out of a persistent volume? Explain?

The mapping between persistentVolume and persistentVolumeClaim is always one to one. Even When you delete the claim, PersistentVolume still remains as we set persistentVolumeReclaimPolicy is set to Retain and It will not be reused

by any other claims. Below is the spec to create the Persistent Volume.

```
apiVersion: v1
kind: PersistentVolume
metadata:
name: mypv
spec:
capacity:
storage: 5Gi
volumeMode: Filesystem
accessModes:
- ReadWriteOnce
persistentVolumeReclaimPolicy: Retain
```

What kind of object do you create, when your dashboard like application, queries the Kubernetes API to get some data?

You should be creating serviceAccount. A service account creates a token and tokens are stored inside a secret object. By default Kubernetes automatically mounts the default service account. However, we can disable this property by setting automountServiceAccountToken: false in our spec. Also, note each namespace will have a service account

```
apiVersion: v1
kind: ServiceAccount
metadata:
name: my-sa
automountServiceAccountToken: false
```

What is the difference between a Pod and a Job? Differentiate the answers as with examples)

A Pod always ensure that a container is running whereas the Job ensures that the pods run to its completion. Job is to do a finite task.

Examples:

```
kubectl run mypod1 --image=nginx --restart=Never
kubectl run mypod2 --image=nginx --restart=onFailure
○ → kubectl get pods
NAME          READY STATUS   RESTARTS AGE
mypod1        1/1 Running   0 59s
○ → kubectl get job
NAME     DESIRED SUCCESSFUL   AGE
mypod1   1 0           19s
```

How do you deploy a feature with zero downtime in Kubernetes?

By default Deployment in Kubernetes using RollingUpdate as a strategy. Let's say we have an example that creates a deployment in Kubernetes

kubectl run nginx --image=nginx **# creates a deployment**
○ → kubectl get deploy
NAME DESIRED CURRENT UP-TO-DATE AVAILABLE AGE
nginx 1 1 1 0 7s

Now let's assume we are going to update the nginx image

kubectl set image deployment nginx nginx=nginx:1.15 **# updates the image**

Now when we check the replica sets

kubectl get replicasets **# get replica sets**
NAME DESIRED CURRENT READY AGE
nginx-65899c769f 0 0 0 7m
nginx-6c9655f5bb 1 1 1 13s

From the above, we can notice that one more replica set was added and then the other replica set was brought down

kubectl rollout status deployment nginx

 # check the status of a deployment rollout

kubectl rollout history deployment nginx

 # check the revisions in a deployment

○ → kubectl rollout history deployment nginx
deployment.extensions/nginx
REVISION CHANGE-CAUSE
1 <none>
2 <none>

How to monitor that a Pod is always running?

We can introduce probes. A liveness probe with a Pod is ideal in this scenario. A liveness probe always checks if an application in a pod is running, if this check fails the container gets restarted. This is ideal in many scenarios where the container is running but somehow the application inside a container crashes.

```
spec:
containers:
- name: liveness
image: k8s.gcr.io/liveness
args:
- /server
livenessProbe:
    httpGet:
      path: /healthz
```

Having a Pod with two containers, can I ping each other? like using the container name?

Containers on same pod act as if they are on the same machine. You can ping them using localhost:port itself. Every container in a pod shares the same IP. You can `ping localhost` inside a pod. Two containers in the same pod share an IP and a network namespace and They are both localhost to each other. Discovery works like this: Component A's pods -> Service Of Component B -> Component B's pods and Services have domain names servicename.namespace.svc.cluster.local, the dns search path of pods by default includes that stuff, so a pod in namespace Foo can find a Service bar in same namespace Foo by connecting to `bar`

Does the rolling update with state full set replicas =1 makes sense?

No, because there is only 1 replica, any changes to state full set would result in an outage. So rolling update of a StatefulSet would need to tear down one (or more) old pods before replacing them. In case 2 replicas, a rolling update will create the second pod, which it will not be succeeded, the PD is locked by first (old) running pod, the rolling update is not deleting the first pod in time to release the lock on the PDisk in time for the second pod to use it. If there's only one that rolling update goes 1 -> 0 -> 1.f the app can run with multiple identical instances concurrently, use a Deployment and roll 1 -> 2 -> 1 instead.

Different Ways to provide API-Security on Kubernetes?

Use the correct auth mode with API server authorization-mode=Node,RBAC Ensure all traffic is protected by TLS Use API authentication (smaller cluster may use certificates but larger multi-tenants may want an AD or some OIDC authentication).
Make kubeless protect its API via authorization-mode=Webhook. Make sure the kube-dashboard uses a restrictive RBAC role policy Monitor RBAC failures Remove default ServiceAccount permissions Filter egress to Cloud API metadata APIs Filter out all traffic coming into kube-system namespace except DNS.
A default deny policy on all inbound on all namespaces is good practice. You explicitly allow per deployment.Use a podsecurity policy to have container restrictions and protect the Node Keep kube at the latest version.

what does kube-proxy do?

kube-proxy does 2 things

- for every Service, open a random port on the node and proxy that port

to the Service.
- install and maintain iptables rules which capture accesses to a virtual ip:port and redirect those to the port in (1)

The kube-proxy is a component that manages host sub-netting and makes services available to other components.Kubeproxy handles network communication and shutting down master does not stop a node from serving the traffic and kubeproxy works, in the same way, using a service. The iptables will route the connection to kubeproxy, which will then proxy to one of the pods in the service.kube-proxy translate the destination address to whatever is in the endpoints.

What runs inside the kubernetes worker nodes?

Container Runtime

- Kubelet
- kube-proxy

Kubernetes Worker node is a machine where workloads get deployed. The workloads are in the form of containerised applications and because of that, every node in the cluster must run the container run time such as docker in order to run those workloads. You can have multiple masters mapped to multiple worker nodes or a single master having a single worker node. Also, the worker nodes are not gossiping or doing leader election or anything that would lead to odd-quantities. The role of the container run time is to start and managed containers. The kubelet is responsible for running the state of each node and it receives commands and works to do from the master. It also does the health check of the nodes and make sure they are healthy. Kubelet is also responsible for metric collectins of pods as well. The kube-proxy is a component that manages host subnetting and makes services available to other components.

Is there a way to make a pod to automatically come up when the host restarts?

Yes using replication controller but it may reschedule to another host if you have multiple nodes in the cluster
A replication controller is a supervisor for long-running pods. An RC will launch a specified number of pods called replicas and makes sure that they keep running. Replication Controller only supports the simple map-style `label: value` selectors. Also, Replication Controller and ReplicaSet aren't very different. You could think of ReplicaSet as Replication Controller. The only thing that is different today is the selector format. If pods are managed by a replication controller or replication set you can kill the pods and they'll be restarted automatically. The yaml definition is as given below:

- apiVersion: v1
- kind: ReplicationController

- metadata:
- name: test
- spec:
- replicas: 3
- selector:
- app: test
- template:
- metadata:
- name: test
- labels:
- app: test
- spec:
- containers:
- name: test
- image: image/test
- ports:
- containerPort: 80

Is there any other way to update configmap for deployment without pod restarts?

well you need to have some way of triggering the reload. ether do a check every minute or have a reload endpoint for an api or project the configmap as a volume, could use inotify to be aware of the change. Depends on how the configmap is consumed by the container. If env vars, then no. If a volumeMount, then the file is updated in the container ready to be consumed by the service but it needs to reload the file
The container does not restart. if the configmap is mounted as a volume it is updated dynamically. if it is an environment variable it stays as the old value until the container is restarted.volume mount the configmap into the pod, the projected file is updated periodically. NOT realtime. then have the app recognise if the config on disk has changed and reload

Do rolling updates declared with a deployment take effect if I manually delete pods of the replica set with kubectl delete pods or with the dashboard? Will the minimum required a number of pods be maintained?

Yes, the scheduler will make sure (as long as you have the correct resources) that the number of desired pods are met. If you delete a pod, it will recreate it. Also deleting a service won't delete the Replica set. if you remove Service or deployment you want to remove all resources which Service created. Also having a single replica for a deployment is usually not recommended because you cannot scale out and are treating in a specific way
Any app should be `Ingress` -> `Service` -> `Deployment` -> (volume mount or 3rd-party cloud storage)
You can skip ingress and just have `LoadBalancer (service)` -> `Deployment`

(or Pod but they don't auto restart, deployments do)

what is the difference between externalIP and loadBalancerIP ?

loadBalancerIP is not a core Kubernetes concept, you need to have a cloud provider or controller like metallb set up the loadbalancer IP. When MetalLB sees a Service of type=LoadBalancer with a ClusterIP created, MetalLB allocates an IP from its pool and assigns it as that Service's External LoadBalanced IP.the externalIP, on the other hand, is set up by kubelet so that any traffic that is sent to any node with that externalIP as the final destination will get routed.`ExternalIP` assumes you already have control over said IP and that you have correctly arranged for traffic to that IP to eventually land at one or more of your cluster nodes and its is a tool for implementing your own load-balancing. Also you shouldn't use it on cloud platforms like GKE, you want to set `spec.loadBalancerIP` to the IP you preallocated. When you try to create the service using .`loadBalancerIP` instead of `externalIP`, it doesn't create the ephemeral port and the external IP address goes to `<pending>` and never updates.

In Kubernetes - A Pod is running 2 containers, when One container stops - another Container is still running, on this event, I want to terminate this Pod?

You need to add a liveness and readiness probe to query each container, if the probe fails, the entire pod will be restarted .add liveness object that calls any api that returns 200 to you from another container and both liveness and readiness probes run in infinite loops for example, If X depended to Y So add liveness in X that check the health of Y.Both readiness/liveness probes always have to run after the container has been started .kubelet component performs the liveness/readiness checks and set initialDelaySeconds and it can be anything from a few seconds to a few minutes depending on app start time. Below is the configuration spec

- livenessProbe spec:
- livenessProbe:
- httpGet:
- path: /path/test/
- port: 10000
- initialDelaySeconds: 30
- timeoutSeconds: 5
- readinessProbe spec:
- readinessProbe:
- httpGet:
- path: /path/test/
- port: 10000
- initialDelaySeconds: 30

- timeoutSeconds: 5

what is the ingress, is it something that runs as a pod or on a pod?

An ingress is an object that holds a set of rules for an ingress controller, which is essentially a reverse proxy and is used to (in the case of nginx-ingress for example) render a configuration file. It allows access to your Kubernetes services from outside the Kubernetes cluster. It holds a set of rules. An Ingress Controller is a controller. Typically deployed as a Kubernetes Deployment. That deployment runs a reverse proxy, the ingress part, and a reconciler, the controller part. the reconciler configures the reverse proxy according to the rules in the ingress object. Ingress controllers watch the k8s api and update their config on changes. The rules help to pass to a controller that is listening for them. You can deploy a bunch of ingress rules, but nothing will happen unless you have a controller that can process them.
LoadBalancer service -> Ingress controller pods -> App service (via ingress) -> App pods

What happens if daemonset can be set to listen on a specific interface since the Anycast IP will be assigned to a network interface alias

Yes, hostnetwork for the daemonset gets you to the host, so an interface with an Anycast IP should work. You'll have to proxy the data through the daemonset.Daemonset allows you to run the pod on the host network, so anycast is possible.Daemonset allows us to run the pod on the host network At the risk of being pedantic, any pod can be specified to run on the host network. The only thing special about DaemonSet is you get one pod per host. Most of the issues with respect to IP space is solved by daemonsets. As kube-proxy is run as daemonset, the node has to be Ready for the kube-proxy daemonset to be up.

How to forward port `8080 (container) -> 8080 (service) -> 8080 (ingress) -> 80 (browser)` how is it done?

The ingress is exposing port 80 externally for the browser to access, and connecting to a service that listens on 8080. The ingress will listen on port 80 by default. An "ingress controller" is a pod that receives external traffic and handles the ingress and is configured by an ingress resource For this you need to configure ingress selector and if no 'ingress controller selector' is specified then no ingress controller will control the ingress.
simple ingress Config will look like

- host: abc.org
- http:
- paths:

- backend:
- serviceName: abc-service
- servicePort: 8080
- Then the service will look like
- kind: Service
- apiVersion: v1
- metadata:
- name: abc-service
- spec:
- ports:
- protocol: TCP
- port: 8080 # this is the port the service listens on
- targetPort: 8080

Are deployments with more than one replica automatically doing rolling updates when a new deployment config is applied?

The Deployment updates Pods in a rolling update fashion when .spec.strategy.type==RollingUpdate .You can specify maxUnavailable and maxSurge to control the rolling update process. Rolling update is the default deployment strategy.kubectl rolling-update updates Pods and ReplicationControllers in a similar fashion. But, Deployments are recommended, since they are declarative, and have additional features, such as rolling back to any previous revision even after the rolling update is done.So for rolling updates to work as one may expect, a readiness probe is essential. Redeploying deployments is easy but rolling updates will do it nicely for me without any downtime. The way to make a rolling update of a Deployment and kubctl apply on it is as below

- spec:
- minReadySeconds: 180
- replicas: 9
- revisionHistoryLimit: 20
- selector:
- matchLabels:
- deployment: standard
- name: standard-pod
- strategy:
- rollingUpdate:
- maxSurge: 1
- maxUnavailable: 1
- type: RollingUpdate

If you have multiple containers in a Deployment file, does use the HorizontalPodAutoscaler scale all of the containers?

Yes, it would scale all of them, internally the deployment creates a replica set (which does the scaling), and then a set number of pods are made by that replica set. the pod is what actually holds both of those containers. and if you want to scale them independently they should be separate pods (and therefore replica sets, deployments, etc).so for hpa to work You need to specify min and max replicas and the threshold what percentage of cpu and memory you want your pods to autoscale..without having the manually run kubectl autoscale deployment ,you can use the below yaml file to do the same

- apiVersion: autoscaling/v1
- kind: HorizontalPodAutoscaler
- metadata:
- annotations:
- name: app
- spec:
- maxReplicas: 15
- minReplicas: 10
- scaleTargetRef:
- apiVersion: autoscaling/v1
- kind: Deployment
- name: app targetCPUUtilizationPercentage: 70

Suppose you have to use database with your application but well, if you make a database container-based deployment. how would the data persist?

Deployments are for stateless services, you want to use a StatefulSet or just define 3+ pods without a replication controller at all. If you care about stable pod names and volumes, you should go for StatefulSet.Using statefulsets you can maintain which pod is attached to which disk.StatefulSets make vanilla k8s capable of keeping Pod state (things like IPs, etc) which makes it easy to run clustered databases. A stateful set is a controller that orchestrates pods for the desired state. StatefulSets formerly known as PetSets will help for the database if hosting your own. Essentially StatefulSet is for dealing with applications that inherently don't care about what node they run on, but need unique storage/ state.

If a pod exceeds its memory "limit" what signal is sent to the process?

SIGKILL as immediately terminates the container and spawns a new one with OOM error. The OS, if using a cgroup based containerisation (docker, rkt, etc), will do the OOM killing. Kubernetes simply sets the cgroup limits but is not ultimately responsible for killing the processes.`SIGTERM` is sent to PID 1 and k8s waits for (default of 30 seconds) `terminationGracePeriodSeconds` before sending the `SIGKILL` or you can change that time with

terminationGracePeriodSeconds in the pod. As long as your container will eventually exit, it should be fine to have a long grace period. If you want a graceful restart it would have to do it inside the pod. If you don't want it killed, then you shouldn't set a memory `limit` on the pod and there's not a way to disable it for the whole node. Also, when the liveness probe fails, the container will SIGTERM and SIGKILL after some grace period.

What is a multinode cluster and single-node cluster in Kubernetes?
Ans: When the controller program and the container node are running on the same machine or on the same operating system then it is called a single-node cluster MultiNodeCluster-when the controller program of k8s in running on the one machine (masternode) and the container is running on the different container host machine (workernode).

How and where is Reverse Proxy used in Kubernetes?
Ans: 'service' kind of resource will actually make use of loadbalancing and reverse proxy in k8s. This program will actually acts like the client for backend PODs and forwards the requests from actual end users to backend and then gives the response to end users as soon as it gets the response from backend servers.

28. What are the different types of services in Kubernetes?
Ans:
There are particularly 3 main services in k8s. They are: 1) Cluster IP 2) Node IP 3) External (Literally, we call it as load balancer) Cluster IP is the default load balancer in k8s. The drawback is that any node within the cluster can connect to LB but from outside no one can connect. In Node port type, LB has access to outside world or even internet. If you had created PODs in your k8s, and want to have the load balancer with them then we need to connect to ELB of 3rd party like AWS.

How does the NodePort service work?
Ans: Node port works on the logic of doing the reverse proxy two times. Assume that k8s is running on minikube and try to randomly assume any empty port as 30k.Now in the yaml file you need to mention node port as 30k. What exactly happens is if anyone tries to connect the minikube ip with 30k as port then the node port program will does reverse proxy to the actual loadbalancer 'service' Again load balancer will internally do reverse proxy from the requests (That is coming from node port program) to the pods. In this way, because we are doing two times reverse proxy hence we are able to make our PODs or LB's to have internet access.

31. What is the difference between port, target port and Nodeport?
A’ns: target port' is the port number at which our application inside POD is running. 'port' is the port number(on load balancer 'service') at which our load balancer receives the request from clients and then forwards them to the

backend. 'node port' is the port on the container host level that helps for giving access to the LB 'service' to internet world. Because of this port node identifies it as a request to actual load balancer

What are PVC, PV and SC in kubernetes?
ANS: User can assign PVC as per their requirement however in order to access storage class, PV act as API for that. PV --> A PersistentVolume (PV) is a piece of storage in the cluster that has been provisioned by an administrator. It is a resource in the cluster just like a node is a cluster resource. PVs are volume plugins like Volumes but have a lifecycle independent of any individual pod that uses the PV. This API object captures the details of the implementation of the storage, be that NFS, iSCSI, or a cloud-provider-specific storage system. PVC --> A PersistentVolumeClaim (PVC) is a request for storage by a user. It is similar to a pod. Pods consume node resources and PVCs consume PV resources. Pods can request specific levels of resources (CPU and Memory). Claims can request specific size and access modes (e.g., can be mounted once read/write or many times read-only). SC --> A StorageClass provides a way for administrators to describe the "classes" of storage th