Devops

Jenkins Interview Questions

By Devops - March 27, 2017

I wrote a blog post to help prepare people for their DevOps interview. These questions are all gathered from different blogs and sites. I have mentioned the site name, from where I have copied. Thanks to all those authors and blog writers.
If you have your own interview experiences, please share it with me in the comments section or functionh@gmail.com along with the company name.

We will give the credits to you

http://www.kerneltraining.com/blog/top-10-devops-interview-questions-answers
Here are Top 10 interview questions and answers, which will help you to face devops interview:
1.Explain the working of HTTP?
This is the most frequently asked devops interview questions. This question will help the interviewer to check your software ability. You can simply answer .Like other protocols, HTTP also works on the client-server model. A web server software which responds to the request is called a server and a web browser which initiates the request is called a client. HTTP enhances its request and response with the help of intermediates such as tunnel, proxy or gateway. URL helps in allocating the resources, that are requested using HTTP. The connection to the application layer port of HTTP is provided by TCP.
2.How you would prepare for migration?

This question shows your experience in real projects, which brings complexity, awkwardness and include other things like roll back and forward, feature toggles, cut over, DNS solutions, branch by abstraction, dress rehearsals and automation in your response. Dealing with legacy configuration and components are always tougher than developing a greenfield system with no existing technology.

3.How you would make software deployable?

The ability to script the reconfiguration and installation of software systems is essential towards automated and controlled change. Older products and systems are supposing that the changes would be minor and infrequent and so the automated changes become difficult, even though there is an increasing style for new software to enable this. In order to expose settings and configuration in a way accessible to automation, the professional has to work with concepts such as scripted installation, separation of concerns, infrastructure as a code, command – tools, test harnesses, dependency injection and inversion of control.

4.What is the difference between RAID0 and RAID1?

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

5.What is an MX record in DNS?

MX records are nothing but the mail exchange records which is used to determine the priority of email servers for a domain. The priority of email servers is divided into two categories, lowest priority email servers and higher priority email servers. The first destination for email is known as the lowest priority email servers, the mail will be sent to the highest priority email servers if the lowest priority email server is unavailable.

6.How would you ensure traceability?

This question examines the candidate's attitude to transaction journeys, reporting, logging and metrics. The candidate must know that monitoring, logging and metrics are the core parts of the software system and without them the software is just like undiagnosable and unmaintainable. Add the words like SCOM, Error tracking, Splunk, SYSLOG, Avicode, Nagios and so on in your response.

7.What is a PTR in DNS?

PTR is nothing but the pointer records which is used to map a network interface to a host name. The setup of reverse DNS is similar to the forward DNS but PTR's are primarily used for reverse DNS. When you assign the DNS forward, the domain owner informs the registrar to let your domain to use specific server names.

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

The advantages of the NoSQL database over RDBMS are

- Ability to scale horizontally
- Breadth of functionality
- Support for unstructured text
- Choice of vendors
- Less need for ETL
- Ability to handle change over time
- Support for multiple data structures

9.What types of testing are needed?

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

10.Are you more DEV or Ops?

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

HTTPS://LINUXAWS.WORDPRESS.COM/2016/06/04/IMPORTANT-AWS-DEVOPS-INTERVIEW-QUESTIONS-TO-ASK/

WHAT IS VPC ?

A ***virtual private cloud*** (VPC) is a virtual network dedicated to your AWS account. You can configure or create your VPC as per requirement like select region, create subnets (IP- CIDR), configure route tables, security groups, Internet gateway etc to your AWS account By which you can launch your AWS resources, such as Amazon EC2, RDS instances etc, into your VPC.
So basically you can say that Amazon VPC is the networking layer for AWS Infrastructure.

WHAT IS VPC PEERING?

A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them using private IP addresses. And instances which is in VPC can communicate with each other as if they are within the same network.
You can create a VPC peering connection between your own VPCs, or with a VPC in another AWS account within a single region.
If you have more than one AWS account within a same region and wants to share or transfer the data, you can peer the VPCs across those accounts to create a file sharing network. You can also use a VPC peering connection to allow other VPCs to access resources you have in one of your VPCs.
A VPC peering connection can help you to facilitate the transfer of data.

WHAT IS VPC ENDPOINTS?

A VPC endpoint enables you to create a private connection between your VPC with another AWS service without requiring access over the Internet, through a NAT device, a VPN connection, or AWS Direct Connect. They are horizontally scaled, redundant, and highly available VPC components that allow communication between instances in your VPC and AWS services without imposing availability risks or bandwidth constraints on your network traffic.
An endpoint enables instances in your VPC to use their private IP addresses to communicate with resources in other services. Don't require public IP addresses to your instances, and you don't need an Internet gateway, a NAT device, or a virtual private gateway in your VPC.

WHAT IS EBS (ELASTIC BLOCK STORAGE)? WHAT TYPE OF PERFORMANCE CAN YOU EXPECT? HOW DO YOU BACK IT UP? HOW DO YOU IMPROVE PERFORMANCE?

AMAZON ELASTIC BLOCK STORAGE

EBS is a virtualized SAN or storage area network. Elastic Block Store (Amazon EBS) provides persistence block level storage volumes for use with EC2 instances. EBS volumes are highly available and reliable storage volumes that can be attached to any running instance that is in the same Availability Zone.

Performance that we can expect: Performance on EBS can exhibit variability. That is it can go above the SLA performance level, then drop below it. The SLA provides you with an average disk I/O rate you can expect. This can frustrate some folks especially performance experts who expect reliable and consistent disk throughput on a server. Traditional physically hosted servers behave that way. Virtual AWS instances do not.
Amazon EBS offering high avialibilty & durability. And it offers the consistent & low-latency performence needed to run your workloads.

EBS Magnetic volumes : You can create EBS Magnetic volumes from 1 GiB to 1 TiB in size

EBS General Purpose SSD (gp2) : You can create **EBS General Purpose SSD (1 GiB – 16 TiB)**
Provisioned IOPS SSD (io1): Highest-performance SSD volume designed for mission-critical applications (4 GiB – 16 TiB)
Cold HDD (sc1): Lowest cost HDD volume designed for less frequently accessed workloads (**500 GiB – 16 TiB**)
Amazon EBS Encryption: You can use encrypted EBS volumes to meet a wide range of data-at-rest encryption requirements for regulated/audited data and applications.
Amazon EBS Snapshots: You can create point-in-time snapshots of EBS volumes, which are persisted to Amazon S3. Snapshots protect data for long-term durability, and they can be used as the starting point for new EBS volumes. The same snapshot can be used to instantiate as many volumes as you wish. These snapshots can be copied across AWS regions.

Performance metrics, such as bandwidth, throughput, latency, and average queue length, are available through the AWS Management Console. These **metrics, provided by AmazonCloudWatch,** allow you to monitor the performance of your volumes to make sure that you are providing enough performance for your applications without paying for resources you don't need.

WHAT IS S3? WHAT IS IT USED FOR? SHOULD ENCRYPTION BE USED IN S3?

Amazon S3 is stand for Simple storage service that is storage for the Internet. It as a, "simple storage service that offers software developers a highly-scalable, reliable, and **low-latency** data storage infrastructure at very low costs". Amazon S3 provides a simple web service interface which you can use to store and retrieve any amount of data, at any time, from anywhere on the web. Using this web service, developers can easily build applications that make use of Internet storage.

You can think of it like ftp storage, where you can move files to and from there, but not mount it like a file system. AWS automatically puts your snapshots there, as well as AMIs there. Encryption should be considered for sensitive data, as S3 is a proprietary technology developed by Amazon themselves, and as yet unproven vis-a-vis a security standpoint.

Encryption should be considered for sensitive data, as S3 is a proprietary technology developed by Amazon themselves, and yet to be proven from a security standpoint.

WHAT IS AN AMI?

AMI stands for Amazon Machine Image. It is effectively a snapshot of the **root filesystem.** AWS AMI provides the information required to launch an instance, which is a virtual server in the cloud. You specify an AMI when you launch an instance, and you can launch as many instances from the AMI as you need. You can also launch instances from as many different AMIs as you need.
An AMI includes the following:

- A template for the root volume for the instance (such as an operating system, an application server, and applications)
- Launch permissions that control which AWS accounts can use the AMI to launch instances
- A block device mapping that specifies the volumes to attach to the instance when it's launched

Build a new AMI by first spinning up and instance from a trusted AMI. Then adding packages and components as required. Be wary of putting sensitive data onto an AMI. For instance your access credentials should be added to an instance after spinup. With a database, mount an outside volume that holds your MySQL data after spinup as well.

Ref: http://docs.aws.amazon.com

WHAT IS THE RELATION BETWEEN INSTANCE AND AMI?

An Amazon Machine Image (AMI) is a template that contains a software configuration (for example, an operating system, an application server, and applications). From an AMI, you launch an instance, which is a copy of the AMI running as a virtual server in the cloud.

You can launch different types of instances from a single AMI. An instance type determines the hardware of the host computer used for your instance. Each instance type offers different compute and memory capabilities.

WHAT AUTOMATION TOOLS CAN YOU USE TO SPINUP SERVERS?

Here below many types tools given any of the following tools can be used:

- Roll-your-own scripts, and use the AWS API tools. Such scripts could be written in bash, perl or other language or your choice.
- Use a configuration management and provisioning tool like Ansible, puppet or its successor Opscode Chef etc.
- You might also look towards a tool like Scalr. Lastly you can go with a managed solution such as Rightscale.

WHAT ARE THE DIFFERENT DEPLOYMENT MODELS FOR CLOUD?

The different models are:

- Private Cloud
- Public Cloud
- Hybrid Clouds

WHAT IS AUTO-SCALING? HOW DOES IT WORK?

- Horizontally Scaling
- Vertically Scaling

Auto scaling is a feature of AWS which allows you to configure and automatically provision and spinup new instances without the need for your intervention. You can do this by setting thresholds and metrics to monitor. When those thresholds are crossed, a new instance of your choosing will be spun up, configured, and rolled into the load balancer pool. You've scaled horizontally without any operator intervention!

Vertically Scaling: This is an incredible feature of AWS and cloud virtualization. Spinup a new larger instance than the one you are currently running. Pause that instance and detach the root ebs volume from this server and discard. Then stop your live instance, detach its root volume. Note the unique device ID and attach that root volume to your new server. And the start it again. You have scaled vertically in-place!!

WHAT IS THE DIFFERENCE BETWEEN SCALABILITY AND ELASTICITY?

Scalability is the ability of a system to increase the workload on its current hardware resources to handle variability in demand.

Elasticity is the ability of a system to increase the workload on its current and additional hardware resources, thereby enabling businesses to meet demand without investing in infrastructure up-front.

LIST OUT DIFFERENT LAYERS WHICH DEFINE CLOUD ARCHITECTURE?

There are five layers:

- Cloud Controller (CLC)
- Walrus
- Cluster Controller
- Storage Controller (SC)
- Node Controller (NC)

WHAT ARE THE SECURITY LAWS WHICH ARE IMPLEMENTED TO SECURE DATA IN A CLOUD?

The security laws which are implemented to secure data in cloud are:

- Processing
- File
- Output reconciliation
- Input Validation
- Security and Backup

WHY API'S HAVE IN CLOUD SERVICES?

Application Programming Interface (API) has the following uses:

- It eliminates the need to write fully fledged programs
- It provides the instructions to set up communication between one or more applications
- It allows easy creation of applications and links the cloud services with other systems

HOW MANY DATA CENTERS ARE DEPLOYED FOR CLOUD COMPUTING? WHAT ARE THEY?

There are two data centers in cloud computing:

- Containerized Data centers
- Low Density Data centers

WHAT ARE THE SECURITY FOR AMAZON EC2?

There are several best practices for secure Amazon EC2. A few of them are given below:

- Use AWS Identity and Access Management (IAM) to control access to your AWS resources.
- Restrict access by only allowing trusted hosts or networks to access ports on your instance.
- Review the rules in your security groups regularly, and ensure that you apply the principle of least
- Privilege – only open up permissions that you require.
- Disable password-based logins for instances launched from your AMI. Passwords can be found or cracked, and are a security risk.

HOW YOU WOULD SIMULATE PERIMETER SECURITY USING AMAZON WEB SERVICES MODEL?

Traditional perimeter security that we're already familiar with using firewalls and so forth is not supported in the Amazon EC2 world.

AWS supports security groups. One can create a security group for a jump box with ssh access – only port 22 open. From there a web server group and database group are created.

The web server group allows 80 and 443 from the world, but port 22 *only* from the jump box group. Further the database group allows port 3306 from the web server group and port 22 from the jump box group. Add any machines to the web server group and they can all hit the database.
No one from the world can, and no one can directly ssh to any of your boxes.

Want to further lock this configuration down? Only allow ssh access from specific IP addresses on your network, or allow just your subnet.

HOW IS BUFFER USED IN AMAZON WEB SERVICES?

Buffer is used to make the system more resilient to burst of traffic or load by synchronizing different components. The components always receive and process the requests in an unbalanced way. Buffer keeps the balance between different components and makes them work at the same speed to provide faster services.

WHAT IS THE FUNCTION OF AMAZON ELASTIC COMPUTE CLOUD?

Amazon Elastic compute cloud also known as Amazon EC2 is an Amazon web service that provides scalable resources and makes the computing easier for developers. The main functions of Amazon EC2 are:

- It provides easy configurable options and allow user to configure the capacity.
- It provides the complete control of computing resources and let the user run the computing environment according to his requirements.
- It provides a fast way to run the instances and quickly book the system hence reducing the overall time.
- It provides scalability to the resources and changes its environment according to the requirement of the user.
- It provides varieties of tools to the developers to build failure resilient applications.

WHAT ARE THE DIFFERENT COMPONENTS USED IN AWS?

The components that are used in AWS are:

- Amazon S3: it is used to retrieve input data sets that are involved in making a cloud architecture and also used to store the output data sets that is the result of the input.
- Amazon SQS: it is used for buffering requests that is received by the controller of the Amazon. It is the component that is used for communication between different controllers.
- Amazon Simple DB: it is used to store intermediate status log and the tasks that are performed by the user/
- Amazon EC2: it is used to run a large distributed processing on the Hadoop cluster. It provides automatic parallelization and job scheduling.

EXPLAIN THE FUNCTION OF AN AMAZON EC2 INSTANCE LIKE STOPPING, STARTING AND TERMINATING?

- **Stopping and Starting** an instance: When an instance is stopped, the instance performs a normal shutdown and then transitions to a stopped state. All of its Amazon EBS volumes remain attached, and you can start the instance again at a later time. You are not charged for additional instance hours while the instance is in a stopped state.
- **Terminating** an instance: When an instance is terminated, the instance performs a normal shutdown, then the attached Amazon EBS volumes are deleted unless the volume's delete OnTermination attribute is set to false. The instance itself is also deleted, and you can't start the instance again at a later time.Hope it would be very helpful to understand and crack the interview.

http://www.ittrainersonline.com/ant-interview-videos/

1 . Are ANT properties immutable ? IS there a way to tweak it ?
2. How do you run SVN tasks with ANT ?
3. What are SVN Hooks
4. How is exception handling done in ANT ?
5. How do you run tasks in parallel ?
6. Explain the differences between task and target ?
7. what is taskdef in ANT ?
8. How do you write custom ant tasks in ANT ?
9. How do you set classpath ref in ANT ?
10. How to run Junit reports with ANT ?
11. What is Cobetura ?

http://www.bogotobogo.com/DevOps/DevOps_CI_CD_Pipeline_Sample.php
http://w3devops.com/top-best-important-linux-shell-script-interview-question-devops/
https://softwareengineering.stackexchange.com/questions/tagged/jenkins
https://github.com/chassing/linux-sysadmin-interview-questions

https://dzone.com/articles/10-devops-interview-questions-to-gauge-a-candidate

What have you been doing over the last one to two years?
Interviews do not have to adhere to a specific framework and can be dynamic in nature. To get a general perspective on what a 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, a DevOps manager, to understand 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 and 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?" "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?" "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 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 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 (i.e., 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 that offers the open source 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? 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.

http://masterneed.com/jenkins-interview-questions/

Jenkins interview questions

What is Jenkins ?

- Jenkins is an open source continuous integration tool written in Java. It keeps a track on version control system and to initiate and monitor a build system if changes occur.

What are the benefits of using Jenkins?

- At integration stage, build failures are cached.
- For each change in the source code an automatic build report notification is generated.
- To notify developers about build report success or failure, it is integrated with LDAP mail server.
- Achieves continuous integration agile development and test driven development.
- With simple steps, maven release project is automated.
- Easy tracking of bugs at early stage in development environment than production.

Explain what is continuous integration?

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

What is the difference between Maven, Ant and Jenkins?

The most basic difference is:

- Maven and Ant are Build Technologies whereas Jenkins is a continuous integration tool.

What is continuous integration in Jenkins?

- In software development, multiple developers or teams work on different segments of same web application so you have to perform integration test by integrating all modules. In order to do that an automated process for each piece of code is performed on daily bases so that all your codes get tested. This process is known as continuous integration.

Mention some of the useful plugins in Jenkin?

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

jenkins interview questions

What is the relation between hudson and Jenkins ?

- Hudson was the earlier name and version of current Jenkins. After some issue , the project name was changed from Hudson to Jenkins.

What is the requirement for using Jenkins?

- For using Jenkins, you have to need a source code repository which is accessible. For example, a Git repository and a working build script, e.g., a Maven script, checked into the repository.

Explain how you can set up Jenkins job?

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

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

- Jenkins saves all the setting, build artifacts and logs in its home directory, to create a back-up of your Jenkins setup, just copy this directory. You can also copy a job directory to clone or replicate a job or rename the directory.

How can you move or copy Jenkins from one server to another?

Follow these steps to move or copy Jenkins from one server to another:

- First, copy the related job directory and slide a job from one installation of Jenkins to another.

- Make a copy of an already existing job by making clone of a job directory by a different name.
- Renaming an existing job by rename a directory.

What you do when you see a broken build for your project in Jenkins ?

- I will open the console output for the build and will try to see if any file changes were missed. If not able to find the issue that way, Will clean and update my local workspace to replicate the problem on my local and will try to solve it.

Which SCM tools Jenkins supports ?

- AccuRev, CVS, Subversion, Git, Mercurial, Perforce, Clearcase and RTC

How can you clone a Git repository via Jenkins?

- If you want to clone a Git repository via Jenkins, you have to enter the e-mail and user name for your Jenkins system. Switch into your job directory and execute the “git config” command for that.

What are the various ways in which build can be scheduled in Jenkins ?

- By source code management commits
- After completion of other builds
- Can be scheduled to run at specified time (crons)
- Manual Build Requests

https://intellipaat.com/interview-question/devops-interview-questions/

1. What is DevOps?

Characteristics	**DevOps**
Basic premise	Agile, lean, collaboration of IT development and operations which is more of a cultural shift
Related to	Agile methodology
Priorities	Resource management, communication and teamwork
Benefits	Speed, functionality, stability and innovation

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

Wish to Learn DevOps? Click Here

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.

Download DevOps Interview questions asked by top MNCs in 2017 ?

Download Sending ...

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.

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.

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

Q1. What is Jenkins?

My suggestion is to start this answer by giving a definition of Jenkins.

Jenkins is an open source automation tool written in Java with plugins built for Continuous Integration purpose. Jenkins is used to build and test your software projects continuously making it easier for developers to integrate changes to the project, and making it easier for users to obtain a fresh build. It also allows you to continuously deliver your software by integrating with a large number of testing and deployment technologies.

Once you have defined Jenkins give an example, you can refer the below mentioned use case:

- First, a developer commits the code to the source code repository. Meanwhile, the Jenkins server checks the repository at regular intervals for changes.
- Soon after a commit occurs, the Jenkins server detects the changes that have occurred in the source code repository. Jenkins will pull those changes and will start preparing a new build.
- If the build fails, then the concerned team will be notified.
- If built is successful, then Jenkins deploys the built in the test server.
- After testing, Jenkins generates a feedback and then notifies the developers about the build and test results.
- It will continue to check the source code repository for changes made in the source code and the whole process keeps on repeating.

Jenkins Architecture - DevOps Interview Questions Jenkins - Edureka

Interviewer now knows what is Jenkins but why we use it, there are many other CI tools as well, so why Jenkins?, the next question in this Jenkins interview questions will deal with that answer.

Q2. What are the benefits of using Jenkins?

I will suggest you to include the following benefits of Jenkins, if you can recall any other benefit apart from the below mentioned points you can include that as well.

- At integration stage, build failures are cached.
- For each change in the source code an automatic build report notification is generated.
- To notify developers about build report success or failure, it is integrated with LDAP mail server.
- Achieves continuous integration agile development and test driven development.
- With simple steps, maven release project is automated.
- Easy tracking of bugs at early stage in development environment than production.

Interviewer: Okay Jenkins looks like a really cool tool, but what are the requirements for using Jenkins?

Q3. What are the pre-requisites for using Jenkins?

Answer to this is pretty straightforward To use Jenkins you require:

- A source code repository which is accessible, for instance, a Git repository.
- A working build script, e.g., a Maven script, checked into the repository.

Remember, you have mentioned Plugins in your previous answer, so next question in this Jenkins interview questions blog will be regarding Plugins.

Q4. Mention some of the useful plugins in Jenkins?

Below I have mentioned some important Plugins:

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

Jenkins Plugins - Jenkins Interview Questions - Edureka

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

Q15. Which SCM tools Jenkins supports?

Below are Source code management tools supported by Jenkins:

- AccuRev
- CVS,
- Subversion,
- Git,
- Mercurial,
- Perforce,
- Clearcase
- RTC

Now, the next set of Jenkins interview questions will test your experience with Jenkins.

Q4. Mention what are the commands you can use to start Jenkins manually?

For this answer I will suggest you to go with the below mentioned flow:

To start Jenkins manually open Console/Command line, then go to your Jenkins installation directory. Over there you can use the below commands:

To start Jenkins: **jenkins.exe start**

To stop Jenkins: **jenkins.exe stop**

To restart Jenkins: **jenkins.exe restart**

Q6. Explain how you can set up Jenkins job?

My approach to this answer will be to first mention how to create Jenkins job.

Go to Jenkins top page, select "New Job", then choose "Build a free-style software project".

Now you can tell the elements of this freestyle job:

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

Q7. Explain how to create a backup and copy files in Jenkins?

Answer to this question is really direct.

To create a backup all you need to do is to periodically back up your JENKINS_HOME directory. This contains all of your build jobs configurations, your slave node configurations, and your build history. To create a back-up of your Jenkins setup, just copy this directory. You can also copy a job directory to clone or replicate a job or rename the directory.

Q8. How will you secure Jenkins?

The way I secure Jenkins is mentioned below, if you have any other way to do it than mention that:

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

I hope you have enjoyed the above set of Jenkins interview questions, the next set of questions will be more challenging, so be prepared.

Q9 Explain how you can deploy a custom build of a core plugin?

Below are the steps to deploy a custom build of a core plugin:

- Stop Jenkins.
- Copy the custom HPI to **$Jenkins_Home/plugins**.
- Delete the previously expanded plugin directory.
- Make an empty file called **<plugin>.hpi.pinned**.
- Start Jenkins.

Q10. What is the relation between Hudson and Jenkins?

You can just say Hudson was the earlier name and version of current Jenkins. After some issue, the project name was changed from Hudson to Jenkins.

Q11. What you do when you see a broken build for your project in Jenkins?

There can be multiple answers to this question I will approach this task in the following way:

I will open the console output for the broken build and try to see if any file changes were missed. If I am unable to find the issue that way, then I will clean and update my local workspace to replicate the problem on my local and try to solve it.

If you do it in a different way then just mention that in your answer.

Q12. Explain how you can move or copy Jenkins from one server to another?

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

You can:

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

Q13. What are the various ways in which build can be scheduled in Jenkins ?

You can schedule a build in Jenkins in the following ways:

- By source code management commits
- After completion of other builds
- Can be scheduled to run at specified time (crons)
- Manual Build Requests

Q14. What is the difference between Maven, Ant and Jenkins?

Maven and Ant are Build Technologies whereas Jenkins is a continuous integration tool.

Q16. What are the two components Jenkins is mainly integrated with?

According to me Jenkins is mainly integrated with the following:

- Version Control system like GIT,SVN.
- Build tools like Apache Maven.

If you have anything else in your mind then mention that as well but make sure you include the above two components in your answer.

Q1. What is Jenkins ?

Ans. It is a continuous integration tool written in Java.

Q2. What is the difference between Maven, Ant and Jenkins ?

Ans. Maven and Ant are Build Technologies whereas Jenkins is a continuous integration tool.

Q3. Which SCM tools Jenkins supports ?

Ans. AccuRev, CVS, Subversion, Git, Mercurial, Perforce, Clearcase and RTC

Q4. What are the various ways in which build can be scheduled in Jenkins ?

Ans. Builds can be triggered by source code management commits.
Can be triggered after completion of other builds.
Can be scheduled to run at specified time (crons)
Manual Build Requests

Q5. What is the relation between hudson and Jenkins ?

Ans. Hudson was the earlier name and version of current Jenkins. After some issue , the project name was changed from Hudson to Jenkins.

Q6. What you do to make sure that your project build doesn't break in Jenkins ?

Ans. I make sure that I perform successful clean install on my local machine with all unit tests.
Then I make sure that I check in all code changes.
Then I do a Synchronize with repository to make sure that all required config and POM changes and any difference is checked into the repository.

Q7. What you do when you see a broken build for your project in Jenkins ?

Ans. I will open the console output for the build and will try to see if any file changes were missed.
If not able to find the issue that way, Will clean and update my local workspace to replicate the problem on my local and will try to solve it.

---

http://tekslate.com/jenkins-interview-questions-and-answers

Mention Jenkins?

Jenkins is an open source tool with plugin built for continuous integration purpose. The principle functionality of Jenkins is to keep a track of version control system and to initiate and monitor a build system if changes occur. It monitors the whole process and provides reports and notifications to alert.

Explain what is continuous integration?

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

What is the requirement for using Jenkins?

To use Jenkins you require

– A source code repository which is accessible, for instance, a Git repository

– A working build script, e.g., a Maven script, checked into the repository

Mention what are the advantages of Jenkins?

Advantage of Jenkins include

– At integration stage, build failures are cached

– For each code commit changes an automatic build report notification generates

– To notify developers about build report success or failure, it is integrated with LDAP mail server

– Achieves continuous integration agile development and test driven development

– With simple steps, maven release project is automated

– Easy tracking of bugs at early stage in development environment than production

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

– Slide a job from one installation of Jenkins to another by copying the related job directory

– Make a copy of an already existing job by making clone of a job directory by a different name

– Renaming an existing job by renaming a directory.

Aspired to become Jenkins Developer? Explore the post to discover the know-hows on

Jenkins Training Videos.

Mention what are the commands you can use to start Jenkins manually?

To start Jenkins manually, you can use either of the following

– (Jenkins_url)/restart: Forces a restart without waiting for builds to complete

– (Jenkin_url)/safeRestart: Allows all running builds to complete

Mention some of the useful plugins in Jenkin?

Some of the important plugins in Jenkin includes

– Maven 2 project

– Amazon EC2

– HTML publisher

– Copy artifact

– Join

– Green Balls

Explain how you can deploy a custom build of a core plugin?

To deploy a custom field of a core plugin, you have to do following things

– Stop Jenkins

– Copy the custom HPI to $Jenkins_Home/plugins

– Delete the previously expanded plugin directory

– Make an empty file called <plugin>.hpi.pinned

– Start Jenkins

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

Jenkins saves all the setting, build artifacts and logs in its home directory, to create a back-up of your Jenkins setup, just copy this directory. You can also copy a job directory to clone or replicate a job or rename the directory.

Explain how you can clone a Git repository via Jenkins?

To clone a Git repository via Jenkins, you have to enter the e-mail and user name for your Jenkins system. For that, you have to switch into your job directory and execute the “git config” command.

Explain how you can set up Jenkins job?

To create a project that is handled via jobs in Jenkins. Select New item from the menu, once this done enter a name for the job and select free-style job. Then click OK to create new job in Jenkins. The next page enables you to configure your job.

Mention what are the two components Jenkins is mainly integrated with?

Jenkin is mainly integrated with two components

– Version Control system like GIT, SVN

– And build tools like Apache Maven.

What is the difference between Maven, Ant and Jenkins ?

Maven and Ant are Build Technologies whereas Jenkins is a continuous integration tool.

Which SCM tools Jenkins supports ?

AccuRev, CVS, Subversion, Git, Mercurial, Perforce, Clearcase and RTC

What are the various ways in which build can be scheduled in Jenkins ?

Builds can be triggered by source code management commits.

Can be triggered after completion of other builds.

Can be scheduled to run at specified time (crons)

Manual Build Requests

What is the relation between hudson and Jenkins ?

Hudson was the earlier name and version of current Jenkins. After some issue , the project name was changed from Hudson to Jenkins.

What you do to make sure that your project build doesn’t break in Jenkins ?

I make sure that I perform successful clean install on my local machine with all unit tests.

Then I make sure that I check in all code changes.

Then I do a Synchronize with repository to make sure that all required config and POM changes and any difference is checked into the repository.

What you do when you see a broken build for your project in Jenkins ?

I will open the console output for the build and will try to see if any file changes were missed.

If not able to find the issue that way, Will clean and update my local workspace to replicate the problem on my local and will try to solve it.

Interested in mastering Jenkins? Learn more about** Jenkins Tutorials **in this blog post.

What is the requirement for using Jenkins?

For using Jenkins, you have to need a source code repository which is accessible. For example, a Git repository and a working build script, e.g., a Maven script, checked into the repository.

How to create a backup and copy files in Jenkins?

If you want to create a back-up of your Jenkins setup, just copy the directory that saves all the setting, build artifacts and logs of Jenkins in its home directory. You can also copy a job directory to clone or replicate a job or rename the directory.

How can you clone a Git repository via Jenkins?

If you want to clone a Git repository via Jenkins, you have to enter the e-mail and user name for your Jenkins system. Switch into your job directory and execute the “git config” command for that.

How can you setup Jenkins jobs?

Follow these steps:

Select new item from the menu.

After that enter a name for the job and select free-style job.

Then click OK to create new job in Jenkins.

The next page enables you to configure your job.

What are the two components Jenkins is mainly integrated with?

Jenkins is integrated with these two components:

Version Control system like GIT, SVN

And build tools like Apache Maven.

How can you move or copy Jenkins from one server to another?

Follow these steps to move or copy Jenkins from one server to another:

First, copy the related job directory and slide a job from one installation of Jenkins to another.

Make a copy of an already existing job by making clone of a job directory by a different name.

Renaming an existing job by rename a directory.

http://www.javatpoint.com/jenkins-interview-questions

1) What is Jenkins?

Jenkins is an open source continuous integration tool written in Java. It keeps a track on version control system and to initiate and monitor a build system if changes occur.

---

2) What is the difference between Maven, Ant and Jenkins?

The most basic difference is:

Maven and Ant are Build Technologies whereas Jenkins is a continuous integration tool.

---

3) Which SCM tools does Jenkins support?

Jenkins supports the following SCM tools:

- AccuRev
- CVS
- Subversion
- Git
- Mercurial
- Perforce
- Clearcase
- RTC

---

4) What is continuous integration in Jenkins?

In software development, multiple developers or teams work on different segments of same web application so you have to perform integration test by integrating all modules. In order to do that an automated process for each piece of code is performed on daily bases so that all your codes get tested. This process is known as continuous integration.

---

5) What is the relation between Hudson and Jenkins?

Hudson was the earlier name and version of current Jenkins. After some issue, the project name was changed from Hudson to Jenkins.

---

6) What is the requirement for using Jenkins?

For using Jenkins, you have to need a source code repository which is accessible. For example, a Git repository and a working build script, e.g., a Maven script, checked into the repository.

---

7) What are the advantages of Jenkins?

Advantage of Jenkins includes:

- Bugs tracking are easy at early stage in development environment.
- Provides a large numbers of plugin support.
- Iterative improvement to the code.
- Build failures are cached at integration stage.
- For each code commit changes an automatic build report notification generates.
- To notify developers about build report success or failure, it is integrated with LDAP mail server.
- Achieves continuous integration agile development and test driven development.
- With simple steps, maven release project is automated.

8) How to make sure that your project builds doesn?t break in Jenkins?

You must follow these steps to make sure that your project builds doesn?t break in Jenkins:

- First, perform successful clean install on your local machine with all unit tests.
- Check all your code changes.
- Synchronize with repository to make sure that all required config and POM changes and any difference is checked into the repository.

9) How can you move or copy Jenkins from one server to another?

Follow these steps to move or copy Jenkins from one server to another:

- First, copy the related job directory and slide a job from one installation of Jenkins to another.
- Make a copy of an already existing job by making clone of a job directory by a different name.
- Renaming an existing job by rename a directory.

10) Which commands can be used to start Jenkins manually?

You can use any one of the following commands to start Jenkins manually:

- (Jenkins_url)/restart: Forces a restart without waiting for builds to complete.
- (Jenkin_url)/safeRestart: Allows all running builds to complete.

11) What are the most useful plugins in Jenkins?

Some most useful plugins in Jenkins:

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

12) How to create a backup and copy files in Jenkins?

If you want to create a back-up of your Jenkins setup, just copy the directory that saves all the setting, build artifacts and logs of Jenkins in its home directory. You can also copy a job directory to clone or replicate a job or rename the directory.

13) How can you clone a Git repository via Jenkins?

If you want to clone a Git repository via Jenkins, you have to enter the e-mail and user name for your Jenkins system. Switch into your job directory and execute the "git config" command for that.

14) How can you setup Jenkins jobs?

Follow these steps:

- Select new item from the menu.
- After that enter a name for the job and select free-style job.

- Then click OK to create new job in Jenkins.
- The next page enables you to configure your job.

---

15) What are the two components Jenkins is mainly integrated with?

Jenkins is integrated with these two components:

- Version Control system like GIT,SVN
    - And build tools like Apache Maven.

http://ec2-54-213-232-205.us-west-2.compute.amazonaws.com/index.php/2017/02/22/devops-jenkins-interview-questions/

1) What is continuous integration (CI) ?
2) What is continuous delivery (CD) ?
3) What is continues deployment ?
4) What is difference between continuous integration(CI)/Continuous delivery/Continuous Deployment?
5) What is Jenkins ?
6) What are the advantages of Jenkins?
7) How you will change the Jenkins default port i.e 8080?
8) How you will run Jenkins behind an Apache Server ?
9) What are the two ways to run Jenkins server ?
10) What are prerequisites of setting up Jenkins Server?
11) What are the commands you can use to start Jenkins manually when you are running Jenkins in Standalone mode ?
12) What's in the Jenkins Home Directory?
13) How you will upgrade your Jenkins server ?
14) What is build jobs and how many different types of build jobs can be created in Jenkins?
15) Explain how you install plugin manually in Jenkins server?
16) What are the most useful plugins in Jenkins?
17)How can you move or copy Jenkins from one server to another?
18)What are version control system/ SCM (software configuration management) support does jenkins support ?
19)What is Quiet period in jenkins?
20)What is "Block build when upstream project is building" in Jenkins & When it useful?
21)What is build trigger ? What are ways to configure Jenkins to start/trigger a build Job?
22)How you will secure the Jenkins?
23)What is multiconfiguration Build Job in Jenkins?
24)How to change workspace and build directory location in Jenkins?
25)Explain master-slave concept of Jenkins?
26)How you will troubleshoot failed build jobs in Jenkins?
27)You are no longer able to login/re-configure Jenkins due to wrong set up security realm/authorization. how you will reset these security configuration to regain access of Jenkins?

http://www.rvhtechguru.com/top-most-important-devops-interview-questions-and-answers-by-experts/

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.automationtestingworld.com/devops-jenkins-must-interview-question-top-10/?i=1

Devops and Jenkins Interview Questions

1. **Explain Jenkins -pipeline highlights?**

Pipeline item type for new jobs (instead of Freestyle)

Entire pipeline as text code in SCM (GitHub)

Multiple SCM repositories in each job

Pausable: Jobs can wait for manual user input before continuing

Jobs share global library to share scripts, functions, variables for DRY (Do not Repeat Yourself) – Reusable components and flow

Extendable DSL with loops, logic

Visualized: Pipeline StageView provides status at-a-glance dashboard and trending

Parallel execution of arbitrary build states

Jobs starting in one agent can switch (be joined) to another (fork/join)

2.What is CI (Continuous integration) & CD (Continuous Delivery) in Jenkins

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

Code done->Unit Tests ->Integration->Acceptance Test->Deployment fully process automatically through jenkins is a part of Continuous delivery or continuous deployment process.

3. **What is the requirement for using Jenkins?**

To use Jenkins you require:

- A source code repository which is accessible, for instance, a Git repository
- A working build script, e.g., a Maven script, checked into the repository

4. **Mention what are the commands you can use to start Jenkins manually?**

To start Jenkins manually, you can use either of the following

- (Jenkins_url)/restart: Forces a restart without waiting for builds to complete
- (Jenkin_url)/safeRestart: Allows all running builds to complete

5. **Mention some of the useful plugins in Jenkin?**

Some of the important plugins in Jenkin includes

- Powershell script
- Window execute script
- Env inject
- VSTS plugin
- Maven 2 project
- Amazon EC2
- HTML publisher
- Copy artifact
- Publish artifact

6. **Mention what are the two components Jenkins is mainly integrated with?**

Jenkin is mainly integrated with two components

- Version Control system like GIT, SVN
- And build tools like Apache Maven.

7. **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.

8. How do all these tools work together?

Below is a generic logical flow where everything gets automated for seamless delivery.

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

9. How to make sure that your project builds does not break in Jenkins?

1. You must follow these steps to make sure that your project builds doesn?t break in Jenkins:
2. First, perform successful clean install on your local machine with all unit tests.
3. Check all your code changes.
4. Synchronize with repository to make sure that all required config and POM changes and any difference is checked into the repository.

10 . Explain some Jenkins Job with example ?

- **Freestyle Project:**

Freestyle build jobs are general-purpose build jobs, which provides maximum flexibility. The freestyle build job is the most flexible and configurable option, and can be used for any type of project. It is relatively straightforward to set up, and many of the options we configure here also appear in other build jobs.

- **Multiconfiguration Job:**

The "multi configuration project" (also referred to as a "matrix project") allows you run the same build job on different environments. It is used for testing an application in different environments, with different databases, or even on different build machines.

- **Monitor an External Job:**

The "Monitor an external job" build job lets you keep an eye on non-interactive processes, such as cron jobs.

---

http://www.automationtestingworld.com/devopsinterviewquestion/

1) How can you define DEVOPS in your own words?

Most companies that implement DevOps methods today still have a development team and an operations team in place. You can think of DevOps as the processes and individuals that build the bridges between these teams to improve the business and enhance the end-customer experience. Various tools and platforms facilitate the work of DevOps, but they do not define it. Organizations that embrace DevOps might have all IT resources within a traditional data center, all resources in an offsite cloud, or distribute their resources in a hybrid environment.

The DevOps movement is not defined nor led by traditional IT software, hardware, or management vendors. In addition, there are currently no codified rules or manuals for DevOps, only generally accepted guidelines. With that said, adoption and implementation of DevOps vary greatly from one organization to the next.

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 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 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.

5. **List out some popular tools for DevOps?**

Some of the popular tools for DevOps are

– Jenkins

– Nagios

– Docker

– Ansible

– Git

6) 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

7 which language you used for development and implementation process?

-Explain about scripting language with example.

8) 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.

9) 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.

10) 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.

---

http://www.mastertheboss.com/soa-cloud/devops/devops-interview-questions

What is DevOps ?

Devops is in a nutshell a cultural movement which aims to remove through collaboration and communication unnecessary silos in an organization. In less abstract terms, Devops can be seen as a number of software development practices that enable automation and accelerate delivery of products. The last element, automation, in turn requires a programmable dynamic platform.

Which are the components of DevOps ?

Operations: which is is responsible for the infrastructure and operational environments that support application deployment, including the network infrastructure. In most cases we can say this is the Sys Admin

Devs: which is responsible for software engineering development. In most case Developers, Architects fall in this category.

Quality Assurance: which are responsible for verifying the quality of the product such as Product Testers.

Do you think Devs and Ops will radically change their working routine ?

In most cases not. Ops will still be Ops and Devs will still be Devs. The difference is, these teams need to begin working closely together.

How can you improve DevOps culture ?

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

Which technologies can act as driver to enable DevOps ?

- **Paas**: which is a category of cloud computing services that provides a platform allowing customers to develop, run, and manage applications without the complexity of building and maintaining the infrastructure
- **Iaas**: which is a category of cloud computing services that abstract the user from the details of infrastructure like physical computing resources, location, data partitioning, scaling, security, backup etc.
- **Configuration automation**: Automation is a big win in part because it eliminates the labor associated with repetitive tasks. Codifying such tasks also means documenting them and ensuring that they're performed correctly, in a safe manner, and repeatedly across different infrastructure types.
- **Microservices**: which consists in a particular way of designing software applications as suites of independently deployable services.
- **Containers**: Containers modernize IT environments and processes, and provide a flexible foundation for implementing DevOps. At the organizational level, containers allow for appropriate ownership of the technology stack and processes, reducing hand-offs and the costly change coordination that comes with them.

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

Microservices are a product of software architecture and programming practices. Microservices architectures typically produce smaller, but more numerous artifacts that Operations is responsible for regularly deploying and managing. For this reason microservices have an important impact on Operations.The term that describes the responsibilities of deploying microservices is microdeployments. So, what DevOps is really about is bridging the gap between microservices and microdeployments.

Which tools are typically integrated in DevOps workflow ?

Many different types of tools are integrated into the DevOps workflow at this point. For example:

- Code repositories: like Git

- Container development tools: to convert code in a repository into a portable containerized image that includes any required dependencies
- Virtual machines software: like Vagrant for creating and configuring lightweight, reproducible, and portable development environments
- IDE: like Eclipse which has integration with DevOps platforms like Openshift
- Continuous Integration and Delivery software: like Jenkins which automates pushing the code directly to production once it has passed automated testing.

What is automation ?

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

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

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

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

At which level can applied automation in DevOps ?

At three levels:

1) Automate the application lifecycle: in terms of software features, version control, build management, integration frameworks

2) Automate the middleware platform automation: such as installing middleware, autoscaling and resources optimization of middlware components

3) At infrastructure by provisioning operating system resources and virtualizing them

Which scripting language is most important for a DevOps engineer?

Software development and Operational automation requires programming. In terms of scripting

Bash is the most frequently used Unix shell which should be your first automation choice. It has a simple syntax, and is designed specifically to execute programs in a non-interactive manner. The same stands for **Perl** which owes great deal of its popularity to being very good at manipulating text and storing data in databases.

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

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

Explain how DevOps is helpful to developers?

DevOps brings faster and more frequent release cycles which allows developers to identify and resolve issues immediately as well as implementing new features quickly.

Since DevOps is what makes people do better work by making them wear different hats, Developers who collaborate with Operations will create software that is easier to operate, more reliable, and ultimately better for the business.

How Database fits in a DevOps ?

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

Which are the reasons against using an RDBMS?

In a nutshell, if your application is all about storing application entities in a persistent and consistent way, then an RDBMS could be an overkill. A simple Key-Value storage solution might be perfect for you. Note that the Value is not meant to be a simple element but can be a complex entity in itself!

Another reason could be if you have hierarchical application objects and need some query capability into them then most **NoSQL** solutions might be a fit. With an RDBMS you can use ORM to achieve the same result, but at the cost of adding extra complexity.

RDBMS is also not the best solution if you are trying to store large trees or networks of objects. Depending on your other needs a **Graph Database** might suit you.

If you are running in the Cloud and need to run a distributed database for durability and availability then you could check **Dynamo** and Big Table based datastores which are built for this core purpose.

Last but not least, if your data grows to large to be processed on a single machine, you might look into **Hadoop** or any other solution that supports distributed Map/Reduce.

What is 2 factors authentication ?

In terms of authentication, when you have to enter only your username and one password, that's considered a single-factor authentication. 2 factors authentication requires the user to have two out of three types of credentials before being able to access an account. The three types are:

- Something you know, such as a personal identification number (PIN), password
- Something you have, such as a digital ATM card, phone
- Something you are, such as a biometric like voice or a fingerprint

What is a PTR record and how to add one?

While a record points a domain name to an IP address, the PTR record resolves the IP address to a domain/hostname. PTR records are used for the reverse DNS (Domain Name System) lookup. Using the IP address you can get the associated domain/hostname. A record should exist for every PTR record.

You can check whether there is a PTR record set for a defined IP address. The syntax of the commands on a Linux OS are:

```
$ dig -x IP
```

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

Push vs Pull Strategy:

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

Server Nodes:

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

Language and Extensibility:

- Puppet uses its own DSL language which is a subset of Ruby. Thus adding extra complex functionality is done through Ruby modules. That being said there's a more strict control on what you are doing with Ruby.
- Ansible playbooks are YAML files. In terms of extensibility, Ansible is built upon Python for which most organization will have some experience.
- Chef: uses Ruby as programming language that is the authoring syntax for Chef cookbooks. Put it straight Chef lets you run wild with Ruby.

Resources & Ordering

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

Resource Dependency

- Puppet internally creates a directed graph of all of the defined resources along with the order they should be applied in. Puppet can even generate a graph file so that one can visualize everything that Puppet manages. On the other hand, building this graph is susceptible to “multiple resource definition” errors or conflicts due to circular dependencies.
- Ansible is merely a thin-wrapper for executing commands over SSH, so there is no resource dependency graph built internally.
- Chef is also able to declare dependencies between resources. Dependency failures are breakages in your dependency graph, which keep the current project’s pipeline from being able to ship safely. These failures are tracked because through Chef Automation

DevOps Tool Support

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

What is an MX record ?

An MX record tells senders how to send email for your domain. When your domain is registered, it’s assigned several DNS records, which enable your domain to be located on the Internet. These include MX records, which direct the domain’s mail flow. Each MX record points to an email server that’s configured to process mail for that domain. There’s typically one record that points to a primary server, then additional records that point to one or more backup servers. For users to send and receive email, their domain's MX records must point to a server that can process their mail.

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

https://jivoi.github.io/2016/01/22/linux-sysadm-devops-interview-questions/

https://github.com/chassing/linux-sysadmin-interview-questions

How to move Jenkins from one PC to another?

you'll have to :

- install a fresh jenkins in the new server
- be sure the old and the new jenkins are stopped
- archive all the content of the JENKINS_HOME of the old jenkins instance
- extract the archive into the new JENKINS_HOME directory
- launch the new Jenkins
- do not forget to change documentation/links to your new instance of Jenkins :)

EDIT: JENKINS_HOME is by default on Linux installation located in `~/.jenkins` , yet to exactly find where it is located, go on the http://your_jenkins_url/configure page and check the value of the first parameter : `Home directory` , this is the JENKINS_HOME.

How to configure Git post commit hook

As mentioned in "Polling must die: triggering Jenkins builds from a git hook", you can notify Jenkins of a new commit:

> With the latest Git plugin 1.1.14 (that I just release now), you can now do this more >easily by simply executing the following command:
>
> ```
> curl http://yourserver/jenkins/git/notifyCommit?url=<URL of the Git repository>
> ```
>
> This will scan all the jobs that’s configured to check out the specified URL, and if they are also configured with polling, it’ll immediately trigger the polling (and if that finds a change worth a build, a build will be triggered in turn.)
>
> This allows a script to remain the same when jobs come and go in Jenkins.
> Or if you have multiple repositories under a single repository host application (such as Gitosis), you can share a single post-receive hook script with all the repositories. Finally, this URL doesn’t require authentication even for secured Jenkins, because the server doesn’t directly use anything that the client is sending. It runs polling to verify that there is a change, before it actually starts a build.

As mentioned here, make sure to use the right address for your Jenkins server:

> since we're running Jenkins as standalone Webserver on port 8080 the URL should have been without the `/jenkins` , like this:
>
> ```
> http://jenkins:8080/git/notifyCommit?url=git@gitserver:tools/common.git
> ```

To reinforce that last point, ptha adds in the comments:

> It may be obvious, but I had issues with:

> ```
> curl http://yourserver/jenkins/git/notifyCommit?url=<URL of the Git repository>.
> ```
>
> The **url** parameter should match exactly what you have in **Repository URL** of your Jenkins job.
> When copying examples I left out the protocol, in our case `ssh://` , and it didn't work.

You can also use a simple post-receive hook like in "Push based builds using Jenkins and GIT"

```
#!/bin/bash
/usr/bin/curl --user USERNAME:PASS -s \

http://jenkinsci/job/PROJECTNAME/build?token=1qaz2wsx
```

> Configure your Jenkins job to be able to “Trigger builds remotely” and use an authentication token (`1qaz2wsx` in this example).

However, this is a project-specific script, and the author mentions a way to generalize it.
The first solution is easier as it doesn't depend on authentication or a specific project.

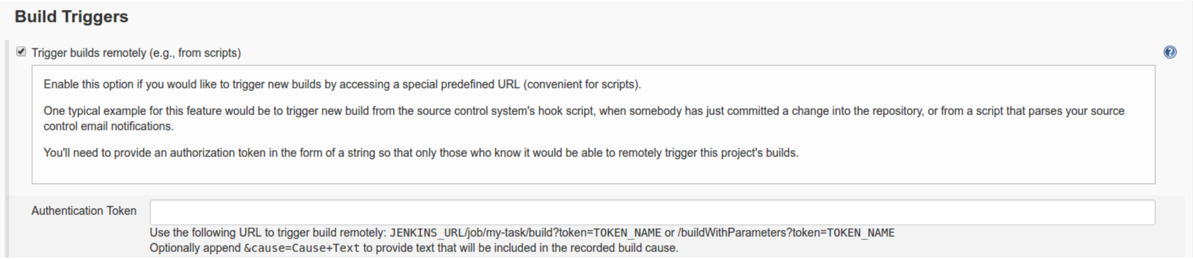
> I want to check in change set whether at least one java file is there the build should start.
> Suppose the developers changed only XML files or property files, then the build should not start.

Basically, you build script can:

- put a 'build' notes (see `git notes`) on the first call
- on the subsequent calls, grab the list of commits between `HEAD` of your branch candidate for build and the commit referenced by the `git notes` 'build' (`git show refs/notes/build`): `git diff --name-only SHA_build HEAD` .
- your script can parse that list and decide if it needs to go on with the build.
- in any case, create/move your `git notes` ' `build` ' to `HEAD` .

May 2016: cwhsu points out in the comments the following possible url:

> you could just use `curl --user USER:PWD http://JENKINS_SERVER/job/JOB_NAME/build?token=YOUR_TOKEN` if you set trigger config in your item



Build Triggers

☑ Trigger builds remotely (e.g., from scripts)

Enable this option if you would like to trigger new builds by accessing a special predefined URL (convenient for scripts).

One typical example for this feature would be to trigger new build from the source control system's hook script, when somebody has just committed a change into the repository, or from a script that parses your source control email notifications.

You'll need to provide an authorization token in the form of a string so that only those who know it would be able to remotely trigger this project's builds.

Authentication Token

Use the following URL to trigger build remotely: `JENKINS_URL`/job/my-task/build?token=`TOKEN_NAME` or /buildWithParameters?token=`TOKEN_NAME`
Optionally append `&cause=Cause+Text` to provide text that will be included in the recorded build cause.

June 2016, polaretto points out in the comments:

> I wanted to add that with just a little of shell scripting you can avoid manual url configuration, especially if you have many repositories under a common directory.
> For example I used these parameter expansions to get the repo name
>
> ```
> repository=${PWD%/hooks};
> repository=${repository##*/}
> ```
>
> and then use it like:
>
> ```
> curl $JENKINS_URL/git/notifyCommit?url=$GIT_URL/$repository
> ```

http://stackoverflow.com/questions/12794568/how-to-configure-git-post-commit-hook/12794930#12794930

Q:: moving jobs in jenkins?

Moving/copying/renaming jobs

You can:

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

Those operations can be done even when Jenkins is running. For changes like these to take effect, you have to click "reload config" to force Jenkins to reload configuration from the disk.

https://wiki.jenkins-ci.org/display/JENKINS/Administering+Jenkins#AdministeringJenkins-Moving%2Fcopying%2Frenamingjobs

Option 2::

```
java -jar jenkins-cli.jar -s http://server get-job myjob > myjob.xml
java -jar jenkins-cli.jar -s http://server create-job newmyjob < myjob.xml
```

```
Option 3:: Job Import plugin
```

Jenkins / Hudson environment variables

When Jenkins connects to a computer, it goes to the `sh` shell, and not the `bash` shell (at least this is what I have noticed - I may be wrong). So any changes you make to $PATH in your bashrc file are not considered.

Also, any changes you make to $PATH in your local shell (one that you personally ssh into) will not show up in Jenkins.

To change the path that Jenkins uses, you have two options (AFAIK):

1) Edit your `/etc/profile` file and add the paths that you want there

2) Go to the configuration page of your slave, and add environment variable `PATH` , with value: `$PATH:/followed-by/paths/you/want/to/add`

If you use the second option, your System Information will still not show it, but your builds will see the added paths.

http://stackoverflow.com/questions/5818403/jenkins-hudson-environment-variables/5819768#5819768

Reset Jenkins Configuration Command Line

config.xml can not be found at

> /var/lib/jenkins/

Its available in

> ~/.jenkins

then after that as other mentioned open the *config.xml* file and make the following changes

- In this replace `<useSecurity>true</useSecurity>` with `<useSecurity>false</useSecurity>`
- Remove `<authorizationStrategy>` and `<securityRealm>`
- Save it and restart the jenkins(*sudo service jenkins restart*

  Option 2::

To reset it **without disabling security** if you're using matrix permissions (probably easily adaptable to other login methods):

- In `config.xml`, set **`disableSignup`** to `false`.
- **Restart** Jenkins.
- Go to the Jenkins web page and **sign up with a new user**.
- In `config.xml`, duplicate one of the `<permission>hudson.model.Hudson.Administer:username</permission>` lines and replace *`username`* with the new user.
- If it's a private server, set **`disableSignup`** back to `true` in `config.xml`.
- **Restart** Jenkins.
- Go to the Jenkins web page and **log in as the new user**.
- **Reset the password** of the original user.
- **Log in** as the original user.

Optional cleanup:

1. Delete the new user.
2. Delete the temporary `<permission>` line in `config.xml`.

No securities were harmed during this answer.

Is there a way to keep Hudson / Jenkins configuration files in source control?

Option1:: SCM Sync configuration plugin
Option 2: write a .gitignore file
The way I prefer is to exclude everything in the Jenkins home folder *except* the configuration files you really want to be in your VCS. Here is the `.gitignore` file I use:

```
*
!.gitignore
!/jobs/*/*.xml
!/*.xml
!/users/*/config.xml
!*/
```

This ignores everything (`*`) except (`!`) `.gitignore` itself, the jobs/projects, the plugin and other important and user configuration files.

It's also worth considering to include the `plugins` folder. Annoyingly updated plugins should be included...

Basically this solution makes it easier for future Jenkins/Hudson updates because new files aren't automatically in scope. You just get on the screeen what you really want.

http://stackoverflow.com/questions/2087142/is-there-a-way-to-keep-hudson-jenkins-configuration-files-in-source-control

Q::Skip a submodule during a maven build

A::
Sure, this can be done using profiles. You can do something like the following in your parent pom.xml.

```
...
 <modules>
    <module>module1</module>
    <module>module2</module>
    ...
</modules>
...
<profiles>
   <profile>
     <id>ci</id>
        <modules>
          <module>module1</module>
```

```
           <module>module2</module>
           ...
           <module>module-integration-test</module>
         </modules>
     </profile>
 </profiles>
...
```

In your CI, you would run maven with the `ci` profile, i.e. `mvn -P ci clean install`

Option::2
Maven version 3.2.1 added this feature, you can use the -pl switch with the "!" to exclude certain submodules.

```
mvn -pl '!submodule-to-exclude' install
```

Be careful in bash the character ! is a special character, so you either have to single quote it (like I did) or escape it with the backslash character.

The syntax to exclude multiple module is the same as the inclusion

```
mvn -pl '!submodule1,!submodule2' install
```

EDIT Windows does not seem to like the single quotes, but it is necessary in bash ; in Windows, use double quotes (thanks @awilkinson)

```
mvn -pl "!submodule1,!submodule2" install
```

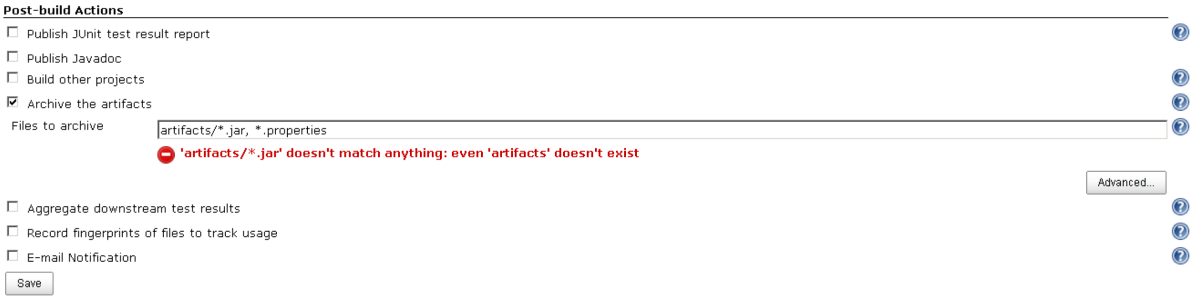
Q:: Archive the artifacts in hudson/jenkins

I have the workspace directory where I check out the code to, compile, and run my ant scripts etc. At the end, in my case, I get a jar file thats ready to install. Is that considered to be the artifact?

Where should I tell my build script to put the jar file? In the workspace directory? My jar file gets a unique filename depending on variables like `BUILD_ID` and such, how can I tell Jenkins which jar file to pick?

I am a little confused here – who can explain?

EDIT: Okay, so i try to do something like this:



Post-build Actions

☐ Publish JUnit test result report

☐ Publish Javadoc

☐ Build other projects

☑ Archive the artifacts

Files to archive: artifacts/*.jar, *.properties

'artifacts/*.jar' doesn't match anything: even 'artifacts' doesn't exist

☐ Aggregate downstream test results

☐ Record fingerprints of files to track usage

☐ E-mail Notification

an artifact in the Jenkins sense is the result of a build - the intended output of the build process.

A common convention is to put the result of a build into a `build` , `target` or `bin` directory.

The Jenkins archiver can use globs (`target/*.jar`) to easily pick up the right file even if you have a unique name per build.

====
http://stackoverflow.com/questions/5739099/how-to-design-and-architect-a-java-java-ee-web-application/5739259#5739259
====

Q:: How to start jenkins on different port rather than 8080 using command prompt in Windows?

Option 1::

Open the `jenkins.xml` in the jenkins home folder (usually `C:\Program Files (x86)\Jenkins`) and change the port number:

```
httpPort=xxxx
```

to

```
httpPort=yyyy
```

then restart the service. it should change the setting permanently.

Option 2::

Use the following command at command prompt:

```
java -jar jenkins.war --httpPort=9090
```

If you want to use https use the following command:

```
java -jar jenkins.war --httpsPort=9090
```

Ref: https://wiki.jenkins-ci.org/display/JENKINS/Starting+and+Accessing+Jenkins

DevOps Online Training *June 14, 2017 at 12:37 AM*

Very good blog on Jenkins Interview questions. Literally prepared questions by covering every concept on Jenkins. this is so help for our DevOps beginner students. Thanks for writing this awesome useful post. Check Out DevOps Git Interview Questions Blog

Best Regards,
Disha, Trainer at DevOps Training Institute in Hyderabad

REPLY

soumya Teja *March 12, 2018 at 7:49 AM*

I would like to say thanks to you for sharing latest updates on **Devops Online Training Hyderabad**

REPLY

Enter your comment...

Popular posts from this blog

Puppet Interview Questions

By Devops - April 07, 2017

Image

READ MORE

Docker Interview Questions

By Devops - March 12, 2017

I wrote a blog post to help prepare people for their DevOps interview. These questions are all gathered from different blogs and sites. I have mentioned the site name, from where I have copied. Thanks to all those authors and blog writers. …

READ MORE

Powered by Blogger

Theme images by Veronica Olson

←

Important

ArchiveList

Search This Blog

Search this blog

DEVOPS

VISIT PROFILE

Archive

Report Abuse