Jenkins Interview Questions & Answers

**Jenkins Interview Questions**

|  |  |
| --- | --- |
|  | |
| Jenkins is an | Open source software |
| Jenkins is an | Automation server |
| Jenkins can | Help to automate the software development process. |
| Jenkins can | Automate the process with continuous integration and facilitate technical aspects of continuous delivery. |
| Jenkins developed by | Jenkins is a fork of a project called Hudson. |
| Jenkins License | MIT |
| Jenkins has written in | Java |

**1) What is Jenkins?**

Jenkins is an open source automation server. Jenkins is a continuous integration tool developed in Java. **Jenkins** helps to automate the non-human part of software development process, with **continuous integration** and facilitating technical aspects of continuous delivery.

**2) Why do we use Jenkins?**

Jenkins is an **open-source** continuous integration software tool written in the Java programming language for testing and reporting on isolated changes in a larger code base in real time. The **Jenkins software** enables developers to find and solve defects in a code base rapidly and to automate testing of their builds.

**3) What is Maven and what is Jenkins?**

**Maven is a build tool**, in short a successor of ant. It helps in build and version control. However, **Jenkins is continuous integration system**, where in maven is used for build. Jenkins can be used to automate the deployment process.

**4) What is the difference between Hudson and Jenkins?**

**Jenkins is the new Hudson**. It really is more like a rename, not a fork, since the whole development community moved to Jenkins. (Oracle is left sitting in a corner holding their old ball “**Hudson**“, but it’s just a soul-less project now.). In a nutshell **Jenkins CI** is the leading open-source continuous integration server.

**5) What is meant by continuous integration in Jenkins?**

**Continuous integration** is a process in which all development work is integrated as early as possible. The resulting artifacts are automatically created and tested. This process allows to identify errors as early as possible. **Jenkins is a popular open source tool** to perform continuous integration and build automation.

**Continuous Integration Interview Questions 6) Why do we use Jenkins with selenium?**

Running **Selenium tests in Jenkins** allows you to run your tests every time your software changes and deploy the software to a new environment when the tests pass. Jenkins can schedule your tests to run at specific time.

**Jenkins CI CD Interview Questions 7) What are CI Tools?**

Here is the list of the top 8 **Continuous Integration tools**:

Jenkins

TeamCity

Travis CI

Go CD

Bamboo

GitLab CI

CircleCI

Codeship

**Jenkins Pipeline Interview Questions 8) What is a CI CD pipeline?**

A **continuous integration** and deployment pipeline (**CD/CI**) is such an important aspect of a software project. It saves a ton of manual, error-prone deployment work. It results in higher quality software for continuous integration, **automated tests**, and code metrics.

**Jenkins Tough Interview Questions 9) What is build** pipeline **in Jenkins?**

Job chaining in **Jenkins** is the process of automatically starting other job(s) after the execution of a job. This approach lets you build **multi-step build pipelines** or trigger the rebuild of a project if one of its dependencies is updated.

**Jenkin Interview Questions 10) What is a Jenkins pipeline?**

The **Jenkins Pipeline plugin** is a game changer for Jenkins users. Based on a *Domain Specific Language (DSL)* in Groovy, the Pipeline plugin makes pipelines scriptable and it is an incredibly powerful way to develop complex, multi-step **DevOps pipelines**.

**And s For Experienced**

**11) What is a DSL** Jenkins**?**

The Jenkins “Job DSL / Plugin” is made up of two parts: The Domain Specific Language (DSL) itself that allows users to describe jobs using a Groovy-based language, and a Jenkins plugin which manages the scripts and the updating of the Jenkins jobs which are created and maintained as a result.

**For Devops 12) What is continuous integration and deployment?**

**Continuous Integration (CI)** is a development practice that requires developers to integrate code into a shared repository several times a day. Each check-in is then verified by an automated build, allowing teams to detect problems early.

**Jenkins Real Time Interview Questions 13) What is the tool used for provisioning and configuration?**

Ansible is an agent-less configuration management as well as orchestration tool. In Ansible, the configuration modules are called “Playbooks”. Like other tools, Ansible can be used for cloud provisioning.

**Jenkins Questions And s 14) What is the difference between Maven, Ant** and **Jenkins?**

Maven and ANT are build tool but main difference is that maven also provides dependency management, standard project layout and project management. On difference between Maven, ANT and Jenkins, later is a continuous integration tool which is much more than build tool.

**Jenkins Questions 15) Which SCM tools Jenkins supports?**

Jenkins supports version control tools, including AccuRev, CVS, Subversion, Git, Mercurial, Perforce, ClearCase and RTC, and can execute Apache Ant, Apache Maven and sbt based projects as well as arbitrary shell scripts and Windows batch commands.

**For Testers**

**Devops Interview Questions Jenkins 16) How schedule a build in Jenkins?**

In Jenkins, under the job configuration we can define various build triggers. Simple find the ‘Build Triggers’ section, and check the ‘ Build Periodically’ checkbox. With the periodically build you can schedule the build definition by the date or day of the week and the time to execute the build.

The format of the ‘Schedule’ textbox is as follows:

MINUTE (0-59), HOUR (0-23), DAY (1-31), MONTH (1-12), DAY OF THE WEEK (0-7)

**Continuous Integration Interview Questions 17) Why do we use Pipelines in Jenkins?**

Pipeline adds a powerful set of automation tools onto Jenkins, supporting use cases that span from simple continuous integration to comprehensive continuous delivery pipelines. By modeling a series of related tasks, users can take advantage of the many features of Pipeline:

Code: Pipelines are implemented in code and typically checked into source control, giving teams the ability to edit, review, and iterate upon their delivery pipeline.

Durable: Pipelines can survive both planned and unplanned restarts of the Jenkins master.

Pausable: Pipelines can optionally stop and wait for human input or approval before continuing the Pipeline run.

Versatile: Pipelines support complex real-world continuous delivery requirements, including the ability to fork/join, loop, and perform work in parallel.

Extensible: The Pipeline plugin supports custom extensions to its DSL and multiple options for integration with other plugins.

**Questions on Jenkins 18) What is a** Jenkinsfile**?**

A Jenkinsfile is a text file that contains the definition of a Jenkins Pipeline and is checked into source control.

Creating a Jenkinsfile, which is checked into source control, provides a number of immediate benefits:

Code review/iteration on the Pipeline

Audit trail for the Pipeline

Single source of truth for the Pipeline, which can be viewed and edited by multiple members of the project.

**Interview Questions on Jenkins 19) How do you create Multibranch Pipeline in Jenkins?**

The Multibranch Pipeline project type enables you to implement different Jenkinsfiles for different branches of the same project. In a Multibranch Pipeline project, Jenkins automatically discovers, manages and executes Pipelines for branches which contain a Jenkinsfile in source control.

**Devops 20) What is** blue ocean **in Jenkins?**

Blue Ocean is a project that rethinks the user experience of Jenkins, modelling and presenting the process of software delivery by surfacing information that’s important to development teams with as few clicks as possible, while still staying true to the extensibility that is core to Jenkins.

**For Automation Testers**

**For DevOps 21) What are the important plugins in Jenkins?**

s Here is the list of some important **Plugins in Jenkins**:

Maven 2 project

Git

Amazon EC2

HTML publisher

Copy artifact

Join

Green Balls

**Interview Questions on Maven and Jenkins 22) What are Jobs in Jenkins?**

**Jenkins** can be used to perform the typical build server work, such as doing continuous/official/nightly builds, run tests, or perform some repetitive batch tasks. This is called “**free-style software project**” in Jenkins.

**Jenkins Advanced Interview Questions 23) How do you create a Job in Jenkins?**

Go to **Jenkins** top page, select “New Job”, then choose “Build a free-style software project”. This job type consists of the following elements:

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.

**Selenium 24) How do you configuring automatic builds in Jenkins?**

**Builds in Jenkins** can be triggered periodically (on a schedule, specified in configuration), or when source changes in the project have been detected, or they can be automatically triggered by requesting the URL:

http://YOURHOST/jenkins/job/PROJECTNAME/build

**Jenkins CI Interview Questions And s 25) How to create a backup and copy files in Jenkins?**

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.

**Jenkins Real-Time Interview Questions**

**26) What is the trustAnchors parameter must be non-empty error and how can you solve it?**

A) This trustAnchors parameter must be non-empty error means that the truststore you specified was not found, or couldn’t be opened due to access permissions for example.

EJP basically ed the question (and I realize this has an accepted ) but I just dealt with this edge-case gotcha and wanted to immortalize my solution. I had the InvalidAlgorithmParameterException error on a hosted jira server that I had previously set up for SSL-only access.

The issue was that I had set up my keystore in the PKCS12 format, but my truststore was in the JKS format. In my case, I had edited my server.xml file to specify the keystoreType to PKCS, but did not specify the truststoreType, so it defaults to whatever the keystoreType is. Specifying the truststoreType explicitly as JKS solved it for me.

**27) What are the feature differences between Jenkins and Hudson?**

A) Jenkins is the recent fork by the core developers of Hudson. To understand why, you need to know the history of the project. It was originally open source and supported by Sun. Like much of what Sun did, it was fairly open, but there was a bit of benign neglect. The source, trackers, website, etc. were hosted by Sun on their relatively closed java.net platform.

Then Oracle bought Sun. For various reasons Oracle has not been shy about leveraging what it perceives as its assets. Those include some control over the logistic platform of Hudson, and particularly control over the Hudson name. Many users and contributors weren’t comfortable with that and decided to leave.

So it comes down to what Hudson vs Jenkins offers. Both Oracle’s Hudson and Jenkins have the code. Hudson has Oracle and Sonatype’s corporate support and the brand. Jenkins has most of the core developers, the community, and (so far) much more actual work.

In fact, arguably it was Oracle who did the forking! And technically, too, that’s kinda what happened.

It’s interesting to see what comes out of “Hudson” though. While the “Winston summarizes the state and rosy future of the Hudson project” stuff they posted on the (new) Hudson website originally seemed like odd humour to me, perhaps this was a purposeful takeover, and the Sonatype guys actually have some big ideas up their sleeve. This analysis, suggesting a deliberate strategy by Oracle/Sonatype to oust Kohsuke and crew to create a more “enterprisy” Hudson is a very interesting read!

In any case, this brief comparison a fortnight after the split—while not exactly scientific—shows Jenkins to be by far more active of the two projects.

Jenkins has continued the path well-trodden by the original Hudson with frequent releases including many minor updates.

Oracle seems to have largely delegated work on the future path for Hudson to the Sonatype team, who has performed some significant changes, especially with respect to Maven. They have jointly moved it to the Eclipse foundation.

I would suggest that if you like the sound of:

Less frequent releases but ones that are more heavily tested for backwards compatibility (more of an “enterprise-style” release cycle)

A product focused primarily on strong Maven and/or Nexus integration (i.e., you have no interest in Gradle and Artifactory etc)

Professional support offerings from Sonatype or maybe Oracle in preference to Cloudbees etc

You don’t mind having a smaller community of plugin developers etc.  
, then I would suggest Hudson.

Conversely, if you prefer:

More frequent updates, even if they require a bit more frequent tweaking and are perhaps slightly riskier in terms of compatibility (more of a “latest and greatest” release cycle)

A system with more active community support for e.g., other build systems / artifact repositories

Support offerings from the original creator et al. and/or you have no interest in professional support (e.g., you’re happy as long as you can get a fix in next week’s “latest and greatest”)

A classical OSS-style witches’ brew of a development ecosystem

then I would suggest Jenkins.

**Jenkins CI Interview Questions**

**28) How to trigger a build remotely from Jenkins? How to configure Git** post commit **hook?**

The requirement is whenever changes are made in the Git repository for a particular project it will automatically start Jenkins build for that project.

A) 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, your 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.

**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 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.

**Which Scm Tools Does Jenkins Support?**

**Jenkins supports the following SCM tools:**

AccuRev

CVS

Subversion

Git

Mercurial

Perforce

Clearcase

RTC

**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.

**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.

**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.

**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.

**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.

**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.

**What Are The Most Useful Plugins In Jenkins?**

**Some most useful plugins in Jenkins:**

Maven 2 project ,Amazon EC2,HTML publisher ,Copy artefact ,Join ,Green Balls

**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 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.

**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.

**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.

**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.

**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 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.

**Tell Me Jenkins Installation Requirements?**

Jenkins requires Java7 or above and Servlet 3.1 to function.

Java8 is recommended.

Jenkins requires a fair amount of memory to operate well.

Smaller installations should start around 256MB1GB.

**Define Few Plugins Of Jenkins?**

**Here are some plugins that can be used with Jenkins:**

Delivery Pipeline

Join Plugin

Copy Artifact

Git

Android Emulator

Cobertura

Email-ext

Docker

Amazon EC2

Openstack Cloud

CloudBees Folders

**1. What is Jenkins ?**

Jenkins is an apparent source device including plugin established for constant alliance purpose. The main functionality of Jenkins is to retain a record of version controller system and to admit and observe a build system if variations occur. It controls the entire process and contributes reports and warnings to inform.

**2. What is continuous integration ?**

within software growth, when many developers or crews are operating on various sections of the corresponding web application, we require to implement integration test by combining all modules. In sequence to do that an electronic method ([pivotal training](https://svrtechnologies.com/pivotal-training)) for every piece of (pivotal training) code is executed on periodic bases so that all your code gets examined.

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

To practice Jenkins you need

A source code container which is accessible, concerning instance, a Git container

A practical build script, example: A Maven script, indicated within the container

**4. What are the benefits of Jenkins?**

Benefits of Jenkins are

⦁ In integration platform, create crashes are cached  
⦁ Toward every code commit develops to warning build report produces a warning  
⦁ To admonition developers regarding build report progress or ([Salesforce Certification Training](https://www.svrtechnologies.com/salesforce)) abortion, it is combined with LDAP mail server  
⦁ Succeeds constant integration agile improvement and test-driven progress.  
⦁ By easy steps, maven discharge project is electronic  
⦁ Simple tracking of faults at an initial stage in a progressive atmosphere than production

**5. Describe how to move or copy Jenkins from one server to another?**

⦁ Drive a work from one installation of Jenkins to another by copying the relevant work directory.

⦁ By making the clone of a job directory by another name Create a copy of a previously existing job.

⦁ Renaming an existing job by renaming a directory.

**6. What are the commands that are used to begin Jenkins manually?**

To begin Jenkins manually, the commands used are either of the following

(Jenkin\_url)/safeRestart: Allows all running builds to complete

(Jenkins\_url)/restart: Forces a restart without waiting for builds to complete

**7.Discuss some of the valuable plugins in Jenkin?**

Some of the significant plugins in Jenkin involves

HTML publisher

Copy artifact

Maven 2 project

Amazon EC2

Join

Green Balls

**8. Describe how to deploy a custom build of a core plugin?**

To deploy a custom field of a core plugin, these are the following things to do

Stop Jenkins

Tracing the custom HPI to $Jenkins\_Home/plugins

Eliminate the previously expanded plugin directory

Make an empty file called .hpi.pinned

Start Jenkins

**9.Demonstrate how to build a backup and copy files in Jenkins?**

Jenkins preserves all the setting, build artifacts and logs in its home directory, to formulate a back-up of your Jenkins settings, simply copy this directory. And also copy a job directory rename the directory to replicate a job.

**10. Define how to clone a Git repository via Jenkins?**

Just enter the e-mail and username for your Jenkins system. For that, you need to switch to your job directory and execute the “git config” command with this you can clone a Git repository via Jenkins.

**11. 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.

**12. 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.

**13. 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.

**14. Which SCM tools does Jenkins support ?**

Jenkins supports the following SCM tools:

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

**15. 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 ([sap training](https://svrtechnologies.com/sap-training)) process is known as continuous integration.

**16. 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.

**17. Why is Continuous Integration important ?**

Two important reasons:

Defects found early cost less to fix : When a defect is found immediately after a developer codes it, it takes 10x times less time to fix it compared to finding the defect a month later.

Reduced Time to Market : Software is always tested. So, it is always ready to move to further environments.

**18. How is Continuous Integration Implemented ?**

Different tools for supporting Continuous Integration are Hudson, Jenkins and Bamboo. Jenkins is the most popular one currently. They provide integration with various version control systems and build tools.

**19. 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.

**20. Why is the necessity of DevOps ?**

Corporations are now facing the necessity to carrying more and quicker and improved requests to see the ever more persistent demands of mindful users to decrease the Time to Marketplace. Devops often benefits placement to occur very profligate.

**Top 15 Devops Ansible Interview Questions and s Pdf**

**1. What Is Ansible ?**

Ansible is a software tool to deploy application using ssh without sny downtime.It is also used to manage and configure software applications. Ansible is developed by Python language.

**2. What Are The Advantages Of Ansible ?**

Agent-less

Verylow overhead

Good performance

**3. How Ansible Works ?**

There are many similar automation tools available like Puppet, Capistrano, Chef, Salt, Space Walk etc, but Ansible categorize into two types of server: controlling machines and nodes.

The controlling machine, where Ansible is installed and Nodes are managed by this controlling machine over SSH. The location of nodes are specified by controlling machine through its inventory.

The controlling machine (Ansible) deploys modules to nodes using SSH protocol and these modules are stored temporarily on remote nodes and communicate with the Ansible machine through a JSON connection over the standard output.

**How Ansible Works:**

Ansible is agent-less, that means no need of any agent installation on remote nodes, so it means there are no any background daemons or programs are executing for Ansible, when it’s not managing any nodes.

Ansible can handle 100’s of nodes from a single system over SSH connection and the entire operation can be handled and executed by one single command ‘ansible’. But, in some cases, where you required to execute multiple commands for a deployment, here we can build playbooks.

Playbooks are bunch of commands which can perform multiple tasks and each playbooks are in YAML file format.

**4. What’s The Use Of Ansible ?**

Ansible can be used in IT infrastructure to manage and deploy software applications to remote nodes. For example, let’s say you need to deploy a single software or multiple software to 100’s of nodes by a single command, here ansible comes into picture, with the help of Ansible you can deploy as many as applications to many nodes with one single command, but you must have a little programming knowledge for understanding the ansible scripts.

We’ve compiled a series on Ansible, title ‘Preparation for the Deployment of your IT Infrastructure with Ansible IT Automation Tool‘, through parts 1-4 and covers the following topics.

**5. Is There A Web Interface / Rest Api / Etc ?**

Yes, Ansible, Inc makes a great product that makes Ansible even more powerful and easy to use. See Ansible Tower.

**6. How Do I Submit A Change To The Documentation ?**

Documentation for Ansible is kept in the main project git repository, and complete instructions for contributing can be found in the docs.

**7. When Should I Use {{ }} ? Also, How To Interpolate Variables Or Dynamic Variable Names ?**

A steadfast rule is ‘always use {{ }} except when when:‘. Conditionals are always run through Jinja2 as to resolve the expression, so when: failed\_when: and changed\_when: are always templated and you should avoid adding {{}}.

In most other cases you should always use the brackets, even if previouslly you could use variables without specifying (like with\_ clauses), as this made it hard to distinguish between an undefined variable and a string.

Another rule is ‘moustaches don’t stack’. We often see this:

{{ somevar\_{{other\_var}} }}

The above DOES NOT WORK, if you need to use a dynamic variable use the hostvars or vars dictionary as appropriate:

{{ hostvars[inventory\_hostname][‘somevar\_’ + other\_var] }}

**8. How To Install Ansible ?**

Installation of Ansible Ubuntu 14.04

The best way to get Ansible for Ubuntu is to add the project’s PPA (personal package archive) to your system.

To do this effectively, we need to install the software-properties-common package, which will give us the ability to work with PPAs easily. (This package was called python-software-properties on older versions of Ubuntu.)

sudo apt-get update

sudo apt-get install software-properties-common

**Once the package is installed, we can add the Ansible PPA by typing the following command:**

sudo apt-add-repository ppa:ansible/ansible

Press ENTER to accept the PPA addition.

**Next, we need to refresh our system’s package index so that it is aware of the packages available in the PPA. Afterwards, we can install the software:**

sudo apt-get update

sudo apt-get install ansible

We now have all of the software required to administer our servers through Ansible.

**9. How Do I Generate Crypted Passwords For The User Module ?**

The mkpasswd utility that is available on most Linux systems is a great option:

mkpasswd –method=sha-512

If this utility is not installed on your system (e.g. you are using OS X) then you can still easily generate these passwords using Python. First, ensure that the Passlib password hashing library is installed.

pip install passlib

Once the library is ready, SHA512 password values can then be generated as follows:

python -c “from passlib.hash import sha512\_crypt; import getpass; print sha512\_crypt.encrypt(getpass.getpass())”

Use the integrated Hashing filters to generate a hashed version of a password. You shouldn’t put plaintext passwords in your playbook or host\_vars; instead, use Vault to encrypt sensitive data.

**10. Desired To Gain Proficiency On Ansible ?**

Explore the blog post on Ansible training to become a pro in Ansible.

**11. How Do I Get Ansible To Reuse Connections, Enable Kerberized Ssh, Or Have Ansible Pay Attention To My Local Ssh Config File ?**

Switch your default connection type in the configuration file to ‘ssh’, or use ‘-c ssh’ to use Native OpenSSH for connections instead of the python paramiko library. In Ansible 1.2.1 and later, ‘ssh’ will be used by default if OpenSSH is new enough to support ControlPersist as an option.

Paramiko is great for starting out, but the OpenSSH type offers many advanced options. You will want to run Ansible from a machine new enough to support ControlPersist, if you are using this connection type. You can still manage older clients. If you are using RHEL 6, CentOS 6, SLES 10 or SLES 11 the version of OpenSSH is still a bit old, so consider managing from a Fedora or openSUSE client even though you are managing older nodes, or just use paramiko.

We keep paramiko as the default as if you are first installing Ansible on an EL box, it offers a better experience for new users.

**12. What Is The Best Way To Make Content Reusable/redistributable ?**

If you have not done so already, read all about “Roles” in the playbooks documentation. This helps you make playbook content self-contained, and works well with things like git submodules for sharing content with others.

If some of these plugin types look strange to you, see the API documentation for more details about ways Ansible can be extended.

**13. How Do I See All The Inventory Vars Defined For My Host ?**

You can see the resulting vars you define in inventory running the following command:

ansible -m debug -a “var=hostvars[‘hostname’]” localhost.

**14. How Do I Copy Files Recursively Onto A Target Host ?**

The “copy” module has a recursive parameter, though if you want to do something more efficient for many files, look at the “synchronize” module instead, which wraps rsync. See the module index for info on both modules.

**15. How Do I Access A Variable Name Programmatically ?**

An example may come up where we need to get the ipv4 address of an arbitrary interface, where the interface to be used may be supplied via a role parameter or other input. Variable names can be built by adding strings together, like so:

{{ hostvars[inventory\_hostname][‘ansible\_’ + which\_interface][‘ipv4’][‘address’] }}

The trick about going through hostvars is necessary because it’s a dictionary of the entire namespace of variables. ‘inventory\_hostname’ is a magic variable that indicates the current host you are looping over in the host loop.