start with questions about yourself they'll have a resume in front of them as obviously you would have sent your resume to them so they'll look at that and they will ask you to quickly walk them through your resume and tell them what you've been doing in your previous project so you'll have to Summarize this.

in a way in which you know kind of walk through all three layers and you have to tell them that you work on a team where you kind of managed aw the AWS part and the automation part of it so that can also include telling them about any infrastructure deployment scripts which you have written it could also include any configuration management that you've done and deployment automation you might have done so it would also include things like setting up a monitoring

**Generic Questions**

1. DevOps! How can you define it in your words?

Ans - Its highly effective collaboration between software developers and IT operations/web operation engineers to produce a working system or release software. A DevOps implementation is generally aligned with Agile methodologies where deploying working software to Production is generally the highest priority. On Agile implementations, the emphasis is placed on people over processes, so a DevOps engineer must be willing to work very closely with Agile development teams to ensure they have an environment necessary to support functions such as Continuous Integration, and Continuous Delivery. On a traditional implementation practice, without DevOps, the operations team is often isolated from developers, often working under a help desk model under general service level agreements where the system operations team treats developers as a customer. This is a proven model which obviously can work very well, but in a DevOps environment, development and operations are streamlined and barriers between the two groups should not exist.

The definition of DevOps each person and will actually define in its own way the reason is because there is no particular definition for DevOps right so but the actual understanding is if you ask me I would say that DevOps is a practice; which brings developers, Testers, operations and HR or their activities together in a collaborative fashion using proper processes and techniques to achieve the fastest and reliable software releases The ultimate purpose is collaboration; of course DevOps is for collaboration and DevOps is actually using the right tools and techniques but the ultimate purpose of the DevOps is to actually give a faster releases with high quality that's the main purpose and that's basically where the return on investment is there right so if you bravely principles and practices into various segments so here is a small example source code management is one of the area continuous integration is another area continuous testing is one of the areas configuration management which also has an infrastructure as code is one area continuous monitoring is another area all these areas together combined and can be called as a DevOps practices

1. Why we need DevOps?

Ans- Companies are now facing the need for delivering more and faster and better applications to meet the ever more pressing demands of conscious users to reduce the " Time to Market ". DevOps often helps deployment to happen very fast.

1. What is the use of DevOps

Ans- DevOps (development and operations) is an enterprise software development phrase used to mean a type of agile relationship between development and IT operations. The goal of DevOps is to change and improve the relationship by advocating better communication and collaboration between these two business units.

1. How do you implement DevOps?

Ans – 7 steps to choosing the right DevOps tools. ...

Step 1: Understand the collaboration and shared tools strategy for the Dev, QA, and infrastructure automation teams. ...

Step 2: Use tools to capture every request. ...

Step 3: Use agile Kanban project management for automation and DevOps requests that can be dealt with in the tooling.

https://techbeacon.com/7-steps-choosing-right-devops-tools

1. What is agile development and Scrum?

Ans-Agile development used as an alternative to Waterfall development practice. In Agile, the development process is more iterative and incremental, there are more testing and feedback at every stage of development as opposed to only the last stage in Waterfall.  
Scrum is used to manage complex software and product development, using iterative and incremental practices. Scrum has three roles i.e. product owner, scrum master, and team.

1. Can we consider DevOps as an agile methodology?

Ans - Of course! DevOps is a movement to reconcile and synchronize development and production start through a set of good practices. Its emergence is motivated by a deep changing demand of business, who want to speed up the changes to stick closer to the requirements of business and the customer.

1. What is DevOps engineer's duty with regards to Agile development?

Ans - DevOps engineer works very closely with Agile development teams to ensure they have an environment necessary to support functions such as automated testing, Continuous Integration, and Continuous Delivery. DevOps engineer must be in constant contact with the developers and make all required parts of environment work seamlessly.

1. What does the term "DevOps" mean to you?
   1. "Hard"/hands-on/SRE vs
   2. "Soft"/Three Ways/Theory of Constraints/philosophy of DevOps
2. Describe your experience with task management
   1. Agile
   2. Kanban
   3. Waterfall
3. What drew you to DevOps?
4. What are the main benefits of DevOps?

Ans – The benefits of DevOps are as below –

* 1. Release Velocity
  2. Development Cycle
  3. Deployment Rollback
  4. Defect detection
  5. Recovery from failure
  6. Collaboration
  7. Performance Oriented

1. What is continuous delivery in DevOps?

Ans - Continuous delivery is a DevOps software development practice where code changes are automatically built, tested, and prepared for a release to production. It expands upon continuous integration by deploying all code changes to a testing environment and/or a production environment after the build stage.

1. What is CI and CD in DevOps?

Ans - DevOps isn't some mystical process or development philosophy fad. It consists of five clearly defined activities that you can't afford to overlook: Continuous Integration, Continuous Delivery, Cloud Infrastructure, Test Automation and Configuration Management.

https://www.techaspect.com/blog/decoding-cicd-devops/

1. What is the role of testing in DevOps?

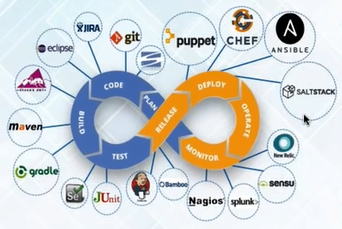
In a pure DevOps world, Quality Assurance is no longer the gatekeeper between development and operations – the bottleneck. QA is an enabler (remember everyone contributes to raising the child). On one side, QA works together with development in trying to push more of their tests into the continuous integration system.

1. What is a continuous monitoring system?

Ans - Continuous monitoring is the process and technology used to detect compliance and risk issues associated with an organization's financial and operational environment. The financial and operational environment consists of people, processes, and systems working together to support efficient and effective operations.

1. Describe the most challenging situation that you were faced with and how did you fix it?
2. How do you stay current?
3. Which Software Stack do you use and why?
4. Describe your worst Fear?
5. What modern technology would you like to learn?
6. Have you worked on any side projects?
7. Do you contribute to any open source projects?
8. Can you describe your workflow when you create a script?

**Popular Tools Used in DevOps**



1. What is JIRA

Ans - JIRA is basically an issue or a defect tracking tool and It will be used for requirement gathering tool as well. Basically, this tool is used to track the development activities; for example, you want to define or you want to develop a project or one of the tasks would be the development of a GUI application. That application development can be divided into multiple small tasks; for example, one of the tasks would be; to design a GUI interface was a login ID and password screen so this can be recorded as a task in JIRA and assigned to a developer and mention the estimated time in the JIRA. So, the developer will start working on the activity like start coding it and once the coding is completed it will change the status of that particular task in the JIRA as close or probably development complete or something so it's basically a tracking of tasks. JIRA is Atlassian and licensed tool. But it can be used for free for a small organization like five to ten developers. But we need to procure the license from Atlassian for organization-wide usage.

1. What is Eclipse?

Ans - Eclipse is an integrated development environment used in computer programming, and is the most widely used Java IDE. It contains a base workspace and an extensible plug-in system for customizing the environment.

1. What are different cycles of Maven Build life cycle?

Ans- The Different Phases in Maven Build Lifecycle are -

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.

Package - Take the compiled code and package it in its distributable format.

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 - Copy the final package to the remote repository for sharing with other

developers and projects.

1. What is Gradle?

Ans-

* + Gradle is a relatively new open source build tool for the Java Virtual Machine.
  + Build scripts for Gradle are written in a Domain Specific Language based on Groovy.
  + The concise nature of Groovy scripting lets you write very expressive build scripts with very little code.

1. What is GIT? What is Bit Bitbucket? What is Envision?
2. What are Maven, Ant, and Gradle?
3. What is Jenkins/TeamCity/GoCI used for?
4. What is Chef, Puppet, Ansible, and Salt lake? Used for
5. What is Docker?
6. What is Nagios/Zenoss/NewRelic used for?
7. What is Slunk?
8. What is a dynamically/statically linked file?
9. What does "./configure && make && make install" do?
10. What is the difference between Containers and VMs?
11. What is a virtual IP address? What is a cluster?
12. How do you print all strings of printable characters present in a file?
13. How do you find shared library dependencies?
14. What are Automake and Autoconf?
15. ./configure shows an error that libfoobar is missing on your system, how could you fix this, what could be wrong?
16. What are the advantages/disadvantages of script vs compiled program?
17. What's the relationship between continuous delivery and DevOps?
18. What are the important aspects of a system of continuous integration and deployment?
19. How would you enable network file sharing within AWS that would allow EC2 instances in multiple availability zones to share data?

**OS questions**

**Linux Questions**

1. How can you view running processes?
2. How do you check server uptime?
3. How do you start/stop services?
4. How do you display the shell’s environment variables?
5. What does #!/bin/bash at the top of a script do?
6. What does "&" after a command do?
7. What does piping commands mean?
8. What distributions have you used on servers?Why?
9. What is CHRONTAB.

the crontab the command is crontab - L

1. What is an alias in Linux

Ans - An alias and Linux is something that tells you the shortcuts on that system these are defined in the et Cie batch RC file

1. What does the chmod command do

Ans- The CHMOD command basically allows you to change the permission of a file in Linux it can be changed from read-write executable mode so it could be changed from reading to read-write or to read write execute depending on the use case

1. What is SSH port forwarding?

Ans- it's a way to actually forward your ports through SSH protocols so it allows you to bypass firewalls and also tunnel ports through strictly guarded environments so this is one of the ways which you can connect to instances or services in your private subnets in AWS or your data center the

1. what is a zombie process (now this is about the most common Linux question asked in an interview zombie process is a process)?

Ans- Which is in a terminated state but which has not yet released the resources so basically it's most commonly a child process where the parent has exited but the child is still there basically its entry is in the process table

1. How check the top CPU consuming processes

Ans- Use the top command to check that the top command gives you a very nice view of the entire Linux process table looking at the top

1. Difference between RAID 0, 1 and 5?
2. What’s the advantage of one RAID over another?
3. Alternative to init.d in Linux?
4. How to view running processes in Linux?
5. How to check DNS records in Linux?
6. Describe your experience with scripting

**Windows Questions**

1. Are you familiar with just Linux or have you worked with Windows environments as well?
2. If yes to windows do you use PowerShell? Octopus Deploy? TeamCity? Active Directory? Azure?

**Version Control questions**

1. Describe a dev/test/production workflow using GIT
   1. Feature branching vs trunk based development
   2. Advantages of requiring pull requests and approvals
2. What types of source control programs are you comfortable with?
3. What are the benefits of using source control?
4. Describe some branching strategies that you have used?
5. Give an example of a good commit message?
6. How will you run a script automatically when a developer commits a change into GIT?

Ans - Wherever a developer making some changes that need to traced back and link with JIRA ticket or want to link with Checks. this can be automated with git events. based on the events we can create some custom scripts that can be executed and this feature is called hooks.

* 1. Client-side hooks.
  2. Server-side hooks.

Git hook - For this case, we can we can write a Client-side post-commit hook. This hook will execute a custom script in which we can add the message and code that we want to run automatically with each commit.

Generally, hooks are stored in hooks subdirectory of Git directory.

Location: .git/hooks

Automate Git script is -

* Commit workflow hooks - committing a file or adding or pushing it.
* Email workflow scripts - Sending emails once push or merge happened
* Other scripts

1. What is a trunk in software?

Ans - In the world of software development, "trunk" means main development branch under a version control system. It’s the base of a project, where all improvements are being merged together.

1. What is code branching and merging?

Ans - Branches are created from the base project, in order to develop a new feature, fix a bug, or simply do some refactoring. They prevent software developers from disturbing each other and enable them to work in parallel. Once a change is finished and tested, the branch is merged back to the trunk.

1. What is a branch in version control?

Ans - A branch in a version control system is a duplicate of the base project. It’s created so that changes can happen in parallel across other branches. It essentially solves the problem of working on the same files at the same time.

1. What is a feature branch?

Ans - The feature branch has a clear and highly-focused purpose—to add specific functionality to a project. It shouldn’t contain any other changes that fix bugs, introduce other features, or are part of a refactoring.

1. What is a version control system?

Ans - Version control systems trace changes occurring in files in a project. They can be recalled or reviewed later. It’s extremely useful for bringing back previous versions as well. This allows developers to find bugs with less effort, as they can see and track all the changes as they occurred.

**CI Questions**

1. What is continuous integration and DevOps?

Ans – Continuous integration is Development practice. Developers are required to check-in the code several times a day. Each check-in is verified by an automated build teams to detect problems early. A continuous integration system usually involves Version control system is being monitored. When a commit is detected, a build will be triggered automatically. Whenever a change to version control system detected, the system would automatically build and test the application. If build or test is not green the system immediately notifies the developers to fix the issue right away.

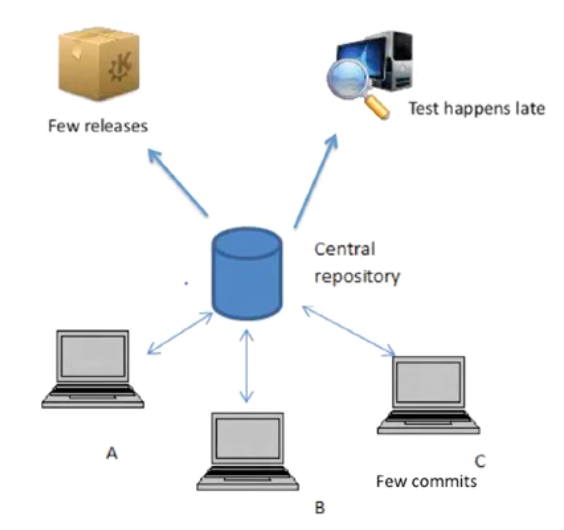
1. Why we need it?

Ans - One of the motivating forces behind CI is to detect problems or bugs as early as possible in the development lifecycle. The entire code base is integrated, built, and tested constantly. The potential box and arrows are called earlier in the life cycle which results in better quality software.

1. Distinct phases of Continuous integration?

Ans – let's talk about distinct stages of adopting continuous integration introducing continuous integration into our organization is not something can be achieved in just one step. Usually it takes you on a path the progress is through several distinct stages.

Let's take a look and approximate picture of each stage.

Stage 1 There are no build servers at all for the team. The application is built manually   
on a developer's machine.

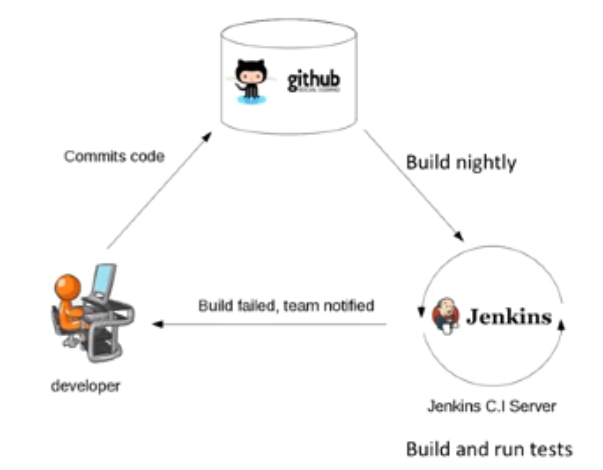
The source code is stored in a central repository but is not enforced.

The developers need to commit their changes on a regular basis.

What happens is that before release a developer or Release Manager would mentally integrate the changes

during this stage, the code changes made by all developers were brought together and forged into a product.

This sometimes involves integrating months of conflict and changes which makes it very hard to anticipate the types of issues that would come up. Also, it takes a lot of effort to resolve those issue because a lot of the code might be written months ago.   
This painful process can cause significant release delays

Stage 2 The team has a build server and automated builds are scheduled on a regular basis typically nightly basis.

This spills script would compile the application and runs a set of automated tests developers now commit the changes regularly usually at the end of every day. If developers commit conflict with another developer’s work. The build server would alert the team members

stage 3 The team is now starting to take continuous integration and automated testing to a further step. The server is configured to start a build whenever new code is committed to the central repository. The boot process usually involves compiling the application and runs a set of the automated unit or integration tests. The boot server alerts team members immediately if a bill fails. Broken bills are usually treated as a high priority issue and are fixed quickly.

Stage 4 automated code quality and code coverage metrics are now run along with unit tests to continuously evaluate the code quality. Gradually the CI process gathers useful information about our code base will be able to answer questions such as is the code coverage in. Do we have fewer and fewer failures?   
This helps teams keep the quality bar of the code base high will be notified if good testing practices are slipping.

Stage 5 confidence has been built up in the automated unit integration and acceptance tests so that automated deployment techniques can be used to push out new changes directly into production.

1. What is continuous integration?

Ans – Continuous Integration is a software development practice where members of a team integrate their work frequently, usually each person integrates at least daily–leading to multiple integrations per day. Each integration is verified by an automated build (including test) to detect integration errors as quickly as possible. Many teams find that this approach leads to significantly reduced integration problems and allows a team to develop cohesive software more rapidly.

Or

Continuous integration is the practice of merging development work with the main branch constantly so that the code is tested as often as possible to catch issues early.

Otherwise the cold quality wouldn't be guaranteed in the release. So how can we implement continuous integration. There are various products and tools which can help you implement continuous integration or organization. Some tools let your host see servers in your own network infrastructure such as Jenkins. There are some other hosted see-I products such as Circle C.I. which is completely hosted in the cloud. Integration is also a mindset when cadenzas integration is introducing into an organization. It dramatically changes the way how organizations think about the whole development process. Continuous integration is not only a toolset but also a mindset to get the most out of continues integration. A team needs to adopt a CIO mentality. Fixing broken builds should be treated as a high priority issue for all team members. The deployment process should be automated with no manual steps involved. All team members should focus on contribute into high quality tests. Each of these requires improvements in the practices and engineering culture of continuous integration across the organization.

1. What is the function of CI (Continuous Integration) server?

Ans- CI server function is to continuously integrate all changes being made and committed to the repository by different developers and check for compile errors. It needs to build code several times a day, preferably after every commit so it can detect which commit made the breakage if the breakage happens.  
Note: Other available and popular CI tools are Jenkins, TeamCity, CircleCI, Hudson, Buildbot etc

1. What is Continuous delivery?

Ans - The continual delivery is that, delivery of code to an environment. Once the code is ready to ship this could be staging or production. But the idea behind is that, the product as delivered to a user base for review and inspection which can be codebase. where a customer's unit tests during continuous integration cannot catch all the bugs in business logic particularly design issues. This is why we need a QA or staging environment for testing cadenzas deployment is the deployment or release of code to production as soon as it's ready continues deployment requires continuous integration and continuous delivery.

1. What is Continuous deployment?

Ans – The deployment or release of code to production as soon as it is ready.

1. What CI tools have you used? Example: Jenkins, Circle-CI.
2. Are you familiar with CI tools? Which ones?
3. Describe your experience implementing continuous deployment
4. How does your setup an end-to-end pipeline from dev to deployment? (long answer)
5. How can Docker help in this case?
6. How frequently have you been deploying?
7. Have you been able to improve the frequency of deployments? If so, how?
8. What are tools and Stages followed with Microsoft Visual Studio Team Foundation Server 2012?

Ans - TFS provides software development teams with the ability to collaborate on their projects. Anyone who is in­volved with creating software will find tools and capabilities that will help them perform their jobs. By anyone, we mean not just programmers, but testers, architects, program managers, business managers, and others who contribute to the development and release of software. This book stresses the following capabilities:

• Version control. TFS provides a place to store and version source code as well as any other artifacts that impact your software project. Examples of these artifacts include scripts, configuration files, and docu­mentation.

• Test case management. Microsoft Test Management (MTM) stores all the testing artifacts it uses, such as test plans, test cases, bugs, and the results of tests runs in TFS.

• Build automation. TFS lets you automate your builds, which means you assemble your application into a product without human intervention. An automated build can include many activities such as compiling source code, packaging binaries, and running tests. In this guidance, we use the TFS build automation system as the basis for the release pipeline’s orchestration, stages and steps.

• Reporting. TFS provides many types of reports and metrics that give your insight into all aspects of your project.

• Environment management. TFS, in conjunction with Lab Management, helps you manage and provision your environments. In this book, we concentrate on using Lab Management’s standard environments as a way of providing consistent environments for everyone involved in the software project.

**Automated Testing Questions**

1. Talk about different types of automated testing you have implemented?
2. What is the importance of software testing?
3. What testing paradigms do you use, if any?
4. Explain the difference between unit and end to end testing?
5. Have you used any testing frameworks? If so, which ones?

**Deployment-related questions**

1. what is a Bluegreen deployment?

Ans - Bluegreen deployment is something that in which you have X number of resources running your application and say that number is 10 so you have 10 servers in a web form now in a Bluegreen deployment what you do is take out half of those from the actual production state you deploy the new code on those now in the meantime the other five or the other half would be serving the production traffic and these 5 would not be hindered or hampered in any way now the first 5 in which you complete the deployment which you've taken out of the actual production load you put those back in and you wait for those to come back into service and then you take the other 5 out of the actual load and you start deploying on those so in a Bluegreen deployment what happens is you never let the end user see the downtime it's an always up environment

1. how do you do a hot deployment (this is just a rephrasing of the previous question)

Ans- Really a hot deployment can be done either by having two environments of the same size and then you redirect traffic through a load balancer or a proxy service to one of the environments deploy it on the other and then redirect to the first one and apply it to the second and just like a Bluegreen deployment you never show the end user the downtime and what is your roll back strategy this is something you need to be very confident about because every deployment should have a roll back associated with it so let's say your deployment fails how do you roll back the system it has to be linked to a blue green or a hot deployment so you have to say that you have a Jenkins job or a script which basically does a blue green deployment and in the middle of that blue green deployment it checks to see whether or not services are up and running and at the least the sgtp endpoints of your application are running or not.

1. Have you used Jenkins for deployment?

Ans – yes I've used it and used it for a couple of strategies; one with plugins and secondly with script so my plugins used to deploy code in the environment using build or publish or SSH and I have also had ansible or chef or puppet code which used to do the deployment for me so we used to use Jenkins as an Orchestrator and puppet and chef and Ansible for the actual deployment of the target service

1. one of the other frequent questions from a deployment perspective is what Jenkins plugins have you used you would

Ans - Jenkins plugins I've used

1. maven and Gradle
2. kebechet for testing PMD for programmatic mistake detection Aztek for Ruby on Rails testing.
3. karma for angularjs testing and integration with s3
4. git plug-in for checking out code
5. SVN plug-in for checking out SVN repositories
6. upstream downstream plugin for connecting the builds
7. in addition to this have also used archive artifacts plug-in publish HTML reports plug-in to publish test reports
8. have you ever used user data for deployment?

Ans- yes I've used it in AWS so when I have an instance behind load balancer I used to use it will define the deployment script so that every time a new instance comes up in the auto scaling group it has got the latest code on it so that latest code is checked out using the SH command which is specified inside the user data.

1. What is Continuous Delivery and why is it important?
2. What is Continuous Deployment?
3. Explain the importance of A/B testing and how it relates to software delivery? o What are your favorite deployment techniques and why? Examples include: 0 downtime, Canary, or Blue/Green deployments.
4. Explain a time when a software release has gone poorly
5. How can you reduce the time to test and Deploy code?

**Cloud questions**

1. Which cloud providers are you familiar with?
2. What are the benefits of using a cloud provider?
3. Have you managed production systems in the past?
4. what is a VPC

Ans - A VPC is a service in the Amazon Cloud

A virtual private cloud (VPC) is a virtual network dedicated to your AWS account. It is logically isolated from other virtual networks in the AWS Cloud. You can launch your AWS resources, such as Amazon EC2 instances, into your VPC.

1. What is the difference between a public and private subnet

Ans - A public subnet is a subnet which is directly accessible from the internet and a private subnet is a subnet that is not accessible from the internet it's only accessible within the VPC.

1. what is reserved instance

Ans- a reserved instance is an instance which is reserved for you by Amazon for a year and they give you significant price reductions on that you can buy that no upfront partial upfront or full upfront payments and you get discounts from 20 to 60 percent based on the payment type and Terms

1. what is the difference between spot instance and reserved instance

Ans- the spot instance is like a bid instance where you have a specific price which you have and based on that you were assigned instances now the moment your bid is lower than the next highest bid your instance is terminated and it's assigned to the next highest bidder a reserved instance on the other hand is not a bit Abul instance you have to buy it for a specific terms

1. what is cloud formation

Ans – AWS CloudFormation is a service that helps you model and set up your Amazon Web Services resources so that you can spend less time managing those resources and more time focusing on your applications that run in AWS.

Cloud formation is an orchestration or an infrastructure or a server deployment tool a managed turbines deployment tool or service which Amazon provides so it takes the JSON as an input in that JSON you provide everything that is required to build up an environment from VPS to servers to buckets everything so it's an Amazon provided service that lets you build entire application stacks from scratch

1. What is DevOps and AWS?

Ans - AWS provides services that help you practice DevOps and that are built within AWS. These tools automate manual tasks, help teams manage complex environments at scale, and keep engineers in control of the high velocity that is enabled by DevOps. Explore our solutions and tooling for DevOps.

Those are –

1. Infrastructure as a code - CloudFormation - We use AWS cloudformation to create and deploy AWS resources by using templates. We can describe our dependencies and pass special parameters in these templates. CloudFormation can read these templates and deploy the application on AWS cloud.
2. OpsWorks - AWS OpsWorks is a configuration management service that helps you build and operate highly dynamic applications, and propagate changes instantly.

OpsWorks is AWS service, used for configuration management by utilising Chef framework. We can automate server configuration, deployment and management by using OpsWorks. It help as in managing EC2 instances in AWS as well as any on-premise servers.

1. Amazon Machine image - AMI is most important part of deployment in AWS. We use AMI to create images of our servers. These images are used for deployment to multiple servers in production and QA environments.
2. What are the key features of AWS OpsWorks stacks?

Ans - The key features of OpsWorks as follows -

a. server support - With OpsWorks Stack we can automate operational tasks on any server in AWS as well as our own data center.

b. scalable automation - We get automated scaling support with AWS OpsWorks stacks. each new instance in AWS can read configuration from OpsWorks.

c. Dashboard

d. Configureation as code

e. Universal application support

1. have you used route 53

Ans - Yes, I've used route 53. I used it for managing DNS I use it to redirect our code ID DNS to route53 using Amazon name servers and then from there in route53 we used to create CNM entries a records MX records txt records and we used to manage the entire DNS from there

1. what is the best feature of AWS which you like?

Ans - I like the auto scaling group and the elastic load balancer because it allows you to infinitely scale your application.

1. Have you used AWS or other cloud platforms?
   1. How long for?
   2. In production or just at home on personal projects?
2. How to keep logs on servers or containers with ephemeral storage?
3. Where to look when trying to reduce cloud costs without reducing capacity?
4. Name the "Big Three" cloud providers
   1. AWS
   2. GCE
   3. Azure
5. Describe the advantages/disadvantages of using CloudFormation to manage your resources
6. Would you use CloudFormation to create a RDS database?
7. Describe EC2 spot instances and which use cases it can be used to reduce costs
8. Talk about IAM roles
9. Talk about VPC's
   1. Subnets
   2. Internet Gateways
   3. NATing
   4. NACL's
   5. VPN/VPC Peering

**Configuration management**

1. Which Configuration Management tools are you most comfortable with? Examples include: Chef, Salt, Puppet, or Ansible?
2. which configuration management tool have you used?

Ans- so you can say that you've used all three chef ansible and puppet in one project or another I have good hands-on experience with each of them and I've written chef cookbooks, puppet modules and ansible play books

1. what is the difference between chef and ansible

Ans - So chef is a ruby based tool. Ruby is programming language it's open source just like ansible chef uses a client server and client only architecture so the difference is that ansible is simply an agentless tool so there is no agent in ansible whatsoever in chef there are two methodologies one is agent based and the other is agent list but if you talk about the difference one of the differences is that there is no agent in ansible and the other difference is ansible uses YAML for defining the state of the systems in its playbooks whereas chef  uses  Ruby for defining   the state of the systems in  its cookbooks  and it's more of  a programming language when it comes to chef and  YAML in ansible is not really a programming  language it's more of a  statically  typed code have you written any cookbooks  or modules and the  playbooks yes I've used both from  the repositories which each of  these configuration management tools provide so for  chef's I've used the  chef's supermarket cookbooks  and I've also written custom cookbooks for puppet  I have used modules from puppet Forge  and I've written my own modules for custom applications for ansible I've used the community provided Play Books   as well as I've written my own custom play books for provisioning of instances on the cloud as well as managing

1. Have you used Puppet, Chef, Salt or Ansible?
   1. How long have you used it for?
   2. Have you used it in production?
2. Describe the size of the environment that you automated (how many servers, small scale or large scale)

**Jenkins Questions**

1. What is Jenkins master Slave concept?
2. What Jenkins can do?
3. Jenkins plugin to build
4. Which branch does Jenkins use - when we don't mention default branch
5. How to update Jenkin Branches from repository
6. What is node
7. Please suggest the monitoring tool according to you, and why
8. What is ELK Stack
9. What is Beat and its functionality

**Nagios**

1. Nagios Plugin Directory
2. /usr/lib/nagios/plugins/
3. at our client and add the Nagios Server IP in the
4. edit the nrpe.cfg file and add the host address

**NGINX**

1. Why we use Nginx?
2. Nginx is a light weight application, as it uses less amount of memory as compared to Tomcat
3. What is nginx Re-write

**Production support**

1. what is the biggest issue you've faced in a production environment

Ans- so you can say that there's an application we used to run  on the cloud and there was some application deployment  error it was a logical  error in the application which caused an error behind  the elastic load balancer  so we had an elastic load balancer  which was tied to the auto scaling group so it kind of  snowballed the application and we were not able to control  it so the application scaled infinitely and it was snowballing the instances so we had to go in and manually freeze the size of the auto scaling group  once the size was  frozen we were able  to go into the instances check the logs  fix the issue   rebuild the AMI and restart the  auto scaling process so that was one of   the biggest issues in production I faced recently what is your DR our strategy  and a live website so a  DR our strategy is that  you   do a failover check using route 53 DNS so one of them is  a DNS based failover in which you've got  an exact replica of your environment your web servers your database your cache  so your database may not be  sync in real time with  a DR so it could be a one  day behind sync so the traffic is switched to DR whenever there's a downtime  in production automatically through a  DNS switch the other DR strategy is that  you  can have   instances in to availability  zones and then you can have  your load balancer   to switch traffic based on  the available those two zones  but that is a same region DNR  so from a multi-region DNR the best option is to  have   a  DNS   base to switch and you  can also have an interface tool interface proxy which can route traffic  based on the health of your endpoints how do you  scale a production web service  so if you're on the cloud you can use the auto scaling group  which will scale automatically based on demand if you don't want to go into instances and you don't want to manage instances you can use the   elastic beanstalk for a managed stack which takes care  of the  load balancing itself if you have an application which is already  containerized  it's a better practice to  run them on EECS on the cloud  which would allow you to scale your applications based on demand  and you can create an application load balancer there if  you're in house you can scale  your application   by having a container management  system in which you can  have some base hosts on which you run  multiple services so   you can keep on adding multiple hosts to that cluster and you don't have to worry about the scaling  part because the  hosts can be added on  the fly to that hardware so your docker cluster like a swarm or a kubernetes can take care  of getting those new machines into your cluster and running the containers which are existing on your cluster   on those new machines  thanks  a   lot everyone this is all from the interview question Hey want to become an expert in cloud computing  then subscribe to   simpler Channel and click here to watch more such videos  turn it   up and get certified in cloud computing click here

**Architecture Questions**

1. How to scale a database without just increasing capacity of a single machine while maintaining ACID?
2. How to choose between relational database and noSQL?
3. What advantages a NoSQL database like MongoDB has, comparing to MySQL?
4. How to manage API versions?
5. How to reduce load time of a dynamic website?
6. How to reduce load time of a static website?
7. What's the use case for a database read replica?
8. Explain what a RESTful architecture is?
9. How would you scale a slow website?
10. Your application just got millions of users overnight, what do you do?
11. What are some ways to increase client-side performance?
12. Can you explain the significance of a primary/replica database architecture?
13. What are some caching strategies you can implement to increase performance?
14. What are the advantages and disadvantages of implementing a Microservices architecture?
15. What is the difference between vertical and horizontal scalability?

**Security Questions**

1. Difference between authorization and authentication?
2. Describe two-factor authentication
3. Describe how would you secure a web application
4. HTTP vs HTTPS
5. Talk about PKI/your experience with SSL/Certificates
6. What is the importance of SSL?
7. What is a SQL injection?
8. What is cross-site scripting (XSS)?
9. Why shouldn’t you roll your own crypto?
10. How are passwords stored on databases?
11. What is a Man-in-the-middle attack?
12. How do you safely manage environment variables in a cloud environment?
13. How do you manage security updates?
14. How do you keep encryption keys and credentials secure but make them available to machines that need them?

**Network Questions**

1. Say I open a web browser and enter an address. I hit enter. Describe how the connection the works in as much detail as possible. Trying to hear that they understand:
   1. DNS
   2. Network routing
   3. Load Balancing
   4. Ports on server
   5. Service that is serving port
2. What’s a PTR in DNS?
3. What’s a MX record in DNS?
4. How a CDN chooses the closest host to serve a client?
5. In which cases would you choose to not implement a CDN?
6. How do you measure the performance of a server/web application? (tools, methods)
7. What are secure ways to SSH to a server inside a private network from a public location?
8. What is the difference between a vlan and subnet?
9. What is a broadcast domain?
10. What is the difference between icmp, tcp and udp?
11. Explain how a tcp session is set up?
12. How are tcp options negotiated and selected?
13. How would you use tcpdump to determine if one remote host was making a connection to your server?

**Database Questions**

1. What is the difference between a relational and non-relational database?
2. When do we optimize the database?
3. Can you give some ways to optimize the performance of a database?
4. What does A.C.I.D mean?
5. How can you customize the query plan to increase throughput?
6. Explain how you would do Database Continuous Delivery?
7. How do you create a user?
8. How do you provide privileges to a user?
9. What is the difference between a "left" and a "right" join?
10. Explain briefly the differences between InnoDB and MyISAM.
11. Describe briefly the steps you need to follow in order to create a simple master/slave cluster.
12. Why should you run "mysql\_secure\_installation" after installing MySQL?
13. How do you check which jobs are running?
14. How would you take a backup of a MySQL database?
15. How do you create a new postgress user?

**Docker and Container**

1. How does Docker improve scalability, distributed computing, and efficiency vs. traditional cloud virtual machines?
2. List some Docker use cases
3. What is Vagrant and what is it used for?

Ans- Vagrant is a tool that can create and manage virtualized (or containerized) environments for testing and developing software. At first, Vagrant used VirtualBox as the hypervisor for virtual environments, but now it supports also KVM.

1. Have you worked on containers?

Ans-Containers are form of lightweight virtualization, heavier than chroot but lighter than hypervisors. They provide isolation among processes while using same kernel as the host machine, and cgroups functionality within kernel. But container formats differ among themselves in a way that some provide more VM-like experience while other containerize only application.  
LXC containers are most VM-like and most heavy weight, while Docker used to be more light weight and was initially designed for single application container. But in more recent releases Docker introduced whole machine containerization features so now Docker can be used both ways. There is also rkt from CoreOS and LXD from Canonical, which builds upon LXC.

1. What is Kubernetes? Explain

Ans-It is massively scalable tool for managing containers, made by Google. It is used internally on huge deployments and because of that it is maybe the best option for production use of containers. It supports self-healing by restating non-responsive containers, it pack containers in a way that they take less resources and has many other great features.