

Jfrog_project1 Config [Jenkins] - Mozilla Firefox

Your JFrog Artifactory | Artifactory | Jfrog_project1 Config | sr1n1v4s/jfrog_project1 | JFrogDev/project-exa... | +

192.168.1.5:8080/job/Jfrog_project1/configure | Search | Star | Home | Back | Forward | Stop | Help | More

Most Visited | Getting Started | paltalk

Jenkins > Jfrog_project1 >

General Source Code Management Build Triggers Build Environment Build Post-build Actions

Source Code Management

None Git

Repositories

Repository URL: https://github.com/sr1n1v4s/jfrog_project1.git ?

Credentials: - none - Add

Advanced...

Add Repository

Branches to build

Branch Specifier (blank for 'any'): */master X ?

Add Branch

Repository browser (Auto) ?

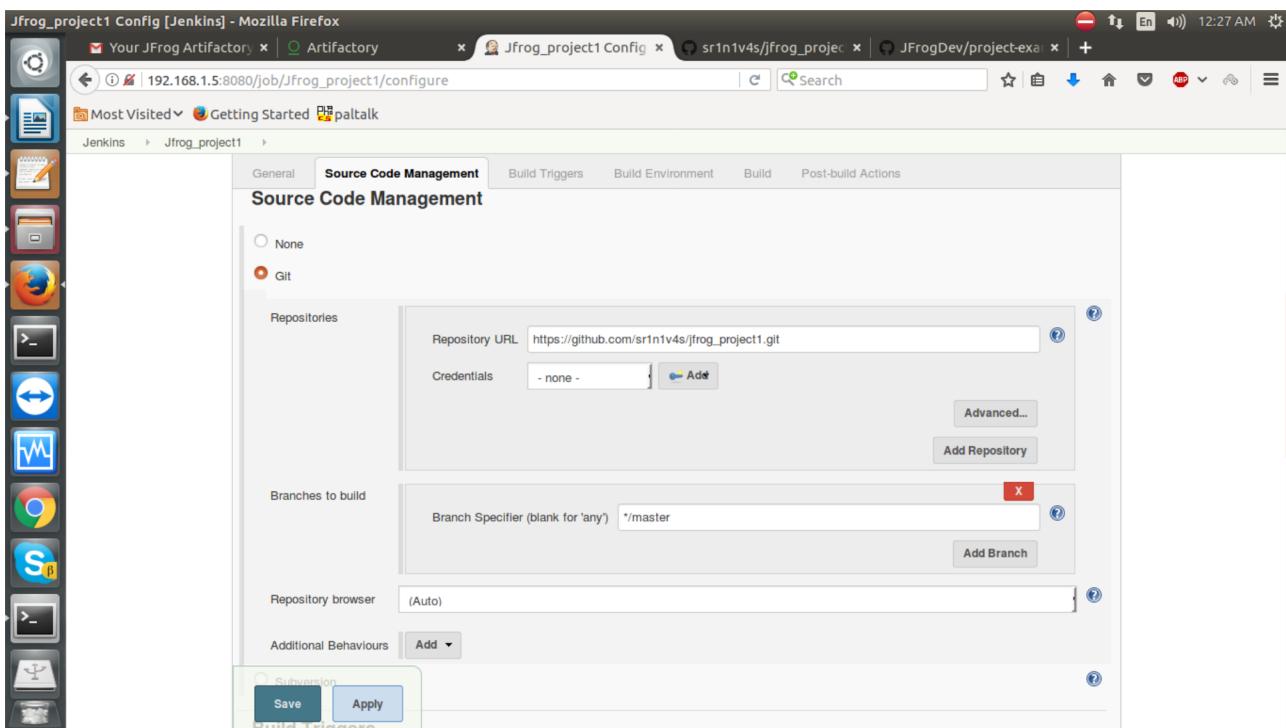
Additional Behaviours Add ▾

?

Subversion ?

Save Apply

Build Triggers



Jfrog_project1 Config [Jenkins] - Mozilla Firefox

Your JFrog Artifactory | Artifactory | Jfrog_project1 Config | sr1n1v4s/jfrog_project1 | JFrogDev/project-exa... | +

192.168.1.5:8080/job/Jfrog_project1/configure | Search | Star | Home | Back | Forward | Stop | Help | More

Most Visited | Getting Started | paltalk

Jenkins > Jfrog_project1 >

General Source Code Management Build Triggers **Build Environment** Artifactory Configuration Build Post-build Actions

Build Environment

Delete workspace before build starts
 Provide Configuration files
 Abort the build if it's stuck
 Add timestamps to the Console Output
 Ant/Ivy-Artifactory Integration
 Generic-Artifactory Integration
 Gradle-Artifactory Integration
 Maven3-Artifactory Integration ?

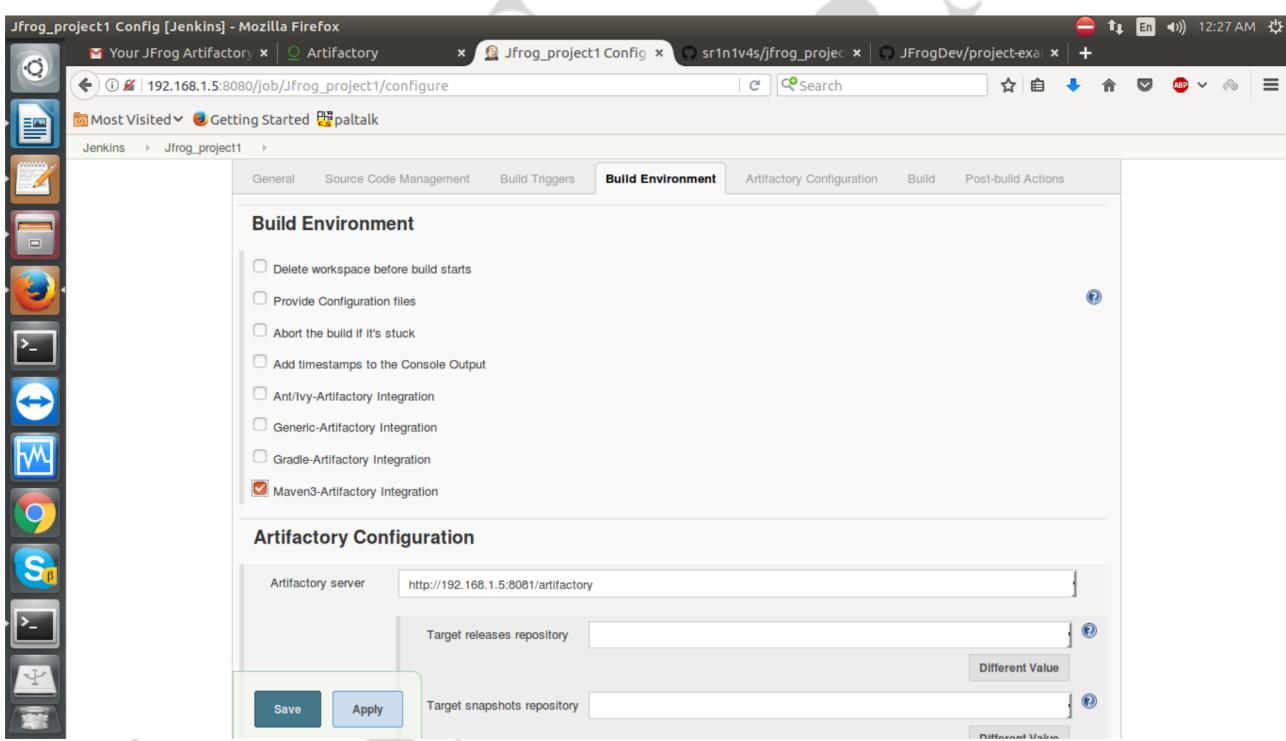
Artifactory Configuration

Artifactory server: http://192.168.1.5:8081/artifactory ?

Target releases repository: Different Value

Target snapshots repository: Different Value

Save Apply



Now Save & click on Build Now
Check the Output

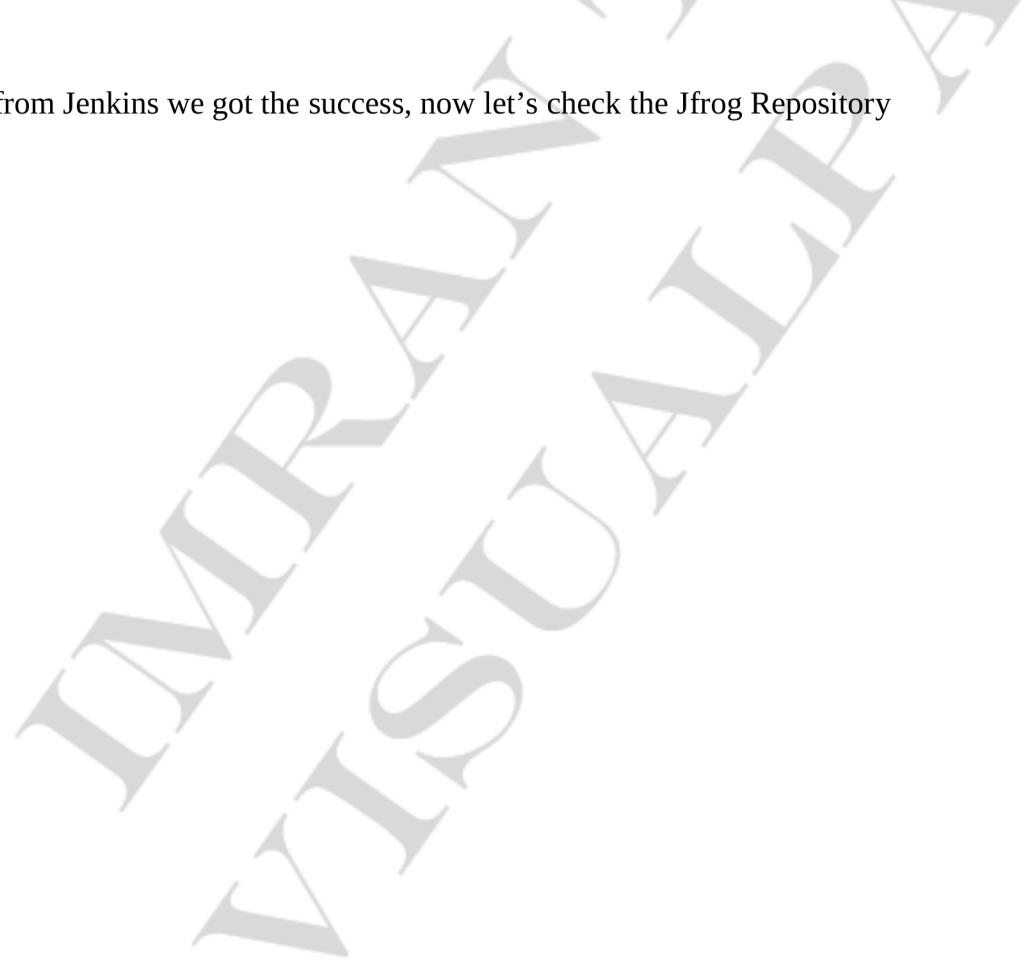
```

Started by user srinivas
Building in workspace /var/lib/jenkins/workspace/Jfrog_project1
Cloning the remote Git repository
  > git init /var/lib/jenkins/workspace/Jfrog_project1 # timeout=10
Cloning repository https://github.com/srlnlv4s/jfrog_project1.git
  > git fetch -t --progress https://github.com/srlnlv4s/jfrog_project1.git +refs/heads/*:refs/remotes/origin/*
  > git config remote.origin.url https://github.com/srlnlv4s/jfrog_project1.git # timeout=10
  > git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
  > git config remote.origin.url https://github.com/srlnlv4s/jfrog_project1.git # timeout=10
Fetching upstream changes from https://github.com/srlnlv4s/jfrog_project1.git
  > git fetch -t --progress https://github.com/srlnlv4s/jfrog_project1.git +refs/heads/*:refs/remotes/origin/*
  > git rev-parse refs/remotes/origin/master^{commit} # timeout=10
  > git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 6b9040df074d7caa647f3fa13943d96f08957d21 (refs/remotes/origin/master)
  > git config core.sparsecheckout # timeout=10
  > git checkout -f 6b9040df074d7caa647f3fa13943d96f08957d21
First time build. Skipping changelog.
Jenkins Artifactory Plugin version: 2.11.0
Artifactory integration is enabled
[Jfrog_project1] $ /var/lib/jenkins/tools/hudson.model.JDK/jdk8/bin/java -classpath /var/lib/jenkins/tools/hudson.tasks.Maven.MavenInstallation/mvn3/boot/plexus-classworlds-2.5.2.jar -Dmaven.home=/var/lib/jenkins/tools/hudson.tasks.Maven.MavenInstallation/mvn3 -DbuildInfoConfig.propertiesFile=/tmp/buildInfo9000479368004298483.properties -Dmaven.plugin.lib=/var/lib/jenkins/cache/artifactory-plugin/2.11.0 -Dclassworlds.conf=/tmp/classworlds5017735712468899130conf -Dmaven.multiModuleProjectDirectory=/var/lib/jenkins/workspace/Jfrog_project1/pom.xml org.codehaus.plexus.classworlds.launcher.Launcher -f pom.xml clean install
[main] INFO org.apache.maven.cli.event.ExecutionEventLogger - Scanning for projects...
[main] INFO org.jfrog.build.extractor.maven.BuildInfoRecorder - Initializing Artifactory Build-Info Recording
[main] INFO org.apache.maven.cli.event.ExecutionEventLogger -

```

Visualpath Training & Consulting.

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: - +91-970 445 5959, 961 824 5689 E-Mail ID : online.visualpath@gmail.com, Website : www.visualpath.in.



Jfrog_project1 #1 Console [Jenkins] - Mozilla Firefox
Your JFrog Artifactory | Artifactory | Jfrog_project1 #1 Con... | srinivas/jfrog_proje... | JFrogDev/project-exa... | +
192.168.1.5:8080/job/Jfrog_project1/1/console | Search | ☆ | ☰ | +
Most Visited | Getting Started | paltalk
Jenkins > Jfrog_project1 > #1

[main] INFO org.jfrog.build.extractor.maven.BuildInfoClientBuilder - Deploying artifact: http://192.168.1.5:8081/artifactory/jfrog_project1/org/jfrog/test/multi2/2.17-SNAPSHOT/multi2-2.17-SNAPSHOT.jar
[main] INFO org.jfrog.build.extractor.maven.BuildInfoClientBuilder - Deploying artifact: http://192.168.1.5:8081/artifactory/jfrog_project1/org/jfrog/test/multi2/2.17-SNAPSHOT/multi2-2.17-SNAPSHOT.pom
[main] INFO org.jfrog.build.extractor.maven.BuildDeploymentHelper - Artifactory Build Info Recorder: Deploying build info ...
[main] INFO org.jfrog.build.extractor.maven.BuildInfoClientBuilder - Deploying build descriptor to: http://192.168.1.5:8081/artifactory/api/build
[main] INFO org.jfrog.build.extractor.maven.BuildInfoClientBuilder - Build successfully deployed. Browse it in Artifactory under http://192.168.1.5:8081/artifactory/webapp/builds/Jfrog_project1/1
[main] INFO org.apache.maven.cli.event.ExecutionEventLogger -
.....
[main] INFO org.apache.maven.cli.event.ExecutionEventLogger - Reactor Summary:
[main] INFO org.apache.maven.cli.event.ExecutionEventLogger -
[main] INFO org.apache.maven.cli.event.ExecutionEventLogger - Simple Multi Modules Build
.....
[main] INFO org.apache.maven.cli.event.ExecutionEventLogger - SUCCESS [02:56 min]
[main] INFO org.apache.maven.cli.event.ExecutionEventLogger - Multi 1
.....
[main] INFO org.apache.maven.cli.event.ExecutionEventLogger - SUCCESS [01:05 min]
[main] INFO org.apache.maven.cli.event.ExecutionEventLogger - Multi 2
.....
[main] INFO org.apache.maven.cli.event.ExecutionEventLogger - SUCCESS [3.052 s]
[main] INFO org.apache.maven.cli.event.ExecutionEventLogger - Multi 3
.....
[main] INFO org.apache.maven.cli.event.ExecutionEventLogger - SUCCESS [22.801 s]
[main] INFO org.apache.maven.cli.event.ExecutionEventLogger -
.....
[main] INFO org.apache.maven.cli.event.ExecutionEventLogger - BUILD SUCCESS
[main] INFO org.apache.maven.cli.event.ExecutionEventLogger -
.....
[main] INFO org.apache.maven.cli.event.ExecutionEventLogger - Total time: 04:35 min
[main] INFO org.apache.maven.cli.event.ExecutionEventLogger - Finished at: 2017-06-07T00:34:26+05:30
[main] INFO org.apache.maven.cli.event.ExecutionEventLogger - Final Memory: 40M/243M
[main] INFO org.apache.maven.cli.event.ExecutionEventLogger -
.....
Finished: SUCCESS

from Jenkins we got the success, now let's check the Jfrog Repository

The screenshot shows the JFrog Artifactory interface. The top navigation bar includes tabs for 'Your JFrog Artifactory' (selected), 'Artifactory', 'Jfrog_project1 #1 Con...', 'sr1n1v4s/jfrog_proje...', and 'JFrogDev/project-exa...'. The address bar shows the URL as '192.168.1.5:8081/artifactory/webapp/#/artifacts/browse/tree/General/libs-release'. The main content area is titled 'Artifact Repository Browser' and displays the 'libs-release' repository under the 'General' tab. The 'Info' section shows details: Name: libs-release, Package Type: Maven, Repository Path: libs-release/, and Repository Layout: maven-2-default. A 'Maven' badge is present. Below this, the 'Included Repositories' section lists 'libs-release-local' and 'jcenter'. On the left, there's a sidebar with various icons and a tree view showing 'libs-release' selected. The bottom right corner has a 'Trash Can' icon.

The screenshot shows the JFrog Artifactory interface. The left sidebar contains various icons for navigation and management. The main area is titled "Artifact Repository Browser". On the left, there's a tree view of the repository structure under "Jfrog_project1", including "org/jfrog/test/multi", "multi1", "multi2", and "multi3" with their respective sub-directories and metadata files. A "libs-release-local" entry is also visible. At the bottom of the tree view, there are links for "Trash Can" and "Create". To the right, a detailed configuration page for "libs-release" is displayed. It includes sections for "General" (with fields for Name, Package Type, Repository Path, and Repository Layout), "Info" (with Maven logo), and "Included Repositories" (listing "libs-release-local" and "jcenter"). A "Actions" button is located at the top right of the configuration panel.

Visualpath Training & Consulting.

Visualpath Training & Consulting
Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: - +91-970 445 5959, 961 824 5689 E-Mail ID : online.visualpath@gmail.com, Website : www.visualpath.in.

Conclusion:

Continuous Integration is a mandatory procedure we need to do to setup code delivery pipelines and that's the first thing we should automate in DevOps Lifecycle.

Jenkins is the most famous CI tool.

Jenkins provides you with so many features and can be used for other purposes also like Deployments and cloud automation.

Jenkins gets integrated with almost every Devops or developer's tool in the market.

Later in the book we will see ansible, puppet and docker, Jenkins gets integrated with these tool very nicely through its plugins.

I say its one of the most important tool in DevOps.

XII. Ansible Tutorial

Ansible is a very powerful open source automation tool. It comes in the category of Configuration Management tools like Puppet, Chef, Saltstack etc. Its simplest among all the configuration management tool in terms of its easy to read & write approach and Masterless/serverless architecture.

Being simple in use it can handle most complex of the task when it comes to Orchestration, where you run automation tasks in chain and order on several different servers and devices.

1. Configuration Management

In simple terms, tools that manages configuration of IT infrastructures like OS, network devices, Application, softwares and Cloud computing services are called as Configuration Management tools.

Configuration of IT Infrastructure like Softwares & OS changes happens every now and then. We add, remove and update them for good. Doing such changes to hundreds of servers and devices is very time consuming and error prone task if you do it manually.

These tools help you manage and automate all that changes with ease and from a centralised place.

Some features

Centralised configuration

Configuration of various servers and devices are managed from a central server. Like you would be managing webservers, dbservers or switches all of their configurations like softwares, patches, config files etc can be setup in the central server. When the node wants to update the latest changes, it can fetch from the server or pushed to the node from server.

Enforcement.

Configuration enforcement may be the single most important feature of a configuration management tool. By running regularly and ensuring the machine is configured to the desired state, configuration management tools prevent configuration to go out of sync. Configuration drift can happen in a variety of ways: Package updates, live debugging, "helpful" co-workers, etc. Whatever the cause, being able to say with confidence, "This is how this machine is configured," is a great way to shorten incident resolution time and reduce surprises.

Abstraction.

Few sysadmins maintain completely homogeneous environments. Even if you're an all-Linux shop, you probably have multiple distros that you support, or at least multiple versions of a distro. With configuration management tools, many of the operating-system-specific implementations of a

Visualpath Training & Consulting.

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: +91-970 445 5959, 961 824 5689 E-Mail ID : online.visualpath@gmail.com, Website : www.visualpath.in.

configuration are abstracted away for you. The same configuration file can be used to manage, for example, the installation of Apache HTTPD on both Red Hat and Ubuntu systems.

Version control friendly.

Of course, the best way to enable cooperation is to have everything in a version control system. All of the tools listed below use some form of text for configuration. This means you can take advantage of the benefits of your favourite version control system.

Replication.

Configuration management makes it easy to replicate environments with the exact same software and configurations. This enables you to effectively build a multistage ecosystem, with production, development, and testing servers.

Some Configuration Management Tools

- Puppet
- Chef
- Ansible
- Saltstack
- CFEngine

2. Some Terminologies

Change Management

It is process when any specific configuration of the machine or software is changed. A typical CM process in IT industry goes through a approval process from customer or higher management, of course we are talking about production systems. A single change on 1, 10 or 100's of servers has to be done very precisely and effectively. For example, upgrading a software package which has a bug on hundreds of servers or restarting a service like webservice to take effect of any new changes.

Provisioning

In general, provisioning means "providing" or making something available in the IT infrastructure. For example, provisioning a web server means installing and setting up web service softwares and its configuration on a OS. Provisioning a cloud instance means creating a virtual machine on the cloud.

Prepare a system to make it ready

Transition from one state to a different state

Provisioning

Examples

- Make an FTP Server
- Make an Email Server
- Make a DB Server



Orchestration

Automation is when we group list of tasks and execute them in top to down order through a script generally like installing webserver through a script. Whereas Orchestration is the process where we group multiple automated tasks and execute them into an order.

For example, if we are orchestrating setup of a multi-tier web application, we have to setup all the services like Databases, Webservices, Loadbalancer, monitoring in an order so it gets validate when we have entire setup running.

First, we will setup Database service so when webservice is setup it gets connected to database and gets validated. Next, we may setup Loadbalancer and add webservices under it. Monitoring would be setup at the very end.

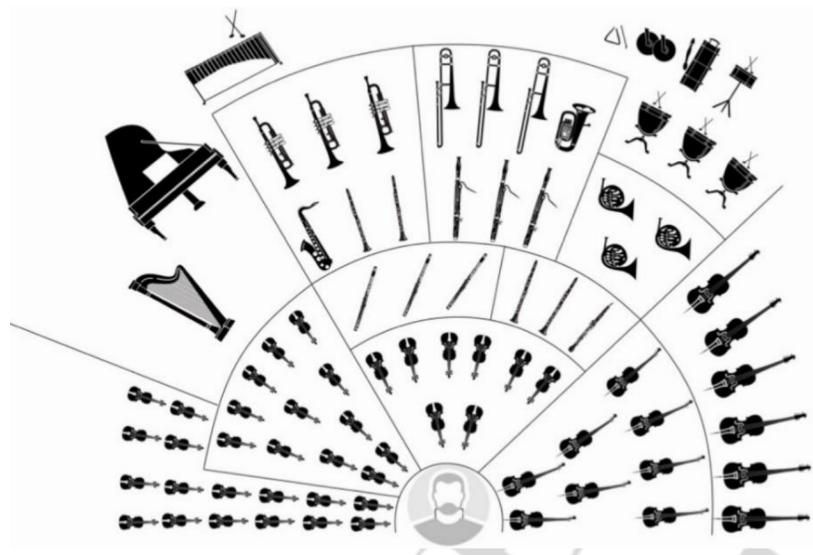
Now monitoring services will return right results from loadbalancer, web service and database service. This process is called as Orchestration but not limited to just this and is done for variety of other IT infrastructure process like cloud computing, network setup etc.

Orchestration

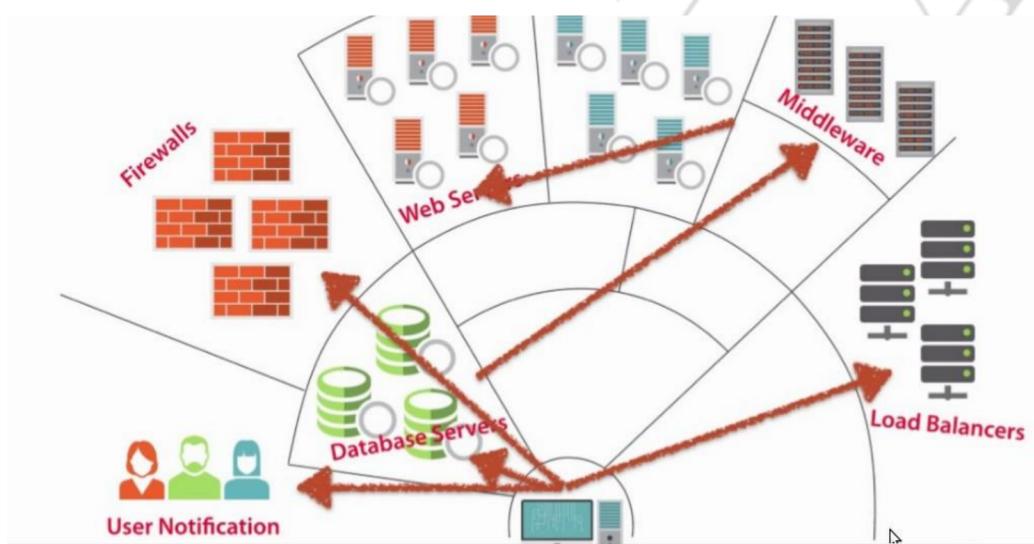
Coordinates automation BETWEEN systems

- Task 1 - System 1
- Task 2 - System 2
- Task 3 - System 3
- Task 4 - System 1

Orchestration word has come from the word Orchestra where different instruments are played in a proper order to generate the beautiful Melody. It's all about order otherwise you don't get music all you get is sound and probably not so good.



Similarly, IT orchestration is all about automating tasks in a proper order.



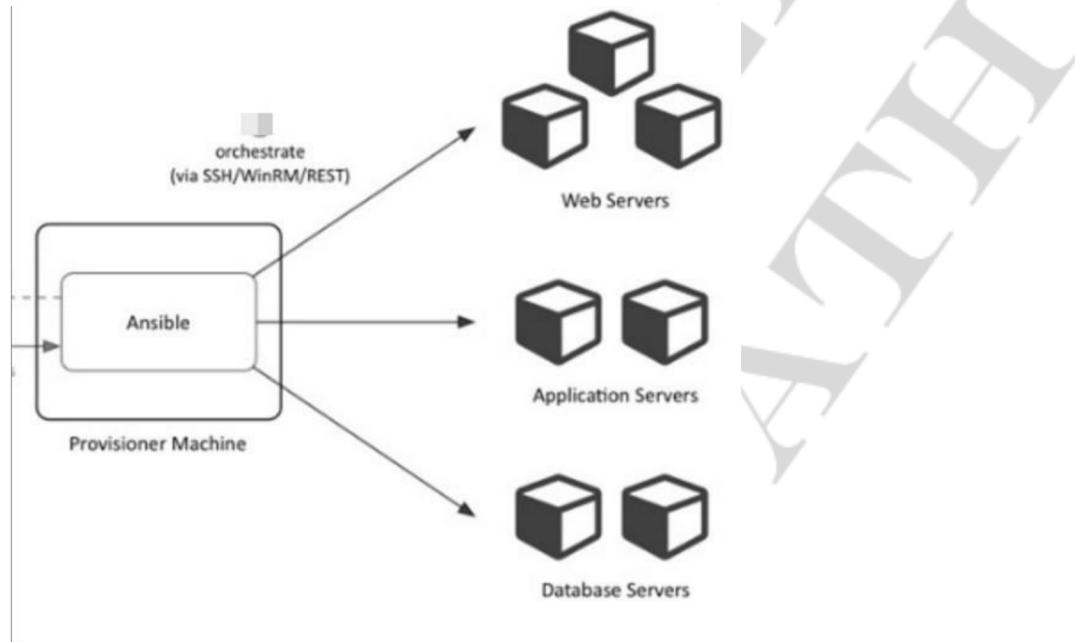
Idempotent Behaviour

Configuration management tools keep track of the state of resources in order to avoid repeating tasks that were executed before. If a package was already installed, the tool won't try to install it again. The objective is that after each provisioning run the system reaches (or keeps) the desired state, even if you run it multiple times. This is what characterizes these tools as having an *idempotent behaviour*. One more example would be if we are trying to push a file to multiple servers and some servers already have the same file with the same content then it's not going to overwrite the file, will simply skip it and push it to their servers where there is a mismatch.

3. Ansible Introduction

Ansible can control large number of servers and eases administration and operations tasks. Ansible can do simple configuration management and complex orchestration, it has all the features that config tools have plus it's very easy to learn and implement.

It communicates over normal SSH channels in order to retrieve information from remote machines, issue commands, and copy files. For windows node it uses winrm.



Since it uses SSH for Linux and winrm for windows there no need to setup anything at the agent side as ssh comes by default in Linux servers and so as winrm in windows server.

4. Installing Ansible

Ansible by default manages machines over the SSH protocol.

Once Ansible is installed, it will not add a database, and there will be no daemons to start or keep running. You only need to install it on one machine (which could easily be a laptop) and it can manage an entire fleet of remote machines from that central point.

Latest Release Via Yum

RPMs are available from yum for [EPEL](#) 6, 7, and currently supported Fedora distributions.

Ansible itself can manage earlier operating systems that contain Python 2.6 or higher (so also EL6). Fedora users can install Ansible directly, though if you are using RHEL or CentOS and have not already done so, [configure EPEL](#)

```
# install the epel-release RPM if needed on CentOS, RHEL, or Scientific Linux
$ sudo yum install ansible
```

Latest Releases Via Apt (Ubuntu)

Ubuntu builds are available [in a PPA here](#).

(Windows isn't supported for the control machine). To configure the PPA on your machine and install ansible run these commands:

```
$ sudo apt-get install software-properties-common
$ sudo apt-add-repository ppa:ansible/ansible
$ sudo apt-get update
$ sudo apt-get install ansible
```

Some Quick Notes:

- Machine, where ansible is installed is called as **Control Machine**.
- Ansible is written in Python Language.
- You should have python 2.6/2.7 to install ansible on control machine.
- Windows isn't supported for the control machine.
- Ansible can automate tasks on Linux and windows Machines

5. Inventory

Inventory is a text file where you define the host information that you want to manage with ansible. The default inventory file location is /etc/ansible/hosts. You can specify a different inventory file using the -i <path> option on the command line.

For this exercise we need two Linux servers, you can spin two centos vm or ec2 instance for practice.

Hosts and Groups

Create a file named inventory-dev(name can be anything) and add below mentioned entry.

Visualpath Training & Consulting.

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: +91-970 445 5959, 961 824 5689 E-Mail ID : online.visualpath@gmail.com, Website : www.visualpath.in.

```

web1 ansible_ssh_host=192.168.1.13 ansible_ssh_user=vagrant ansible_ssh_pass='vagrant'
db1 ansible_ssh_host=192.168.1.14 ansible_ssh_user=vagrant ansible_ssh_pass='vagrant'

[webservers]
web1

[dbservers]
db1

```

Explanation

- web1 and db1 are the names that we have given to the hosts.
- ansible_ssh_host is the variable and its value is the IP address of the server.
- ansible_ssh_user variable holds the username
- ansible_ssh_password holds the password
- [webservers] & [dbservers] is the name of the group which can contain n number hosts. Groupnames are enclosed in square brackets [] .

Note: Mentioning password in the inventory file is not recommended, it's just for initial learning later we will do ssh key exchange.

6. Inventory for Production systems/Real Time.

As we have seen above we put the password in clear text and IP address information also in the inventory. This is a real concern for security, you cannot share this inventory with anyone and also cannot track it in VCS like git. We have better ways to deal with this situation.

1. Since ansible uses SSH, its always recommended to do SSH key exchange and authorize ansible server login to the nodes its managing.

Note: Refer Bash Scripting chapter to learn SSH key exchange.

This way we don't need to mention username and password in the inventory file.

2. Next thing is the IP address, we can manage that with the /etc/hosts file. Map IP to hostname in /etc/hosts file and you can then mention the hostname directly in the inventory.

HOSTS File

```

$ cat /etc/hosts
192.168.1.13 web1
192.168.1.14 db1

```

Visualpath Training & Consulting.

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: - +91-970 445 5959, 961 824 5689 E-Mail ID : online.visualpath@gmail.com, Website : www.visualpath.in.

INVENTORY File

```
$ cat inventory-dev
'
[webservers]
web1

[dbservers]
db1
```

So now our inventory is very simple and just contain the group and hostname, which is safe.

7. Adhoc command

Ansible gives a quick method to communicate and execute commands on remote/local machines through Adhoc commands.

```
imran@DevOps:~/..../exercise1$ ansible -i inventory-dev -m ping web1
web1 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}

imran@DevOps:~/..../exercise1$ ansible -i inventory-dev -m ping db1
db1 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}

imran@DevOps:~/..../exercise1$ ansible -i inventory-dev -m ping webservers
web1 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}

imran@DevOps:~/..../exercise1$ ansible -i inventory-dev -m ping dbservers
db1 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

Visualpath Training & Consulting.

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: +91-970 445 5959, 961 824 5689 E-Mail ID : online.visualpath@gmail.com, Website : www.visualpath.in.

```

}

imran@DevOps:~/..../exercise1$ ansible -i inventory-dev -m ping all
web1 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
db1 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}

```

Explanations

- Adhoc commands are executed by specifying “**ansible**” command.
- -i <inventory name> tells ansible to pick up host information from this file, if not specified ansible will look for the host information in /etc/ansible/hosts
- -m means the module name. **ping** is a module which will login to the host and check the connectivity.
- Web1, db1, webservers, dbservers are the host and group name where we want to execute the task, “all” means all the hosts from the inventory file

Host key checking Error

```

"msg": "ERROR! Using a SSH password instead of a key is not possible because
Host Key checking is enabled and sshpass does not support this. Please add this
host's fingerprint to your known_hosts file to manage this host."
}

```

Solution:

Open /etc/ansible/ansible.cfg file and uncomment “#host_key_checking = False” by removing #
 sudo vi /etc/ansible/ansible.cfg

```
# uncomment this to disable SSH key host checking
host_key_checking = False
```

Group of Groups and Group Variables

Variables can be applied to an entire group

```
[webservers]
web1
```

Visualpath Training & Consulting.

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: +91-970 445 5959, 961 824 5689 E-Mail ID : online.visualpath@gmail.com, Website : www.visualpath.in.

```

[dbservers]
db1

[datacenter:children]
webservers
dbservers

[datacenter:vars]
ansible_ssh_user=vagrant
ansible_ssh_pass='vagrant'

```

Explanation

- [datacenter:children] tells datacenter is a master group and underneath we specify other group names
- [datacenter:vars] is used to define variable at the group level, you can specify variable on any group by specifying [groupname:vars].

Note

Ansible 2.0 has deprecated the “ssh” from `ansible_ssh_user`, `ansible_ssh_host`, and `ansible_ssh_port` to become `ansible_user`, `ansible_host`, and `ansible_port`. If you are using a version of Ansible prior to 2.0, you should continue using the older style variables (`ansible_ssh_*`). These shorter variables are ignored, without warning, in older versions of Ansible.

General for all connections:

ansible_host

The name of the host to connect to, if different from the alias you wish to give to it.

ansible_port

The ssh port number, if not 22

ansible_user

The default ssh user name to use.

Specific to the SSH connection:

Visualpath Training & Consulting.

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: - +91-970 445 5959, 961 824 5689 E-Mail ID : online.visualpath@gmail.com, Website : www.visualpath.in.

ansible_ssh_pass

The ssh password to use (never store this variable in plain text; always use a vault. See [Variables and Vaults](#))

ansible_ssh_private_key_file

Private key file used by ssh. Useful if using multiple keys and you don't want to use SSH agent.

8. About Modules

Modules (also referred to as “task plugins” or “library plugins”) are the ones that do the actual work in ansible, they are what gets executed in each playbook task. But you can also run a single one using the ‘ansible’ command.

List of all the modules

https://docs.ansible.com/ansible/list_of_all_modules.html

```
imran@DevOps:~/.../exercise1$ ansible-doc -l
```

Let's review how we execute three different modules from the command line:

```
ansible webservers -m service -a "name=httpd state=started"  
ansible webservers -m ping  
ansible webservers -m command -a "/sbin/reboot -t now"
```

Installing package

```
imran@DevOps:~/.../exercise1$ ansible -i inventory-dev -m yum -a "name=httpd  
state=installed" web1 --sudo  
  
web1 | SUCCESS => {  
    "changed": false,  
    "msg": "",  
    "rc": 0,  
    "results": [  
        "httpd-2.2.15-59.el6.centos.x86_64 providing httpd is already installed"  
    ]  
}
```

Explanation

- “yum” is a ansible module that manages packages on red hat based systems, for Debian based we use module named “apt”.

- **-a** is used to provide arguments for the module like name=httpd(key=value). Majority of the modules will have arguments, some arguments are mandatory like “name” argument for “yum”.

\$ ansible-doc yum will show you list of all the arguments for yum module.

- **--sudo** tells ansible to execute the module with root privileges, user should have the sudo privileges or else the module will fail.

Starting service.

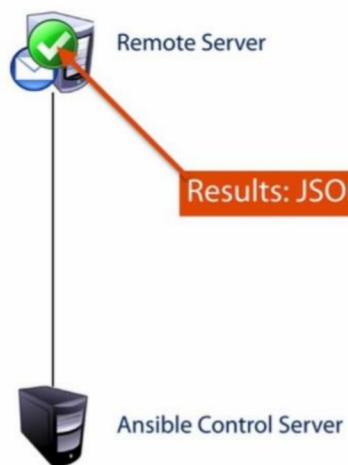
```
imran@DevOps:~/.../exercise1$ ansible -i inventory-dev -m service -a "name=httpd
state=started" web1 --sudo

web1 | SUCCESS => {

    "changed": true,
    "name": "httpd",
    "state": "started"
}
```

Output of adhoc commands

ansible command return output is json format



```
web1 | SUCCESS => {

    "changed": true,
    "name": "httpd",
    "state": "started"
}
```

- web1 is the name of the hosts on which module got executed.
- Status is SUCCESS that means it got executed successfully.
- changed: true means that the module execution made some changes in web1.
- changed: false means that the system is in the same desired state as shown below.

```
imran@DevOps:~/.../exercise1$ ansible -i inventory-dev -m service -a "name=httpd
state=started" web1 --sudo

web1 | SUCCESS => {
    "changed": false,
    "name": "httpd",
    "state": "started"
}
```

- ➔ httpd service was already running on web1 so even executing the adhoc command again will not make any changes this is called the **IDEMPOTENT** behaviour.

Few more sample modules with adhoc commands.

To transfer a file directly to many servers:

```
$ ansible -i inventory-dev -m copy -a "src=/etc/hosts dest=/tmp/hosts"
datacenter
```

The **file** module allows changing ownership and permissions on files. These same options can be passed directly to the **copy** module as well:

```
$ ansible webservers -m file -a "dest=/opt/info.txt mode=600"
$ ansible webservers -m file -a "dest=/opt/info.txt mode=600 owner=devops
group=devops"
```

Ensure a package is installed, but don't update it:

```
$ ansible webservers -m yum -a "name=acme state=present"
```

Ensure a package is installed to a specific version:

```
$ ansible webservers -m yum -a "name=acme-1.5 state=present"
```

Ensure a package is at the latest version:

```
$ ansible webservers -m yum -a "name=acme state=latest"
```

Ensure a package is not installed:

```
$ ansible webservers -m yum -a "name=acme state=absent"
```

9. Ansible Configuration

Certain settings in Ansible are adjustable via a configuration file. The stock configuration should be sufficient for most users, but there may be reasons you would want to change them.

Changes can be made in global ansible.cfg file /etc/ansible/ansible.cfg or you can create your own ansible.cfg (current working directory) which will have higher precedence over the global file.

```
$ cat ansible.cfg
[defaults]
hostfile = inventory_prod
host_key_checking=False
#ask_sudo_pass = True
```

This ansible.cfg file is located at the same place where you have the inventory file.

Explanation

- [defaults] is the main section of ansible.cfg
- hostfile will have the value where inventory file is located if it's in the current working directory specify the name or else complete path of the file should be specified. After mentioning inventory path in ansible.cfg its not required to pass the inventory path with -i option.
- host_key_checking=False tells ansible to not check the host fingerprints before doing ssh to the host.

There long list of ansible config parameters that you can choose from. List of ansible configuration is specified in ansible documentation.

https://docs.ansible.com/ansible/intro_configuration.html

10. setup - Gathers facts about remote hosts

It can be executed directly by /usr/bin/ansible to check what variables are available to a host. Ansible provides many *facts* about the system, automatically.

```
$ ansible -m setup web1
web1 | SUCCESS => {
    "ansible_facts": {
        "ansible_all_ipv4_addresses": [
            "10.0.2.15",
            "192.168.1.13"
    }
}
```

Visualpath Training & Consulting.

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: +91-970 445 5959, 961 824 5689 E-Mail ID : online.visualpath@gmail.com, Website : www.visualpath.in.

```

] ,
"ansible_all_ipv6_addresses": [
    "fe80::a00:27ff:fe15:d519",
    "fe80::a00:27ff:febff:5936"
],
"ansible_architecture": "x86_64",
"ansible_bios_date": "12/01/2006",
"ansible_bios_version": "VirtualBox",
"ansible_cmdline": {
output trimmed
"ohai_uptime": "1 hours 38 minutes 26 seconds",
"ohai_uptime_seconds": 5906,
"ohai_virtualization": {
    "role": "guest",
    "system": "vbox"
}
},
"changed": false

```

```

# Display facts from all hosts and store them indexed by I(hostname) at
C(/tmp/facts).

# ansible all -m setup --tree /tmp/facts

# Display only facts regarding memory found by ansible on all hosts and output
them.

# ansible all -m setup -a 'filter=ansible_*_mb'

# Display only facts returned by facter.

# ansible all -m setup -a 'filter=facter_*'

# Display only facts about certain interfaces.

# ansible all -m setup -a 'filter=ansible_eth[0-2]'

# Restrict additional gathered facts to network and virtual.

# ansible all -m setup -a 'gather_subset=network,virtual'

```

Visualpath Training & Consulting.

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: +91-970 445 5959, 961 824 5689 E-Mail ID : online.visualpath@gmail.com, Website : www.visualpath.in.