

# DevOps UNIT-5

- ⇒ Introduction to testing
- ⇒ Automation of testing pro's and Con's
- ⇒ Selenium - Introduction and features
- ⇒ Java Script Testing
- ⇒ Testing Backend Integration points
- ⇒ Test Driven Development & REPL Driven Development
- ⇒ Deployment Systems, Virtualization Stacks
- ⇒ Executing code on the client
- ⇒ The puppet master and puppet agents
- ⇒ Ansible, chef, Salt Stack

# Introduction to testing

①

⇒ If we are going to release our code frequently, we need to be sure its good in quality. That's why we need automation testing.

## Manual Testing:-

Even though automating test is more beneficial for DevOps, manual testing will always be important. Before we can automate tests, we need to perform manual test at least once, this ensures that everything is functioning correctly.

⇒ Acceptance testing is a type of testing where the final product is tested to ensure it meets the requirements and works as expected for the end users. And Acceptance testing is hard to automate and it is hard to understand for developers. So it is done by quality assurance (QA) team.

⇒ If manual testing is done easily, then automated testing also done easily. To make QA team happy, you should

- 1) manage test data, especially what's in databases, so tests give consistent results.

2) quickly deploy new code and keep software up-to-date after testing is successful.

⇒ We cannot copy large databases for testing. They might also have user data that needs protection. In these situations, you should remove any personal details in the data before using it for testing.

⇒ Each company is different, one solution will not work for all companies. A good rule to follow is KISS Rule:- "Keep it Simple, Stupid"; means make things as simple as possible.

## Automation of testing pros and Con's

when people talk about automation test, they get excited because it has many benefits like:

- ① Better Software quality
- ② more confidence that Software releases will work as expected
- ③ Less boring when compare to manual testing

⇒ But in real world Companies with Complex projects, they don't do automation testing.

⇒ Testing Software is essential to make sure it works well. manual testing is too slow for Continuous Delivery, So automation test is needed. But there are problems.

### ① Cheap Tests Have Lower Value :-

Unit tests are easy and cheap to write, but people think manual testing finds more bugs. So, they might skip Unit tests.  
↳ it is a small ~~automated~~ test that checks if a specific part of the code works correctly

### ② Difficult Test Environment :-

It's easy to test small parts of the code, but hard to create realistic test Setups for the whole system. This can be due to lack of hardware, licenses or people

### ③ Constant changes in Software :-

⇒ immediately whenever we change code, we need to perform testing which takes time. This makes automation test extra work.

## ④ Hard to write good test code:-

writing good automated test code can be difficult. However, when you succeed, you gain a deeper understanding of how automated testing works. This knowledge helps you improve the project and catch issues early.

## Selenium - Introduction & Features

⇒ Selenium is one of the most widely used Open Source website testing tool, developed by Jason Huggings in 2004

### why Selenium :-

#### ① Automated Testing :-

Selenium allows you to write scripts that can perform tasks on a website, such as clicking buttons, filling out forms or checking if certain text is visible. This saves time because you don't have to do these tasks manually.

#### ② Cross-Browser Testing :-

Selenium can work with different web browsers like Chrome, Firefox, Safari. This means you can test your web site on all popular browsers to ensure it works well.

### ③ Supports multiple Languages.

You can write Selenium test code in several programming languages like python, java, C#, JavaScript, so many Developers can use it.

### ④ Open Source:- Supports multiple Operating Systems (like windows, mac, Linux, android etc...)

many Companies and Developers use Selenium because it is free to download & use.

### How Selenium work:-

- ⇒ Selenium uses a Special driver for each browser. When you write a test script, Selenium sends commands to the browser through these drivers. It supports parallel testing, which reduces time.
- ⇒ For example, I want to check my website title is 'ns lectures' or not. So I will write script to check my title using any programming language like python or java or C#. When I execute my script, my Special driver will check my title is 'ns lectures' or not.
- ⇒ In DevOps, we can integrate Selenium with automation tools like Jenkins and Docker to achieve continuous testing.
- ⇒ Selenium is an essential tool to check whether website is working correctly or not. It automates repetitive tasks and save time. So developers can test their websites on different browsers.

# JavaScript Testing

(3)

Today many software applications or services that people use have web interfaces, which means they are used through web browsers like chrome or Firefox. Examples: Amazon, Flipkart and we will run these interfaces using JavaScript. It's important to have good tools to test the JavaScript code that runs these interfaces. Here are some important JavaScript testing tools.

## ① Karma:-

This is a tool that runs unit tests for JavaScript. Unit tests check small parts of your code to make sure they work correctly.

## ② Jasmine:-

This tool tests how software behaves in different situations. It's like checking if a car drives smoothly on different roads.

## ③ Protractor:-

Popular tool for creating and testing web applications. Specially designed for testing application build with AngularJS.

=> Although you can use general tools like Selenium to test web applications, Protractor is better for AngularJS apps. It understands how AngularJS works, making it easier to test these type of applications.

=> These tools help developers make sure their web applications work well and behave correctly, which is very important in today's software development.

## Testing Backend Integration points

- ⇒ Backend testing is like checking the parts of a machine that you can't see. When you use a website or an app, the front end is what you interact with like buttons, images, text. The backend is everything behind the scenes that makes the front end work (like databases, servers, applications)
- ⇒ Backend testing ensures that all these hidden parts are working correctly, storing data properly and communicating with each other without any errors. It involves checking the databases, servers, and APIs to make sure everything functions correctly.
- ⇒ SOAP (Simple Object Access Protocol) and REST (Representation State Transfer) are two different methods used to automate backend testing, especially for websites and API's.

SoapUI is a tool that helps you to write and run backend testing. Using Soap UI you can test step by step and it is easy to use.

Benefits of Soap UI:-

- ① Easy for testers:- you can easily write and run tests
- ② Good for Developers- Developers can add testing without using SoapUI's main screen, directly they can use command line or directly they can integrate testing in CI/CD process
- ③ Free Version! - SoapUI is free to download and use.

## How SoapUI works:

5

SoapUI interface displays a list of tests (called test cases) on one side of the screen. This makes it easy to select the tests you want to run. You can choose to run a single test or multiple tests simultaneously. The results of the test are displayed on the other side of the interface.

- ⇒ SoapUI saves scripts in XML format. This XML format makes it easy to organize, read and modify the tests. Similar to how code is managed in programming. Since the tests are stored as XML, you can easily update or change the test cases as needed.

## Test Driven Development

- ⇒ Test Driven Development (TDD) is a way of developing software where you write tests before you write the actual code. It became popular in the 1990's because of Extreme programming.

### How Test Driven Development (TDD) works:

- ① write the test first :- At first you need to understand all requirement and features of new software that you want to create. and start writing test that need to be performed on your code.
- ② check the Test fails:- initially i did not written code to develop my software, so when i perform testing, my test should fail. This is expected because there is no code to pass your test.

### ③ Write the code :

Next, Start writing code by selecting basic features of your software. In this stage, it doesn't have to be perfect, it just needs to work and pass the test.

### ④ Run All Tests :

After writing complete code, run all existing tests. If we pass all existing tests, then testing is successful.

### ⑤ Improve the code (This step is called Refactoring)

It means taking the code you just wrote and make it better.

Organize your code by removing unnecessary parts.

→ TDD helps you understand what the software should do before you start coding. It also creates tests that make sure everything works correctly, even when you make changes later. This is useful for continuous integration and development.

# REPL - Driven Development

(8)

- ⇒ REPL Stands for Read Eval Print loop. It is a Software, where you can write and test your code, using languages like Lisp, python, Ruby and Javascript.
- ⇒ In this Software, you can write code, immediately run and test it and see the results. This allows for quick testing and feedback as you develop.

## How REPL works:

### ① write Small function:

You create small, focused functions that do one specific task. These functions should be independent, meaning they don't depend heavily on other parts of the code.

### ② Immediate Testing:

As soon as you write a function, you can run it in REPL. This will check if it works correctly. If there's a problem, you can fix it.

## Different from Test Driven Development:

In Test - Driven Development (TDD), you write tests before writing the actual code. Whereas in REPL, you can write a test code simultaneously. This can be more easy for some developers.

## Combining with Unit Testing!

while Using REPL, you can also Create Unit tests for your functions. Unit tests check if individual piece of code work correctly.

### Benefits of REPL:

- ① Quick feedback: you get instant feedback on your code, which helps you identify and fix problems faster.
- ② Simplicity: writing small functions makes your code more straightforward and manageable.
- ③ Enhanced Learning:

For beginners, using a REPL can be a great way to learn programming concepts, as you can experiment without a complex setup.

→ REPL driven development is an effective approach, especially for those new to programming. it allows writing clear, independent functions and allows for immediate testing making it a useful strategy in the Software development process.

# Deployment Systems

(6)

⇒ A Deployment System in DevOps is a tool or set of tools that automatically move new software from the developer's computer to the production server. Once the software is placed on the production server, all users can access it.

why are there so many Deployment Systems :-

There are many deployment systems because deploying software can be complex.

Imagine we have an application with three main parts

web Server :- manages website requests

Application Server :- Runs the main program

Database Server :- Stores data

if we only had one server handling all these parts and if we went to update this software once a year we could install this software manually. but

⇒ manual work is boring

⇒ manual work can lead to mistakes.

In real life :-  
Big companies have many computers and servers and applications and each server and application might need different methods for updating.

we can run our application on :-

→ A physical Server

→ A Virtual machine

→ A Container (a light weight, portable way to run apps)

If you want to running an application on physical Server, you might need one type of deployment System. If you want to run an application on Virtual machines, you might need other type of deployment System.

### Main Duties of a Deployment System

- ① Automatically place the application on the Server
- ② Apply updates or new versions of the application
- ③ Set up the necessary configuration to run the application properly

## Virtualization Stacks

### Virtualization:-

- ⇒ Virtualization makes you create Virtual Computers (called virtual machines) inside your real computer. These virtual machines acts like real computers with their own virtual hardware (like CPUs, and network cards).
- ⇒ You can use Virtualization to test different types of hardware on your computer. For example, you can test an android phone on your computer using tool like QEMU to test mobile apps.  
↳ tool to create and run virtual machines.

### Use of Virtualization

- ⇒ Imagine you have a big computer that can run many programs. Virtualization will create separate "mini-computers" inside your big computer. If one "mini-computer" has problem or uses too much power, it won't affect the other "mini-computers".
- ⇒ For example, your big computer is powerful with 64 cores (like having 64 small CPUs). You can create Virtual machine that uses only 2 cores out of 64 cores. If this Virtual machine has issues, only those 2 cores are affected, not the entire big computer.

### Hypervisors : The key to Virtualization

A hypervisor is a special program that creates and manages Virtual machines (VMs). It shares the computer resources like CPU and memory among these VMs.

There are two main types of hypervisors

① Bare Metal Hypervisor:-

⇒ It runs directly on the Computer hardware. It Does not need an Operating System to run. Example :- VMware ESX.

② Hosted Hypervisor:-

⇒ Runs on top of an existing Operating System. Needs an Operating System to function. Example:- KVM on Linux.

Popular Virtualization Tools:-

① VMware:- A well-known tool that comes in desktop and Server versions. VMware ESX is used directly on hardware.

② KVM:- (Kernel-based Virtual machine) A free tool for Linux. Popular because it doesn't have extra costs.

③ Xen:- A tool that can run modified Operating Systems for better performance.

④ Virtual Box:- A free tool from Oracle, good for developers to run different Operating Systems on their computers.

Automating with APIs

Virtualization tools come with APIs that help manage VMs automatically. One popular API is libvirt which works with various hypervisors like KVM, Xen, LXC.

## Executing Code on the client

⇒ By using some tools you can run commands on servers and manage them.

for example, if you want to check files on all your web servers, you can use these tools to do it.

Puppet is a tool used in IT to manage and automate setups on servers.

MCollective is a tool that works with puppet to manage and execute commands on multiple servers at once.

By running commands you can collect information for all servers.

⇒ these tools make it easy to run commands on many servers at the same time, which can be useful for performing various operations.

### The Puppet master and puppet agents

Puppet is a popular tool used in large organizations to handle setup and updates on servers.

(configure)

In puppet software, there are two main components:

① puppet master:-  
It is a central server that sends server updates to the puppet Agents.

## Puppet Agents

they are individual servers, check in regularly with puppet master to see if anything need to be updated in their servers.

⇒ Since servers run many applications, it is essential to keep servers up-to-date. This can be achieved using puppet.

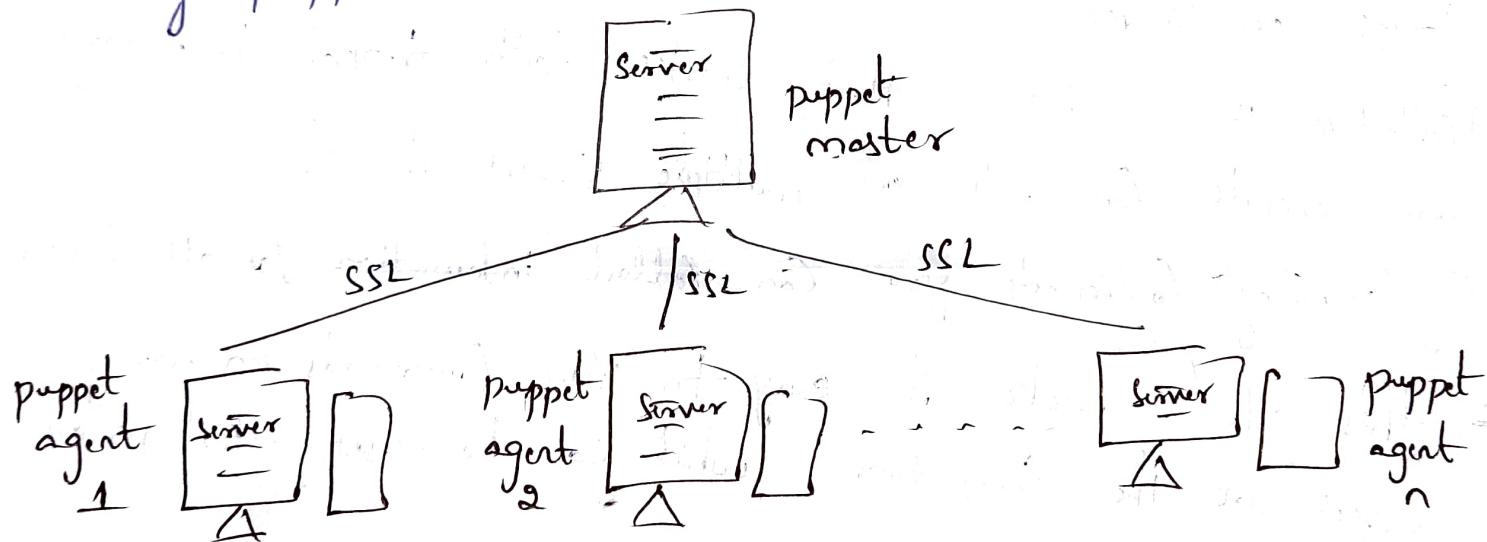


fig. puppet Architecture

⇒ master and client communicate through SSL (Secure Socket layer)

# Ansible

⇒ Ansible is a tool that helps manage and deploy Software to multiple Servers in a simple way. Unlike some other tools, Ansible doesn't need Special Software running on each Server to manage Servers. Instead, Ansible connects to the Servers using SSH and runs Commands to set them. You only need to install ansible on the main Control machine.

## Keypoints about Ansible:

### ① No Extra Software Needed :-

When compared to other tools, Ansible doesn't need Special Software on the Servers to manage it.

### ② Uses SSH :-

Ansible uses SSH to Connect and Run Commands on the Servers.

### ③ Needs python :-

The Servers being managed by Ansible need to have python installed but any version of python usually works fine.

⇒ Ansible performance Speed is less than other tools.

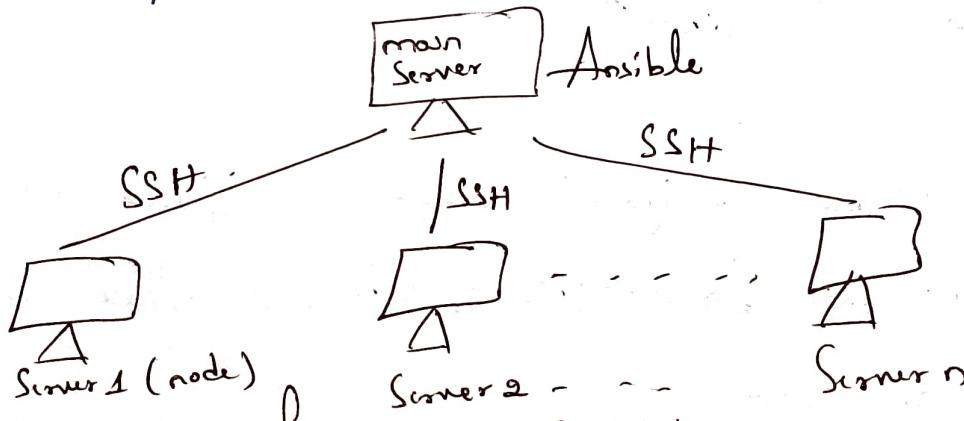


Fig Ansible Architecture

## How Ansible works:

⇒ Ansible uses Special files called "playbooks" to define what you want to do on the Servers. These playbooks are written in a Straight-forward language called YAML.

For example, I want to check Servers are working or not. So i will Create playbook to perform this Operation and Use Some playbook in all Servers.

## Chef

⇒ chef is a tool used to manage and deploy Software on multiple Servers. It uses Ruby programming language and helps automate the Setup of Servers.

## How chef works:

### ① Recipes and Cookbooks:

chef uses files called "Recipes" to define what should be installed and Configured on your Servers. A Collection of recipes is called a "Cookbook".

### ② chef Servers:

All the Cookbooks and Configuration data are stored on a Central Server Called the chef Server. This is the main place where everything is organized.

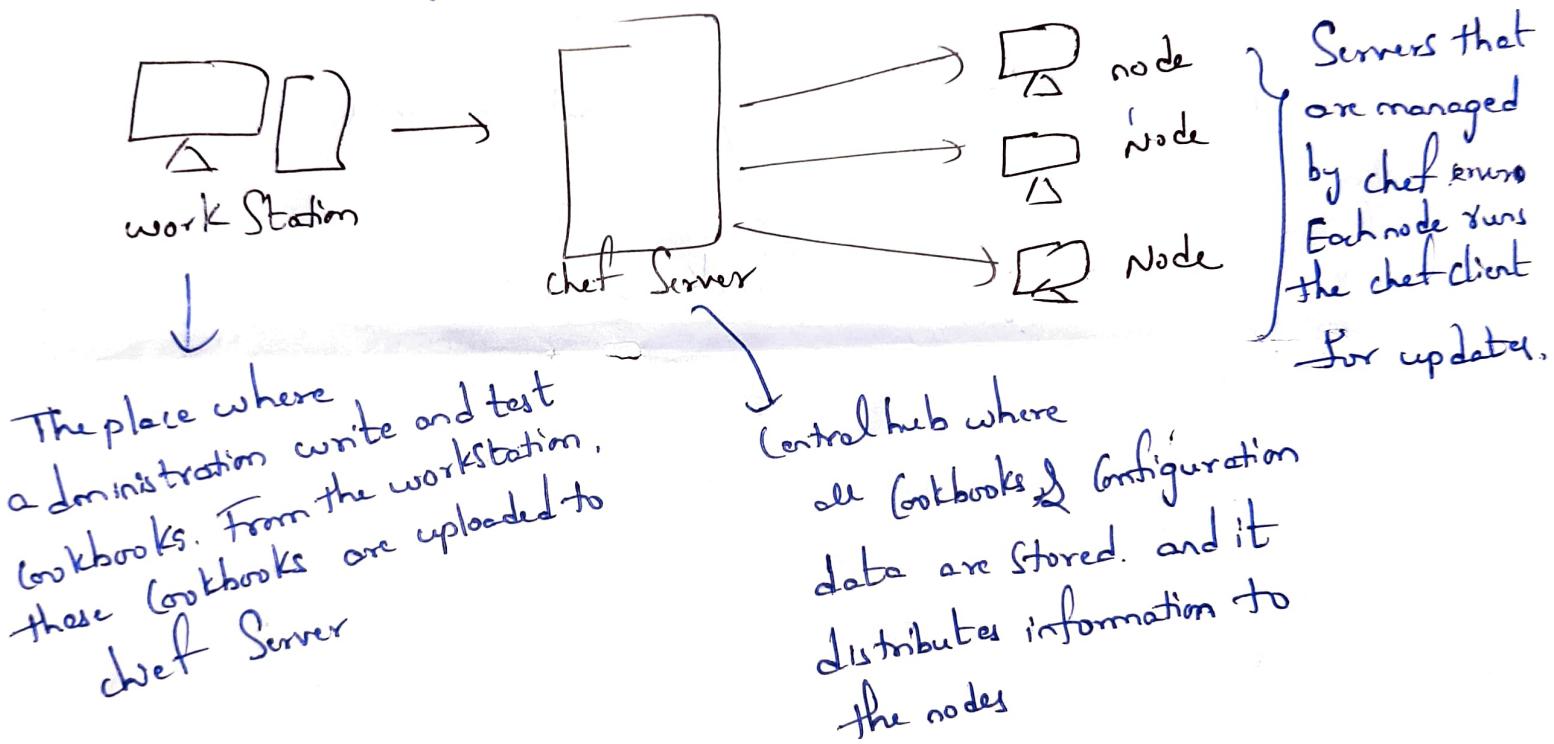
### ③ Nodes:

these are the servers you want to manage

### ④ Chef Client:

This program runs on each server (node) and checks which node need to be configured.

This process automates server setup, ensuring all servers are consistently configured without manual intervention.



## Salt Stack

Salt Stack is a tool that helps manage and control multiple servers. It works by using a central server called "Salt master", which sends instructions to the servers to manage and configure ~~salt~~ called "Salt minions".

These instructions are defined in files called "State files".

⇒ Salt Stack automates the process of Setting up, Configuring, managing Servers, making it easier and faster to handle many Servers at once.

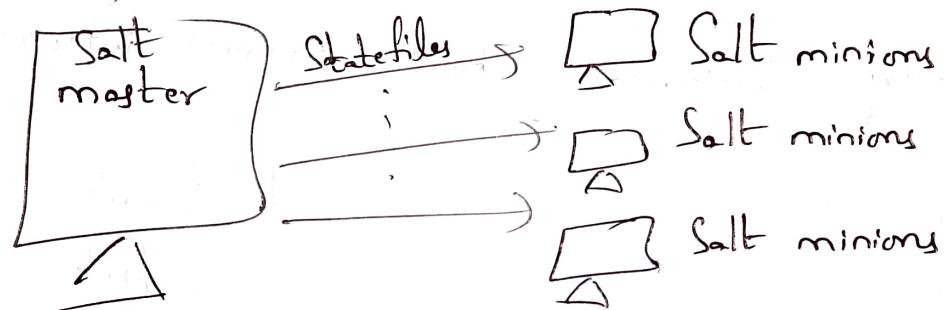


Fig: Salt Stack Architecture