**2. List the essential tools used in Devops.**

* Git
* Jenkins
* Selenium
* Puppet
* Chef
* Ansible
* Nagios
* Docker
* Monit
* ELK –Elasticsearch, Logstash, Kibana
* Collectd/Collect
* Git(GitHub)

**3. What are the core operations of DevOps in terms of development and Infrastructure?**

**The core operations of DevOps**

* Application development
* Code developing
* Code coverage
* Unit testing
* Packaging
* Deployment With infrastructure
* Provisioning
* Configuration
* Orchestration
* Deployment

**4. What are the advantages of DevOps with respect to Technical and Business perspective?**

**Technical benefits:**

* Software delivery is continuous.
* Reduces Complexity in problems.
* Faster approach to resolve problems
* Manpower is reduced.

**Business benefits:**

* High rate of delivering its features
* Stable operating environments
* More time gained to Add values.
* Enabling faster feature time to market

**5. The scope for SSH?**

* SSH is a Secure Shell which provides users with a secure, encrypted mechanism to log into systems and transfer files.
* To log out a remote machine and work on command line.
* To secure encrypted communications between two hosts over an insecure network.

**6. Which are the areas where DevOps are implemented?**

* Production Development
* Creation of the production feedback and its development
* IT Operations development

**7. List the agile methodology of DevOps.**

* DevOps is a process
* Agile is same as DevOps.
* Separate group for are framed.
* It is problem solving.
* Developers managing production
* DevOps is development-driven release management

**8. List the major difference between the Agile and DevOps.**

**Agile:**

1. Agile is about software development

**Devops:**

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

**9. Name the popular scripting language of DevOps.**

Python

**10. How DevOps is helpful to developers?**

* To fix the bug and implement new features quickly.
* It provides the clarity of communication among team members.

**11. What are Vagrant and its uses.**

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

Are you interested in learning DevOps? We have the comprehensive [*DevOps Training Courses*](https://intellipaat.com/devops-certification-training/#about-course)to give you a head start in your career.

**12. What are the major difference between the Linux and Unix operating systems?**

**Unix:**

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

**Linux:**

* Linux has probably been home to every programming language known to humankind.
* These are used for personal computers.
* The LINUX is based on the kernel of UNIX operating system.

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

* Backup System
* Recovery plans
* Load Balancing
* Monitoring
* Centralized logging

**14. What are the benefits of the NoSQL?**

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

**15. What are adoptions of DevOps in industry?**

1. Use of agile and other development processes and methods .
2. Demand for an increased rate of production releases from application and business.
3. Wide availability of virtual and cloud infrastructure from both internal and external providers;
4. Increased usage of data center ,automation and configuration management tools;
5. Increased focus on test automation and continuous integration methods;
6. Best practices on critical issues.

**16. What are the advantages of NoSQL database over RDBMS?**

**The advantages are:**

1. There is very less scope of ETL
2. Support is given for unstructured text
3. Changes are handle over period of time
4. Main objectives are functionality.
5. It has the ability to scale horizontally
6. Multiple data structures are given support.
7. Vendors can be chosen.

Give your career a big boost by going through our [*DevOps Training Videos*](https://intellipaat.com/devops-certification-training/#course-preview)*!*

**17. The top 10 skills the person should be having for the DevOp’s position?**

* Excellent in System Admin
* Virtualization Experience
* Good Technical Skills
* Excellent Scripting
* Good Developing skills
* Chef in Automation Tool Experience
* People Management
* Customer Service
* Real time Cloud operations
* Who care about someone

**18. Explain how the implementation of “Infrastructure as code” is processed or executed in terms of AWS.**

**In AWS,**

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

**19. What measures we have taken to handle revision (version) control?**

To handle revision control, post your code on SourceForge or GitHub so everyone can view it and ask the viewers to give suggestions for the better improvement of it.

**20. What are the types of HTTP requests?**

The types of Http requests are

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

<http://interviewquestionstutorials.com/tag/50-top-devops-interview-questions-and-answers-pdf/>

**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 manged 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 anategory ?

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

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

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

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 striped down linux distribution meant for running containters, 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 is 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.

Examples are Amazon AWS Lambda and Auth0 serveless platform.

**Devops Example : Deploying Applications with Ansible**

Ansible is a lightweight, extensible solution for automating your application provisioning. Ansible has no dependencies other than Python and SSH. It doesn’t require any agents to be set up on the remote hosts and it doesn’t leave any traces after it runs either. It allows you to significantly simplify our operations by creating easy YAML based playbooks. It’s good for configuration automation, deployments and orchestration.

Components of Ansible

**Playbooks :** Ansible playbooks are a way to send commands to remote computers in a scripted way. Instead of using Ansible commands individually to remotely configure computers from the command line, you can configure entire complex environments by passing a script to one or more systems.

Ansible playbooks are written in the YAML data serialization format. If you don't know what a data serialization format is, think of it as a way to translate a programmatic data structure (lists, arrays, dictionaries, etc) into a format that can be easily stored to disk. The file can then be used to recreate the structure at a later point. JSON is another popular data serialization format, but YAML is much easier to read.

Let's look at a basic playbook that allow us to install a web application (nginx) in a multiple hosts :

hosts: webservers  
tasks:  
- name: Installs nginx web server  
apt: pkg=nginx state=installed update\_cache=true  
notify:  
- start nginx

handlers:  
- name: start nginx  
service: name=nginx state=started

**The hosts file :** (by default under /etc/ansible/hosts) this is the Ansible Inventory file, and it stores the hosts, and their mappings to the host groups (webservers ,databases etc)

[webservers] 10.0.15.22  
# example of setting a host inventory by IP address.  
# also demonstrates how to set per-host variables.

[repository\_servers] example-repository  
#example of setting a host by hostname. Requires local lookup in /etc/hosts  
# or DNS.  
[dbservers] db01

**The SSH key :** For the first run, we'll need to tell ansible the SSH and Sudo passwords, because one of the thing that the common role does is to configure passwordless sudo, and deploy a SSH key. So in this case ansible can execute the playbook’s commands in the remote nodes (hosts ) and deploy the web application nginx.

**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 intervewer 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 questions 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.  
**31) 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.  
**32) 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  
**33) 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  
**34) 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  
**35) 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.

**36) 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.  
**37) 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  
**38) 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.  
**39) 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.  
**40) Mention what are the types of Http requests?**  
The types of Http requests are  
• GET  
• HEAD  
• PUT  
• POST  
• PATCH  
• DELETE  
• TRACE  
• CONNECT  
• OPTIONS

<http://www.interviewquestionspdf.com/2017/01/top-60-aws-devops-real-time-interview.html>

**1. What have you been doing over the last 1-2 years?**

Interviews do not have to adhere to a specific framework and can be dynamic in nature. To get a general perspective on what an candidate has done, it’s a good idea to start an interview with a general query into the engineer’s recent professional activities.

This will help you, as a DevOps manager, to understand with what specific tools and technologies the engineer has been working over the past few years (these can include Git, Puppet, Jenkins, Docker, Ansible, and scripting languages). Also, it will reveal the candidate’s ability to work in a team as the candidate will most likely divulge whether he or she flew solo or was part of a bigger outfit. If the person’s answer does not include this information, then that is another must-ask question.

It is critical to take note of the roles in which the candidate has served and the tasks that the candidate has performed, even if they are not strictly required in your organization or in the role for which he or she is interviewing. If the prospect does not mention the exact tools that you currently use, follow up with questions about those tools and tasks to get a good feel for his or her ability to assimilate knowledge as well as his or her general operating dependencies. Good candidates will always demonstrate a deep understanding in the field of their operation while others will reply with superficial answers to drill-down follow-up questions.

**2. How do you deploy software?**

This question is critical for any DevOps position. As more and more DevOps teams move towards automating and adopting continuous delivery best practices, it is critical to gauge whether the candidate is comfortable talking about code deployment and whether he or she understands how all of the available [continuous integration tools](https://logz.io/blog/continuous-integration-tools/) and [DevOps tools](https://logz.io/blog/devops-tools/) fit together. If you have a drawing board available, let him or her build a diagram for you.

Depending on the answers that you get, you can develop further lines of questioning dynamically. For example: “Do you have a database in the stack?” “How do you update the schema?” “What tests do you run, and how do you run them?” “If all tests pass, how is the code deployed into production?” “How do you make sure that you do not lose traffic during deployment?”

**3. How have you handled failed deployments?**

Failed deployments, of course, are an all-too-common occurrence when deploying code. DevOps engineers need to be extremely hands-on — they need to know when something has gone wrong and then troubleshoot the issue as quickly as possible.

A good way of assessing the suitability of a candidate is to ask them to tell the story of a failed deployment and how it was handled. Specific, follow-up questions can include: “How do you know there was a deployment failure?” “Do you roll back automatically?” and “What criteria do you use?”

**4. If something breaks in production, how do you know about it?**

Monitoring is a huge component of DevOps work (and this is reflected by the multitude of monitoring tools and platform out there). Regardless of the specific tools that you use and the monitoring system that you employ in your company, you need to know how well-versed the candidate is in planning and executing a monitoring strategy.

Again, you could use the storytelling tactic: “Tell me about a crisis in production that you had, how you became aware of it, and how it was solved.” A good war story is always enlightening — it will help you to assess not only how skilled the candidates are in monitoring but also how they handle crises (assuming that they tell the truth, of course).

Other leading questions: “What monitoring tools do you work with?” “Did you choose them? If so, why?” and “How do you get alerted?” I have found that the best candidates will have plenty to share about their monitoring expertise and specifically about advanced user-experience monitoring techniques.

**5. What happens when you type “mv \*” in a directory with three subdirectories a, b and c?**

Of course, this question — and the responses — can vary, but the idea is to gauge the technical expertise of the engineer in a Linux environment, which is a “must” in almost all DevOps positions.

It’s a good idea to change the bash command as you receive the answers. If you feel the questions are too easy, try raising the bar with more advanced bash questions. For example, what is the difference between **‘cmd1 ; cmd2’** and **‘cmd1 && cmd2’**?

You might want to prepare a quiz sheet with a list of five to ten commands. This way, the candidate will find it easier to answer.

**6. Without using Docker, can you see the processes running inside a container from the outside?**

Ok, we cheated here. Not every company is using [Docker](https://logz.io/blog/what-is-docker/) or even containers at all, so this question is a bit technology-specific. Based on our expertise and on the data in [The 2016 DevOps Pulse survey](https://logz.io/blog/the-devops-pulse-2016-results/) that we recently released, more and more companies are moving to microservices and containerized architectures. So, we added this question to the list.

Of course, this question is meant to figure out whether the candidate understands how containerization works. Instead of asking “How do containers work?” or “What is a Docker image?”, the answer to the question above will inform you whether the person “gets” it. Other questions may include “How does container linking work?” or “How and why would you optimize a Dockerfile?”

**7. Describe the Linux boot process.**

This is another question meant to gauge the candidate’s system understanding and Linux expertise.

A good candidate will be able to detail the correct order and significance of at least some of the various stages (e.g., BIOS, MBR, bootloader, kernel, initialization, and runlevel). To drill down further, I’d recommend a follow-up question such as “What information needs to be provided to the bootloader?”

**8. How does ‘traceroute’ work?**

Any DevOps interview has to include networking questions.

Many candidates will not know the answer to this question while others will offer only a partial answer. A good way to separate the DevOps wheat from the chaff is to see if the candidate only explains that the command prints the route that packets take to the network host or if he or she also delves into the “how.”

Even if you do not receive a correct and complete answer, this question is a good starting point for a deeper conversation in which you can brainstorm with the candidate. In this process, you can try to come up with valid possibilities and discount invalid ones based on a solid understanding of IP routing.

Another example of a good networking question that I often use: “What is the difference between trying to connect to a port that is not being listened to as opposed to one that is firewalled in terms of TCP?”

**9. Do you consider seven to be a high load average?**

Logz.io is an [AI-powered log analysis platform](https://logz.io/product/) that offers the open source [ELK Stack](https://logz.io/learn/complete-guide-elk-stack/) as a cloud service, so we do our healthy share of performance testing and tuning. We need our DevOps engineers to understand the fundamentals of system performance monitoring for both planning purposes and troubleshooting issues in production.

This question enables you to learn whether the candidate understands the meaning of load average in the first place. If they understand and explain that it is not CPU usage, it is a great opening for a deeper discussion on troubleshooting performance.

Useful follow-ups: “Is it possible to observe high load with low CPU usage?” “If so, what may be the reasons?” and “How would you check?”

**10. Do a FizzBuzz coding test.**

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

The candidate should:

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

Most good developers should be able to write such a program on paper within a couple of minutes. See how they write the code, ask them why they wrote specific parts in certain ways, and then check the validity of the code.

**Explain what is DevOps?**  
DevOps 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 aims at establishing a culture and environment where building, testing, and releasing software, can happen rapidly, frequently, and more reliably.  
  
**DevOps focuses on 4 primary areas within IT?**   
1. Culture.  
2. Organization (style including roles).  
3. Processes.  
4. Tools.  
  
**What are the nine things that make up Dev & Ops?**  
  
- 1 step build & deploy  
- Automated infrastructure  
- Shared Version control  
- Shared metrics  
- Break up code with flags  
- IM IRC Bots  
- Healthy attitude towards failure  
- Trust and respect  
- Don't Blame Others

**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.

**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

|  |  |
| --- | --- |
| 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



|  |  |
| --- | --- |
| 1 | /usr/bin/canberra-gtk-play—id= “desktop-login”—description= “play login sound” |

* Logout and then login once you are done

**Y**ou 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.

GIT is a very popular version control tool in software community. Many fortune 500 organizations use GIT. This book contains basic to expert level GIT interview questions that an interviewer asks. Each question is accompanied with an answer so that you can prepare for job interview in short time.

We have compiled this list of GIT questions after attending dozens of technical interviews in top-notch companies like- Google, Facebook, Ebay, Amazon etc.

Some of the sample questions are:  
How can we know if a branch is already merged into master in GIT?  
What is the purpose of git stash drop?  
What is the HEAD in GIT?  
What is the most popular branching strategy in GIT?  
What is SubGit?  
What is the use of git instaweb?  
What are git hooks?  
What is GIT?  
What is a repository in GIT?  
What are the main benefits of GIT?  
What are the disadvantages of GIT?  
What are the main differences between GIT and SVN?  
How will you start GIT for your project?  
What is git clone in GIT?  
How will you create a repository in GIT?  
What are the different ways to start work in GIT?  
GIT is written in which language?  
What does ‘git pull’ command in GIT do internally?  
What does ‘git push’ command in GIT do internally?  
What is git stash?  
What is the meaning of ‘stage’ in GIT?  
What is the purpose of git config command?  
How can we see the configuration settings of GIT installation?  
How will you write a message with commit command in GIT?  
What is stored inside a commit object in GIT?  
How many heads can you create in a GIT repository?  
Why do we create branches in GIT?  
What are the different kinds of branches that can be created in GIT?  
How will you create a new branch in GIT?  
How will you add a new feature to the main branch?  
What is a pull request in GIT?

<https://www.amazon.com/Top-100-Interview-Questions-Answers-ebook/dp/B01L80Z02S>

##### **1. What is ANT?**

**Ans.** ANT full form is Another Needed Tool. Ant is a build tool that is java based. A build tool performs the following tasks:  
  
Compiling java code into byte code  
Placing this byte code in a package  
Deployment to production systems  
Document creation and release notes preparation.

JMS

##### **Ques 1. How many messaging models do JMS provide for and what are they?**

**Ans.** JMS provide for two messaging models, publish-and-subscribe and point-to-point queuing.

##### **Ques 2. What is JMS (Java Messaging Service)?**

**Ans.** JMS is an acronym used for Java Messaging Service. It is Java's answer to creating software using asynchronous messaging. It is one of the official specifications of the J2EE technologies and is a key technology.

##### **Ques 3. How the JMS is different from RPC?**

**Ans.** In RPC the method invoker waits for the method to finish execution and return the control back to the invoker. Thus it is completely synchronous in nature. While in JMS the message sender just sends the message to the destination and continues it's own processing. The sender does not wait for the receiver to respond. This is asynchronous behavior.

##### **Ques 4. What are the basic advantages of JMS?**

**Ans.** JMS is asynchronous in nature. Thus not all the pieces need to be up all the time for the application to function as a whole. Even if the receiver is down the MOM will store the messages on it's behalf and will send them once it comes back up. Thus at least a part of application can still function as there is no blocking.

##### **Ques 5. What are the different types of messages available in the JMS API?**

**Ans.** Message, TextMessage, BytesMessage, StreamMessage, ObjectMessage, MapMessage are the different messages available in the JMS API.

##### **Ques 6. What are the different messaging paradigms JMS supports?**

**Ans.** Publish and Subscribe i.e. pub/suc and Point to Point i.e. p2p.

##### **Ques 7. What is the difference between topic and queue?**

**Ans.** A topic is typically used for one to many messaging i.e. it supports publish subscribe model of messaging. While queue is used for one-to-one messaging i.e. it supports Point to Point Messaging.

##### **Ques 8. What is the use of Message object?**

**Ans.** Message is a light weight message having only header and properties and no payload. Thus if theIf the receivers are to be notified abt an event, and no data needs to be exchanged then using Message can be very efficient.

##### **Ques 9. What is the basic difference between Publish Subscribe model and P2P model?**

**Ans.** Publish Subscribe model is typically used in one-to-many situation. It is unreliable but very fast. P2P model is used in one-to-one situation. It is highly reliable.

##### **Ques 10. What is the use of TextMessage?**

**Ans.** TextMessage contains instance of java.lang.String as it's payload. Thus it is very useful for exchanging textual data. It can also be used for exchanging complex character data such as an XML document.

##### **Ques 12. What is JMS provider?**

**Ans.** An implementation of the JMS interface for a Message Oriented Middleware (MOM). Providers are implemented as either a Java JMS implementation or an adapter to a non-Java MOM.

##### **Ques 13. Wat is JMS client?**

**Ans.** An application or process that produces and/or receives messages.

##### **Ques 14. What is JMS producer?**

**Ans.** A JMS client that creates and sends messages.

##### **Ques 15. What is JMS consumer?**

**Ans.** A JMS client that receives messages.

##### **Ques 16. What is JMS message?**

**Ans.** An object that contains the data being transferred between JMS clients.

##### **Ques 17. What is JMS queue?**

**Ans.** A staging area that contains messages that have been sent and are waiting to be read. Note that, contrary to what the name queue suggests, messages don't have to be delivered in the order sent. If the message driven bean pool contains more than one instance then messages can be processed concurrently and thus it is possible that a later message is processed sooner than an earlier one. A JMS queue guarantees only that each message is processed only once.

##### **Ques 18. What is JMS topic?**

**Ans.** A distribution mechanism for publishing messages that are delivered to multiple subscribers.

##### **Ques 19. What is JMS?**

**Ans.** Java Message Service (JMS): An interface implemented by most J2EE containers to provide point-to-point queueing and topic (publish/subscribe) behavior. JMS is frequently used by EJB's that need to start another process asynchronously.   
For example, instead of sending an email directly from an Enterprise JavaBean, the bean may choose to put the message onto a JMS queue to be handled by a Message-Driven Bean (another type of EJB) or another system in the enterprise. This technique allows the EJB to return to handling requests immediately instead of waiting for a potentially lengthy process to complete.

##### **Ques 20. What type messaging is provided by JMS?**

**Ans.** Both synchronous and asynchronous.

##### **Ques 21. How may messaging models do JMS provide for and what are they?**

**Ans.** JMS provides for two messaging models, publish-and-subscribe and point-to-point queuing.

##### **Ques 22. What is the point-to-point model in JMS?**

**Ans.** A point-to-point model is based on the concept of a message queue: Senders send messages into the queue, and the receiver reads messages from this queue. In the point-to-point model, several receivers can exist, attached to the same queue. However, (Message Oriented Middleware)MOM will deliver the message only to one of them. To which depends on the MOM implementation.

##### **Ques 23. What is the publish-and-subscribe model in JMS?**

**Ans.** A publish-subscribe model is based on the message topic concept: Publishers send messages in a topic, and all subscribers of the given topic receive these messages.

##### **Ques 24. What is JMS administered object?**

**Ans.** A preconfigured JMS object (a resource manager connection factory or a destination) created by an administrator for the use of JMS clients and placed in a JNDI namespace.

##### **Ques 25. What is publish/subscribe messaging?**

**Ans.** With publish/subscribe message passing the sending application/client establishes a named topic in the JMS broker/server and publishes messages to this queue. The receiving clients register (specifically, subscribe) via the broker to messages by topic; every subscriber to a topic receives each message published to that topic. There is a one-to-many relationship between the publishing client and the subscribing clients.

##### **Ques 26. What is the main parts of JMS applications?**

**Ans.** The main parts of JMS applications are:  
--ConnectionFactory and Destination  
--Connection   
--Session   
--MessageProducer  
--MessageConsumer   
--Message

##### **Ques 27. What Is Messaging?**

**Ans.** Messaging is a method of communication between software components or applications. A messaging system is a peer-to-peer facility: A messaging client can send messages to, and receive messages from, any other client. Each client connects to a messaging agent that provides facilities for creating, sending, receiving, and reading messages.   
Messaging enables distributed communication that is loosely coupled. A component sends a message to a destination, and the recipient can retrieve the message from the destination. However, the sender and the receiver do not have to be available at the same time in order to communicate. In fact, the sender does not need to know anything about the receiver; nor does the receiver need to know anything about the sender. The sender and the receiver need to know only what message format and what destination to use. In this respect, messaging differs from tightly coupled technologies, such as Remote Method Invocation (RMI), which require an application to know a remote application's methods.   
Messaging also differs from electronic mail (e-mail), which is a method of communication between people or between software applications and people. Messaging is used for communication between software applications or software components.   
Messaging is a mechanism by which data can be passed from one application to another application.

##### **Ques 28. What is the Role of the JMS Provider?**

**Ans.** The JMS provider handles security of the messages, data conversion and the client triggering. The JMS provider specifies the level of encryption and the security level of the message, the best data type for the non-JMS client.

##### **Ques 29. What is the diffrence between Java Mail and JMS Queue?**

**Ans.** JMS is the ideal high-performance messaging platform for intrabusiness messaging, with full programmatic control over quality of service and delivery options.   
JavaMail provides lowest common denominator, slow, but human-readable messaging using infrastructure already available on virtually every computing platform.

##### **Ques 30. Does JMS specification define transactions?**

**Ans.** JMS specification defines a transaction mechanisms allowing clients to send and receive groups of logically bounded messages as a single unit of information. A Session may be marked as transacted. It means that all messages sent in a session are considered as parts of a transaction. A set of messages can be committed (commit() method) or rolled back (rollback() method). If a provider supports distributed transactions, it's recommended to use XAResource API.

##### **Ques 31. What is synchronous messaging?**

**Ans.** Synchronous messaging involves a client that waits for the server to respond to a message. So if one end is down the entire communication will fail.

##### **Ques 32. What is asynchronous messaging?**

**Ans.** Asynchronous messaging involves a client that does not wait for a message from the server. An event is used to trigger a message from a server. So even if the client is down , the messaging will complete successfully.

##### **Ques 33. How does a typical client perform the communication?**

**Ans.** 1. Use JNDI to locate administrative objects.  
2. Locate a single ConnectionFactory object.  
3. Locate one or more Destination objects.  
4. Use the ConnectionFactory to create a JMS Connection.  
5. Use the Connection to create one or more Session(s).  
6. Use a Session and the Destinations to create the MessageProducers and MessageConsumers needed.  
7. Perform your communication.

##### **Ques 34. What is JMS session?**

**Ans.** A single-threaded context for sending and receiving JMS messages. A JMS session can be nontransacted, locally transacted, or participating in a distributed transaction.

##### **Ques 35. What is the use of JMS? In which situations we are using JMS? Can we send message from one server to another server using JMS?**

**Ans.** JMS is the ideal high-performance messaging platform for intrabusiness messaging, with full programmatic control over quality of service and delivery options.

##### **Ques 36. What is the difference between durable and non-durable subscriptions?**

**Ans.** Point-To-Point (PTP). This model allows exchanging messages via queues created for some purposes. A client can send and receive messages from one or several queues. PTP model is easier than pub/sub model.   
A durable subscription gives a subscriber the freedom of receiving all messages from a topic, whereas a non-durable subscription doesn't make any guarantees about messages sent by others when a client was disconnected from a topic.

##### **Ques 37. What is the difference between Message producer and Message consumer?**

**Ans.** In Publish/Subscribe model:   
A publish/subscribe (pub/sub) messaging system supports an event driven model where information consumers and producers participate in the transmission of messages. Producers "publish" events, while consumers "subscribe" to events of interest, and consume the events. Producers associate messages with a specific topic, and the messaging system routes messages to consumers based on the topics the consumers register interest in.   
  
In Point-To-Point model:   
In point to point messaging systems, messages are routed to an individual consumer which maintains a queue of "incoming" messages. Messaging applications send messages to a specified queue, and clients retrieve messages from a queue.

##### **Ques 38. What is JMS application?**

**Ans.** One or more JMS clients that exchange messages.

##### **Ques 39. What type messaging is provided by JMS?**

**Ans.** Both synchronous and asynchronous are provided by JMS

##### **Ques 40. How JMS is different from RPC?**

**Ans.** In RPC the method invoker waits for the method to finish execution and return the control back to the invoker. Thus it is completely synchronous in nature. While in JMS the message sender just sends the message to the destination and continues it's own processing. The sender does not wait for the receiver to respond. This is asynchronous behavior.

##### **Ques 41. What Is the JMS API?**

**Ans.** The Java Message Service is a Java API that allows applications to create, send, receive, and read messages. Designed by Sun and several partner companies, the JMS API defines a common set of interfaces and associated semantics that allow programs written in the Java programming language to communicate with other messaging implementations.   
The JMS API minimizes the set of concepts a programmer must learn to use messaging products but provides enough features to support sophisticated messaging applications. It also strives to maximize the portability of JMS applications across JMS providers in the same messaging domain.   
The JMS API enables communication that is not only loosely coupled but also  
\* Asynchronous. A JMS provider can deliver messages to a client as they arrive; a client does not have to request messages in order to receive them.  
\* Reliable. The JMS API can ensure that a message is delivered once and only once. Lower levels of reliability are available for applications that can afford to miss messages or to receive duplicate messages.   
The JMS Specification was first published in August 1998. The latest version of the JMS Specification is Version 1.1, which was released in April 2002. You can download a copy of the Specification from the JMS Web site, http://java.sun.com/products/jms/.

##### **Ques 42. What is JMS client?**

**Ans.** A Java language program that sends or receives messages.

##### **Ques 43. Give an example of using the point-to-point model.**

**Ans.** The point-to-point model is used when the information is specific to a single client. For example, a client can send a message for a print out, and the server can send information back to this client after completion of the print job.

##### **Ques 44. What is Producer and Consumer?**

**Ans.** Messaging lets a servlet delegate processing to a batch process either on the same machine or on a separate machine. The servlet creates a message and sends it to a queue. The servlet immediately completes and when the batch process is ready, it processes the message.   
Messaging is therefore comprised of three main components:   
A Producer creates messages and sends them to a Queue. The Producer could be something like a Servlet.   
A Queue stores the messages from the Produces and provides them to a Consumer when ready. The Queue is implemented by the messaging provider.   
A Consumer processes messages as they become available in the Queue. The Consumer is typically a bean implementing the MessageListener interface.

##### **Ques 46. What is the Role of the JMS Provider?**

**Ans.** The JMS provider handles security of the messages, data conversion and the client triggering. The JMS provider specifies the level of encryption and the security level of the message, the best data type for the non-JMS client.

##### **Ques 48. What is the difference between Byte Message and Stream Message?**

**Ans.** Bytes Message stores data in bytes. Thus the message is one contiguous stream of bytes. While the Stream Message maintains a boundary between the different data types stored because it also stores the type information along with the value of the primitive being stored. Bytes Message allows data to be read using any type. Thus even if your payload contains a long value, you can invoke a method to read a short and it will return you something. It will not give you a semantically correct data but the call will succeed in reading the first two bytes of data. This is strictly prohibited in the Stream Message. It maintains the type information of the data being stored and enforces strict conversion rules on the data being read.

##### **Ques 49. Are you aware of any major JMS products available in the market?**

**Ans.** IBM's MQ Series is one of the most popular product used as Message Oriented Middleware. Some of the other products are SonicMQ, iBus etc. Weblogic application server also comes with built in support for JMS messaging.

##### **Ques 50. What are the different types of messages available in the JMS API?**

**Ans.** Message, TextMessage, BytesMessage, StreamMessage, ObjectMessage, MapMessage are the different messages available in the JMS API.

##### **Ques 51. How Does the JMS API Work with the J2EE Platform?**

**Ans.** When the JMS API was introduced in 1998, its most important purpose was to allow Java applications to access existing messaging-oriented middleware (MOM) systems, such as MQSeries from IBM. Since that time, many vendors have adopted and implemented the JMS API, so that a JMS product can now provide a complete messaging capability for an enterprise.   
Since the 1.3 release of the J2EE platform ("the J2EE 1.3 platform"), the JMS API has been an integral part of the platform, and application developers can use messaging with components using J2EE APIs ("J2EE components").   
The JMS API in the J2EE platform has the following features.   
\* Application clients, Enterprise JavaBeans (EJB) components, and Web components can send or synchronously receive a JMS message. Application clients can in addition receive JMS messages asynchronously. (Applets, however, are not required to support the JMS API.)  
\* Message-driven beans, which are a kind of enterprise bean, enable the asynchronous consumption of messages. A JMS provider may optionally implement concurrent processing of messages by message-driven beans.  
\* Message sends and receives can participate in distributed transactions.  
  
The JMS API enhances the J2EE platform by simplifying enterprise development, allowing loosely coupled, reliable, asynchronous interactions among J2EE components and legacy systems capable of messaging. A developer can easily add new behavior to a J2EE application with existing business events by adding a new message-driven bean to operate on specific business events. The J2EE platform's EJB container architecture, moreover, enhances the JMS API by providing support for distributed transactions and allowing for the concurrent consumption of messages.   
Another J2EE platform technology, the J2EE Connector Architecture, provides tight integration between J2EE applications and existing Enterprise Information (EIS) systems. The JMS API, on the other hand, allows for a very loosely coupled interaction between J2EE applications and existing EIS systems.   
At the 1.4 release of the J2EE platform, the JMS provider may be integrated with the application server using the J2EE Connector Architecture. You access the JMS provider through a resource adapter. For more information, see the Enterprise JavaBeans Specification, v2.1, and the J2EE Connector Architecture Specification, v1.5.

##### **Ques 52. What is the role of JMS in enterprise solution development?**

**Ans.** JMS is typically used in the following scenarios  
1. Enterprise Application Integration: - Where a legacy application is integrated with a new application via messaging.  
2. B2B or Business to Business: - Businesses can interact with each other via messaging because JMS allows organizations to cooperate without tightly coupling their business systems.  
3. Geographically dispersed units: - JMS can ensure safe exchange of data amongst the geographically dispersed units of an organization.  
4. One to many applications: - The applications that have to push data in packet to huge number of clients in a one-to-many fashion are good candidates for the use JMS. Typical such applications are Auction Sites, Stock Quote Services etc.

##### **Ques 56. Why do the TopicSession.createDurableSubscriber and TopicSession.unubscribe calls raise JMSException with the message "ORA - 4020 - deadlock detected while trying to lock object"?**

**Ans.** CreateDurableSubscriber and unsubscribe calls require exclusive access to the Topics. If there are pending JMS operations (send/publish/receive) on the same Topic before these calls are issued, the ORA - 4020 exception is raised.   
There are two solutions to the problem:  
1. Try to isolate the calls to createDurableSubscriber and unsubscribe at the setup or cleanup phase when there are no other JMS operations happening on the Topic. That will make sure that the required resources are not held by other JMS operational calls. Hence the error ORA - 4020 will not be raised.  
2. Issue a TopicSession.commit call before calling createDurableSubscriber and unsubscribe call.

##### **Ques 57. Why doesn't AQ\_ADMINISTRATOR\_ROLE or AQ\_USER\_ROLE always work for AQ applications using Java/JMS API?**

**Ans.** In addition to granting the roles, you would also need to grant execute to the user on the following packages:  
\* grant execute on sys.dbms\_aqin to <userid>  
\* grant execute on sys.dbms\_aqjms to <userid>

##### **Ques 59. What are the three components of a Message?**

**Ans.** A JMS message consists of three parts:  
Message header - For message identification. For example, the header is used to determine if a given message is appropriate for a "subscriber"   
Properties - For application-specific, provider-specific, and optional header fields   
Body - Holds the content of the message. Several formats are supported, including TextMessage, which wrap a simple String, that wrap arbitrary Java objects (which must be serializable). Other formats are supported as well.

##### **Ques 60. What are the types of messaging?**

**Ans.** There are two kinds of Messaging. Synchronous messaging involves a client that waits for the server to respond to a message. Asynchronous messaging involves a client that does not wait for a message from the server. An event is used to trigger a message from a server.

##### **Ques 61. What is the difference between Point to Point and Publish/Subscribe?**

**Ans.** Point-to-point (P2P)   
In point-to-point, messages are sent via queues. Messages are put onto the queues by the message producers (the clients). The message consumer is responsible for pulling the message from the queue. Point-to-point is typically used when a given message must be processed (received) only once by a given consumer. In this way, there is only one consumer of the given message.   
Publish-and-subscribe (pub/sub)  
In publish-and-subscribe, messages are sent through topics. Messages are published to topics by the message producers. The messages may be received by any consumers that subscribe to the given topic. In this way, a message may be received, or processed, by multiple consumers.

##### **Ques 62. Why doesn’t the JMS API provide end-to-end synchronous message delivery and notification of delivery?**

**Ans.** Some messaging systems provide synchronous delivery to destinations as a mechanism for implementing reliable applications. Some systems provide clients with various forms of delivery notification so that the clients can detect dropped or ignored messages. This is not the model defined by the JMS API. JMS API messaging provides guaranteed delivery via the once-and-only-once delivery semantics of PERSISTENT messages. In addition, message consumers can insure reliable processing of messages by using either CLIENT\_ACKNOWLEDGE mode or transacted sessions. This achieves reliable delivery with minimum synchronization and is the enterprise messaging model most vendors and developers prefer. The JMS API does not define a schema of systems messages (such as delivery notifications). If an application requires acknowledgment of message receipt, it can define an application-level acknowledgment message.

##### **Ques 63. What are the core JMS-related objects required for each JMS-enabled application?**

**Ans.** Each JMS-enabled client must establish the following:  
\* A connection object provided by the JMS server (the message broker)  
\* Within a connection, one or more sessions, which provide a context for message sending and receiving  
\* Within a session, either a queue or topic object representing the destination (the message staging area) within the message broker  
\* Within a session, the appropriate sender or publisher or receiver or subscriber object (depending on whether the client is a message producer or consumer and uses a point-to-point or publish/subscribe strategy, respectively). Within a session, a message object (to send or to receive)

##### **Ques 64. How does the Application server handle the JMS Connection?**

**Ans.** 1. App server creates the server session and stores them in a pool.  
2. Connection consumer uses the server session to put messages in the session of the JMS.  
3. Server session is the one that spawns the JMS session.  
4. Applications written by Application programmers creates the message listener.

##### **Ques 1. What is Log4j?**

**Ans.** Log4j (Log for Java) is logging framework provided by apache foundation for java based applications.  
In the applicaitons, if you want to log some information, like any event triggered, or any Database updated is happened, we have the need to log the specific information or error for the useful of the application.  
To debug any issues in applications, we have to log the error/exceptions in the logs. For this we will use log4j mechanism .  
Log4j logs the information and shows this information in different targets. The different targets are called appenders (console, file etc ).

##### **Ques 2. How do you define logging for your application?**

**Ans.** To define logging for your application, you have to download the log4j framework (log4j.jar) from the apache site.  
Once log4j jars are downloaded, make sure that these jars are in classpath of your application. let say you have web-application need to be added log4j. In this case, log4j jars are copied to WEB-INFO/lib folder of your webapplication  
create new file either logging.properties or log4j.xml which will be copied to WEB-INF/classes folder  
logging.properties/log4j.xml contains the all the configuration related to logging mechanism and logger level and package that you want to define logger level.

Example:

**logging.properties:**

logger.level=INFO

logger.handlers=CONSOLE,FILE,RejRec

handler.RejRec=org.jboss.logmanager.handlers.PeriodicRotatingFileHandler

handler.RejRec.level=WARN

handler.RejRec.formatter=RejRec

handler.RejRec.properties=append,autoFlush,enabled,suffix,fileName

handler.RejRec.append=true

handler.RejRec.autoFlush=true

handler.RejRec.enabled=true

handler.RejRec.suffix=.yyyy-MM-dd

handler.RejRec.fileName=E\:\\Docs\\WithoutBook\\DID\\jboss-eap-6.2\\standalone\\log\\RejectedRecords.log

**log4j.xml:**

<log4j:configuration>

<appender name="ASYNC" class="org.apache.log4j.AsyncAppender">

        <appender-ref ref="LOG"/>

</appender>

 <appender name="MEETING-APP-LOG" class="org.apache.log4j.DailyRollingFileAppender"> <param name="File" value="E:/Softwares/glassfish\_dump/glassfish/domains/domain1/logs/meeting\_app.log" /> <param name="Append" value="true"/> <param name="DatePattern" value="'.'yyyy-MM-dd-a"/> <layout class="org.apache.log4j.PatternLayout"> <param name="ConversionPattern" value="%d{yyyy-MM-dd} %d{HH:mm:ss,SSS} %-5p [%t] %c - %m%n" /> </layout> </appender>

 <logger name="com.withoutbook" additivity="false">

        <level value="warn"/>

        <appender-ref ref="MEETING-APP-LOG"/>

</logger>

    <root>

        <priority value="debug"/>

        <appender-ref ref="ASYNC"/>

        <appender-ref ref="ERROR-LOG"/>

    </root>

</log4j:configuration>

##### **Ques 3. What are the different logging levels?**

**Ans.** There are several logging levels that you can configure in you applicaiton   
Those are FATAL,ERROR,WARN,TRACE,DEBUG,INFO OR ALL in apache logging. Default logging level is INFO.

##### **Ques 4. What are different types of logs?**

**Ans.** Usually in any application there two types of logs  
**1.Application server logs :-**These are the logs configured at the application server level. for example in tomcat,  
we have log files called localhost.log. tomcat.log,catalina.log, stdout.log, sterr.log. all these logs are showing with default settings defined in logging.properites located in your tomcat installation folder/conf folder.  
if you want custom settings, we have to change the different parameters in logging.properties in conf folder of tomcat directory.  
  
**2.Application logs:-**We can define logging at each applicaiton level, For this we have to create log4j.xml or logging.properties in WEB-INF/classes folder.

##### **Ques 5. What are different appenders that can configure in log4j?**

**Ans.** There are different appenders that we can configure in log4j are: **CONSOLE, FILES, DATABASE, JMS, EVENT LOGGING**  
  
**Mainly used appenders:**

**CONSOLE in log4j:-**If we use this as appenders in your application, log4j logs the information in the console or command prompt window that is started with startup script.  
**Files in log4j:-**Files Appender is used to log the information into our custom name files. when we are configuring this appender, we can specify the file name.

##### **Ques 6. What is the importance of logging applications?**

**Ans.** Logging helps in debugging as well. Although debuggers are available but frankly it takes time to debug an application using a debugger. An application can be debugged more easily with few well-placed logging messages. So we can safely say that logging is a very good debugging tool. If logging is done sensibly and wisely, it can provide detailed context for application failures.  
  
In distributed applications (e.g. web/remote applications), the logging is very important. The administrator can read logs to learn about the problems that occurred during some interval.  
  
Java's built-in APIs provide logging options but they are not that flexible. Another option is to use Apache’s open source logging framework called log4j.   
  
**Drawbacks of logging applications:**  
There are some drawbacks of use logging in your application. For example: logging will  
- pollute the code  
- increase the size of the code  
- reduce the speed  
  
These are important points because, we ultimately want efficient applications. Log4J has the solution to this. You may turn on or turn off the logging at runtime by changing the configuration file. This means no change in the Java source code (binary).

To reduce the code now in Spring Framework you can use of AOP where you can configure log mechanisms easily.

##### **Ques 7. Describe about logger levels in log4j.**

**Ans.** Before using log4j framework, one should be aware of different categories of log messages. Following are 5 categories:  
  
**DEBUG**  
The DEBUG Level is used to indicate events that are useful to debug an application. Handling method for DEBUG level is: debug().  
  
**INFO**  
INFO level is used to highlight the progress of the application. Handling method for INFO level is: info().  
  
**WARN**  
The WARN level is used to indicate potentially harmful situations. Handling method for WARN level is: warn().  
  
**ERROR**  
The ERROR level shows errors messages that might not be serious enough and allow the application to continue. Handling method for ERROR level is: error().  
  
**FATAL**  
The Fatal level is used to indicate severe events that will may cause abortion of the application. Handling method for FATAL level is: fatal().  
  
If you declare log level as debug in the configuration file, then all the other log messages will also be recorded.   
If you declare log level as info in the configuration file, then info, warn, error and fatal log messages will be recorded.  
If you declare log level as warn in the configuration file, then warn, error and fatal log messages will be recorded.  
If you declare log level as error in the configuration file, then error and fatal log messages will be recorded.  
If you declare log level as fatal in the configuration file, then only fatal log messages will be recorded.

##### **Ques 8. What are the three main components in log4j?**

**Ans.** There are 3 main components that are used to log messages based upon type and level. These components also control the formatting and report place at runtime. These components are:  
  
- loggers  
- appenders  
- layouts

##### **Ques 2. How many different types of JDBC drivers are present? Discuss them.**

**Ans.** There are four JDBC driver types.   
Type 1: JDBC-ODBC Bridge plus ODBC Driver:   
The first type of JDBC driver is the JDBC-ODBC Bridge. It is a driver that provides JDBC access to databases through ODBC drivers. The ODBC driver must be configured on the client for the bridge to work. This driver type is commonly used for prototyping or when there is no JDBC driver available for a particular DBMS.   
Type 2: Native-API partly-Java Driver:   
The Native to API driver converts JDBC commands to DBMS-specific native calls. This is much like the restriction of Type 1 drivers. The client must have some binary code loaded on its machine. These drivers do have an advantage over Type 1 drivers because they interface directly with the database.   
Type 3: JDBC-Net Pure Java Driver:   
The JDBC-Net drivers are a three-tier solution. This type of driver translates JDBC calls into a database-independent network protocol that is sent to a middleware server. This server then translates this DBMS-independent protocol into a DBMS-specific protocol, which is sent   
to a particular database. The results are then routed back through the middleware server and sent back to the client. This type of solution makes it possible to implement a pure Java client. It also makes it possible to swap databases without affecting the client.  
Type 4: Native-Protocol Pure Java Driver   
These are pure Java drivers that communicate directly with the vendor's database. They do this by converting JDBC commands directly into the database engine's native protocol. This driver has no additional translation or middleware layer, which improves performance tremendously.   
What does the "static" keyword mean in front of a variable? A method? A class? Curly braces {}?   
static variable   
- means a class level variable   
static method:   
-does not have "this". It is not allowed to access the not static members of the class.   
can be invoked enev before a single instance of a class is created.   
eg: main   
static class:   
no such thing.   
static free floating block:   
is executed at the time the class is loaded. There can be multiple such blocks. This may be useful to load native libraries when using native methods.   
eg:   
native void doThis(){   
static{   
System.loadLibrary("myLibrary.lib");   
}

##### **Ques 1. What are needed to set-up a cluster with JBoss ?**

**Ans.** Basically starting JBoss with the \'all\' configuration contains everything needed for clustering:  
It has all the libraries for clustering:  
  
JGroups.jar, jboss-cache.jar  
  
Clustered beans (cluster-service.xml)  
  
HA-JNDI  
  
HTTP session replications (tc5-cluster-service.xml)  
  
Farming  
  
HA-JMS

##### **Ques 2. Which component handles cluster communication in JBoss?**

**Ans.** The JGroups framework provides services to enable peer-to-peer communications between nodes in a cluster. It is built on top a stack of network communication protocols that provide transport, discovery, reliability and failure detection, and cluster membership management services.

##### **Ques 3. Is it possible to put a JBoss server instance into multiple cluster at the same time ?**

**Ans.** It is technically possible to put a JBoss server instance into multiple clusters at the same time, this practice is generally not recommended, as it increases the management complexity.

##### **Ques 4. What\'s JBoss cache in short?**

**Ans.** JBossCache enables easy distribution of datasets across your computing environments. It is based on JGroups and enables clustering and high availability of that data. You may choose to distribute the data with JBoss Messaging to move it where it is needed for computation or event-based programming.

##### **Ques 5. What is about Seam?**

##### **Ans.** Built on the standards JavaServer Faces and EJB 3.0, JBoss Seam unifies component and programming models and delivers a consistent and powerful framework for rapid creation of web applications with Java EE 5.0. Seam simplifies web application development and enables new functionality that was difficult to implement by hand before, such as stateful conversations, multi-window operation, and handling concurrent fine-grained AJAX requests. Seam also unifies and integrates popular open source technologies like Facelets, Hibernate, iText, and Lucene. **Ques 6. Does Seam run on other application servers besides JBoss?**

**Ans.** Seam runs beautifully on other application servers - just like everything else the Hibernate team does, this is not a JBoss-only thing.

##### **Ques 7. What\'s JBoss JBPM?**

**Ans.** JBoss JBPM is a platform for process languages. At the base there is a java library to define and execute graphs. The actual process constructs like e.g. send email, user task and update database are defined on top of this. Every process language is made up of a set of such process constructs. And that is what is pluggable in this base library. On top of the JBoss jBPM base library, there are implemented several process languages as a set of process constructs: jPDL, BPEL and SEAM pageflow:  
  
jPDL is a process language with a clean interface to Java and very sophisticated task management capabilities. There is no standard for Java process languages, so it is proprietary.  
  
BPEL is a service orchestration language. As said before, in BPEL, you can write new services as a function of other services. This is typically a component of an Enterprise Service Bus (ESB).  
  
SEAM pageflow is a language that allows for the graphically define the navigation between pages in a SEAM web application.

##### **Ques 9. What is JBoss?**

**Ans.** JBoss is a popular open source application server based on Java EE technology. Being Java EE based, the JBoss supports cross-platform java applications. It was embedded with Apache Tomcat web server. It runs under any JVM of 1.3 or later versions. JBoss supports JNDI, Servlet/JSP (Tomcat or Jetty), EJB, JTS/JTA, JCA, JMS, Clustering (JavaGroups), Web Services (Axis), and IIOP integration (JacORB).

##### **Ques 10. What version of JBoss AS do I need to run Seam?**

**Ans.** For Seam 1.3: Seam was developed against JBoss 4.2. Seam can still be run against JBoss 4.0. The seam documentation contains instructions for configuring JBoss 4.0.  
  
For Seam 1.2: Since Seam requires the latest edition of EJB3, you need to install JBoss AS from the latest JEMS installer. Make sure that you select the “ejb3″ or “ejb3+clustering” profile to include EJB3 support. Also, the jboss-seam.jar library file from the Seam distribution must be included in each Seam application you deploy. Refer to examples in Seam distribution (inside the examples directory) to see how to build and package Seam applications.

##### **Ques 11. Can I run Seam outside of JBoss AS?**

**Ans.** Yes, you can run Seam applications in plain Tomcat 5.5+ or in the Sun GlassFish application server. To run Seam application in Tomcat, you need a number of additional library files and a few configuration files to bootstrap the JBoss EJB3 inside Tomcat. Please refer to the deploy.tomcat ANT build target for the Seam booking example (in the examples/booking directory of the Seam distribution) for more on how to build a Tomcat WAR for Seam applications. Refer to this blog post on how to run Seam in Sun’s Glassfish application server.

##### **Ques 12. Can I run Seam in a J2EE environment?**

**Ans.** Yes, as of Seam 1.1, you can use Seam in any J2EE application server, with one caveat: you will not be able to use EJB 3.0 session beans. However, you can use either Hibernate or JPA for persistence, and you can use Seam JavaBean components instead of session beans.

##### **Ques 13. Can I run Seam with JDK 1.4 and earlier?**

**Ans.** No, Seam only works on JDK 5.0 and above. It uses annotations and other JDK 5.0 features.

##### **Ques 14. How to build clusters in JBoss?**

**Ans.** here is a nutshell summary from the example given in the .org quick start:   
  
from your jboss bin directory:   
$ ./run.sh -c all -g DocsPartition -u 239.255.100.100 -b 192.168.0.101 -Djboss.messaging.ServerPeerID=1   
  
./run.sh = the executable   
-c all = use server in jboss\_home/server/all directory   
-g DocsPartition = this is the partition name, use whatever but should be the same across nodes   
-u 239.255.100.100 = this is the multicast address, ususally just pick anything in the 239.255.x.y range and you should be ok   
-b 192.168.0.101 = this is the ip of the machine the node is on   
-Djboss.messaging.ServerPeerID=1 = this is the peerid, it must be unique for each node,   
  
so the start script that you would run on your second machine for the second node would look like:   
$ ./run.sh -c all-node2 -g DocsPartition -u 239.255.100.100 -b 192.168.0.102 -Djboss.messaging.ServerPeerID=2   
  
you can also use nohup to send the process to the background...   
  
now that both nodes should be running, you have to enable and configure the sticky sessions on thw webserver and each server.

##### **Ques 2. What are the components of logical database structure of Oracle database?**

**Ans.** There are tablespaces and database's schema objects.

##### **Ques 3. What is a tablespace?**

**Ans.** A database is divided into Logical Storage Unit called tablespaces. A tablespace is used to grouped related logical structures together.

##### **Ques 4. What is SYSTEM tablespace and when is it created?**

**Ans.** Every Oracle database contains a tablespace named SYSTEM, which is automatically created when the database is created. The SYSTEM tablespace always contains the data dictionary tables for the entire database.

##### **Ques 6. What is schema?**

**Ans.** A schema is collection of database objects of a user.

##### **Ques 7. What are Schema Objects?**

**Ans.** Schema objects are the logical structures that directly refer to the database's data. Schema objects include tables, views, sequences, synonyms, indexes, clusters, database triggers, procedures, functions packages and database links.

##### **Ques 10. What is Oracle table?**

**Ans.** A table is the basic unit of data storage in an Oracle database. The tables of a database hold all of the user accessible data. Table data is stored in rows and columns.

##### **Ques 11. What is an Oracle view?**

**Ans.** A view is a virtual table. Every view has a query attached to it. (The query is a SELECT statement that identifies the columns and rows of the table(s) the view uses.)

##### **Ques 21. What are the type of Synonyms?**

**Ans.** There are two types of Synonyms Private and Public.

##### **Ques 25. What are Clusters?**

**Ans.** Clusters are groups of one or more tables physically stores together to share common columns and are often used together.

##### **Ques 26. What is an Integrity Constrains?**

**Ans.** An integrity constraint is a declarative way to define a business rule for a column of a table.

##### **Ques 27. What is an Index?**

**Ans.** An Index is an optional structure associated with a table to have direct access to rows, which can be created to increase the performance of data retrieval. Index can be created on one or more columns of a table.

##### **Ques 2. Why is software development life cycle important?**

**Ans.** SDLC serves as a guide to the project and provides a flexible and consistent medium to accommodate changes, and perform the project to meet client’s objectives. SDLC phases define key schedule and delivery points which ensure timely and correct delivery to the client within budget and other constraints and project requirements. SDLC co-operates project control and management activities as they must be introduced within each phase of SDLC.

##### **Ques 3. What are the different phases in SDLC?**

**Ans.**

There are 5 phases in Software Development Life Cycle:

1. Requirement & analysis  
2. Design  
3. Coding  
4. Testing  
5. Maintenance

##### **Ques 4. What is SDLC model? What are the most well known SDLC models?**

**Ans.**

An SDLC model defines implementation of an approach to the project. It defines the various processes, and phases that would be carried out throughout the project to produce the desired output. There are a variety of SDLC models that exist catering to different needs and characteristics of a project. Some are of iterative nature (Prototyping), whereas some are sequential (waterfall). Some of the well known SDLC models are:

* Waterfall Model
* Iterative Model
* Spiral Model
* V-Model
* RAD Model
* Agile Model

##### **Ques 5. Describe waterfall software development life cycle model.**

**Ans.**

Waterfall is a sequential and non iterative SDLC model which describes flowing of phases downwards one by one. The process does not start a phase unless the previous phase is completed once and for all completely. The waterfall model consists of the following phases:

Requirements gathering  
Design  
Implementation  
Testing  
Maintenance

##### **Ques 6. Briefly describe the phases in the waterfall model.**

**Ans.** **Requirements gathering:** All the requirements are gathered and analysis is performed for the complete system.  
**Design:** Various design models are created for the complete system after the requirements gathering phase has been completed and ended.  
**Implementation:** The complete system is implemented once the design for the system has been frozen.  
**Testing:**The complete system is tested after all the construction and integration has completed.  
**Maintenance:**Post implementation support carries out after implementation of the system.

##### **Ques 7. Explain the strengths of the waterfall model.**

**Ans.** Strengths of the waterfall model are:  
a) No planning needed  
b) Works well for small projects with fixed and clear requirements.  
c) Lesser cost as planning overhead is less  
d) Quickest delivery of the complete system

##### **Ques 8. Explain the weaknesses of the waterfall model.**

**Ans.**

Weaknesses of waterfall model are:

a) It is Inflexible  
b) Accommodating changes is very hard  
c) Longest tangible delivery time. The customer does not see anything but the whole product when it’s ready.  
d) Unsuitable for large projects and where requirements are not clear.

##### **Ques 9. Explain when to use the waterfall model.**

**Ans.** One should use the waterfall model only when:

* Requirements are very clear and fixed.
* There are no ambiguous requirements.
* Ample resources with required expertise are available freely.
* The client has high confidence in the organization.
* The organization has experience of similar projects.
* The project is short.

##### **Ques 10. Describe V-Shaped software development life cycle model.**

**Ans.**

The V-shaped SDLC model is an extension of the waterfall model. The typical waterfall moves linearly downwards, whereas, in V-shaped model phases are turned upwards after coding phase to form the V shape. It demonstrates relationship between each phase of SDLC and its respective testing phase. Unlike waterfall model, the V-Shape includes early test planning.

Requirement analysis---------------------------------------Acceptance testing  
              System design------------------------------------System testing  
               Architecture design -----------------------------Integration testing  
                            Module design----------------------------Unit testing  
                                                                  **Coding**

##### **Ques 11. Briefly describe the phases in the V-Shaped model.**

**Ans.**

Phases in V-Shaped model:  
**Verification phases**are on the left side of the V-shape. It consists of:

**Requirements analysis:**Requirements are gathered and analysis is performed to understand the problem and propose a solution.  
**System Design:** Engineers analyze the requirements gathered and propose ways the system can be created or built from a feasibility point of view.  
**Architecture design:** Architecture of the system is designed consisting of various modules, depicting their relationships and communication between them.   
**Module design:** This is a low level design where modules are designed individually and in a detailed manner.

**Coding:** This is at the bottom of the V-Shape model. Module design is converted into code by developers.

**Validation phases** are on the right side of the V-shape. It consists of:  
**Unit testing:** Testing by analysis of the code by developers for their independent modules is done.  
**Integration testing:** Independent modules are tested together to validate interface and expose errors in them.  
**System testing:** The system is tested against the system specifications.  
**User Acceptance testing:**Testing is performed by end users to validate that the requirements mentioned in requirements phase have been met by the system or not before accepting it for production.

##### **Ques 12. Explain the strengths of the V-Shaped model.**

**Ans.**

Strengths of V-Shaped model:

a) Simple and easy to use model.  
b) Every phase has clear and fixed deliverables.  
c) Higher chances of success as test planning starts early in the SDLC cycle.  
d) Quickest for project where requirements are fixed and clearly defined.

##### **Ques 13. Explain the weaknesses of the V-Shaped model.**

**Ans.** Weaknesses of V-Shaped model:  
a) It is inflexible.  
b) Changes in requirements are very hard to accommodate  
c) No early prototypes are available  
d) Needs ample skilled resources.

##### **Ques 14. Explain when to use the V-Shaped model.**

**Ans.** The V-shaped model should be used for small to medium sized projects where requirements are clearly defined and fixed. The model accommodates more planning for test than waterfall but makes accommodation of changes harder than other models. The V-Shaped model should be chosen when ample technical resources are available with needed technical expertise. Since, no prototypes are produced, there is a very high risk involved in meeting customer expectations, therefore, confidence of customer should be very high in order for choosing the V-Shaped model approach.

##### **Ques 15. Describe prototype software development life cycle model.**

**Ans.** Prototype SDLC models is based upon creation of a software prototype of the complete system and then refine and review it continuously till the complete acceptable system is built.

##### **Ques 16. Briefly describe the phases in the Prototype model.**

**Ans.** **Phases in prototype model:**  
**Identify some requirements to begin with:** Get a list of some major requirements which define the need for the new system including the main input output information.  
**Develop initial prototype:** Develop a basic initial prototype which only has UI screens.  
**Review the prototype:** End users and SME’s work and examine the prototype and provide feedback for improvements/enhancements.  
**Revise and enhance the prototype:** Scope is changed based on feedback from end users and the prototype is enhanced and refined to accommodate user feedback.

##### **Ques 17. Explain the strengths of the prototype model.**

**Ans.**

Strengths of prototype model are:

a) Gains customer’s confidence as developers and customers are in sync with each other’s expectations continuously.  
b) Ideal for online systems where high level of human computer interaction is involved.  
c) Very flexible, as changes in requirements can be accommodated much more easily with every new review and refining.  
d) Helps the developers and users both understand the system better.  
e) Software built through prototyping needs minimal user training as users get trained using the prototypes on their own from the very beginning of the project.  
f) Integration requirements are very well understood and deployment channels are decided at a very early stage

##### **Ques 18. Explain the weaknesses of the prototype model.**

**Ans.**

Weaknesses of the Prototype model are:

a) Focusing on the prototype can mislead developers from understanding the actual desired system.  
b) End users get confused, believing the prototype to be the complete system  
c) Developers might misunderstand end users objectives.  
d) Developer might get too involved in prototype and deviate from the actual system that the prototype must be converted into.  
e) Expensive as prototypes need a lot of effort and time. It may take a lot of work to be done for very less needed work to be achieved.

##### **Ques 19. Explain when to use the Prototype model.**

**Ans.** Prototype model should be used when the desired system needs to have a lot of interaction with the end users. Typically, online systems, web interfaces have a very high amount of interaction with end users, are best suited for Prototype model. It might take a while for a system to be built that allows ease of use and needs minimal training for the end user. Prototyping ensures that the end users constantly work with the system and provide a feedback which is incorporated in the prototype to result in a useable system. They are excellent for designing good human computer interface systems.

##### **Ques 20. Describe rapid application development (RAD) software development life cycle model.**

**Ans.** RAD involves iterative development along with creation of prototypes. It uses interactive use of techniques and prototypes to define user’s requirements and system design clearly. Structured techniques are used to create initial design models based on user input and prototypes are built on top of that. The end users and analysts use the prototypes to validate and enhance the requirements and design models. The process lasts till a set of final technical requirements and design models have been created.

##### **Ques 21. Briefly describe the phases in the rapid application development (RAD) model.**

**Ans.**

**Phases in RAD:**

**Business modeling:** The information flow is identified between various business functions.  
**Data modeling:** Information gathered from business modeling is used to define data objects that are needed for the business.  
**Process modeling:**Data objects defined in data modeling are converted to achieve the business information flow to achieve some specific business objective. Description are identified and created for CRUD of data objects.  
**Application generation:**Automated tools are used to convert process models into code and the actual system.  
**Testing and turnover:**Test new components and all the interfaces

##### **Ques 22. Explain the strengths of the rapid application development (RAD) model.**

**Ans.**

Strengths of RAD:

a) Reduced development time.  
b) Increases reusability of components  
c) High modularization achieves a more flexible and maintainable system  
d) Quick initial reviews occur.

e) Encourages customer feedback  
f) Integration from very beginning solves a lot of integration issues.  
g) Business owners actively participate

##### **Ques 23. Explain the weaknesses of the Rapid Application Development (RAD) model.**

**Ans.**

Weaknesses of RAD:

a) Depends on strong team and individual performances for identifying business requirements.  
b) Only system that can be modularized can be built using RAD  
c) Requires highly skilled developers/designers.  
d) High dependency on modeling skills  
e) Inapplicable to cheaper projects as cost of modeling and automated code generation is very high for cheaper budgeted projects to befit.

##### **Ques 24. Explain when to use the rapid application development (RAD) model.**

**Ans.** RAD should be used when there is a need to create a system that can be modularized in 2-3 months of time. It should be used if there’s high availability of designers for modeling and the budget is high enough to afford their cost along with the cost of automated code generating tools. RAD SDLC model should be chosen only if resources with high business knowledge are available and there is a need to produce the system in a short span of time (2-3 months).

##### **Ques 25. Describe incremental software development life cycle model.**

**Ans.** Incremental SDLC approach suggests construction of a partial system rather than the complete system and then builds more functionality into it. Requirements and features are prioritized and categorized and then implemented in phases, each phase based on the waterfall model. The process continues till the complete system is achieved.

##### **Ques 26. Briefly describe the phases in the Incremental model.**

**Ans.** Phases of incremental model are same as waterfall i.e. Requirements, design, implementation, testing, maintenance. However, instead of following the waterfall once and for all linearly, incremental model takes a different approach. In this phases are repeated incrementally as business value is delivered incrementally as well.   
For every single phase and increment a waterfall model is followed. The waterfall model is then put in a cycle of increments along with verification of requirements, and design.

##### **Ques 27. Explain the strengths of the Incremental model.**

**Ans.**

Strengths of incremental model are:

a) Develop high risk business features first  
b) Every increment delivers an operational product  
c) Customer’s confidence is high as they validate every increment and provide feedback  
d) Low initial delivery cost  
e) Changes in requirements can be accommodated for easily.  
f) More flexible than waterfall

##### **Ques 28. Explain the weaknesses of the Incremental model.**

**Ans.** Weaknesses of incremental model are:  
a) Needs good planning and design.  
b) Needs a clear and complete definition of the complete system before it can be broken down and built incrementally.  
c) Integration needs are very high  
d) Total cost is higher than waterfall.

##### **Ques 29. Explain when to use the Incremental model.**

**Ans.** Incremental model should only be used when:  
Requirements of the complete system are clearly defined and understood.   
Major requirements must be defined; however, some details can evolve with time.   
There is a need to get a product to the market early.  
A new technology is being used  
Resources with needed skill set are not available  
There are some high risk features and goals.

##### **Ques 30. Describe spiral software development life cycle model.**

**Ans.** The spiral SDLC model combines components of both design and prototype in phases. It’s a hybrid of waterfall and prototyping model. One should use spiral SDLC model for large and expensive projects.

##### **Ques 31. Briefly describe the phases in the Spiral model.**

**Ans.**

Phases in spiral model:

a) System requirements are identified in detail.  
b) An initial design is created for the new system based on the requirements in previous phase. All feasible and technical approaches are identified and analyzed to build the system. The design is done on a very broader and deeper scope to identify and deal with potential risks in the system.  
c) A prototype is created depicting a few features of the system.  
d) A second prototype is created using 4 steps: Evaluate first prototype, define requirements for second prototype, planning and designing for second prototype, constructing and testing second prototype.

##### **Ques 32. Explain the strengths of the spiral model.**

**Ans.**

Strengths of the Spiral model:

a) Early identification of potential risk areas.  
b) Customer sees a prototype very early in the SDLC.  
c) Critical and risky features are built first for risk mitigation and clarifying requirements.  
d) Design can evolve with iterations.  
e) Feedback from users helps maintain their expectations.  
f) Cost is assessed frequently, hence better planning.

##### **Ques 33. Explain the weaknesses of the spiral model.**

**Ans.** Weaknesses of Spiral model are:  
a) Not suitable for smaller or low budget projects as cost is high for identifying risks.  
b) Time spent on risks, planning, and prototyping may not be as efficient.  
c) It is complex.  
d) Spiral may continue indefinitely.  
e) Hard to define clear milestones, which allow the SDLC to move to the next phase.  
f) Developers must have other work during non development phases.

##### **Ques 34. Explain when to use the spiral model.**

**Ans.**

Spiral model should be used when:

a) Prototypes are expected/needed.  
b) Large and high budget projects  
c) When risk assessment is very critical  
d) Requirements are not very clearly defined.  
e) Requirements are vague and even complex  
f) The organization does not have much experience with the domain.  
g) Ample time is available.

##### **Ques 35. Describe tailored software development life cycle model.**

**Ans.** There is no specific SDLC model that can be used for all types of projects and situations. If none of the popular SDLC models suit for a specific project then, pick the closest matching SDLC model and modify it as per needs. Identify how important is risk assessment and use spiral’s risk assessment methodology if it’s a risk critical project. Project should be delivered in small chunks, ideally merging the incremental model with V-shaped model. One must spend ample time in choosing the right model or customizing one to suit a project for its successful and efficient completion.

##### **Ques 36. Describe the importance of selecting team members with a mix of personality types for software development.**

**Ans.**

Choose or building the right team is vital for the success of any project. A project needs a variety of skills and qualities which are not present in any individual. However, as a workaround, a team should be built of people with a variety of skill sets to fulfill the project need. The main advantage of choosing team members with a mix of personality types is that it provides a wider range of views towards a project or any specific action item in the project, e.g. : requirements, design, development, testing or even implementation. Different views allow for a broader angle to a problem and solution minimizing the risk of missing requirements or misunderstanding them.

Some of the personality traits that are essential to any project are:

a) Aggressive go getter, contrary, a calm patient and more laid back personality  
b) Risk taker, contrary, a cautious personality  
c) Strategic, contrary, analytical personality  
d) Lateral thinking

Different situations in a project are handled better by different personality types and hence a perfect blend/mix of personality types is essential for the project to complete successfully.

##### **Ques 37. Describe the phases of team development in SDLC.**

**Ans.**

The 4 stages of team building are:

**Forming:** Team members are told what is expected out of them and where do they fit in the team. The team is guided using operating guidelines, and communication within.

**Storming:** During this phase team members show some resistance and frustrations trying to work together. There would be jealousies and ego clashes and the team manager must act as a referee or a coach.

**Norming:** In this phase the team has learnt to function as a whole. Team members find their consistent ways of working and hold their ideas back to avoid problems and conflicts. The team manager guides the team to not hold themselves back by increasing responsibilities and pressures.

**Performing:** In this phase the team has learnt to perform their role as whole, have and resolve conflicts, take risks, make adjustments or compromises, and perform actively to face various challenges.

##### **Ques 38. What is the difference between an Iterative model and the Waterfall model?**

**Ans.** Waterfall Model is a flow based model, in which we pass every phase once, and can not go back to that phase again. Its most eminent drawback is that if there is any change in requirements, we cannot make any changes to the requirement section. Iterative Model is somewhat similar to waterfall model but herein we can always come back to previous phases, and make the changes accordingly.

##### **Ques 39. Explain the Difference between SDLC and STLC?**

**Ans.** SDLC is a software development life Cycle model which is utilized for project management and involves processes from the feasibility Analysis to maintenance of the completed application. STLC is Software testing Life cycle and SDLC work closely together and are almost inseparable under some of the activities. However the stages are very different under sdlc and stlc

##### **Ques 40. What are functional requirements?**

**Ans.** Functional requirement is a document which contains what a certain system has to do to achieve a certain specific objective.This task is carried out during the preliminary stage of SDLC.

##### **Ques 41. What are Non-functional requirements?**

**Ans.** Without Non-functional, a software will never function or will have vital missing information in its output. Response time, security, reliability, accuracy, capacity and availability are examples of Non functional requirement for a software development process. Non functional requirements decides how the Program or the software will function in future.

##### **Ques 42. What is the difference between Incremental model and Spiral model?**

**Ans.** There is not much difference between these two sdlc models. Sdlc spiral Model includes the iterative nature of the prototyping model and the linear nature of the waterfall model. This approach is ideal for developing software that is revealed in various versions.

##### **Ques 43. Give some practical real life examples of Spiral Model.**

**Ans.** The most popular real life examples for sdlc Spiral model are Microsoft Windows operating System, Visual Studio Manager, Adobe Photoshop, WordPress CMS and many more.

##### **Ques 44. Why is Agile so popular?**

**Ans.** Agile methodology is way too advanced and complex than the simple Waterfall model. The feasibility of agile to reshape the entire development structure to suit the most effective outcome is what makes Agile the number 1 choice of developers today.

##### **Ques 45. Can I build a software project without sdlc models?**

**Ans.** Of course. There is no hard and fast requirements for a developer to implement any sdlc model for developing a software project. The ability to simplify project into modules and ascertain correct progression for completion is the only reason for which sdlc models and methodology was designed in the first place. You can sure work without them but the challenges will be more and there won’t be any specific process to organize your work as a whole.