

Who Needs This Book?

This guide will benefit:

- A beginner who has never faced any DevOps interview
- Anyone who wants a brief on DevOps
- Professional who want answers with examples and explanation
- One who don't know what "They" really want to hear....

Need "How To" tips, phrases, and words for answering job interview questions

Question 1:

Explain what is DevOps?

Answer:

It is a newly emerging term in IT field, which is nothing but a practice that emphasizes the collaboration and communication of both software developers and other information-technology (IT) professionals. It focuses on delivering software product faster and lowering the failure rate of releases.

Characteristics

DevOps

Basic premise

Agile, lean, collaboration of IT development and operations which is more of a cultural shift

Related to

Agile methodology

Priorities

Resource management, communication and teamwork

Benefits	Speed, functionality, stability and innovation
-----------------	--

Question 2:

What is the need for DevOps?

Answer:

Instead of releasing big sets of features, companies are trying to see if small features can be transported to their customers through a series of release trains. This has many advantages like quick feedback from customers, better quality of software etc., which in turn leads to high customer satisfaction. To achieve this, companies are required to:

1. Increase deployment frequency
2. Lower failure rate of new releases
3. Shortened lead time between fixes

4. Faster mean time to recovery in the event of new release crashing

DevOps fulfills all these requirements and helps in achieving seamless software delivery. You can give examples of companies like Etsy, Google and Amazon which have adopted DevOps to achieve levels of performance that were unthinkable even five years ago. They are doing tens, hundreds or even thousands of code deployments per day while delivering world class stability, reliability and security.

Question 3:

What are the Principles of DevOps?

Answer:

- Collaboration
- Breakdown the barriers
- Work as one team end to end

- Treat Infrastructure as code
- Support business and IT agility
- Automate everything
- Test everything
- Measure & monitor everything

- Less complex problems to fix
- Faster resolution of problems

Question 4:

What are the advantages of DevOps?

Answer:

For this answer, you can use your past experience and explain how DevOps helped you in your previous job. If you don't have any such experience, then you can mention the below advantages.

Technical benefits:

- Continuous software delivery

Business benefits:

- Faster delivery of features
- More stable operating environments
- More time available to add value (rather than fix/maintain)

Question 5:

The top 10 skills the person should be having for the DevOp's position?

Answer:

- Excellent in System Admin
- Virtualization Experience
- Good Technical Skills

- Excellent Scripting
- Good Developing skills
- Chef in Automation Tool Experience
- People Management
- Customer Service
- Real time Cloud operations
- Who care about someone

Question 6:

How did the last system that you worked on, as a DevOps engineer changed from the time you started to the time you left, and who drove those changes?

Answer:

Candidate's answer's should display an unquenchable desire to keep

looking for improvements, and the willpower to make sure those improvements happen. "A great candidate will have transformed their systems for the better, by their own initiative," says Brian Kelly, vice president of engineering at the DevSecOps company Conjur. Bonus points if they tell you, they rallied others to their cause rather than just acted as a lone hero.

Candidate's technical knowledge is important, of course, but that innate personal drive to improve the efficiency and velocity of systems is vital, says Kelly, who's hired for DevOps roles at three different companies over the last five years.

Question 7:

What is Build Automation?

Answer:

Build Automation are tools or frameworks that allow source code to be automatically compiled into releasable binaries. Usually includes code-level unit

testing to ensure individual pieces of code behave as expected.

Question 8:

What is Scrum?

Answer:

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.

Question 9:

List the agile methodology of DevOps.

Answer:

- DevOps is a process
- Agile is same as DevOps.
- Separate group for are framed.

- It is problem solving.

- Developers managing production

- DevOps is development-driven release management

Question 10:

List the major difference between the Agile and DevOps.

Answer:

Agile:

1. Agile is about software development

Devops:

1. DevOps is about software deployment and management.
2. DevOps does not replace Agile or Lean. It does this by killing waste, removing handovers, and streamlining deployments to allow faster and

more continuous deployments to PRODUCTION.

Agile is a set of values and principles about how to produce i.e. develop software. Example: if you have some ideas and you want to turn those ideas into working software, you can use the agile values and principles as a way to do that. But, that software might only be working on a developer's laptop or in a test environment. You want a way to quickly, easily and repeatedly move that software into production infrastructure, in a safe and simple way. To do that, you need DevOps tools and techniques.

You can summarize by saying Agile software development methodology focuses on the development of software but DevOps on the other hand is responsible for development as well as deployment of the software in the safest and most reliable way possible.

Question 11:

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

Answer:

DevOps engineer work 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.

Question 12:

What is the most important thing DevOps helps us achieve?

Answer:

The most important thing that DevOps helps us achieve is to get the changes into production as quickly as possible while minimizing risks in software quality assurance and compliance. This is the primary objective of DevOps. However, you can add many other positive effects of DevOps. For

example, clearer communication and better working relationships between teams i.e. both the Ops team and Dev team collaborate together to deliver good quality software which in turn leads to higher customer satisfaction.

Question 13:

Mention what are the key aspects or principle behind DevOps?

Answer:

The key aspects or principle behind DevOps is

- Infrastructure as code
- Continuous deployment
- Automation
- Monitoring
- Security

Question 14:

The second part of the answer has two possibilities:

1. If you have experience with all the above tools then you can say that I have worked on all these tools for developing good quality software and deploying that software easily, frequently, and reliably.
2. If you have experience only with some of the above tools then mention those tools and say that I have specialization in these tools and have an overview about the rest of the tools.

Question 15:

How do all these tools work together?

Answer:

Given below is a generic logical flow where everything gets automated for seamless delivery. However, this flow may vary from organization to organization as per the requirement.

features.

Question 16:

What is Version control?

Answer:

It is a system that records changes to a file or set of files over time so that you can recall specific versions later. Version control systems consist of a central shared repository where teammates can commit changes to a file or set of file. Then you can mention the uses of version control.

Version control allows you to:

- Revert files back to a previous state.
- Revert the entire project back to a previous state.
- Compare changes over time.
- See who last modified something that might be causing a problem.

Which are the top DevOps tools? Which tools have you worked on?

Answer:

The most popular DevOps tools are mentioned below:

- Git : Version Control System tool
- Jenkins : Continuous Integration tool
- Selenium : Continuous Testing tool
- Puppet, Chef, Ansible : Configuration Management and Deployment tools
- Nagios : Continuous Monitoring tool
- Docker : Containerization tool

You can also mention any other tool if you want, but make sure you include the above tools in your answer.

1. Developers develop the code and this source code is managed by Version Control System tools like Git etc.
2. Developers send this code to the Git repository and any changes made in the code is committed to this Repository.
3. Jenkins pulls this code from the repository using the Git plugin and builds it using tools like Ant or Maven.
4. Configuration management tools like puppet deploys & provisions testing environment and then Jenkins releases this code on the test environment on which testing is done using tools like selenium.
5. Once the code is tested, Jenkins send it for deployment on the production server (even production server is provisioned & maintained by tools like puppet).
6. After deployment It is continuously monitored by tools like Nagios.
7. Docker containers provides testing environment to test the build

- Who introduced an issue and when.

Question 17:

What is Puppet?

Answer:

It is a Configuration Management tool which is used to automate administration tasks.

Now, you should describe its architecture and how Puppet manages its Agents. Puppet has a Master-Slave architecture in which the Slave has to first send a Certificate signing request to Master and Master has to sign that Certificate in order to establish a secure connection between Puppet Master and Puppet Slave. Puppet Slave sends request to Puppet Master and Puppet Master then pushes configuration on Slave.

Question 18:

What is Ansible?

Answer:

Ansible is a software tool to deploy application using ssh without any downtime. It is also used to manage and configure software applications. You don't need any agent on server to deploy application using Ansible. It is developed in Python language.

Question 19:

Name few advantages of Ansible?

Answer:

Below are few advantages of Ansible:

- Agent-less
- Very low overhead
- Good Performance

Question 20:

What is Chef?

Answer:

It is a powerful automation platform that transforms infrastructure into code. Chef is a tool for which you write scripts that are used to automate processes. Now you can explain the architecture of Chef, it consists of:

Chef Server: The Chef Server is the central store of your infrastructure's configuration data. The Chef Server stores the data necessary to configure your nodes and provides search, a powerful tool that allows you to dynamically drive node configuration based on data.

Chef Node: A Node is any host that is configured using Chef-client. Chef-client runs on your nodes, contacting the Chef Server for the information necessary to configure the node. Since a Node is a machine that runs the Chef-client software, nodes are sometimes referred to as "clients".

Chef Workstation: A Chef Workstation is the host you use to modify your cookbooks and other configuration data.

Question 21:

What is Nagios?

Answer:

Nagios is one of the monitoring tools. It is used for continuous monitoring

- Ensure IT infrastructure outages have a minimal effect on your organization's bottom line.
- Monitor your entire infrastructure and business processes.

Nagios runs on a server, usually as a daemon or service. Nagios periodically runs plugin residing on the same server; they contact hosts or servers on your network or on the internet. One can view the status information using the web interface. You can also receive email or SMS notifications if something happens.

The Nagios daemon behaves like a scheduler that runs certain scripts at certain moments. It stores the results of those scripts and will run other scripts if these results change.

Question 22:

What are containers?

of systems, applications, services, and business processes etc. in a DevOps culture. In the event of a failure, Nagios can alert technical staff of the problem, allowing them to begin remediation processes before outages affect business processes, end-users, or customers. With Nagios, you don't have to explain why an unseen infrastructure outage affects your organization's bottom line.

Now, once you have defined what Nagios is, you can mention the various things that you can achieve using Nagios.

By using Nagios you can:

- Plan for infrastructure upgrades before outdated systems cause failures.
- Respond to issues at the first sign of a problem.
- Automatically fix problems when they are detected.
- Coordinate technical team responses.
- Ensure your organization's SLAs are being met.

Answer:

Containers are used to provide consistent computing environment from a developer's laptop to a test environment, from a staging environment into production. Container consists of an entire runtime environment: an application, plus all its dependencies, libraries and other binaries, and configuration files needed to run it, bundled into one package. Containerizing the application platform and its dependencies removes the differences in OS distributions and underlying infrastructure.

Question 23:

What is Git?

Answer:

- Git is a Distributed Version Control system (DVCS). It can track changes to a file and allows you to revert back to any particular change.

What is Git?

Answer:

- Git is a Distributed Version Control system (DVCS). It can track changes to a file and allows you to revert back to any particular change.
- Its distributed architecture provides many advantages over other Version Control Systems (VCS) like SVN. One major advantage is that it does not rely on a central server to store all the versions of a project's files. Instead, every developer "clones" a copy of a repository. The local repository has the full history of the project on his hard drive so that when there is a server outage, all you need for recovery is one of your teammate's local Git repositories.
- There is a central cloud repository as well where developers can commit changes and share it with other teammates where all collaborators are committing changes "Remote repository".

Question 24:

Explain some basic Git commands?

Answer:

Command	Function
git config --global user.name "name" git config --global user.email "E-mail"	Configure the author name and email address to be used with your commits
Git init	Create a new local local repository
Git clone /path/to/repository	Create a working copy of a local repository
git clone username@host:/path/to/repository	For a remote server, use
git add <Filename.> git add *	Add one or more file to staging
git commit -m "Commit message"	Commit changes to head
git commit -a	Commit any files you've added with git add, and also commit any files you've changed since then
git push origin master	Send changes to the master branch of your remote repository
git status	List the files you've changed and those you still need to add or commit
git remote add origin <server>	If you haven't connected your local repository to a remote server, add the server to be able to push to it

Question 25:

What are the advantages that Containerization provides over virtualization?

Answer:

Below are the advantages of containerization over virtualization:

- Containers provide real-time provisioning and scalability but VMs provide slow provisioning
- Containers are lightweight when compared to VMs
- VMs have limited performance when compared to containers
- Containers have better resource utilization compared to VMs

Question 26:

How exactly are containers (Docker in our case) different from hypervisor virtualization (vSphere)? What are the benefits?

Answer:

Below are some differences:

Feature	Hypervisor Virtualization	Containers
Default security support	To a great degree	To a slightly less degree
Memory on disk required	Complete OS plus apps	App requirement only
Time taken to start up	Substantially longer because It requires boot of OS plus app loading	Substantially shorter because only apps need to start as kernel is already running
Portability	Portable with proper preparation	Portable within image format; typically smaller
Operating System	Has its own OS	It uses the host OS

Question 27:

What is Docker hub?

Answer:

Docker hub is a cloud-based registry service which allows you to link to code repositories, build your images and test them, stores manually pushed

images, and links to Docker cloud so you can deploy images to your hosts. It provides a centralized resource for container image discovery, distribution and change management, user and team collaboration, and workflow automation throughout the development pipeline.

Question 28:

What is Docker Container?

Answer:

Docker containers include the application and all of its dependencies but share the kernel with other containers, running as isolated processes in user space on the host operating system. Docker containers are not tied to any specific infrastructure: they run on any computer, on any infrastructure, and in any cloud.

Now, how to create a Docker container? Docker containers can be created by either creating a Docker image or then running it or you can use Docker images that are present on the Dockerhub.

Docker containers are basically runtime instances of Docker images.

Question 29:

What is Docker Swarm?

Answer:

It is native clustering for Docker which turns a pool of Docker hosts into a single, virtual Docker host. Docker Swarm serves the standard Docker API, any tool that already communicates with a Docker daemon can use Swarm to transparently scale to multiple hosts.

I will also suggest you to include some supported tools:

- Dokku
- Docker Compose
- Docker Machine
- Jenkins

Question 30:

How can you improve DevOps culture?

Answer:

- Open communication: a new culture is always created through discussions. In the Devops approach, however, the talks are focused on the product through its lifecycle rather discussing about the organization.
- Responsibility: DevOps becomes most effective when its principles pervade all the organization rather than being limited to single roles. Everyone is accountable for building and running an application that works as expected. This turns in assigning wider responsibilities and rewards at various levels.
- Respect: As open communication is necessary so does respect which means respectful discussion and listening to other opinions and experiences
- Trust: In a perfect Devops trust is essential. Operations must trust Development they are doing their best according to the common plan. Development must trust that Quality Assurance is there to improve the quality of their work and Product Manager needs to trust that Operations is going to provide precise metrics and reports on the product deployment

- Containers: Containers modernize IT environments and processes, and provide a flexible foundation for implementing DevOps. At the organizational level, containers allow for appropriate ownership of the technology stack and processes, reducing hand-offs and the costly change coordination that comes with them.

Question 32:

What are the core operations of DevOps with application development and with infrastructure?

Answer:

The core operations of DevOps with Application development

- Code building
- Code coverage
- Unit testing
- Packaging

down. It affected sales for millions of Etsy's users who sold goods through online market place and risked driving them to the competitor. With the help of a new technical management team, Etsy transitioned from its waterfall model, which produced four-hour full-site deployments twice weekly, to a more agile approach. Today, it has a fully automated deployment pipeline, and its continuous delivery practices have reportedly resulted in more than 50 deployments a day with fewer disruptions.

Question 34:

Explain your understanding and expertise on both the software development side and the technical operations side of an organization you have worked with in the past.

Answer:

For this answer, share your past experience and try to explain how flexible you were in your previous job. You can refer the below example:

DevOps engineers almost always work in a 24/7 business-critical online environment. I was adaptable to on-call duties and was available to take up

Question 31:

Which technologies can act as driver to enable DevOps?

Answer:

- Paas: It is a category of cloud computing services that provides a platform allowing customers to develop, run, and manage applications without the complexity of building and maintaining the infrastructure
- Iaas: It is a category of cloud computing services that abstract the user from the details of infrastructure like physical computing resources, location, data partitioning, scaling, security, backup etc.
- Configuration automation: Automation is a big win in part because it eliminates the labor associated with repetitive tasks. Codifying such tasks also means documenting them and ensuring that they're performed correctly, in a safe manner, and repeatedly across different infrastructure types.
- Micro services: It consists in a particular way of designing software applications as suites of independently deployable services.

- Deployment

With infrastructure

- Provisioning
- Configuration
- Orchestration
- Deployment

Question 33:

Explain with a use case where DevOps can be used in an industry / real-life.

Answer:

There are many industries that are using DevOps so you can mention any of those use cases or you can refer the below example:

Etsy is a peer-to-peer e-commerce website focused on handmade or vintage items and supplies, as well as unique factory-manufactured items. Etsy struggled with slow, painful site updates that frequently caused the site to go

real-time, live-system responsibility. I successfully automated processes to support continuous software deployments. I have experience with public/private clouds, tools like Chef or Puppet, scripting and automation with tools like Python and PHP, and a background in Agile.

Question 35:

What do you understand by term Provisioning in DevOps?

Answer:

Provisioning is the process of preparing new systems for users (in a Continuous Delivery scenario, typically development or test teams). The systems are generally virtualized and instantiated on demand. Configuration of the machines to install operating systems, middle-ware, etc. is handled by automated system configuration management tools, which also verify that the desired configuration is maintained.

Question 36:

What check-in standards do you recommend before a build can be scheduled?

Answer:

The goal here is to get the candidate to articulate what source control conditions should be met before a full build. For instance, there may be a requirement that third-party analysis tools be run before a build can be performed or scheduled. A candidate may also suggest that, in some cases, before code can be checked in for a build, it might need a code review.

Question 37:

How do you prevent accidental connections between environments?

Answer:

Environments should have barriers preventing them from talking to one another, so that the production system doesn't use services from the developer workstations or the integration environments.

Question 38:

How do you push configuration changes through environments?

Answer:

Configuration is tricky to manage because the developer adds new

configuration options to the software, and each environment has its own configuration files. A candidate should discuss that, typically, there is a merge process to create final configuration files out of parts that are solution-specific as well as some which are environment-specific.

Question 39:

How do you package your deployments?

Answer:

The answer can be MSI, ZIP, or any suitable mechanism. The intent of the question is to identify the packaging mechanisms that the candidate is familiar with and has used.

Question 40:

How do you perform "smoke tests?"

Answer:

Smoke tests are designed to quickly indicate a successful or failed deployment. The answer here should include automated or semi-automated execution of smoke tests, and a candidate should identify the kinds of tests that should be used to quickly assess whether or not the deployment was successful.

Question 41:

What are the benefits of using version control?

Answer:

Below are the advantages of using version control:

1. With Version Control System (VCS), all the team members are allowed to work freely on any file at any time. VCS will later allow you to merge all the changes into a common version.
2. All the past versions and variants are neatly packed up inside the VCS.

Task branching:

In this model each task is implemented on its own branch with the task key included in the branch name. It is easy to see which code implements which task, just look for the task key in the branch name.

Release branching:

Once the develop branch has acquired enough features for a release, you can clone that branch to form a Release branch. Creating this branch starts the next release cycle, so no new features can be added after this point, only bug fixes, documentation generation, and other release-oriented tasks should go in this branch. Once it is ready to ship, the release gets merged into master and tagged with a version number. In addition, it should be merged back into develop branch, which may have progressed since the release was initiated.

Question 43:

How do you manage roll-backs?

Answer:

A typically overlooked process is what happens when the build fails to

When you need it, you can request any version at any time and you'll have a snapshot of the complete project right at hand.

3. Every time you save a new version of your project, your VCS requires you to provide a short description of what was changed. Additionally, you can see what exactly was changed in the file's content. This allows you to know who has made what change in the project.
4. A distributed VCS like Git allows all the team members to have complete history of the project so if there is a breakdown in the central server you can use any of your teammate's local Git repositories.

Question 42:

Describe branching strategies?

Answer:**Feature branching:**

A feature branch model keeps all of the changes for a particular feature inside of a branch. When the feature is fully tested and validated by automated tests, the branch is then merged into master.

deploy to an environment. Understanding the steps that are taken—and that there are ways to recover—is the key to this question.

Question 44:

How do you know when to deprovision dynamic test environments?

Answer:

Generally, test environments are deprovisioned because there are a maximum number of environments that can be active at any one time—or because a time limit has expired. Conditions may dictate that users are notified before an environment is recycled.

Question 45:

How do you secure certificates and signing information?

Answer:

Because builds should be signed—and impossible to produce a version of the software not created on the build server—keeping the certificates and private keys restricted is important. However, developers need to be able to build the software for their testing. The response should discuss how a candidate

can build for themselves while only the build server creates official builds.

Question 46:

How do you expect you would be required to multitask as a DevOps professional?

Answer:

Focus attention on bridging communication gaps between Development and Operations teams.

Understand system design from an architect's perspective, software development from a developer's perspective, operations and infrastructure from the perspective of a seasoned Systems Administrator.

Execute – to be able to actually do what needs to be done.

Question 47:

What are the goals of Configuration management processes?

Answer:

The purpose of Configuration Management (CM) is to ensure the integrity of a product or system throughout its life-cycle by making the development or

deployment process controllable and repeatable, therefore creating a higher quality product or system. The CM process allows orderly management of system information and system changes for purposes such as to:

- Revise capability,
- Improve performance,
- Reliability or maintainability,
- Extend life,
- Reduce cost,
- Reduce risk and
- Liability or correct defects.

Question 48:

What is the difference between Asset management and Configuration Management?

Answer:

Given below are few differences between Asset Management and

Configuration Management:

Asset Management	Configuration Management
Concerned with finances	Concerned with operations
Scope is everything you own	Scope is everything you deploy
Interfaces to purchasing and leasing	Interfaces to ITIL processes
Maintains data for taxes	Maintains data for troubleshooting
Lifecycle from Purchase to disposal	Lifecycle from deploy to retirement
Only incidental relationships	All operational relationships

Question 49:

What testing is necessary to ensure that a new service is ready for production?

Answer:

DevOps is all about continuous testing throughout the process, starting with development through to production. Everyone shares the testing responsibility. This ensures that developers are delivering code that doesn't have any

errors and is of high quality, and it also helps everyone leverage their time most effectively.

Question 50:

What types of testing are needed?

Answer:

This question exhibits the candidate's knowledge about the real-world failure modes and the level of experience a candidate has. The software teams always believe that the software will always work, but it fails occasionally. It is necessary to find team members who assume that the code will fail and they must be ready to plan for those failures. One must try to integrate early load testing, multi-variate testing, unit test strategy, use of test harnesses, network simulation, A/B etc.

Question 51:

What's a PTR in DNS?

Answer:

Pointer records are used to map a network interface (IP) to a host name.

These are primarily used for reverse DNS. Reverse DNS is setup very similar to how normal (forward) DNS is setup. When you delegate the DNS forward, the owner of the domain tells the registrar to let your domain use specific name servers.

Question 52:

Describe two-factor authentication?

Answer:

Two-factor authentication is a security process in which the user provides two means of identification from separate categories of credentials; one is typically a physical token, such as a card, and the other is typically something memorized, such as a security code.

Question 53:

What other tools might help you in this role?

Answer:

DevOps is so diverse and inclusive that it rarely ends with coding, testing and systems. A DevOps project might rely on database platforms like SQL or

NoSQL, data structure servers like Redis, or configuration and management issue tracking systems like Redmine. Web applications are popular for modern enterprises, making a background with Web servers, like Microsoft Internet Information Services, Apache Tomcat or other Web servers, beneficial. Make sure to bring across that you are familiar with Agile application lifecycle management techniques and tools.

Question 54:

How Database fits in a DevOps ?

Answer:

In a perfect DevOps world, the DBA is an integral part of both Development and Operations teams and database changes should be as simple as code changes. So, you should be able to version and automate your Database scripts. In terms of choices between RDBMS, noSQL or other kind of storage solutions a good database design means less change to your schema of Data and more efficient testing and service virtualization. Treating database management as an afterthought and not choosing the right database during early

stages of the software development lifecycle can prevent successful adoption of the true DevOps movement.

Question 5:

Tell us about the CI tools that you are familiar with?

Answer:

The premise of CI is to get feedback as early as possible because the earlier you get feedback, the less things cost to fix. Popular open source tools include Hudson, Jenkins, CruiseControl and CruiseControl.NET. Commercial tools include ThoughtWorks' Go, Urbancode's Anthill Pro, Jetbrains' Team City and Microsoft's Team Foundation Server.

Question 56:

Explain what do you understand by A3 Problem Solving?

Answer:

A structured problem solving approach that uses a lean tool called the A3 Problem-Solving Report. The term "A3" represents the paper size historically used for the report (a size roughly equivalent to 11" x 17").

Question 57:

What do you understand by LXC?

Answer:

LXC - (LinuX Container) is an operating system-level virtualization method that allows multiple isolated Linux systems to run as single host that acts as the controller. Virtualization is not provided through Virtual machines, but rather through a virtual environment with a process set and network space. Each isolated system gets its own directory structure, network devices, IP addresses and process table. It's written in C, Python, Shell and LUA.

Question 58:

What are microservices and why they have an impact on operations ?

Answer:

Microservices are a product of software architecture and programming practices. Microservices architectures typically produce smaller, but more numerous artifacts that Operations is responsible for regularly deploying and managing. For this reason microservices have an important impact on

Operations. The term that describes the responsibilities of deploying microservices is microdeployments. So, what DevOps is really about is bridging the gap between microservices and microdeployments.

Question 59:

Explain how "Infrastructure of code" is processed or executed in AWS?

Answer:

In AWS,

- The code for infrastructure will be in simple JSON format
- This JSON code will be organized into files called templates
- These templates can be deployed on AWS and then managed as stacks
- Later the CloudFormation service will do the Creating, deleting, updating, etc. operation in the stack

Question 60:

Which scripting language is most important for a DevOps engineer?

Answer:

features quickly. It also helps for clearer communication between the team members.

Question 62:

Will DevOps solve all the problems in the software?

Answers:

No. DevOps contains tools, principles, and techniques to develop the software safely and easily. However, it will not rectify any software issues on its own. It helps the developers to fix the problems in an effective way.

Question 63:

Name some popular companies that are benefited using DevOps.

Answer:

Top IT enterprises like Amazon, Google, and Etsy have reported adopting DevOps and achieved improvement in terms of performance and efficiency. Being a large software manufacturer, they develop tens and thousands of software code deployments every day.

Question 64:

Software development and Operational automation requires programming. In terms of scripting, **Bash** is the most frequently used Unix shell which should be your first automation choice. It has a simple syntax, and is designed specifically to execute programs in a non-interactive manner.

The same stands for **Perl** which owes great deal of its popularity to being very good at manipulating text and storing data in databases.

Next, if you are using Puppet or Chef it's worth learning **Ruby** which is relatively easy to learn, and so many of the automation tools have been specifically with it.

Java has a huge impact in IT backend, although it has a limited spread across Operations.

Python is other popular alternative.

Question 61:

Explain how DevOps is helpful to developers?

Answer:

DevOps can be helpful to developers to fix the bug and implement new

List out some popular tools for DevOps?

Answer:

Some of the popular tools for DevOps are

- Jenkins
- Nagios
- Monit
- ELK (Elasticsearch, Logstash, Kibana)
- io
- Jenkins
- Docker
- Ansible
- Git
- Collectd/Collectl

Question 65:

What is SSH?

Answer:

SSH (Also known as Secure Shell) is a program to log into another computer over a network, to execute commands in a remote machine, and to move files from one machine to another. It provides strong authentication and secure communications over unsecure channels. It is intended as a replacement for rlogin, rsh, and rcp.

Question 66:

Mention at what instance have you used the SSH?

Answer:

- I have used SSH to log into a remote machine and work on the command line.
- Beside this, I have also used it to tunnel into the system in order to facilitate secure encrypted communications between two untrusted hosts over an insecure network.

Question 67:

- We can't do DevOps – We're Unique
- We can't do DevOps – We've got the wrong people

Question 68:

What are Vagrant and its uses?

Answer:

- Vagrant used virtual box as the hypervisor for virtual environments and in current scenario it is also supporting the KVM. Kernel-based Virtual Machine
- Vagrant is a tool that can create and manage environments for testing and developing software.

Question 69:

Explain how would you handle revision (version) control?

Answer:

My approach to handle revision control would be to post the code on

Question 71:

Explain the working of HTTP?

Answer:

Like other protocols, HTTP also works on the client-server model. A web server software which responds to the request is called a server and a web browser which initiates the request is called a client. HTTP enhances its request and response with the help of intermediates such as tunnel, proxy or gateway. URL helps in allocating the resources that are requested using HTTP. The connection to the application layer port of HTTP is provided by TCP.

Question 72:

Explain what would you check If a Linux-build-server suddenly starts getting slow?

Answer:

If a Linux-build-server suddenly starts getting slow, you will check for following three things

What are the anti-patterns of DevOps?

Answer:

A pattern is common usage usually followed. If a pattern commonly adopted by others does not work for your organization and you continue to blindly follow it, you are essentially adopting an anti-pattern. There are myths about DevOps. Some of them include:

- DevOps is a process
 - Agile equals DevOps?
 - We need a separate DevOps group
 - Devops will solve all our problems
 - DevOps means: Developers Managing Production
 - DevOps is Development-driven release management
1. DevOps is not development driven.
 2. DevOps is not IT Operations driven.

SourceForge or GitHub so everyone can view it. Also, I will post the checklist from the last revision to make sure that any unsolved issues are resolved.

Question 70:

Mention what are the types of Http requests?

Answer:

The types of Http requests are

- GET
- HEAD
- PUT
- POST
- PATCH
- DELETE
- TRACE
- CONNECT
- OPTIONS

• Application Level Troubleshooting

RAM related issues, Disk I/O read write issues, Disk Space related Issues, etc.

Check for Application log file OR application server log file, system performance issues, Web Server Log – check HTTP, tomcat log, etc. or check jboss, weblogic logs to see if the application server response/receive time is the issues for slowness, Memory Leak of any application

• System Level Troubleshooting

Antivirus related issues, Firewall related issues, Network issues, SMTP server response time issues, etc.

• Dependent Services Troubleshooting

Question 73:

What are the major differences between the Linux and Unix operating systems?

Answer:**Unix:**

- It belongs to the family of multitasking, multiuser operating systems.
- These are mostly used in internet servers and workstations.
- It is originally derived from AT&T Unix, developed starting in the 1970s at the Bell Labs research center by Ken Thompson, Dennis Ritchie, and others.
- Both the operating systems are open source but UNIX is relatively similar one as compared to LINUX.

Linux:

- Linux has probably been home to every programming language known to

humankind.

- These are used for personal computers.
- The LINUX is based on the kernel of UNIX operating system.

Question 74:

What do you understand by Acceptance Testing?

Answer:

Acceptance Testing is typically high-level testing of the entire system carried out to determine whether the overall quality of both new and existing features is good enough for the system to go to production.

Question 75:

How we can make sure new service is ready for the products launched?

Answer:

- Backup System
- Recovery plans:
- Load Balancing

- Monitoring
- Centralized logging

Question 76:

What are the benefits of the NoSQL?

Answer:

- Non-relational and schema-less data model
- Low latency and high performance
- Highly scalable

Question 77:

What are the advantages of NoSQL database over RDBMS?

Answer:

The advantages are:

1. There is very less scope of ETL
2. Support is given for unstructured text

3. Changes are handle over period of time
4. Main objectives are functionality.
5. It has the ability to scale horizontally
6. Multiple data structures are given support.
7. Vendors can be chosen.

Question 78:

What are the main SQL migration difficulties to NoSQL ?

Answer:

Each record in a relational database according to a schema - with a fixed number of fields (columns) each has a specified object and a data type. Each record is the same. The data is denormalized in several tables. The advantage is that there is less of duplicate data in the database. The downside is that a change in the pattern means performing several "alter table" that require expensive to lock multiple tables simultaneously to ensure that change does not leave the database in an inconsistent state.

With databases data, on the other hand, each document can have a

completely different structure from other documents. No additional management is required on the database to manage changes in the schemes.

Question 79:

How would you know whether your video card can run Unity ?

Answer:

When you use command

Shell

/usr/lib/nux/unity_support_test-p

It will give detailed output about Unity's requirements and if they are met, then your video card can run unity.

Question 80:

Explain how to enable startup sound in Ubuntu?

Answer:

To enable startup sound

- Click control gear and then click on Startup Applications
- In the Startup Application Preferences window, click Add to add an entry

- Then fill the information in comment box like Name, Command and Comment
- Shell

```
/usr/bin/canberra-gtk-play--id=“desktop-login”--description=“play login sound”
```

- Logout and then login once you are done
- You can also open it with shortcut key Ctrl+Alt+T.

Question 81:

What is the quicker way to open an Ubuntu terminal in a particular directory?

Answer:

To open Ubuntu terminal in a particular directory you can use custom keyboard short cut.

To do that, in the command field of a new custom keyboard, type genome

– terminal – working – directory = /path/to/dir.

Question 82:

Explain how you can get the current color of the current screen on the Ubuntu desktop?

Answer:

You can open the background image in The Gimp (image editor) and then use the dropper tool to select the color on the specific point. It gives you the RGB value of the color at that point.

Question 83:

Explain how you create launchers on desktop in Ubuntu?

Answer:

To create launchers on desktop in Ubuntu you can use ALT+F2 then type “ gnome-desktop-item-edit –create-new~/desktop ”, it will launch the old GUI dialog and create a launcher on your desktop

Question 84:

Explain what is Memcached?

- Gaming and entertainment -> Session caching

Question 88:

What does Memcache helps in?

Answer:

- Speed up application processes
- It determines what to store and what not to
- Reduce the number of retrieval requests to the database
- Cuts down the I/O (Input/Output) access (hard disk)

Question 89:

What is drawback of Memcached ?

Answer:

- It is not a persistent data store
- Not a database
- It is not an application specific
- It cannot cache large object

Question 90:

Yes, it is possible to share a single instance of Memcache between multiple projects. Memcache is a memory store space, and you can run memcache on one or more servers. You can also configure your client to speak to a particular set of instances. So, you can run two different Memcache processes on the same host and yet they are completely independent. If you have partitioned your data, then it becomes necessary to know from which instance to get the data from or to put into.

Question 92:

You are having multiple Memcache servers, in which one of the memcacher server fails, and it has your data, will it ever try to get key data from that one failed server?

Answer:

The data in the failed server won't get removed, but there is a provision for auto-failure, which you can configure for multiple nodes. Fail-over can be triggered during any kind of socket or Memcached server level errors and not during normal client errors like adding an existing key, etc.

Answer:

Memcached is a free and open source, high-performance, distributed memory object caching system. The primary objective of Memcached is to enhance the response time for data that can otherwise be recovered or constructed from some other source or database. It is used to avoid the need to operate SQL data base or another source repetitively to fetch data for concurrent request.

Question 85:

What is Memcached used for?

Answer:

- Social Networking -> Profile Caching
- Content Aggregation -> HTML/ Page Caching
- Ad targeting -> Cookie/profile tracking
- Relationship -> Session caching
- E-commerce -> Session and HTML caching
- Location-based services -> Data-base query scaling

Mention some important features of Memcached?

Answer:

Important features of Memcached includes

- CAS Tokens: A CAS token is attached to any object retrieved from cache. You can use that token to save your updated object.
- Callbacks: It simplifies the code
- getDelayed: It reduces the delay time of your script which is waiting for results to come back from server
- Binary protocol: You can use binary protocol instead of ASCII with the newer client
- Igbinary: Previously, client always used to do serialization of the value with complex data, but with Memcached you can use igbinary option.

Question 91:

Explain whether it is possible to share a single instance of a Memcache between multiple projects?

Answer:

Question 93:

Explain how you can minimize the Memcached server outages?

Answer:

- When one instance fails, several of them goes down, this will put larger load on the database server when lost data is reloaded as client make a request. To avoid this, if your code has been written to minimize cache stampedes then it will leave a minimal impact
- Another way is to bring up an instance of Memcached on a new machine using the lost machines IP address
- Code is another option to minimize server outages as it gives you the liberty to change the Memcached server list with minimal work
- Setting timeout value is another option that some Memcached clients implement for Memcached server outage. When your Memcached server goes down, the client will keep trying to send a request till the time-out limit is reached

Question 95:

Explain how you can update Memcached when data changes?

Answer:

When data changes you can update Memcached by

- Clearing the Cache proactively: Clearing the cache when an insert or update is made
- Resetting the Cache: It is similar to the first method but rather than just deleting the keys and waiting for the next request for the data to refresh the cache, reset the values after the insert or update.

Question 96:

Explain what is Dogpile effect? How can you prevent this effect?

Answer:

Dogpile effect is referred to the event when cache expires, and websites are hit by the multiple requests made by the client at the same time. This effect can be prevented by using semaphore lock. In this system when value expires, first process acquires the lock and starts generating new value.

Question 97:

restarted then all the data stored in Memcached is deleted.

Question 99:

Mention what is the difference between Memcache and Memcached?

Answer:

- Memcache: It is an extension that allows you to work through handy object-oriented (OOP's) and procedural interfaces. It is designed to reduce database load in dynamic web applications.
- Memcached: It is an extension that uses libmemcached library to provide API for communicating with Memcached servers. It is used to increase the dynamic web applications by alleviating database load. It is the latest API.

Question 100:

What are containers?

Answer:

Containers are form of lightweight virtualization, more heavy than chroot but lighter than hypervisors. They provide isolation among processes while using same kernel as the host machine, and cgroups functionality within

Explain how Memcached should not be used?

Answer:

- Memcached common misuse is to use it as a data store, and not as a cache
- Never use Memcached as the only source of the information you need to run your application. Data should always be available through another source as well
- Memcached is just a key or value store and cannot perform query over the data or iterate over the contents to extract information
- Memcached does not offer any form of security either in encryption or authentication

Question 98:

When server gets shut down, is the data stored in Memcached still available?

Answer:

Data stored in Memcached is not durable so if server is shut down or

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.

Question 101:

What is Kubernetes? Explain

Answer:

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 restarting non responsive containers, it pack containers in a way that they take less resources and has many other great features.

Question 102:

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

Answer:

CI server's function is to continuously integrate all changes being made and committed to 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

Question 103:

What is Continuous Delivery?

Answer:

Is it practice of delivering the software for testing as soon as it is built by CI (Continuous Integration) servers. It requires heavy use of Versioning Control System for so always available to developers and testers alike.

Question 104:

What is automation ?

Answer:

Automation is the process of removing manual, error-prone operations from your services, ensuring that your applications or services can be repeatedly deployed.

Question 105:

At which level can applied automation in DevOps ?

Answer:

At three levels:

- 1) Automate the application lifecycle: in terms of software features, version control, build management, integration frameworks
- 2) Automate the middleware platform automation: such as installing middleware, autoscaling and resources optimization of middleware components
- 3) At infrastructure by provisioning operating system resources and virtualizing them

Question 106:

In terms of automation, discuss about the differences between Puppet, Ansible and Chef

Answer:**Push vs Pull Strategy:**

- Puppet nodes use a Pull strategy as nodes periodically check into a puppet master server to "pull" resource definitions.
- Ansible uses a Push strategy. The machine where Ansible is installed uses SSH to copy files, remotely install packages, etc. on target machines. The client machine requires no special setup outside of a working installation of Python 2.5+.
- Chef: Chef Client queries Chef server for the latest set of recipes (configuration instructions) that apply to the current node.

Server Nodes:

- Puppet infrastructure is made up of one or more "puppet master" servers, along with a special agent package installed on each client node.
- Ansible has no concept of master/slave server, nor special agent executable to install: just proper SSH keys/credentials in order to connect to the nodes.
- Chef infrastructure uses a Chef Server, the main hub where Chef propagates and stores system configuration information and policies and a Chef Client installed on every node being managed

Language and Extensibility:

- Puppet uses its own DSL language which is a subset of Ruby. Thus adding extra complex functionality is done through Ruby modules. That being said there's a more strict control on what you are doing with Ruby.
- Ansible playbooks are YAML files. In terms of extensibility, Ansible is built upon Python for which most organization will have some experience.

- Chef: uses Ruby as programming language that is the authoring syntax for Chef cookbooks. Put it straight Chef lets you run wild with Ruby.

Resources & Ordering

- Puppet: Resources defined in a Puppet manifest are not applied in order of their appearance (ex: top->bottom). Instead resources are applied randomly, unless explicit resource ordering is used.
- Ansible: The playbooks using a traditional top-to-bottom, as they appear in the file. This is more intuitive for developers coming from other languages.
- Chef: Always executes recipes in the order you specify them. It will not arbitrarily reorder things. So if you want one recipe to be run before another, just load them in that order

Resource Dependency

- Puppet internally creates a directed graph of all of the defined resources

along with the order they should be applied in. Puppet can even generate a graph file so that one can visualize everything that Puppet manages. On the other hand, building this graph is susceptible to "multiple resource definition" errors or conflicts due to circular dependencies.

- Ansible is merely a thin-wrapper for executing commands over SSH, so there is no resource dependency graph built internally.
- Chef is also able to declare dependencies between resources. Dependency failures are breakages in your dependency graph, which keep the current project's pipeline from being able to ship safely. These failures are tracked because through Chef Automation

DevOps Tool Support

Puppet, Ansible and Chef are well supported by other DevOps tools like Vagrant, Packer, and Jenkins.

Question 107:

What is the purpose of CM tools and which ones have you used ?

Answer:

Configuration Management tool's purpose is to automatize deployment and configuration of software on big number of servers. Most CM tools usually use agent architecture which means that every machine being managed needs to have agent installed. My favorite tool is one that uses agentless architecture - Ansible. It only requires SSH and Python. And if raw module is being used, not even Python is required because it can run raw bash commands. Other available and popular CM tools are Puppet, Chef, and SaltStack.

Question 108:

What is OpenStack ?

Answer:

OpenStack is often called Cloud Operating System, and that is not far from the truth. It is the complete environment for deploying IaaS which gives you possibility of making your own cloud similar to AWS. It is highly modular and

consists of many sub-projects so you can pick and choose which functionality you need. OpenStack distribution is available from Red Hat, Mirantis, HPE, Oracle, Canonical and many others. It is completely open source project but some vendors make proprietary distributions.

Question 109:

Classify Cloud Platforms anatogy?

Answer:

Cloud Computing software can be classified as Software as a Service or SaaS, Infrastructure as a Service or IaaS and Platform as a Service or PaaS.

SaaS is peace of software that runs over network on remote server and has only user interface exposed to users, usually in web browser. For example,salesforce.com.

Infrastructure as a service is a cloud environment that exposes VM to user to use as entire OS or container where you could install anything you would install on your server. Example for this would be OpenStack, AWS, and Euca-lryptus.

PaaS allows users to deploy their own application on the preinstalled platform, usually framework of application server and suite of developer tools. Examples for this would be OpenShiftHeroku.

Question 110:

What are easiest ways to build a small cloud ?

Answer:

VMfest is one of the options for making IaaS cloud from VirtualBox VMs in no time. If you want a lightweight PaaS there is Dokku which is basically a bash script that makes PaaS out of Dokku containers.

Question 111:

What is AWS (Amazon Web Services)? Did got chance to work on Amazon tools ?

Answer:

AWS provides a set of flexible services designed to enable companies to create and deliver products with greater speed and reliability using AWS and DevOps practices . These services simplify commissioning and infrastructure

management , application code deployment , automated software release process and monitoring of the application and infrastructure performance. Amazon used tools like AWS CodeCommit, AWS CodeDeploy, AWS CodePipeline etc that helps to make devops easier.

Question 112:

What is EC2?

Answer:

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

The EC2 service is inseparable from the concept of Amazon Machine Image - AMI . The May is Indeed the image of a virtual machine That Will Be Executed . EC2 based on XEN virtualization , that's why it is quite easy to move XEN servers to EC2

Question 113:

What are the main advantages of Git over CVS?

Answer:

The biggest advantage is that Git is distributed while CVS is centralized. Changes in CVS are per file, while changes (commits) in Git they always refer to the whole project. Git offers much more tools than CVS.

Question 114:

What are the difference between containers and virtual machines?

Answer:

Each VM instantiation requires starting a full OS. VMs take up a lot of system resources. This quickly adds up to a lot of RAM and CPU cycles. - cles. tain- host uses the process and file system isolation features of the linux kernel.

Question 115:

What is the role of a configuration management tool in devops ?

Answer:

Automation plays an essential role in server configuration management.

For that purpose we use CM tools , they store information about versions and builds of the software and testware and provide the traceability between software and testware.

Question 116:

Automation is a key point of DevOps, however what is the prerequisite of it ?

Answer:

The necessary prerequisite of it is standardization. Which means both:

- **Technical standardization:** choose standard Operating systems and middleware, develop with a standard set of common libraries
- **Process standardization:** standard systems development life cycle, release management, monitoring and escalation management.

Question 117:

What is CoreOS, and what are alternatives?

Answer:

network monitoring. Those tools are used to monitor network traffic, network quality and detect network problems even before they arise. Of those listed, only Splunk is proprietary other is open source.

Question 120:

What is Juju?

Answer:

Juju is orchestration tool primarily for Ubuntu for management, provision and configuration on Ubuntu systems. It is initially written in Python and since have been rewritten in Go.

Question 121:

Give me an examples of how you would handle projects ?

Answer:

As a DevOps engineer, I would demonstrate a clear understanding of DevOps project management tactics and also work with teams to set objectives, streamline workflow, maintain scope, research and introduce new tools or frameworks, trans requirements into workflow and follow up. I would resort to

CoreOS is stripped down linux distribution meant for running containers, mainly with its own rkt format but others are also supported. It was initially based on ChromeOS and supported Docker. The alternatives to this are canonical's Ubuntu snappy or red hat enterprise linux atomic host. Of course, Containers can also be ran on regular Linux system.

Question 118:

What is Kickstart?

Answer:

It is a way to install Red Hat based systems by automated way. During manual install process, Anaconda installer creates file anaconda-ks.cfg which then can be used with system-config-kickstart tool to install same configuration automatically on multiple systems.

Question 119:

What are tools for network monitoring? List a few.

Answer:

Nagios, Icinga 2, OpenNMS, Splunk and Wireshark are a few tools for

CI, release management and other tools to keep interdisciplinary projects on track.

Question 122:

What is post mortem meetings?

Answer:

It is a meeting where we discuss what went wrong and what steps should be taken so that failure doesn't happen again. Post mortem meetings are not about finding the one to be blamed, they are for preventing outages from recurring and planning redesign of the infrastructure so that downtime can be minimized. It is about learning from mistakes.

Question 122:

What you know about serverless model?

Answer:

Serverless refers to a model where the existence of servers is hidden from developers. It means you no longer have to deal with capacity, deployments, scaling and fault tolerance and OS. It will essentially reducing maintenance

servers.

Question 125:

Do a FizzBuzz coding test.

Answer:

The main idea of the FizzBuzz test is to see how a developer handles an easy coding task. Live simulations are a good way to see how quick engineers are on their feet as well as how they grasp a simple task and then translates it into code.

The candidate should:

- Write a program or script that prints out the numbers between 1 and 100
- For each number that is divisible by three, "Fizz" is printed
- For each number that is divisible by five, "Buzz" is printed
- For each number that is divisible by both three and five, "FizzBuzz" is printed

Most good developers should be able to write such a program on paper

determine the priority of email servers for a domain. The priority of email servers is divided into two categories, lowest priority email servers and higher priority email servers. The first destination for email is known as the lowest priority email servers, the mail will be sent to the highest priority email servers if the lowest priority email server is unavailable.

Question 128:

How would you make software deployable?

Answer:

The ability to script the installation and reconfiguration of software systems is essential towards controlled and automated change. Although there is an increasing trend for new software to enable this, older systems and products suffer from the assumption that changes would be infrequent and minor, and so make automated changes difficult. As a professional who appreciates the need to expose configuration and settings in a manner accessible to automation, I will work with concepts like Inversion of Control (IoC) and Dependency Injection, scripted installation, test harnesses, separation of concerns,

efforts and allow developers to quickly focus on developing codes.

Examples are Amazon AWS Lambda and Autho serverless platform.

Question 123:

Deploy vs. Release in DevOps? What do you understand by these two terms?

Answer:

- You can deploy code to production without releasing
- Release is to make functionality available to end users

Question 124:

What do you understand by NGINX?

Answer:

NGINX is open source software for web serving, reverse proxying, caching, load balancing, media streaming, and more. In addition to its HTTP server capabilities, NGINX can also function as a proxy server for email (IMAP, POP3, and SMTP), as a reverse proxy and load balancer for HTTP, TCP, and UDP

within a couple of minutes. See how they write the code, ask them why they wrote specific parts in certain ways, and then check the validity of the code.

Question 126:

What is the difference between RAID0 and RAID1?

Answer:

RAID0 provides no redundancy and it uses striping, which means the information is divided across all the drives, whereas in RAID1 redundancy is provided through mirroring, that means: the information is written identically to two drives. If anyone of the drives fail, RAID1 also fails and no false tolerance is provided by RAID0.

Question 127:

What is an MX record in DNS?

Answer:

MX records are nothing but the mail exchange records which is used to

command-line tools, and infrastructure as code.

Question 129:

What is the one most important thing DevOps helps do?

Answer:

The most important thing DevOps helps do is to get the changes into production as quickly as possible while minimizing risks in software quality assurance and compliance. That is the primary objective of DevOps. However, there are many other positive side-effects to DevOps. For example, clearer communication and better working relationships between teams which creates a less stressful working environment.

Question 130:

Which scripting languages do you think are most important for a DevOps engineer?

Answer:

As far as scripting languages go, the simpler the better. In fact, the language itself isn't as important as understanding design patterns and development paradigms such as procedural, object-oriented, or functional

programming.

Question 131:

How do you expect you would be required to multitask as a DevOps professional?

Answer:

- Focus attention on bridging communication gaps between Development and Operations teams.
- Understand system design from an architect's perspective, software development from a developer's perspective, operations and infrastructure from the perspective of a seasoned Systems Administrator.
- Execute – to be able to actually do what needs to be done.

Question 132:

Are you more DEV or Ops?

Answer:

This is one of the trickiest questions that you may face in your interview.

This question mainly depends on the skills of people involved, the company in which you are working and the job. You have to talk about the experience and you have to show the ability that you can handle both.

Question 133:

What is Continuous Testing?

Answer:

Continuous Testing is the process of executing automated tests as part of the software delivery pipeline to obtain immediate feedback on the business risks associated with the latest build. In this way, each build is tested continuously, allowing Development teams to get fast feedback so that they can prevent those problems from progressing to the next stage of Software delivery life-cycle. This dramatically speeds up a developer's workflow as there's no need to manually rebuild the project and re-run all tests after making changes.

Question 134:

What is Automation Testing?

Answer:

How to automate Testing in DevOps lifecycle?

Answer:

In DevOps, developers are required to commit all the changes made in the source code to a shared repository. Continuous Integration tools like Jenkins will pull the code from this shared repository every time a change is made in the code and deploy it for Continuous Testing that is done by tools like Selenium as shown in the below diagram.

In this way, any change in the code is continuously tested unlike the traditional approach

Question 137:

Why is Continuous Testing important for DevOps?

Answer:

"Continuous Testing allows any change made in the code to be tested immediately. This avoids the problems created by having "big-bang" testing left to the end of the cycle such as release delays and quality issues. In this way, Continuous Testing facilitates more frequent and good quality releases."

Automation testing or Test Automation is a process of automating the manual process to test the application/system under test. Automation testing involves use of separate testing tools which lets you create test scripts which can be executed repeatedly and doesn't require any manual intervention.

Question 135:

What are the benefits of Automation Testing?

Answer:

Some advantages of automation testing are:

- Supports execution of repeated test cases
- Aids in testing a large test matrix
- Enables parallel execution
- Encourages unattended execution
- Improves accuracy thereby reducing human generated errors
- Saves time and money

Question 136:**Question 138:**

What are the key elements of Continuous Testing tools?

Answer:

Key elements of Continuous Testing are:

1. **Risk Assessment:** It covers risk mitigation tasks, technical debt, quality assessment and test coverage optimization to ensure the build is ready to progress toward next stage.
2. **Policy Analysis:** It ensures all processes align with the organization's evolving business and compliance demands are met.
3. **Requirements Traceability:** It ensures true requirements are met and re-work is not required. An object assessment is used to identify which requirements are at risk, working as expected or require further validation.
4. **Advanced Analysis:** It uses automation in areas such as static code analysis, change impact analysis and scope assessment/prioritization to prevent defects in the first place and accomplishing more within each iteration.

5. **Test Optimization:** It ensures tests yield accurate outcomes and provide actionable findings. Aspects include Test Data Management, Test Optimization Management and Test Maintenance
6. **Service Virtualization:** It ensures access to real-world testing environments. Service visualization enables access to the virtual form of the required testing stages, cutting the waste time to test environment setup and availability.

Question 139:

Which Testing tool are you comfortable with and what are the benefits of that tool?

Answer:

We can state the tool name here what we have used. Here let's use Selenium

Some advantages of Selenium are:

- It is free and open source

points) individually.

Question 141:

What is NRPE (Nagios Remote Plugin Executor) in Nagios?

Answer:

The NRPE add-on is designed to allow you to execute Nagios plugins on remote Linux/Unix machines. The main reason for doing this is to allow Nagios to monitor "local" resources (like CPU load, memory usage, etc.) on remote machines. Since these public resources are not usually exposed to external machines, an agent like NRPE must be installed on the remote Linux/Unix machines.

The NRPE add-on consists of two pieces:

- The check_nrpe plugin, which resides on the local monitoring machine.
- The NRPE daemon, which runs on the remote Linux/Unix machine.

Question 142:

- Asynchronous in nature and cannot be monitored effectively by polling their status on a regularly scheduled basis.
- Located behind a firewall and cannot be checked actively from the monitoring host.

The main features of Active checks are as follows:

- Active checks are initiated by the Nagios process.
- Active checks are run on a regularly scheduled basis.

Question 144:

How does Nagios help with Distributed Monitoring?

Answer:

With Nagios you can monitor your whole enterprise by using a distributed monitoring scheme in which local slave instances of Nagios perform monitoring tasks and report the results back to a single master. You manage all configuration, notification, and reporting from the master, while the slaves do all

- It has a large user base and helping communities
- It has cross Browser compatibility (Firefox, chrome, Internet Explorer, Safari etc.)
- It has great platform compatibility (Windows, Mac OS, Linux etc.)
- It supports multiple programming languages (Java, C#, Ruby, Python, Pearl etc.)
- It has fresh and regular repository developments
- It supports distributed testing

Question 140:

What are the Testing types supported by Selenium?

Answer:

Selenium supports two types of testing:

- Regression Testing: It is the act of retesting a product around an area where a bug was fixed.
- Functional Testing: It refers to the testing of software features (functional

When Does Nagios check for external commands?

Answer:

Nagios check for external commands under the following conditions:

- At regular intervals specified by the command_check_interval option in the main configuration file or,
- Immediately after event handlers are executed. This is in addition to the regular cycle of external command checks and is done to provide immediate action if an event handler submits commands to Nagios.

Question 143:

What is the difference between Active and Passive check in Nagios?

Answer:

The major difference between Active and Passive checks is that Active checks are initiated and performed by Nagios, while passive checks are performed by external applications.

Passive checks are useful for monitoring services that are:

the work. This design takes advantage of Nagios's ability to utilize passive checks i.e. external applications or processes that send results back to Nagios. In a distributed configuration, these external applications are other instances of Nagios.

Question 145:

Explain Main Configuration file of Nagios and its location?

Answer:

The main configuration file contains a number of directives that affect how the Nagios daemon operates. This config file is read by both the Nagios daemon and the CGIs (It specifies the location of your main configuration file).

Now you can tell where it is present and how it is created. A sample main configuration file is created in the base directory of the Nagios distribution when you run the configure script. The default name of the main configuration file is nagios.cfg. It is usually placed in the etc/ subdirectory of your Nagios installation (i.e. /usr/local/nagios/etc/).

Question 146:

Explain how Flap Detection works in Nagios?

Answer:

Flapping occurs when a service or host changes state too frequently, this causes lot of problem and recovery notifications.

Whenever Nagios checks the status of a host or service, it will check to see if it has started or stopped flapping. Nagios follows the below given procedure to do that:

- Storing the results of the last 21 checks of the host or service analyzing the historical check results and determine where state changes transitions occur
- Using the state transitions to determine a percent state change value (a measure of change) for the host or service
- Comparing the percent state change value against low and high flapping thresholds.

A host or service is determined to have started flapping when it's percent

state change first exceeds a high flapping threshold. A host or service is determined to have stopped flapping when its percent state goes below a low flapping threshold.

Question 147:

What are the three main variables that affect recursion and inheritance in Nagios?

Answer:

The variables are:

- Name
- Use
- Register

Name is a placeholder that is used by other objects. Use defines the "parent" object whose properties should be used. Register can have a value of 0 (indicating it's only a template) and 1 (an actual object). The register value is never inherited.

Question 148:

What is meant by saying Nagios is Object Oriented?

Answer:

"One of the features of Nagios is object configuration format in that you can create object definitions that inherit properties from other object definitions and hence the name. This simplifies and clarifies relationships between various components."

Question 149:

What is State Stalking in Nagios?

Answer:

State Stalking is used for logging purposes. When Stalking is enabled for a particular host or service, Nagios will watch that host or service very carefully and log any changes it sees in the output of check results.

It can be very helpful in later analysis of the log files. Under normal circumstances, the result of a host or service check is only logged if the host or service has changed state since it was last checked.

Question 150:

Explain how the implementation of "Infrastructure as code" is processed or executed in terms of AWS.

Answer:

- The code will be in the simple JSON format.
- This JSON code is well organized into files called templates.
- These templates are deployed on AWS and then further managed as stacks.
- Cloud Formation service will help in doing the Creating, deleting, updating, etc. operation in the stack.

Question 151:

What is Dark Launch?

Answer:

It is a go-live strategy in which code implementing new features is released to a subset of the production environment but is not visibly, or only partially, activated. The code is exercised, however, in a production setting without

users being aware of it.

Question 152:

What are the AWS Services for DevOps?

Answers:

AWS Code Commit: AWS CodeCommit is a fully managed source control service that makes it easy for companies to host secure and highly scalable private Git repositories. CodeCommit eliminates the need to operate your own source control system or worry about scaling its infrastructure. You can use CodeCommit to securely store anything from source code to binaries, and it works seamlessly with your existing Git tools.

AWS CodePipeline: AWS CodePipeline is a continuous integration and continuous delivery service for fast and reliable application and infrastructure updates. CodePipeline builds, tests, and deploys your code every time there is a code change, based on the release process models you define. This enables you to rapidly and reliably deliver features and updates. You can easily build out an end-to-end solution by using our pre-built plugins for popular

third-party services like GitHub or integrating your own custom plugins into any stage of your release process.

AWS CodeBuild: AWS CodeBuild is a fully managed build service that compiles source code, runs tests, and produces software packages that are ready to deploy. With CodeBuild, you don't need to provision, manage, and scale your own build servers. CodeBuild scales continuously and processes multiple builds concurrently, so your builds are not left waiting in a queue. You can get started quickly by using prepackaged build environments, or you can create custom build environments that use your own build tools. With CodeBuild, you are charged by the minute for the compute resources you use.

AWS CodeDeploy : AWS CodeDeploy is a service that automates code deployments to any instance, including Amazon EC2 instances and servers running on-premises. AWS CodeDeploy makes it easier for you to rapidly release new features, helps you avoid downtime during application deployment, and handles the complexity of updating your applications. You can use AWS CodeDeploy to automate software deployments, eliminating the need for

error-prone manual operations, and the service scales with your infrastructure so you can easily deploy to one instance or thousands.