

DevOps Interview Questions

General DevOps Interview Questions

This category will include questions that are not related to any particular DevOps stage. Questions here are meant to test your understanding about DevOps rather than focusing on a particular tool or a stage.

Q1. What is the need for DevOps?

According to me, this answer should start by explaining the general market trend. 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.

If I have to test your knowledge on DevOps, you should know the difference between Agile and DevOps. The next question is directed towards that.

Q2. How is DevOps different from Agile / SDLC?

I would advise you to go with the below explanation:

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 repeatably 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. Here's a blog that will give you more information on the evolution of DevOps.

Now remember, you have included DevOps tools in your previous answer so be prepared to answer some questions related to that.

Q3. Which are the top DevOps tools? Which tools have you worked on?

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. 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 those softwares 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.

Q4. How do all these tools work together?

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.

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 build 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 sends 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 features.

Q5. What are the advantages of DevOps?

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
- Less complex problems to fix
- Faster resolution of problems

Business benefits:

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

Q6. What is the most important thing DevOps helps us achieve?

According to me, 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. Learn more in this [DevOps tutorial](#) blog. 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.

Q7. Explain with a use case where DevOps can be used in industry / real-life.

There are many industries that are using DevOps so you can mention any of those use cases, you can also 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 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.

Q8. 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.

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 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.

Q9. What are the anti-patterns of DevOps?

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

Version Control System (VCS) Interview Questions

Now let's look at interview questions on VCS:

Q1. What is Version control?

This is probably the easiest question you will face in the interview. My suggestion is to first give a definition of Version control. 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.
- Who introduced an issue and when.

Q2. What are the benefits of using version control?

I will suggest you to include the following advantages of 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. 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 repository.

Q3. Describe branching strategies you have used.

This question is asked to test your branching experience so tell them about how you have used branching in your previous job and what purpose does it serve, you can refer the below points:

- 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.
- 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.

In the end tell them that branching strategies varies from one organization to another, so I know basic branching operations like delete, merge, checking out a branch etc.

Q4. Which VCS tool you are comfortable with?

You can just mention the VCS tool that you have worked on like this: "I have worked on Git and one major advantage it has over other VCS tools like SVN is that it is a distributed version control system." Distributed VCS tools do not necessarily rely on a central server to store all the versions of a project's files. Instead, every developer "clones" a copy of a repository and has the full history of the project on their own hard drive.

Q5. What is Git?

I will suggest that you attempt this question by first explaining about the architecture of git as shown in the below diagram. You can refer to the explanation given below:

- 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 I have shown in the diagram below with "Local repository" and 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 repository.
- There is a central cloud repository as well where developers can commit changes and share it with other teammates as you can see in the diagram where all collaborators are committing changes "Remote repository".

Q6. Explain some basic Git commands?

Below are some basic Git commands:

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 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

Q7. In Git how do you revert a commit that has already been pushed and made public?

There can be two answers to this question so make sure that you include both because any of the below options can be used depending on the situation:

- Remove or fix the bad file in a new commit and push it to the remote repository. This is the most natural way to fix an error. Once you have made necessary changes to the file, commit it to the remote repository for that I will use **git commit -m "commit message"**
- Create a new commit that undoes all changes that were made in the bad commit.to do this I will use a command **git revert <name of bad commit>**

Q8. How do you squash last N commits into a single commit?

There are two options to squash last N commits into a single commit. Include both of the below mentioned options in your answer:

- If you want to write the new commit message from scratch use the following command
**git reset -soft HEAD~N &&
git commit**
- If you want to start editing the new commit message with a concatenation of the existing commit messages then you need to extract those messages and pass them to Git commit for that I will use
**git reset -soft HEAD~N &&
git commit --edit -m "\$(git log --format=%B --reverse
.HEAD@{N})"**

Q9. What is Git bisect? How can you use it to determine the source of a (regression) bug?

I will suggest you to first give a small definition of Git bisect, Git bisect is used to find the commit that introduced a bug by using binary search.

Command for Git bisect is

git bisect <subcommand> <options>

Now since you have mentioned the command above, explain what this command will do, This command uses a binary search algorithm to find which commit in your project's history introduced a bug. You use it by first telling it a "bad" commit that is known to contain the bug, and a "good" commit that is known to be before the bug was introduced. Then Git bisect picks a commit between those two endpoints and asks you whether the selected commit is "good" or "bad". It continues narrowing down the range until it finds the exact commit that introduced the change.

Q10. What is Git rebase and how can it be used to resolve conflicts in a feature branch before merge?

According to me, you should start by saying git rebase is a command which will merge another branch into the branch where you are currently working, and move all of the local commits that are ahead of the rebased branch to the top of the history on that branch. Now once you have defined Git rebase time for an example to show how it can be used to resolve conflicts in a feature branch before merge, if a feature branch was created from master, and since then the master branch has received new commits, Git rebase can be used to move the feature branch to the tip of master.

The command effectively will replay the changes made in the feature branch at the tip of master, allowing conflicts to be resolved in the process. When done with care, this will allow the feature branch to be merged into master with relative ease and sometimes as a simple fast-forward operation.

Q11. How do you configure a Git repository to run code sanity checking tools right before making commits, and preventing them if the test fails?

I will suggest you to first give a small introduction to sanity checking, A sanity or smoke test determines whether it is possible and reasonable to continue testing.

Now explain how to achieve this, this can be done with a simple script related to the pre-commit hook of the repository. The pre-commit hook is triggered right before a commit is made, even before you are required to enter a commit message. In this script one can run other tools, such as linters and perform sanity checks on the changes being committed into the repository.

Finally give an example, you can refer the below script:

```
#!/bin/sh
files=$(git diff --cached --name-only --diff-filter=ACM | grep '.go$')
if [ -z files ]; then
exit 0
fi
unfmt=$(gofmt -l $files)
if [ -z unfmt ]; then
exit 0
fi
echo "Some .go files are not fmt'd"
exit 1
```

This script checks to see if any .go file that is about to be committed needs to be passed through the standard Go source code formatting tool gofmt. By exiting with a non-zero status, the script effectively prevents the commit from being applied to the repository.

Q12. How do you find a list of files that has changed in a particular commit?

For this answer instead of just telling the command, explain what exactly this command will do so you can say that, To get a list files that has changed in a particular commit use command

git diff-tree -r {hash}

Given the commit hash, this will list all the files that were changed or added in that commit. The -r flag makes the command list individual files, rather than collapsing them into root directory names only.

You can also include the below mention point although it is totally optional but will help in impressing the interviewer.

The output will also include some extra information, which can be easily suppressed by including two flags:

git diff-tree --no-commit-id --name-only -r {hash}

Here --no-commit-id will suppress the commit hashes from appearing in the output, and --name-only will only print the file names, instead of their paths.

Q13. How do you setup a script to run every time a repository receives new commits through push?

There are three ways to configure a script to run every time a repository receives new commits through push, one needs to define either a pre-receive, update, or a post-receive hook depending on when exactly the script needs to be triggered.

- Pre-receive hook in the destination repository is invoked when commits are pushed to it. Any script bound to this hook will be executed before any references are updated. This is a useful hook to run scripts that help enforce development policies.
- Update hook works in a similar manner to pre-receive hook, and is also triggered before any updates are actually made. However, the update hook is called once for every commit that has been pushed to the destination repository.
- Finally, post-receive hook in the repository is invoked after the updates have been accepted into the destination repository. This is an ideal place to configure simple deployment scripts, invoke some continuous integration systems, dispatch notification emails to repository maintainers, etc.

Hooks are local to every Git repository and are not versioned. Scripts can either be created within the hooks directory inside the ".git" directory, or they can be created elsewhere and links to those scripts can be placed within the directory.

Q14. How will you know in Git if a branch has already been merged into master?

I will suggest you to include both the below mentioned commands:
git branch -merged lists the branches that have been merged into the current branch.

git branch -no-merged lists the branches that have not been merged.

Continuous Integration questions

Now, let's look at Continuous Integration interview questions:

Q1. What is meant by Continuous Integration?

I will advise you to begin this answer by giving a small definition of Continuous Integration (CI). It is a development practice that requires developers to integrate code into a shared repository several times a day. Each check-in is then verified by an automated build, allowing teams to detect problems early.

I suggest that you explain how you have implemented it in your previous job. You can refer the below given example:

1. Developers check out code into their private workspaces.
2. When they are done with it they commit the changes to the shared repository (Version Control Repository).
3. The CI server monitors the repository and checks out changes when they occur.
4. The CI server then pulls these changes and builds the system and also runs unit and integration tests.
5. The CI server will now inform the team of the successful build.
6. If the build or tests fails, the CI server will alert the team.
7. The team will try to fix the issue at the earliest opportunity.
8. This process keeps on repeating.

Q2. Why do you need a Continuous Integration of Dev & Testing?

For this answer, you should focus on the need of Continuous Integration. My suggestion would be to mention the below explanation in your answer: Continuous Integration of Dev and Testing improves the quality of

software, and reduces the time taken to deliver it, by replacing the traditional practice of testing after completing all development. It allows Dev team to easily detect and locate problems early because developers need to integrate code into a shared repository several times a day (more frequently). Each check-in is then automatically tested.

Q3. What are the success factors for Continuous Integration?

Here you have to mention the requirements for Continuous Integration. You could include the following points in your answer:

- Maintain a code repository
- Automate the build
- Make the build self-testing
- Everyone commits to the baseline every day
- Every commit (to baseline) should be built
- Keep the build fast
- Test in a clone of the production environment
- Make it easy to get the latest deliverables
- Everyone can see the results of the latest build
- Automate deployment

Q4. Explain how you can move or copy Jenkins from one server to another?

I will approach this task by copying the jobs directory from the old server to the new one. There are multiple ways to do that; I have mentioned them below:

You can:

- Move a job from one installation of Jenkins to another by simply copying the corresponding job directory.
- Make a copy of an existing job by making a clone of a job directory by a different name.
- Rename an existing job by renaming a directory. Note that if you change a job name you will need to change any other job that tries to call the renamed job.

Q5. Explain how can create a backup and copy files in Jenkins?

Answer to this question is really direct. To create a backup, all you need to do is to periodically back up your JENKINS_HOME directory. This contains all of your build jobs configurations, your slave node configurations, and your build history. To create a back-up of your Jenkins

setup, just copy this directory. You can also copy a job directory to clone or replicate a job or rename the directory.

Q6. Explain how you can setup Jenkins job?

My approach to this answer will be to first mention how to create Jenkins job. Go to Jenkins top page, select “New Job”, then choose “Build a free-style software project”.

Then you can tell the elements of this freestyle job:

- Optional SCM, such as CVS or Subversion where your source code resides.
- Optional triggers to control when Jenkins will perform builds.
- Some sort of build script that performs the build (ant, maven, shell script, batch file, etc.) where the real work happens.
- Optional steps to collect information out of the build, such as archiving the artifacts and/or recording javadoc and test results.
- Optional steps to notify other people/systems with the build result, such as sending e-mails, IMs, updating issue tracker, etc..

Q7. Mention some of the useful plugins in Jenkins.

Below, I have mentioned some important Plugins:

- Maven 2 project
- Amazon EC2
- HTML publisher
- Copy artifact
- Join
- Green Balls

These Plugins, I feel are the most useful plugins. If you want to include any other Plugin that is not mentioned above, you can add them as well. But, make sure you first mention the above stated plugins and then add your own.

Q8. How will you secure Jenkins?

The way I secure Jenkins is mentioned below. If you have any other way of doing it, please mention it in the comments section below:

- Ensure global security is on.
- Ensure that Jenkins is integrated with my company’s user directory with appropriate plugin.
- Ensure that matrix/Project matrix is enabled to fine tune access.

- Automate the process of setting rights/privileges in Jenkins with custom version controlled script.
- Limit physical access to Jenkins data/folders.
- Periodically run security audits on same.

Continuous Testing Interview Questions:

Now let's move on to the Continuous Testing questions.

Q1. What is Continuous Testing?

I will advise you to follow the below mentioned explanation: 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 in 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.

Q2. What is Automation Testing?

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.

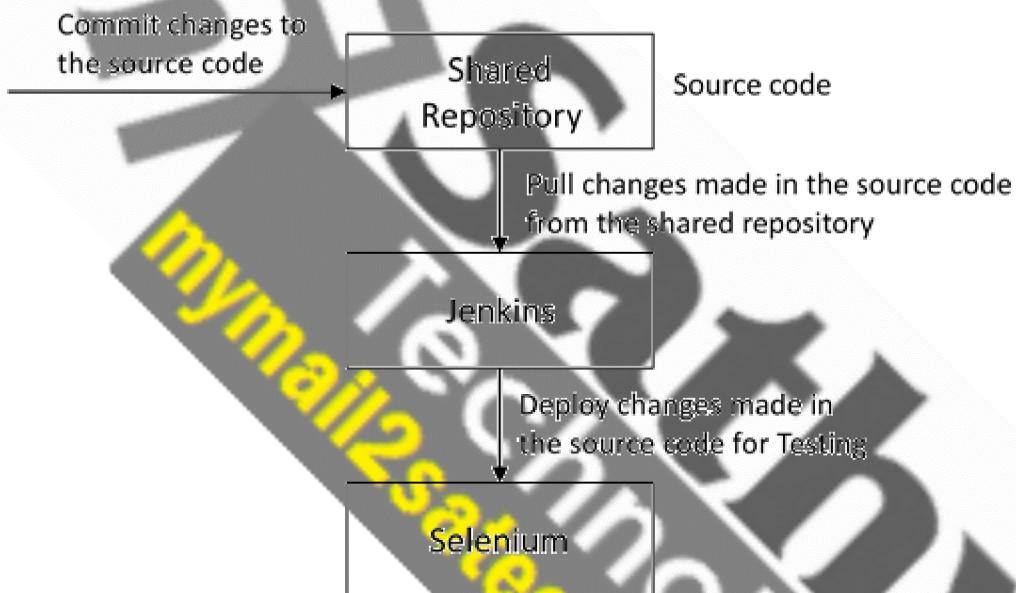
Q3. What are the benefits of Automation Testing?

I have listed down some advantages of automation testing. Include these in your answer and you can add your own experience of how Continuous Testing helped your previous company:

- 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

Q4. How to automate Testing in DevOps lifecycle?

I have mentioned a generic flow below which you can refer to: 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.



Q5. Why is Continuous Testing important for DevOps?

You can answer this question by saying, "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."

Q6. What are the key elements of Continuous Testing tools?

Key elements of Continuous Testing are:

- **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.

- **Policy Analysis:** It ensures all processes align with the organization's evolving business and compliance demands are met.
- **Requirements Traceability:** It ensures true requirements are met and rework is not required. An object assessment is used to identify which requirements are at risk, working as expected or require further validation.
- **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.
- **Test Optimization:** It ensures tests yield accurate outcomes and provide actionable findings. Aspects include Test Data Management, Test Optimization Management and Test Maintenance
- **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.

Q7. Which Testing tool are you comfortable with and what are the benefits of that tool?

Here mention the testing tool that you have worked with and accordingly frame your answer. I have mentioned an example below: I have worked on Selenium to ensure high quality and more frequent releases.

Some advantages of Selenium are:

- It is free and open source
- 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

Q8. What are the Testing types supported by Selenium?

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 points) individually.

Q9. What is Selenium IDE?

My suggestion is to start this answer by defining Selenium IDE. It is an integrated development environment for Selenium scripts. It is implemented as a Firefox extension, and allows you to record, edit, and debug tests. Selenium IDE includes the entire Selenium Core, allowing you to easily and quickly record and play back tests in the actual environment that they will run in.

Now include some advantages in your answer. With autocomplete support and the ability to move commands around quickly, Selenium IDE is the ideal environment for creating Selenium tests no matter what style of tests you prefer.

Q10. What is the difference between Assert and Verify commands in Selenium?

I have mentioned differences between Assert and Verify commands below:

- Assert command checks whether the given condition is true or false. Let's say we assert whether the given element is present on the web page or not. If the condition is true, then the program control will execute the next test step. But, if the condition is false, the execution would stop and no further test would be executed.
- Verify command also checks whether the given condition is true or false. Irrespective of the condition being true or false, the program execution doesn't halt i.e. any failure during verification would not stop the execution and all the test steps would be executed.

Q11. How to launch Browser using WebDriver?

The following syntax can be used to launch Browser:

```
WebDriver driver = new FirefoxDriver();
WebDriver driver = new ChromeDriver();
WebDriver driver = new InternetExplorerDriver();
```

Q12. When should I use Selenium Grid?

For this answer, my suggestion would be to give a small definition of Selenium Grid. It can be used to execute same or different test scripts on multiple platforms and browsers concurrently to achieve distributed test execution. This allows testing under different environments and saving execution time remarkably.

Configuration Management Interview Questions

Now let's check how much you know about Configuration Management.

Q1. What are the goals of Configuration management processes?

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.

Q2. What is the difference between Asset management and Configuration Management?

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

Q3. What is the difference between an Asset and a Configuration Item?

According to me, you should first explain Asset. It has a financial value along with a depreciation rate attached to it. IT assets are just a sub-set of it. Anything and everything that has a cost and the organization uses it for its asset value calculation and related benefits in tax calculation falls under Asset Management, and such item is called an asset. Configuration Item on the other hand may or may not have financial values assigned to it. It will not have any depreciation linked to it. Thus, its life would not be dependent on its financial value but will depend on the time till that item becomes obsolete for the organization.

Now you can give an example that can showcase the similarity and differences between both:

1) Similarity:

Server – It is both an asset as well as a CI.

2) Difference:

Building – It is an asset but not a CI.

Document – It is a CI but not an asset

Q4. What do you understand by “Infrastructure as code”? How does it fit into the DevOps methodology? What purpose does it achieve?

Infrastructure as Code (IAC) is a type of IT infrastructure that operations teams can use to automatically manage and provision through code, rather than using a manual process. Companies for faster deployments treat infrastructure like software: as code that can be managed with the DevOps tools and processes. These tools let you make infrastructure changes more easily, rapidly, safely and reliably.

Q5. Which among Puppet, Chef, SaltStack and Ansible is the best Configuration Management (CM) tool? Why?

This depends on the organization's need so mention few points on all those tools:

Puppet is the oldest and most mature CM tool. Puppet is a Ruby-based Configuration Management tool, but while it has some free features, much

of what makes Puppet great is only available in the paid version. Organizations that don't need a lot of extras will find Puppet useful, but those needing more customization will probably need to upgrade to the paid version.

Chef is written in Ruby, so it can be customized by those who know the language. It also includes free features, plus it can be upgraded from open source to enterprise-level if necessary. On top of that, it's a very flexible product.

Ansible is a very secure option since it uses Secure Shell. It's a simple tool to use, but it does offer a number of other services in addition to configuration management. It's very easy to learn, so it's perfect for those who don't have a dedicated IT staff but still need a configuration management tool.

SaltStack is python based open source CM tool made for larger businesses, but its learning curve is fairly low.

Q6. What is Puppet?

I will advise you to first give a small definition of Puppet. 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 as shown on the diagram below. Puppet Slave sends request to Puppet Master and Puppet Master then pushes configuration on Slave.

Q7. Before a client can authenticate with the Puppet Master, its certs need to be signed and accepted. How will you automate this task?

The easiest way is to enable auto-signing in puppet.conf. Do mention that this is a security risk. If you still want to do this:

- Firewall your puppet master - restrict port tcp/8140 to only networks that you trust.

- Create puppet masters for each ‘trust zone’, and only include the trusted nodes in that Puppet masters manifest.
- Never use a full wildcard such as *.

Q8. Describe the most significant gain you made from automating a process through Puppet.

For this answer, I will suggest you to explain your past experience with Puppet. You can refer the below example: I automated the configuration and deployment of Linux and Windows machines using Puppet. In addition to shortening the processing time from one week to 10 minutes, I used the roles and profiles pattern and documented the purpose of each module in README to ensure that others could update the module using Git. The modules I wrote are still being used, but they've been improved by my teammates and members of the community

Q9. Which open source or community tools do you use to make Puppet more powerful?

Over here, you need to mention the tools and how you have used those tools to make Puppet more powerful. Below is one example for your reference:

Changes and requests are ticketed through Jira and we manage requests through an internal process. Then, we use Git and Puppet's Code Manager app to manage Puppet code in accordance with best practices. Additionally, we run all of our Puppet changes through our continuous integration pipeline in Jenkins using the beaker testing framework.

Q10. What are Puppet Manifests?

It is a very important question so make sure you go in a correct flow. According to me, you should first define Manifests. Every node (or Puppet Agent) has got its configuration details in Puppet Master, written in the native Puppet language. These details are written in the language which Puppet can understand and are termed as Manifests. They are composed of Puppet code and their filenames use the .pp extension. Now give an example. You can write a manifest in Puppet Master that creates a file and installs apache on all Puppet Agents (Slaves) connected to the Puppet Master.

Q11. What is Puppet Module and How it is different from Puppet Manifest?

For this answer, you can go with the below mentioned explanation: A Puppet Module is a collection of Manifests and data (such as facts, files, and templates), and they have a specific directory structure. Modules are useful for organizing your Puppet code, because they allow you to split your code into multiple Manifests. It is considered best practice to use Modules to organize almost all of your Puppet Manifests. Puppet programs are called Manifests which are composed of Puppet code and their file names use the .pp extension.

Q12. What is Facter in Puppet?

You are expected to answer what exactly Facter does in Puppet so according to me, you should say, "Facter gathers basic information (facts) about Puppet Agent such as hardware details, network settings, OS type and version, IP addresses, MAC addresses, SSH keys, and more. These facts are then made available in Puppet Master's Manifests as variables."

Q13. What is Chef?

Begin this answer by defining Chef. 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. What processes? Pretty much anything related to IT. 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.

Q14. What is a resource in Chef?

My suggestion is to first define Resource. A Resource represents a piece of infrastructure and its desired state, such as a package that should be installed, a service that should be running, or a file that should be generated.

You should explain about the functions of Resource for that include the following points:

- Describes the desired state for a configuration item.
- Declares the steps needed to bring that item to the desired state.
- Specifies a resource type such as package, template, or service.
- Lists additional details (also known as resource properties), as necessary.
- Are grouped into recipes, which describe working configurations.

Q15. What do you mean by recipe in Chef?

For this answer, I will suggest you to use the above mentioned flow: first define Recipe. A Recipe is a collection of Resources that describes a particular configuration or policy. A Recipe describes everything that is required to configure part of a system. After the definition, explain the functions of Recipes by including the following points:

- Install and configure software components.
- Manage files.
- Deploy applications.
- Execute other recipes.

Q16. How does a Cookbook differ from a Recipe in Chef?

The answer to this is pretty direct. You can simply say, "a Recipe is a collection of Resources, and primarily configures a software package or some piece of infrastructure. A Cookbook groups together Recipes and other information in a way that is more manageable than having just Recipes alone."

Q17. What happens when you don't specify a Resource's action in Chef?

My suggestion is to first give a direct answer: when you don't specify a resource's action, Chef applies the default action.

Now explain this with an example, the below resource:

```
file 'C:\Users\Administrator\chef-repo\settings.ini' do
  content 'greeting=hello world'
end
```

is same as the below resource:

```
file 'C:\Users\Administrator\chef-repo\settings.ini' do
  action :create
  content 'greeting=hello world'
end
```

because: create is the file Resource's default action.

Q18. What is Ansible module?

Modules are considered to be the units of work in Ansible. Each module is mostly standalone and can be written in a standard scripting language such as Python, Perl, Ruby, bash, etc.. One of the guiding properties of modules is idempotency, which means that even if an operation is repeated multiple times e.g. upon recovery from an outage, it will always place the system into the same state.

Q19. What are playbooks in Ansible?

Playbooks are Ansible's configuration, deployment, and orchestration language. They can describe a policy you want your remote systems to enforce, or a set of steps in a general IT process. Playbooks are designed to be human-readable and are developed in a basic text language. At a basic level, playbooks can be used to manage configurations of and deployments to remote machines.

Q20. How do I see a list of all of the ansible_ variables?

Ansible by default gathers "facts" about the machines under management, and these facts can be accessed in Playbooks and in templates. To see a list of all of the facts that are available about a machine, you can run the "setup" module as an ad-hoc action:

```
Ansible           -m           setup           hostname
```

This will print out a dictionary of all of the facts that are available for that particular host.

Q21. How can I set deployment order for applications?

WebLogic Server 8.1 allows you to select the load order for applications. See the Application MBean Load Order attribute in Application. WebLogic Server deploys server-level resources (first JDBC and then JMS) before deploying applications. Applications are deployed in this order: connectors, then EJBs, then Web Applications. If the application is an EAR, the individual components are loaded in the order in which they are declared in the application.xml deployment descriptor.

Q22. Can I refresh static components of a deployed application without having to redeploy the entire application?

Yes, you can use weblogic.Deployer to specify a component and target a server, using the following syntax:

```
java weblogic.Deployer -adminurl http://admin:7001 -name appname -targets server1,server2 -deploy jsps/*.jsp
```

Q23. How do I turn the auto-deployment feature off?

The auto-deployment feature checks the applications folder every three seconds to determine whether there are any new applications or any changes to existing applications and then dynamically deploys these changes.

The auto-deployment feature is enabled for servers that run in development mode. To disable auto-deployment feature, use one of the following methods to place servers in production mode:

- In the Administration Console, click the name of the domain in the left pane, then select the Production Mode checkbox in the right pane.
- At the command line, include the following argument when starting the domain's Administration Server:
-Dweblogic.ProductionModeEnabled=true
- Production mode is set for all WebLogic Server instances in a given domain.

Q24. When should I use the external_stage option?

Set -external_stage using weblogic.Deployer if you want to stage the application yourself, and prefer to copy it to its target by your own means.

Continuous Monitoring Interview Questions

Let's test your knowledge on Continuous Monitoring.

Q1. Why is Continuous monitoring necessary?

I will suggest you to go with the below mentioned flow: Continuous Monitoring allows timely identification of problems or weaknesses and quick corrective action that helps reduce expenses of an organization. Continuous monitoring provides solution that addresses three operational disciplines known as:

- continuous audit
- continuous controls monitoring
- continuous transaction inspection

Q2. What is Nagios?

You can answer this question by first mentioning that Nagios is one of the monitoring tools. It is used for Continuous monitoring 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 affect your organization's bottom line.

Now once you have defined what is Nagios, 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.
- Ensure IT infrastructure outages have a minimal effect on your organization's bottom line.
- Monitor your entire infrastructure and business processes.

This completes the answer to this question. Further details like advantages etc. can be added as per the direction where the discussion is headed.

Q3. How does Nagios works?

I will advise you to follow the below explanation for this answer: Nagios runs on a server, usually as a daemon or service. Nagios periodically runs plugins 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.

Now expect a few questions on Nagios components like Plugins, NRPE etc..

Q4. What are Plugins in Nagios?

Begin this answer by defining Plugins. They are scripts (Perl scripts, Shell scripts, etc.) that can run from a command line to check the status of a host or service. Nagios uses the results from Plugins to determine the current status of hosts and services on your network. Once you have defined Plugins, explain why we need Plugins. Nagios will execute a Plugin whenever there is a need to check the status of a host or service. Plugin will perform the check and then simply returns the result to Nagios. Nagios will process the results that it receives from the Plugin and take the necessary actions.

Q5. What is NRPE (Nagios Remote Plugin Executor) in Nagios?

For this answer, give a brief definition of Plugins. The NRPE addon 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.

I will advise you to explain the NRPE architecture on the basis of diagram shown below. The NRPE addon 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.

Q6. What do you mean by passive check in Nagios?

According to me, the answer should start by explaining Passive checks. They are initiated and performed by external applications/processes and the Passive check results are submitted to Nagios for processing. Then explain the need for passive checks. They are useful for monitoring services that are Asynchronous in nature and cannot be monitored effectively by polling their status on a regularly scheduled basis. They can also be used for monitoring services that are Located behind a firewall and cannot be checked actively from the monitoring host.

Q7. When Does Nagios Check for external commands?

Make sure that you stick to the question during your explanation so I will advise you to follow the below mentioned flow. 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.

Q8. What is the difference between Active and Passive check in Nagios?

For this answer, first point out the basic difference Active and Passive checks. 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. If your interviewer is looking unconvinced with the above explanation then you can also mention some key features of both Active and Passive checks:

Passive checks are useful for monitoring services that are:

- 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 Actives checks are as follows:

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

Q9. How does Nagios help with Distributed Monitoring?

The interviewer will be expecting an answer related to the distributed architecture of Nagios. So, I suggest that you answer it in the below mentioned format:

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 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.

Q10. Explain Main Configuration file of Nagios and its location?

First mention what this main configuration file contains and its function. 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/).

Q11. Explain how Flap Detection works in Nagios?

I will advise you to first explain Flapping first. Flapping occurs when a service or host changes state too frequently, this causes lot of problem

and recovery notifications.

Once you have defined Flapping, explain how Nagios detects Flapping. 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 its 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.

Q12. What are the three main variables that affect recursion and inheritance in Nagios?

According to me the proper format for this answer should be: First name the variables and then a small explanation of each of these variables:

- Name
- Use
- Register

Then give a brief explanation for each of these variables. 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 its only a template) and 1 (an actual object). The register value is never inherited.

Q13. What is meant by saying Nagios is Object Oriented?

Answer to this question is pretty direct. I will answer this by saying, “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.”

Q14. What is State Stalking in Nagios?

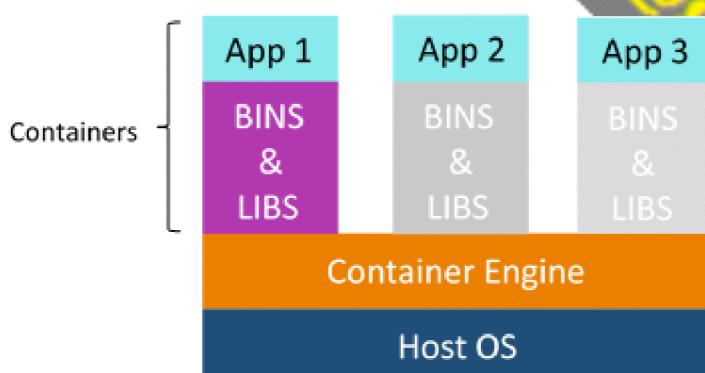
I will advise you to first give a small introduction on State Stalking. It is used for logging purposes. When Stalking is enabled for a particular host or service, Nagios will watch that host or service very carefully and log any changes it sees in the output of check results. Depending on the discussion between you and interviewer you can also add, “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.”

Containerization and Virtualization Interview Questions

Let's see how much you know about containers and VMs.

Q1. What are containers?

My suggestion is to explain the need for containerization first, containers are used to provide consistent computing environment from a developer's laptop to a test environment, from a staging environment into production. Now give a definition of containers, a 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.



Q2. What are the advantages that Containerization provides over virtualization?

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

Q3. How exactly are containers (Docker in our case) different from hypervisor virtualization (vSphere)? What are the benefits?

Given below are some differences. Make sure you include these differences in your answer:

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

Q4. What is Docker image?

I suggest that you go with the below mentioned flow: Docker image is the source of Docker container. In other words, Docker images are used to create containers. Images are created with the build command, and they'll produce a container when started with run. Images are stored in a Docker registry such as registry.hub.docker.com because they can become quite large, images are designed to be composed of layers of other images, allowing a minimal amount of data to be sent when transferring images over the network.

Tip: Be aware of Dockerhub in order to answer questions on pre-available images.

Q5. What is Docker container?

This is a very important question so just make sure you don't deviate from the topic. I advise you to follow the below mentioned format: 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 explain how to create a Docker container, Docker containers can be created by either creating a Docker image and then running it or you can use Docker images that are present on the Dockerhub. Docker containers are basically runtime instances of Docker images.

Q6. What is Docker hub?

Answer to this question is pretty direct. 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.

Q7. How is Docker different from other container technologies?

According to me, below points should be there in your answer: Docker containers are easy to deploy in a cloud. It can get more applications running on the same hardware than other technologies, it makes it easy for developers to quickly create, ready-to-run containerized applications and it makes managing and deploying applications much easier. You can even share containers with your applications. If you have some more points to add you can do that but make sure the above the above explanation is there in your answer.

Q8. What is Docker Swarm?

You should start this answer by explaining Docker Swarm. 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

Q9. What is Dockerfile used for?

This answer according to me should begin by explaining the use of Dockerfile. Docker can build images automatically by reading the instructions from a Dockerfile.

Now I suggest you to give a small definition of Dockerfile. A Dockerfile is a text document that contains all the commands a user could call on the command line to assemble an image. Using docker build users can create an automated build that executes several command-line instructions in succession.

Now expect a few questions to test your experience with Docker.

Q10. Can I use json instead of yaml for my compose file in Docker?

You can use json instead of yaml for your compose file, to use json file with compose, specify the filename to use for eg:
docker-compose -f docker-compose.json up

Q11. Tell us how you have used Docker in your past position?

Explain how you have used Docker to help rapid deployment. Explain how you have scripted Docker and used Docker with other tools like Puppet, Chef or Jenkins. If you have no past practical experience in Docker and have past experience with other tools in similar space, be honest and explain the same. In this case, it makes sense if you can compare other tools to Docker in terms of functionality.

Q12. How to create Docker container?

I will suggest you to give a direct answer to this. We can use Docker image to create Docker container by using the below command:

docker run -t -i <image name> <command name>

This command will create and start container.

You should also add, If you want to check the list of all running container with status on a host use the below command:

docker ps -a

Q13. How to stop and restart the Docker container?

In order to stop the Docker container you can use the below command:

docker stop <container ID>

Now to restart the Docker container you can use:

docker restart <container ID>

Q14. How far do Docker containers scale?

Large web deployments like Google and Twitter, and platform providers such as Heroku and dotCloud all run on container technology, at a scale of hundreds of thousands or even millions of containers running in parallel.

Q15. What platforms does Docker run on?

I will start this answer by saying Docker runs on only Linux and Cloud platforms and then I will mention the below vendors of Linux:

- Ubuntu 12.04, 13.04 et al
- Fedora 19/20+
- RHEL 6.5+
- CentOS 6+
- Gentoo
- ArchLinux
- openSUSE 12.3+
- CRUX 3.0+

Cloud:

- Amazon EC2
- Google Compute Engine
- Microsoft Azure
- Rackspace

Note that Docker does not run on Windows or Mac.

Q16. Do I lose my data when the Docker container exits?

You can answer this by saying, no I won't lose my data when Docker container exits. Any data that your application writes to disk gets preserved in its container until you explicitly delete the container. The file system for the container persists even after the container halts.

Additional Questions:

1. How does HTTP work?

The HTTP protocol works in a client and server model like most other protocols. A web browser using which a request is initiated is called as a client and a web server software which responds to that request is called a server. World Wide Web Consortium and the Internet Engineering Task Force are two important spokes in the standardization of the HTTP protocol. HTTP allows improvement of its request and response with the help of intermediates, for example a gateway, a proxy, or a tunnel. The resources that can be requested using the HTTP protocol, are made available using a certain type of URI (Uniform Resource Identifier) called a URL (Uniform Resource Locator). TCP (Transmission Control Protocol) is used to establish a connection to the application layer port 80 used by HTTP.

2. Explain your understanding and expertise on both the software development side and the technical operations side of an organization you've worked for in the past.

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

3. Discuss your experience building bridges between IT Ops, QA and development.

DevOps is all about effective communication and collaboration. I've been able to deal with production issues from the development and operations

sides, effectively straddling the two worlds. I'm less interested in finding blame or playing the hero than I am with ensuring that all of the moving parts come together.

4. What types of testing are needed?

Software teams will often look for the "fair weather" path to system completion; that is, they start from an assumption that software will usually work and only occasionally fail. I believe to practice defensive programming in a pragmatic way, which often means assuming that the code will fail and planning for those failures. I try to incorporate unit test strategy, use of test harnesses, early load testing; network simulation, A/B and multi-variate testing etc.

5. Give me an example of how you would handle projects?

As a professional with managerial responsibilities, 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, translate requirements into workflow and follow up. I would resort to CI, release management and other tools to keep interdisciplinary projects on track.

6. What's your career objective in your role as a DevOps engineer?

My passion is breaking down the barriers and building and improving processes, so that the engineering and operations teams work better and smarter. That's why I love DevOps. It's an opportunity to be involved in the entire delivery system from start to finish.

7. How would you make software deployable?

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, command-line tools, and infrastructure as code.

8. What is the one most important thing DevOps helps do?

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.

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

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.

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

I believe I'll be expected to:

1. Focus attention on bridging communication gaps between Development and Operations teams.
2. 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.
3. Execute – to be able to actually do what needs to be done.

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

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.

12. What's a PTR in DNS?

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.

13. Describe two-factor authentication?

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.

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

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.

15. What are the advantages of NoSQL database over RDBMS?

The advantages are:

1. Less need for ETL
2. Support for unstructured text
3. Ability to handle change over time
4. Breadth of functionality
5. Ability to scale horizontally
6. Support for multiple data structures
7. Choice of vendors

16. What is an MX record in DNS?

MX records are mail exchange records used for determining the priority of email servers for a domain. The lowest priority email server is the first destination for email. If the lowest priority email server is unavailable, mail will be sent to the higher priority email servers.

17. What is the difference between RAID 0 and RAID 1?

RAID 1 offers redundancy through mirroring, i.e., data is written identically to two drives. RAID 0 offers no redundancy and instead uses striping, i.e., data is split across all the drives. This means RAID 0 offers no fault tolerance; if any of the constituent drives fails, the RAID unit fails.

18. How would you prepare for a migration?

Tips to answer: This question evaluates your experience of real projects with all the awkwardness and complexity they bring. Include terms like cut-over, dress rehearsals, roll-back and roll-forward, DNS solutions, feature toggles, branch by abstraction, and automation in your answer. Developing greenfield systems with little or no existing technology in place is always easier than having to deal with legacy components and configuration. As a candidate if you appreciate that any interesting software system will in effect be under constant migration, you will appear suitable for the role.

19. What's your systems background?

Tips to answer: Some DevOps jobs require extensive systems knowledge, including server clustering and highly concurrent systems. As a DevOps engineer, you need to analyze system capabilities and implement upgrades for efficiency, scalability and stability, or resilience. It is recommended that you have a solid knowledge of OSes and supporting technologies, like network security, virtual private networks and proxy server configuration.

DevOps relies on virtualization for rapid workload provisioning and allocating compute resources to new VMs to support the next rollout, so it is useful to have in-depth knowledge around popular hypervisors. This should ideally include backup, migration and lifecycle management tactics to protect, optimize and eventually recover computing resources. Some environments may emphasize microservices software development tailored for virtual containers. Operations expertise must include extensive knowledge of systems management tools like Microsoft System Center, Puppet, Nagios and Chef. DevOps jobs with an emphasis on

operations require detailed problem-solving, troubleshooting and analytical skills.

20. What DevOp tools have you worked with?

Tips to answer: Software configuration management and build/release (version control) tools, including Apache Subversion, Mercurial, Fossil and others, help document change requests. Developers can more easily follow the company's best practices and policies while software changes.

Continuous integration (CI) tools such as Rational Build Forge, Jenkins and Semaphore merge all developer copies of the working code into a central version. These tools are important for larger groups where teams of developers work on the same codebase simultaneously. QA experts use code analyzers to test software for bugs, security and performance. If you've used HP's Fortify Static Code Analyzer, talk about how it identified security vulnerabilities in coding languages. Also speak about tools like GrammaTech's CodeSonar that you used to identify memory leaks, buffer underruns and other defects for C/C++ and Java code. It is essential that you have adequate command of the principal languages like Ruby, C#, .NET, Perl, Python, Java, PHP, Windows PowerShell, and are comfortable with the associated OS environments Windows, Linux and Unix.

21. How much have you interacted with cloud based software development?

Tips to answer: Share your knowledge around use of cloud platforms, provisioning new instances, coding new software iterations with the cloud provider's APIs or software development kits, configuring clusters to scale computing capacity, managing workload lifecycles and so on. This is the perfect opportunity to discuss container-based cloud instances as an alternative to conventional VMs. Event-based cloud computing, such as AWS Lambda offers another approach to software development, a boon for experienced DevOps candidates. In your interview, mention experience handling big data, which uses highly scalable cloud infrastructures to tackle complex computing tasks.

22. What other tools are you familiar with that might help you in this role?

Tips to 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.

23. Are you familiar with just Linux or have you worked with Windows environments as well?

Tips to answer: Demonstrate as much as you can, a clear understanding of both the environments including the key tools.

24. How can you reduce load time of a dynamic website?

Tips to answer: Talk about Webpage optimization, cached web pages, quality web hosting , compressed text files, Apache fine tuning.

25. Describe your experience implementing continuous deployment?

Tips to answer: Answer with a comprehensive list of all the tools that you used. Include inferences of the challenges you faced and how you tackled them.

26. How would you ensure traceability?

Tips to answer: This question probes your attitude to metrics, logging, transaction journeys, and reporting. You should be able to identify that metric, monitoring and logging needs to be a core part of the software system, and that without them, the software is essentially not going to be able to appear maintained and diagnosed. Include words like SysLog, Splunk, error tracking, Nagios, SCOM, Avicode in your answer.

27. What was your greatest achievement on a recent project?

Tips to answer: Make sure you demonstrate your perfect understanding of both development and operations. Do not let your answer lean towards one particular skillset ignoring the other. Even if you have worked in an environment wherein you had to work more with one skillset, assure the interviewer that you are agile according to the needs of your organization.

28. What problems did you face and how did you solve them in a way that met the team's goals?

Tips to answer: This question aims to find out how much you can handle stress and non-conformity at work. Talk about your leadership skills to handle and motivate the team to solve problems together. Talk about CI, release management and other tools to keep interdisciplinary projects on track.

29. Are you more Dev or Ops?

Tips to answer: This is probably the trickiest question that you might face in the interview. Emphasize the fact that this depends a lot on the job, the company you are working for and the skills of people involved. You really have to be able to alternate between both sides of the fence at any given time. Talk about your experience and demonstrate how you are agile with both.

30. What special training or education did it require for you to become a DevOps engineer?

Tips to answer: DevOps is more of a mind-set or philosophy rather than a skill-set. The typical technical skills associated with DevOps Engineers today is Linux systems administration, scripting, and experience with one of the many continuous integration or configuration management tools like Jenkins and Chef. What it all boils down to is that whatever skill-sets you have, while important, are not as important as having the ability to learn new skills quickly to meet the needs. It's all about pattern recognition, and having the ability to merge your experiences with current requirements. Proficiency in Windows and Linux systems administration, script development, an understanding of structured programming and

object-oriented design, and experience creating and consuming RESTful APIs would take one a long way.

1. What is meant by CI?

Answer: In software development, when multiple developers or teams are working on different segments of same web application, we need to perform integration test by integrating all modules. In order to do that an automated process for each piece of code is performed on daily bases so that all your code get tested.

2. Why we use Jenkins?

Answer: Jenkins is used for the following reasons:

- A Version Control Software
- Working Build Script (ANT/MAVEN/MSBUILD)

3. What are some of the benefits of using Jenkins?

Answer: Benefits of using Jenkins are:

- Complete Build history
- Easy integration of Version Control Software, Build, Reporting and Deployment Tools
- Integration with several plugins
- Customization and Extension via custom scripts and plugins
- Easy tracking of bugs at early stage in development environment than production

4. You are asked to backup Jenkins. How would you approach this task?

Answer: Use the backup plugin. To use this, determine first If you want to back up all data or just a subset. Alternatively, script a physical copy using shell commands/Java CLI for Jenkins.

5. You are required to run 2 jobs in parallel while a 3rd job is to run after these have completed. How will you achieve this in Jenkins?

Answer: Use the Pipeline plugin. (TIP: You may be grilled in more depth on knowledge of Groovy in doing this task). Alternatively, you can use the Build Flow Plugin (TIP: If you answer this question, be prepared to answer questions on the DSL syntax, including passing parameters and error handling. Just as a FYI, the pipeline plugin is motivated by the Build Flow Plugin – you do not need both)

6. As part of the unit testing process, you are required to run unit test cases in parallel. How do you do this?

Answer: The choice of testing suite is important. TestNG and Junit 4 do support running test suites in parallel. Note that this runs test cases in parallel on a single node. To further refine this, you could also use the

Parallel Test Executor plugin for same purposes. In such a case, you can optionally split the cases and have it distributed across to multiple machines.

The Build Pipeline (see above question) has several build slave managed by Puppet and orchestrated by the Docker Swarm Plugin. When new slave is initialized, puppet installs everything to run the tests and slave joins the swarms. There are 2 jobs – a Distribution job and an actual execution job. The 1st job breaks down the test cases into sets (one set per slave) – and is written either in Groovy (Preferred) or Java. HERE IS THE CATCH --- EXECUTION TIME OF EACH TEST IS RECORDED. New Tests are randomly assigned, but the idea is to distribute the tests based on duration so that each set takes roughly the same time. The execution job is triggered on each slave using a Parametrized build (passing name of set). Number of slaves is determined by the Jenkins API and passed as a variable using the envinject Plugin.

7. This is great. What are the drawbacks you faced?

Answer: Dependent tests. If they are distributed to different sets so that they can be standalone

8. Name some common plugins you have used.

Answer: Common Plugins used are:

- Email Ext
- Git
- Build Pipelines
- Pipeline
- Flow
- Deployment Plugin
- OAuth

9. How will you secure Jenkins?

Answer: We will ensure the following:

- Ensure global security is on.
- Ensure that Jenkins is integrated with my company's user directory with appropriate plugin
- Ensure that matrix/Project matrix is enabled to fine tune access
- Automate the process of setting rights/privileges in Jenkins with custom version controlled script.
- Limit physical access to Jenkins data/folders.
- Periodically run security audits on same.

1) Explain what is DevOps?

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.

2) Mention what are the key aspects or principle behind DevOps?

The key aspects or principle behind DevOps is

Infrastructure as code

Continuous deployment

Automation

Monitoring

Security

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

The core operations of DevOps with

Application development

Code building

Code coverage

Unit testing

Packaging

Deployment

With infrastructure

Provisioning

Configuration

Orchestration

Deployment

4) Explain how “Infrastructure of code” is processed or executed in AWS?

In AWS,

The code for infrastructure will be in simple JSON format

This JSON code will be organized into files called templates

This 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

5) Explain which scripting language is most important for a DevOps engineer?

A simpler scripting language will be better for a DevOps engineer. Python seems to be very popular.

2000px-Devops.svg

6) Explain how DevOps is helpful to developers?

DevOps can be helpful to developers to fix the bug and implement new features quickly. It also helps for clearer communication between the team members.

7) List out some popular tools for DevOps?

Some of the popular tools for DevOps are

Jenkins

Nagios

Monit

ELK (Elasticsearch, Logstash, Kibana)

io

Jenkins

Docker

Ansible

Git

Collectd/Collectl

8) Mention at what instance have you used the SSH?

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.

9) Explain how would you handle revision (version) control?

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

10) Mention what are the types of Http requests?

The types of Http requests are

GET

HEAD

PUT

POST

PATCH

DELETE

TRACE

CONNECT

OPTIONS

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

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

Application Level troubleshooting

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

System Level troubleshooting

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

Dependent Services troubleshooting

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

12) Whether your video card can run Unity how would you know?

When you use command

```
/usr/lib/nux/unity_support_test-p
```

```
1
```

```
/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.

13) Explain how to enable startup sound in Ubuntu?

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

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

```
1
```

```
/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.

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

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.

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

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.

16) Explain how you create launchers on desktop in Ubuntu?

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

17) Explain what is Memcached?

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.

Memcached can be used for

- 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
- Gaming and entertainment -> Session caching

Memcache helps in

- 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)

Drawback of Memcached is

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

18) Mention some important features of Memcached?

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.

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

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. Unless, if you have partitioned your data, then it becomes necessary to know from which instance to get the data from or to put into.

20) 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?

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.

21) Explain how you can minimize the Memcached server outages?

- 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

22) Explain how you can update Memcached when data changes?

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.

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

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.

24) Explain how Memcached should not be used?

- 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

25) When server gets shut down does data stored in Memcached is still available?

Data stored in Memcached is not durable so if server is shut down or restarted then all the data stored in Memcached is deleted.

26) Mention what is the difference between Memcache and Memcached?

- 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.

Basic Questions

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

Its highly effective daily 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, 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 automated testing, continuous Integration and continuous Delivery. On a traditional implementation, 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.

2) Why we need DevOps ?

Companies are now facing the need to 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.

3) What is agile development and Scrum ?

Agile development used as an alternative to Waterfall development practice. In Agile, the development process is more iterative and incremental, there is 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 ie product owner, scrum master, and team.

4) Can we consider DevOps as an agile methodology ?

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 demands of business, who want to speed up the changes to stick closer to the requirements of business and the customer.

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

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.

Technical Questions

6) Have you worked on containers ?

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 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.

7) What is Kubernetes? Explain

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.

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

CI server 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

9) What is Continuous Delivery ?

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

10) What is Vagrant and what is it used for ?

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.

11) Do you ever used any scripting language ?

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.

Currently, several scripting languages are available so the question arises : what is the most appropriate language for DevOps approach? Simply everything , it depends on the context of the project and tools used for example if Ansible used its good have knowledge in Python and if its for Chef its on Ruby.

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

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.

13) What is the purpose of CM tools and which one you have used ?

Configuration Management tools' 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, SaltStack.

14) What is OpenStack ?

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 chose which functionality you need. OpenStack distribution are available from Red Hat, Mirantis, HPE, Oracle, Canonical and many others. It is completely open source project but some vendors make proprietary distributions.

15) Classify Cloud Platforms category ?

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 piece 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, Eucalyptus.

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 OpenShift, Heroku.

16) What are easiest ways to build a small cloud ?

VMfest is one 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.

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

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.

18) What is EC2 ?

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 .

19) Do you find any advantage of using NoSQL database over RDBMS ?

Typical web applications are built with a three-tier architecture. To carry the load, more Web servers are simply added behind a load balancer to support more users. The ability to scale out is a key principle in the world of cloud computing, more and more important in which VM instances can be easily added or removed to meet demand.

However, when it comes to the data layer, relational databases (RDBMS) does not allow a passage to the simple scale and do not provide a flexible data model. Manage more users means adding more servers and large servers are very complex, owners and disproportionately expensive, in

contrast to low-cost hardware, the "commodity hardware", architectures in the cloud. Organizations are beginning to see performance issues with their relational databases for existing or new applications. Especially as the number of users increases, they realize the need for a faster and more flexible basis. This is the time to begin to assess and adopt NoSQL database like in their Web applications.

20) What are the main SQL migration difficulties NoSQL ?

Each record in a relational database according to a schema - with a fixed number of fields (columns) each having 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.

21) What are the benefits of NoSQL databases Documents ?

The main advantages of document databases are the following :

flexible data model data can be inserted without a defined schema and format of the data that is inserted can change at any time , providing extreme flexibility , which ultimately allows a significant agility to business

Consistent , high-performance Advanced NoSQL database technologies are putting cache data , transparently, in system memory ; a behavior that is completely transparent to the developer and the team in charge of operations .

Some easy scalability NoSQL databases automatically propagate data between servers , requiring no participation applications. Servers can be added and removed without disruption to applications , with data and I/O spread across multiple servers.

22) What are the main advantages of Git over CVS ?

The biggest advantage is that Git is distributed while CVS is centralised. 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.

23) Difference between containers and virtual machines ?

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. Container host uses the process and file system isolation features of the linux kernel.

24) What is CoreOS, and what are alternatives ?

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.

25) What is Kickstart ?

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.

26) What are tools for network monitoring? List few

For example, Nagios, Icinga 2, OpenNMS, Splunk and Wireshark. 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 are open source.

27) What is Juju ?

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

28) Give me an examples of how you would handle projects ?

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, translate requirements into workflow and follow up. I would resort to CI, release management and other tools to keep interdisciplinary projects on track.

29) What is post mortem meetings ?

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 reoccurring and planing redesign of the infrastructure so that downtime can be minimised. It is about learning from mistakes.

30) What you know about serverless model ?

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 efforts and allow developers to quickly focus on developing codes.

