Name: Deep Chavan
Batch: T11
Roll No.: 15
Subject: Security Lab

# Assignment 1

**Aim:** Shift cipher and mono alphabet substitution cipher.

**Theory:**

Monoalphabetic Cipher is a part of the substitution technique in which a single cipher alphabet is used per message (mapping is done from plain alphabet to cipher alphabet). Monoalphabetic cipher converts plain text into cipher text and re-convert a cipher text to plain text.

Monoalphabetic Cipher eliminates the brute-force techniques for cryptanalysis. Moreover, the cipher line can be a permutation of the 26 alphabetic characters.

Implementation:

# Breaking the Mono-alphabetic Substitution Cipher

PART I

Decrypt the following cipher text. A tool to simulate the Mono-Alphabetic Subsitution cipher is provided beneath for your assistance.

Here is the table of frequencies of English alphabets for your reference:

| a | b | c | d | e | f | g | h | i | j | k | l | m |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 8.167 | 1.49 | 2.782 | 4.253 | 12.702 | 2.228 | 2.015 | 6.094 | 6.966 | 0.153 | 0.772 | 4.025 | 2.406 |
| **n** | **o** | **p** | **q** | **r** | **s** | **t** | **u** | **v** | **w** | **x** | **y** | **z** |
| 6.749 | 7.507 | 1.929 | 0.095 | 5.987 | 6.327 | 9.056 | 2.758 | 0.978 | 2.360 | 0.150 | 1.974 | 0.074 |

```
qeehn el xuu nwmrn. nkr lwtqn x nfxuu orb ve x qeeh vee nfxuu leh krh
ve lwv, civ vkheipk gkwdk nkr nrrn xt xvvhxdvwsr pxhqrt. nkr vkrt
qwndesrhn x cevvur uxcruurq 'qhwto fr', vkr detvrtvn el gkwdk dxinr
krh ve nkhwto vee nfxuu ve hrxdk vkr orb. x dxor gwvk 'rxv fr' et wv
dxinrn krh ve pheg ve nidk x vhrfrtqein nwmr krh krxq kwvn vkr
drwuwtp.
```

Next Ciphertext

Calculate Frequencies in ciphertext

Ciphertext Frequencies:

| a | b | c | d | e | f | g | h | i | j | k | l | m |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.000 | 1.037 | 2.282 | 3.942 | 8.091 | 1.452 | 3.112 | 5.602 | 2.075 | 0.000 | 8.506 | 1.452 | 0.415 |
| **n** | **o** | **p** | **q** | **r** | **s** | **t** | **u** | **v** | **w** | **x** | **y** | **z** |
| 7.469 | 1.867 | 1.452 | 3.32 | 11.618 | 0.622 | 4.979 | 5.602 | 9.959 | 6.639 | 7.884 | 0.622 | 0.000 |

PART II

Note that the *cipher text is in lower case* and when you replace any character, the final character of replacement, i.e., *plaintext is changed to upper case* automatically in the following scratchpad.

Scratchpad:

```
CHAPTER 1 - DOWN THE RABBIT HOLE: ALICE IS BORED SITTING ON THE
RIVERBANK WITH HER SISTER, WHEN SHE NOTICES A TALKING, CLOTHED WHITE
RABBIT WITH A POCKET WATCH RUN PAST. SHE FOLLOWS IT DOWN A RABBIT HOLE
WHEN SUDDENLY SHE FALLS A LONG WAY TO A CURIOUS HALL WITH MANY LOCKED
DOORS OF ALL SIZES. SHE FINDS A SMALL KEY TO A DOOR TOO SMALL FOR HER TO
FIT, BUT THROUGH WHICH SHE SEES AN ATTRACTIVE GARDEN. SHE THEN DISCOVERS
A BOTTLE LABELLED 'DRINK ME', THE CONTENTS OF WHICH CAUSE HER TO SHRINK
TOO SMALL TO REACH THE KEY. A CAKE WITH 'EAT ME' ON IT CAUSES HER TO
GROW TO SUCH A TREMENDOUS SIZE HER HEAD HITS THE CEILING.
```

Modify the text above (in scratchpad):

This is case *insensitive* function and replaces only cipher text (lower case) by plain text (upper case).

Replace cipher character m by plaintext character z Modify

Use the following function to undo any unwanted exchange by giving an uppercase character and a lower case. This is a case sensitive function.

Replace character n by character ns Replace these exact characters

Your replacement history:

You replaced d by C You replaced k by H You replaced x by A You replaced y by P You replaced v by T You replaced r by E You replaced h by R You replaced c by B You replaced w by I You replaced g by W You replaced e by O You replaced u by L You replaced q by D You replaced t by N You replaced ns by N You replaced n by ns You replaced n by S You replaced d by C You replaced k by H You replaced x by A You replaced y by P You replaced v by T You replaced r by E You replaced h by R You replaced q by D You replaced e by O You replaced g by W You replaced t by N You replaced c by B You replaced w by I You replaced u by L You replaced n by S You replaced p by G You replaced s by V You replaced o by K You replaced l by F You replaced i by U You replaced b by Y You replaced f by M You replaced m by Z

## PART III

Enter your solution plaintext here:

```
TO FIT, BUT THROUGH WHICH SHE SEES AN ATTRACTIVE GARDEN. SHE THEN
DISCOVERS A BOTTLE LABELLED 'DRINK ME', THE CONTENTS OF WHICH CAUSE
HER TO SHRINK TOO SMALL TO REACH THE KEY. A CAKE WITH 'EAT ME' ON IT
CAUSES HER TO GROW TO SUCH A TREMENDOUS SIZE HER HEAD HITS THE
CEILING.
```

Solution Key = xcdqrlpkwzoufteyahnvisgjbm

Check Answer!

CORRECT!!

Steps for Installation of Git

Download Git from https://git-scm.com/

Install latest version

Git 2.32.0.2 Setup
Select Components
Which components should be installed?
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.
Additional icons
On the Desktop
Windows Explorer integration
Git Bash Here
Git GUI Here
Git LFS (Large File Support)
Associate .git* configuration files with the default text editor
Associate .sh files to be run with Bash
Check daily for Git for Windows updates
Current selection requires at least 261.1 MB of disk space.
https://gitforwindows.org/
Back
Next
Cancel

Git 2.32.0.2 Setup
Select Start Menu Folder
Where should Setup place the program's shortcuts?
Setup will create the program's shortcuts in the following Start Menu folder.
To continue, click Next. If you would like to select a different folder, click Browse.
Browse...
Don't create a Start Menu folder
https://gitforwindows.org/
Back
Next
Cancel

Git 2.32.0.2 Setup
Choosing the default editor used by Git
Which editor would you like Git to use?
Use the Nano editor by default
GNU nano is a small and friendly text editor running in the console window.
This is the recommended option for end users if no GUI editors are installed.
https://gitforwindows.org/
Back
Next
Cancel

Git 2.32.0.2 Setup
Adjusting the name of the initial branch in new repositories
What would you like Git to name the initial branch after "git init"?
Let Git decide
Let Git use its default branch name (currently: "master") for the initial branch in newly created repositories. The Git project intends to change this default to a more inclusive name in the near future.
Override the default branch name for new repositories
NEW! Many teams already renamed their default branches; common choices are "main", "trunk" and "development". Specify the name "git init" should use for the initial branch:
main
This setting does not affect existing repositories.
https://gitforwindows.org/
Back
Next
Cancel

Git 2.32.0.2 Setup
Adjusting your PATH environment
How would you like to use Git from the command line?
Use Git from Git Bash only
This is the most cautious choice as your PATH will not be modified at all. You will only be able to use the Git command line tools from Git Bash.
Git from the command line and also from 3rd-party software
(Recommended) This option adds only some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools.
You will be able to use Git from Git Bash, the Command Prompt and the Windows PowerShell as well as any third-party software looking for Git in PATH.
Use Git and optional Unix tools from the Command Prompt
Both Git and the optional Unix tools will be added to your PATH.
Warning: This will override Windows tools like "find" and "sort". Only use this option if you understand the implications.
https://gitforwindows.org/
Back
Next
Cancel

Git 2.32.0.2 Setup
Choosing HTTPS transport backend
Which SSL/TLS library would you like Git to use for HTTPS connections?
Use the OpenSSL library
Server certificates will be validated using the ca-bundle.crt file.
Use the native Windows Secure Channel library
Server certificates will be validated using Windows Certificate Stores.
This option also allows you to use your company's internal Root CA certificates distributed e.g. via Active Directory Domain Services.
https://gitforwindows.org/
Back
Next
Cancel

Git 2.32.0.2 Setup
Configuring the line ending conversions
How should Git treat line endings in text files?
Checkout Windows-style, commit Unix-style line endings
Git will convert LF to CRLF when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Windows ("core.autocrlf" is set to "true").
Checkout as-is, commit Unix-style line endings
Git will not perform any conversion when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Unix ("core.autocrlf" is set to "input").
Checkout as-is, commit as-is
Git will not perform any conversions when checking out or committing text files. Choosing this option is not recommended for cross-platform projects ("core.autocrlf" is set to "false").
https://gitforwindows.org/
Back
Next
Cancel

Git 2.32.0.2 Setup
Configuring the terminal emulator to use with Git Bash
Which terminal emulator do you want to use with your Git Bash?
Use MinTTY (the default terminal of MSYS2)
Git Bash will use MinTTY as terminal emulator, which sports a resizable window, non-rectangular selections and a Unicode font. Windows console programs (such as interactive Python) must be launched via `winpty` to work in MinTTY.
Use Windows' default console window
Git will use the default console window of Windows ("cmd.exe"), which works well with Win32 console programs such as interactive Python or node.js, but has a very limited default scroll-back, needs to be configured to use a Unicode font in order to display non-ASCII characters correctly, and prior to Windows 10 its window was not freely resizable and it only allowed rectangular text selections.
https://gitforwindows.org/
Back
Next
Cancel

Git 2.32.0.2 Setup
Choose the default behavior of `git pull`
What should `git pull` do by default?
Default (fast-forward or merge)
This is the standard behavior of `git pull`: fast-forward the current branch to the fetched branch when possible, otherwise create a merge commit.
Rebase
Rebase the current branch onto the fetched branch. If there are no local commits to rebase, this is equivalent to a fast-forward.
Only ever fast-forward
Fast-forward to the fetched branch. Fail if that is not possible.
https://gitforwindows.org/
Back
Next
Cancel

Git 2.32.0.2 Setup
Choose a credential helper
Which credential helper should be configured?
Git Credential Manager Core
(NEW!) Use the new, cross-platform version of the Git Credential Manager.
See more information about the future of Git Credential Manager here.
Git Credential Manager
(DEPRECATED) The Git Credential Manager for Windows handles credentials e.g. for Azure DevOps and GitHub (requires .NET framework v4.5.1 or later).
None
Do not use a credential helper.
https://gitforwindows.org/
Back
Next
Cancel

Git 2.32.0.2 Setup
Configuring extra options
Which features would you like to enable?
Enable file system caching
File system data will be read in bulk and cached in memory for certain operations ("core.fscache" is set to "true"). This provides a significant performance boost.
Enable symbolic links
Enable symbolic links (requires the SeCreateSymbolicLink permission). Please note that existing repositories are unaffected by this setting.
https://gitforwindows.org/
Back
Next
Cancel

Git 2.32.0.2 Setup
Configuring experimental options
These features are developed actively. Would you like to try them?
Enable experimental support for pseudo consoles.
(NEW!) This allows running native console programs like Node or Python in a Git Bash window without using winpty, but it still has known bugs.
Enable experimental built-in file system monitor
(NEW!) Automatically run a built-in file system watcher, to speed up common operations such as `git status`, `git add`, `git commit`, etc in worktrees containing many files.
https://gitforwindows.org/
Back
Install
Cancel
A4 BDA LAB

Git 2.32.0.2 Setup
Installing
Please wait while Setup installs Git on your computer.
Extracting files...
C:\Program Files\Git\mingw64\lib\tcl8.6\encoding\cp874.enc
https://gitforwindows.org/
Cancel

Git 2.32.0.2 Setup
Completing the Git Setup Wizard
Setup has finished installing Git on your computer. The application may be launched by selecting the installed shortcuts.
Click Finish to exit Setup.
Launch Git Bash
View Release Notes
Finish

Govt...
pay
RPS
Report_fin...
kvs_maths
kvsmaths1
QA_RPS_Ve...
Atom
SAG
FDP_STT...
QA_RPS_Ve...
Microsoft
Teams
Git Bash
kvs SPOKEN
HINDI 030...
ALMAS
Electricals
DevOps STTP
Saee Gondal
Fees Receipt

AICTESigner
SAG
CNSS_2020...
PanCard_A...
Gondal
State-Govt...
7th pay
RPS
Report_fin...
kvs_maths
kvsmaths1
Google
Chrome
McAfee
Security Sc...
R@1n
NTA-EC-P-...
AICTE_202...
Personal
Inspection-...
DBMS_viva...
Gazette_Se...
Survey
papers
Wild Animals
by Saee Go...
TSEC TE
Soft_Engg
MINGW64:/c/Users/avinash
avinash@DESKTOP-MM79UFH MINGW64 ~
$
Search the web and Windows

**Deep Chavan**
**T11-15**

# LAB ASSIGNMENT 3

AIM: To perform various Git Operations.

LAB OUTCOME:

LO1, LO2 Mapped.

THEORY:

Git is a distributed version control system that enables collaboration among developers by tracking changes in code. It allows multiple people to work on the same project simultaneously, maintaining different versions of the code through commits and branches. This system facilitates efficient code merging, conflict resolution, and the ability to work on new features independently. Git also provides a remote repository for sharing and backup, making it an essential tool for managing and tracking code changes in software development projects.

Git commands are instructions that you give to the Git version control system to perform various tasks and operations on your source code repository. These commands allow you to interact with your code, track changes, collaborate with others, and manage your project's history effectively. Each Git command corresponds to a specific action, such as initialising a repository, staging changes, committing code, creating branches, merging changes, fetching remote updates, and more. Some of these are given below:

1. git init: Initialises a new Git repository. Example: git init
2. git clone: Copies a remote repository to your local machine. Example: git clone https://github.com/username/repo.git
3. git add: Stages changes for commit. Example: git add file.txt
4. git commit: Records staged changes with a message. Example: git commit -m "Added newfeature"
5. git status: Shows the status of your working directory. Example: git status
6. git log: Displays commit history. Example: git log
7. git branch: Lists, creates, or deletes branches. Example: git branch new-feature 8. git checkout: Switches to a different branch. Example: git checkout new-feature
9. git merge: Combines changes from different branches.
10. Example: git merge feature-branch
11. git pull: Fetches remote changes and merges them into the current branch. Example: git pull origin main
12. git push: Uploads local changes to a remote repository. Example: git push origin main
13. git remote: Manages remote repositories.
14. Example: git remote add origin https://github.com/username/repo.git

15. git fetch: Retrieves remote changes but doesn't merge them. Example: git fetch origin
16. git diff: Shows differences between working directory and last commit. Example: git diff
17. git reset: Unstaged files or discards changes. Example: git reset file.txt
18. COMMANDS & OUTPUT:

Lab1002@MU2024 MINGW64 ~

$ git config usage: git config

[<options>]

Config file location

--global use global config file

--system use system config file

--local use repository config file

--worktree use per-worktree config file

-f, --file <file> use given config file

--blob <blob-id> read config from given blob object

Action

--get get value: name [value-pattern]

--get-all get all values: key [value-pattern]

--get-regexp get values for regexp: name-regex [value-pattern]

--get-urlmatch get value specific for the URL: section[.var] URL

--replace-all replace all matching variables: name value [value-pattern]

--add add a new variable: name value

--unset remove a variable: name [value-pattern]

--unset-all remove all matches: name [value-pattern]

--rename-section rename section: old-name new-name

--remove-section remove a section: name

-l, --list list all

--fixed-value use string equality when comparing values to 'value-pattern'

-e, --edit open an editor

--get-color find the color configured: slot [default]

--get-colorbool find the color setting: slot [stdout-is-tty]

Type

-t, --type <type> value is given this type

--bool value is "true" or "false"

--int value is decimal number

--bool-or-int value is --bool or --int

--bool-or-str value is --bool or string

--path value is a path (file or directory name)

--expiry-date value is an expiry date

Other

-z, --null terminate values with NUL byte

--name-only show variable names only

--includes respect include directives on lookup

--show-origin show origin of config (file, standard input, blob, command line)

--show-scope show scope of config (worktree, local, global, system, command)

--default <value>  with --get, use default value when missing entry

Lab1002@MU2024 MINGW64 ~

$ git config --global user.email kamath.s.shreya@gmail.com

Lab1002@MU2024 MINGW64 ~

$ git status fatal: not a git repository (or any of the parent

directories):

.git

Lab1002@MU2024 MINGW64 ~

$ git config --global user.name shreya

Lab1002@MU2024 MINGW64 ~

$ git init

Initialized empty Git repository in C:/Users/Lab1002/.git/

Lab1002@MU2024 MINGW64 ~ (master)

$ git status

On branch master No commits yet

Lab1002@MU2024 MINGW64 ~ (master)

$ git --version git version

2.41.0.windows.3

Lab1002@MU2024 MINGW64 ~ (master)

$ git remote add origin "https://github.com/ShreyaKamath09/DevOps.git"

Lab1002@MU2024 MINGW64 ~ (master)

$ git pull origin main remote: Enumerating objects: 9, done. remote: Counting

objects: 100% (9/9), done. remote: Compressing objects: 100% (4/4), done.

remote: Total 9 (delta 0), reused 0 (delta 0), pack-reused 0 Unpacking objects: 100% (9/9), 1.83 KiB | 5.00 KiB/s, done. From https://github.com/ShreyaKamath09/DevOps

* branch main -> FETCH_HEAD * [new

branch] main -> origin/main

CONCLUSION:

In conclusion, I've effectively used different Git commands, demonstrating their crucial role in version control and teamwork. This hands-on experience highlights Git's essential role in making coding processes more organised and collaborative.

**Deep Chavan**

**T11-15**

**LAB ASSIGNMENT 4**

**AIM: To understand Continuous integration and Jenkins installation.**

**LAB OUTCOME:**

**LO1, LO3 Mapped.**

**THEORY:**

**Continuous Integration (CI) is a software development practice that involves automatically integrating code changes from multiple contributors into a shared repository. The goal is to detect and address integration issues early, ensuring that new code doesn't break existing functionality. Jenkins is a popular open-source automation server used for implementing CI/CD (Continuous Integration/Continuous Deployment) pipelines. Jenkins is an opensource automation server that facilitates Continuous Integration (CI) and Continuous Delivery (CD) in software development. It automates the process of building, testing, and deploying code changes, helping development teams streamline their workflows and ensure software quality. Jenkins provides a platform for creating and managing pipelines, which are a series of steps that outline how software changes should be built, tested, and delivered. These pipelines are often defined using a domain-specific language (DSL) or through a graphical interface.**

**Steps for Jenkins Installation on Windows:**

1. **Install Java: Make sure Java is installed on the system.**
2. **Download Jenkins: Visit the official Jenkins website (https://www.jenkins.io/) and download**

**the Windows installer.**

3. **Run Installer: Execute the downloaded Jenkins installer.**
4. **Follow Setup: Follow the on-screen instructions to complete the installation.**
5. **Start Jenkins: If it doesn't start automatically, initiate Jenkins as a service.**
6. **Access Web Interface: Open a web browser and go to http://localhost:8080.**
7. **Initial Admin Password: Check the Jenkins logs for the initial admin password.**
8. **Complete Setup: Follow prompts to install plugins and set up an admin user.**
9. **Install Plugins: If needed, install additional plugins for integration.**
10. **Create Pipelines: Configure CI/CD pipelines to automate tasks.**

11. **Remember, Jenkins on Windows operates on port 8080 by default. Make sure Java is properly configured and accessible from the command line.**

**SCREENSHOTS:**

Jenkins 2.417 Setup
Welcome to the Jenkins 2.417 Setup Wizard
The Setup Wizard will install Jenkins 2.417 on your computer.
Click Next to continue or Cancel to exit the Setup Wizard.
Back
Next
Cancel

Jenkins 2.417 Setup
Destination Folder
Jenkins
Click Next to install to the default folder or click Change to choose another.
Install Jenkins 2.417 to:
C:\Program Files\Jenkins\
Change...
Back
Next
Cancel

Jenkins 2.417 Setup
Service Logon Credentials
Enter service credentials for the service.
Jenkins
Jenkins 2.417 installs and runs as an independent Windows service. To operate in this manner, you must supply the user account credentials for Jenkins 2.417 to run successfully.
Logon Type:
Run service as LocalSystem (not recommended)
Run service as local or domain user:
Account:
admin
Password:
Test Credentials
Back
Next
Cancel

Jenkins 2.417 Setup
Port Selection
Choose a port for the service.
Jenkins
Please choose a port.
Port Number (1-65535):
8080
Test Port
It is recommended that you accept the selected default port.
Back
Next
Cancel

Jenkins 2.417 Setup
Jenkins 2.417 Setup
Select Java home directory (JDK or JRE)
Jenkins
Please select the path of a Java Development Kit or Java Runtime Environment.
Only Java 11, 17 and 21 are supported by Jenkins.
C:\Program Files\Java\
Change...
Back
Next
Cancel

Jenkins 2.417 Setup
Custom Setup
Select the way you want features to be installed.
Jenkins
Click the icons in the tree below to change the way features will be installed.
Jenkins
Start Service
Firewall Exception
The required Jenkins components
This feature requires 86MB on your hard drive. It has 1 of 2 subfeatures selected. The subfeatures require 0KB on your hard drive.
Browse...
Reset
Disk Usage
Back
Next
Cancel

Jenkins 2.417 Setup
Jenkins
Ready to install Jenkins 2.417
Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.
Back
Install
Cancel

Jenkins 2.417 Setup
Jenkins
Installing Jenkins 2.417
Please wait while the Setup Wizard installs Jenkins 2.417.
Status:
Back
Next
Cancel

Jenkins 2.417 Setup
Completed the Jenkins 2.417 Setup Wizard
Click the Finish button to exit the Setup Wizard.
Back
Finish
Cancel

Getting Started
Unlock Jenkins
To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:
C:\ProgramData\Jenkins\.jenkins\secrets\initialAdminPassword
Please copy the password from either location and paste it below.
Administrator password
Continue

**CONCLUSION:**

**Through this assignment, I have learnt the concept of Continuous Integration and successfully installed Jenkins on a Windows System.**

**Deep Chavan**

**T11-15**

# LAB ASSIGNMENT 5

**AIM:** To build a Java program using Jenkins.

**LAB OUTCOME:**
LO1, LO3 Mapped.

**THEORY:**
Building programs using Jenkins involves automating the development workflow, particularly for Java applications. Through Jenkins, developers create a structured process that automatically manages tasks such as code compilation, testing, and potential deployment.

When a Jenkins job is established, it constructs a framework that guides the progression of software development activities. Jenkins can be configured to monitor version control repositories, subsequently triggering a build each time code changes are committed. This preemptive identification of errors during automated builds aids in identifying and addressing issues before they propagate further within the development cycle.

**Steps to build Java Programs using Jenkins:**

1. Install Jenkins: Install Jenkins on the system.
2. Create Job: Make a new job in Jenkins.
3. Setup Code: Configure version control (e.g., Git) for your job.
4. Build Steps: Add steps to compile your Java program, e.g., use "Execute shell" with javac.
5. Post-Build: Optionally, add post-build actions like archiving artefacts.
6. Save and Build: Save the job and trigger a build.
7. Check Results: Review build results, including console output and test reports.
8. Test Steps: If applicable, add test steps using tools like JUnit.
9. Automate: Set up auto-build triggers for continuous integration.

localhost:8080/newJob
Gmail
YouTube
Maps
Dashboard
Enter an item name
shreya
» Required field
Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
Folder
OK
Type here to search
ENG IN
10:26 PM
8/20/2023

Jenkins
Search (CTRL+K)
admin
log out
Dashboard
shreya
Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
General
Enabled
Description
DevOps Lab
Simple Java Program on Jenkins
Date: 21st August, 2023
Plain text
Preview
Discard old builds
This project is parameterized
Execute concurrent builds if necessary
Advanced
Save
Apply
Type here to search
ENG IN
10:27 PM
8/20/2023

Dashboard > shreya > Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
Execute Windows batch command
Command
See the list of available environment variables
javac Helloworld.java
java Helloworld
Advanced
Add build step
Post-build Actions
Add post-build action
Save
Apply
Type here to search
ENG IN
10:27 PM
8/20/2023

Helloworld.java - Notepad
File Edit Format View Help
public class Helloworld
{
public static void main(String[] args)
{
System.out.println("Hello World!");
}
}
Ln 7, Col 2
100%
Windows (CRLF)
UTF-8

Jenkins
Search (CTRL+K)
admin
log out
Dashboard > shreya >
Status
Changes
Workspace
Build Now
Configure
Delete Project
Rename
Build History
trend
No builds
shreya
DevOps Lab
Simple Java Program on Jenkins
Date: 21st August, 2023
Edit description
Disable Project
Permalinks
Type here to search
ENG IN
10:31 PM
8/20/2023

Jenkins

Search (CTRL+K) admin log out

Dashboard > shreya > #2

- Status
- Changes
- Console Output
- Edit Build Information
- Delete build '#2'
- Previous Build

## #2 (Aug 20, 2023, 10:33:34 PM)

Keep this build forever

Add description

Started 6.4 sec ago
Took 1.2 sec

No changes.

Started by user admin

REST API Jenkins 2.417

Type here to search ENG IN 10:33 PM 8/20/2023

Jenkins

Search (CTRL+K) admin log out

Dashboard > shreya > #2 > Console Output

- Status
- Changes
- Console Output
- View as plain text
- Edit Build Information
- Delete build '#2'
- Previous Build

## Console Output

```
Started by user admin
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\shreya
[shreya] $ cmd /c call C:\WINDOWS\TEMP\jenkins7542321336782794327.bat

C:\ProgramData\Jenkins\.jenkins\workspace\shreya>javac Helloworld.java

C:\ProgramData\Jenkins\.jenkins\workspace\shreya>java Helloworld
Hello World!

C:\ProgramData\Jenkins\.jenkins\workspace\shreya>exit 0
Finished: SUCCESS
```

REST API Jenkins 2.417

Type here to search ENG IN 10:33 PM 8/20/2023

Jenkins

Search (CTRL+K) admin log out

Dashboard > shreya >

- Status
- Changes
- Workspace
- Build Now
- Configure
- Delete Project
- Rename

Build History trend

Filter builds... /

#2
Aug 20, 2023, 10:33 PM

## shreya

DevOps Lab
Simple Java Program on Jenkins
Date: 21st August, 2023

Edit description

Disable Project

### Permalinks

- Last build (#2), 44 sec ago
- Last stable build (#2), 44 sec ago
- Last successful build (#2), 44 sec ago
- Last failed build (#1), 2 min 37 sec ago
- Last unsuccessful build (#1), 2 min 37 sec ago
- Last completed build (#2), 44 sec ago

Type here to search ENG IN 10:34 PM 8/20/2023

Dashboard > shreya > Configuration

## Configure

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

**Execute Windows batch command**

Command

See the list of available environment variables

```
javac Helloworld.java
java Helloworld
javac Forloop.java
java Forloop
```

Advanced

Add build step

Post-build Actions

Add post-build action

Save Apply

Type here to search

ENG IN 10:36 PM 8/20/2023

Forloop.java - Notepad

File Edit Format View Help

```
public class Forloop{
        public static void main(String[] args){
                for(int i=0; i<5; i++){
                        System.out.println(i);
}
}
}
```

Ln 7, Col 2 | 100% | Windows (CRLF) | UTF-8

**Jenkins** Search (CTRL+K) admin log out

Dashboard > shreya > #5

- Status
- Changes
- Console Output
- Edit Build Information
- Delete build '#5'
- Previous Build

## #5 (Aug 20, 2023, 10:39:00 PM)

Keep this build forever

Add description

Started 6.4 sec ago
Took 1.3 sec

No changes.

Started by user admin

REST API Jenkins 2.417

Microsoft Store

Type here to search

ENG IN 10:39 PM 8/20/2023

**CONCLUSION**:

Hence, through this assignment, I have successfully built a Hello World and a program with for loops in Java using Jenkins.

**Deep Chavan**

**T11-15**

# LAB ASSIGNMENT 6

**AIM:** To build a Pipeline using Jenkins.

**LAB OUTCOME:**
LO1, LO3 Mapped.

**THEORY:**

**A Jenkins Pipeline** is a method of defining software development workflows as code. It lets you describe your entire build, test, and deployment process in a structured manner. With Jenkins Pipelines, you can choose between Declarative and Scripted syntax to specify your automation steps. These pipelines are versioned alongside your application code in your version control system, ensuring consistency and reproducibility. Pipelines offer features like parallel execution of tasks, integration with numerous plugins, and the ability to encapsulate reusable components. They bring transparency through detailed logs, visualise the flow of tasks, and facilitate advanced automation practices like Continuous Integration and Continuous Delivery.

**Continuous Delivery (CD)** pipelines automate the steps involved in getting software changes from development to production. They include building code, running tests, deploying to staging, and potentially deploying to production. CD pipelines ensure that software updates are thoroughly tested, reducing errors and allowing for rapid and reliable delivery. They involve stages such as automated testing, staging environment validation, and production deployment, all supported by automation. By automating these steps, CD pipelines streamline software delivery, enhance collaboration, and enable faster response to user needs.

**A Jenkinsfile** is a text file used to define and describe the stages and steps of a Jenkins Pipeline. It's written in Groovy, a versatile scripting language that allows you to express complex automation workflows. The Jenkinsfile is stored within your version control repository alongside your application code, enabling versioning and consistency between code and pipeline changes.

There are two main syntax options for writing Jenkinsfiles: Declarative and Scripted.

**Declarative Pipeline Syntax:**

Declarative pipelines offer a simplified way to define pipelines. They focus on describing the high-level structure of the pipeline and its stages. Declarative pipelines are less verbose and are designed to be easy to read and understand. They follow a structured syntax and provide predefined directives for commonly used stages and steps.

Example:

```
pipeline {
agent     any
    stages {

        stage('Build') {
            steps {
```

```
sh 'echo "Building"'

            }

        }

        stage('Test') { steps

            {

                sh 'echo "Testing"'

            }

        }

        stage('Deploy') { steps

            {

                sh 'echo "Deploying"'

            }

        }

    }
}
```

**Scripted Pipeline Syntax:**

Scripted pipelines provide more flexibility and control over your pipeline workflow. With Scripted pipelines, you write custom Groovy scripts to define each stage and step. This syntax is more powerful but can be more verbose and complex compared to the Declarative syntax. Scripted pipelines are suitable for scenarios where you need precise control over the order and execution of tasks.

Example:

```
node {
stage('Build') {
echo "Building"

    }
stage('Test') {
```

```
echo "Testing"

    }
stage('Deploy') {
echo "Deploying"
```

}

}

**Benefits of Jenkins Pipeline:**

1. Consistency: Pipelines provide a uniform process for building, testing, and deploying code, reducing variability and ensuring reliability.

2. Efficiency: Automation streamlines tasks, minimising manual effort and accelerating software delivery.

3. Versioned Control: Jenkinsfiles are versioned, keeping automation in sync with code changes and aiding collaboration.

4. Adaptability: Pipelines offer adaptable syntax options (Declarative and Scripted) to match your workflow complexity.

## OUTPUT:

Dashboard > MyPipeline > Configuration
Configure
General
Advanced Project Options
Pipeline
Definition
Pipeline script
Script
pipeline {
agent any
stages {
stage('Hello') {
steps {
echo 'Hello World'
}
}
}
}
Use Groovy Sandbox
Pipeline Syntax
Save
Apply
REST API
Jenkins 2.420
28°C
Mostly cloudy
Search
ENG
IN
11:21 AM
8/28/2023

localhost:8080/job/MyPipeline/
Jenkins
Search (CTRL+K)
Shreya Kamath
log out
Dashboard > MyPipeline >
Status
Changes
Build Now
Configure
Delete Pipeline
Full Stage View
Rename
Pipeline Syntax
Build History
trend
Filter builds...
#1
28-Aug-2023, 11:16 am
Atom feed for all
Atom feed for failures
MyPipeline
First pipeline
DevOps Lab 28/08/2023
Edit description
Disable Project
Stage View
Hello
Average stage times:
(Average full run time: ~7s)
793ms
#1
Aug 28
11:16
No Changes
793ms
Permalinks
REST API
Jenkins 2.420
28°C
Mostly cloudy
Search
ENG
IN
11:20 AM
8/28/2023

Jenkins
Search (CTRL+K)
Shreya Kamath
log out
Dashboard > MyPipeline > #2
Status
Changes
Console Output
Edit Build Information
Delete build '#2'
Restart from Stage
Replay
Pipeline Steps
Workspaces
Previous Build
Build #2 (28-Aug-2023, 11:20:36 am)
Keep this build forever
Add description
Started 10 sec ago
Took 3.7 sec
Started by user Shreya Kamath
REST API
Jenkins 2.420
28°C
Mostly cloudy
Search
ENG
IN
11:20 AM
8/28/2023

Dashboard > MyPipeline > Configuration

## Configure

- General
- Advanced Project Options
- Pipeline

### Advanced Project Options

Advanced

### Pipeline

Definition

Pipeline script

Script ?

try sample Pipeline...

```
pipeline {

agent any

stages { stage('Build') { steps { echo 'Hi, GeekFlare. Starting to build the App.' stage('Test') { stage('Deploy start ') {

}

}

steps {

input('Do you want to proceed?')

}

}
```

Save Apply

28°C Mostly cloudy | Search | ENG IN | 11:23 AM 8/28/2023

Jenkins | Search (CTRL+K) | 1 | Shreya Kamath | log out

Dashboard > MyPipeline > #2

- Status
- Changes
- Console Output
  - View as plain text
- Edit Build Information
- Delete build '#2'
- Restart from Stage
- Replay
- Pipeline Steps
- Workspaces
- Previous Build

## Console Output

```
Started by user Shreya Kamath
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\.jenkins\workspace\MyPipeline
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Hello)
[Pipeline] echo
Hello World
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

REST API Jenkins 2.420

28°C Mostly cloudy | Search | ENG IN | 11:21 AM 8/28/2023

## CONCLUSION:

Through this assignment, I have learnt the concept of Pipeline and Continuous Delivery (CD) on Jenkins, and successfully executed a pipeline printing “Hello World”.

**Deep Chavan**
**T11-15**

## Lab Assignment 7

**AIM:** To understand Docker architecture and container life cycle, install docker , deploy container in docker.

**LAB OUTCOME:**
LO1, LO5 Mapped.

**THEORY:**
**Docker** is a technology that allows you to package and run applications and their dependencies in a consistent and isolated environment called a container. Think of it like a shipping container for your software – it contains everything your application needs to run, such as code, libraries, and settings, all bundled together.

**Docker Architecture:**

1. **Docker Engine**: This is like the core of Docker. It's a program that runs on your computer or server and manages containers. It consists of the Docker daemon (a background service) and the Docker command-line interface (CLI).

2. **Images**: Containers start from images. An image is like a blueprint or template for a container. It includes all the files and instructions needed to create a container. Images can be shared and used to create multiple containers.

3. **Containers**: These are the instances of images. When you run an image, it becomes a container. Containers are isolated environments that contain your application and its dependencies, making sure it runs consistently across different systems.

4. **Registry**: A registry is like a library of Docker images. Docker Hub is a popular public registry, but you can also set up private registries. You can push (upload) and pull (download) images to/from registries.

**Life Cycle of a Container:**

1. Create: You start by creating a container from an image using the `docker run` command. This creates an isolated instance of your application.

2. Run: Once created, you can start the container with `docker start`. Your application runs within the container as if it's on its own little computer.

3. Pause and Resume: You can pause a running container with `docker pause` and then resume it with `docker unpause`. This can be handy for saving resources when a container isn't actively in use.

4. Stop: When you're done with a container, you can stop it with `docker stop`. This gracefully shuts down your application.

5. Start: You can later start the container again with `docker start`, and it will resume from where it left off.

6. Remove: If you no longer need a container, you can remove it with `docker rm`. This deletes the container, but not the image it was created from.

7. Cleanup: You can also clean up unused images with `docker image prune` to free up storage space.

**Installation steps of docker with screenshot.**

1. Download Docker Desktop for Windows:

- Visit https://www.docker.com/products/docker-desktop and download the installer.

2. Run the Installer:

- Double-click the installer file to begin installation.

3. Configuration Options:

- After installation, access Docker settings by right-clicking the Docker icon on the desktop. Now, you're ready to configure Docker Desktop for Windows.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\rudra> docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
719385e32844: Pull complete
Digest: sha256:dcba6daec718f547568c562956fa47e1b03673dd010fe6ee58ca806767031d1c
Status: Downloaded newer image for hello-world:latest
docker.io/library/hello-world:latest

What's Next?
  View summary of image vulnerabilities and recommendations → docker scout quickview hello-world
PS C:\Users\rudra> docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

**CONCLUSION:**

Docker simplifies the process of developing, testing, and deploying applications because it ensures that what works on your development machine will also work in other environments, like a production server, without the "it works on my machine" problem. It's especially valuable in modern software development and deployment workflows, where consistency and scalability are essential.

**Deep Chavan**

**T11-15**

# LAB ASSIGNMENT 8

**AIM:** : Deploy static web application on docker.

**LAB OUTCOME:**
LO1, LO5 Mapped.

**THEORY:**
To deploy a static web application on Docker, you can follow these steps:

**1. Install Docker Desktop:**
If you haven't already, download and install Docker Desktop for Windows. You can get it from the official Docker website: https://www.docker.com/products/docker-desktop

**2. Verify Docker Installation:**
After installation, open Docker Desktop to ensure that it's running correctly. You should see the Docker icon in your system tray.

**3. Create a Dockerfile:**
Create a Dockerfile in the root directory of your web application. This file is used to define how your application should be built and run within a Docker container. Here's a simple example of a Dockerfile for a static web application:

Dockerfile

```
# Use an official Nginx image as the base image
FROM nginx:alpine

# Copy your static web application files to the container
COPY ./path/to/your/app /usr/share/nginx/html

# Expose port 80 to the host
EXPOSE 80
```

**4. Build the Docker Image:**
Open a terminal and navigate to the directory containing your Dockerfile. Run the following command to build a Docker image:

```
docker build -t my-web-app .
```

Replace `my-web-app` with your desired image name, and don't forget the period at the end, which indicates the current directory.

**5. Run the Docker Container:**
After building the image, you can start a Docker container based on that image using the following command:

```
docker run -d -p 8080:80 my-web-app
```

This command runs the container in detached mode (`-d`) and maps port 8080 on your host to port 80 in the container. You can choose a different port if you like.

**6. Access Your Web Application:**

Open a web browser and navigate to `http://localhost:8080` (or the port you specified in step 5). You should be able to access your static web application running inside the Docker container.

**7. Manage Docker Containers:**

You can manage your Docker containers using Docker commands like `docker ps` to list running containers, `docker stop <container_id>` to stop a container, and `docker rm <container_id>` to remove a container.

**SCREENSHOTS:**

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\rudra> cd desktop/containerdemo
PS C:\Users\rudra\desktop\containerdemo> docker build -t static-web-app .
[+] Building 8.9s (8/8) FINISHED                                                                                    docker:default
 => [internal] load build definition from Dockerfile                                                                          0.0s
 => => transferring dockerfile: 84B                                                                                           0.0s
 => [internal] load .dockerignore                                                                                             0.0s
 => => transferring context: 2B                                                                                               0.0s
 => [internal] load metadata for docker.io/library/nginx:alpine                                                               3.0s
 => [auth] library/nginx:pull token for registry-1.docker.io                                                                  0.0s
 => [internal] load build context                                                                                             0.0s
 => => transferring context: 4.28kB                                                                                           0.0s
 => [1/2] FROM docker.io/library/nginx:alpine@sha256:4c93a3bd8bf95412889dd84213570102176b6052d88bb828eaf449c56aca55ef        5.7s
 => => resolve docker.io/library/nginx:alpine@sha256:4c93a3bd8bf95412889dd84213570102176b6052d88bb828eaf449c56aca55ef        0.0s
 => => sha256:d571254277f6a0ba9d0c4a08f29b94476dcd4a95275bd484ece060ee4ff847e4 16.69kB / 16.69kB                             0.0s
 => => sha256:96526aa774ef0126ad0fe9e9a95764c5fc37f409ab9e97021e7b4775d82bf6fa 3.40MB / 3.40MB                               1.7s
 => => sha256:34b58b4f5c6d133d97298cbaae140283dc325ff1aeffb28176f63078baeffd14 1.99kB / 1.99kB                               0.0s
 => => sha256:f2004135e416117cc29b9fd1a5c217b19bd25556f8f54f981f1191674080a1f2 1.90MB / 1.90MB                               0.8s
 => => sha256:fbf1cf5026c467c51d6532a304acb35164d5aaee73d59e12def63095f4fe895f 626B / 626B                                   0.3s
 => => sha256:4c93a3bd8bf95412889dd84213570102176b6052d88bb828eaf449c56aca55ef 1.65kB / 1.65kB                               0.0s
 => => sha256:38966af6931dff98fc0ff3f63f490938a895c2739b20e819b60ad6024b6dbfe4 958B / 958B                                   0.7s
 => => sha256:c3ee70732c61e54665d4cd10d75c2962958b72d6dbefe015e76956109d9b5313 370B / 370B                                   1.0s
 => => sha256:7e2fd992447a7940a6090f3c4eb2dd92ad37ae1144d6a9285bf3eb08bbe9be6e 1.21kB / 1.21kB                               1.2s
 => => sha256:76cbc9ea6abf200d8089d7fe3c6ad19d6f0ce9eb05199736fe1d62f711a3d507 1.40kB / 1.40kB                               1.3s
 => => sha256:37f8bcf34db7931f3e1386852d3dde3d244cb54f28aabed22d4a69082078dc59 12.64MB / 12.64MB                             5.0s
 => => extracting sha256:96526aa774ef0126ad0fe9e9a95764c5fc37f409ab9e97021e7b4775d82bf6fa                                    0.1s
 => => extracting sha256:f2004135e416117cc29b9fd1a5c217b19bd25556f8f54f981f1191674080a1f2                                    0.2s
 => => extracting sha256:fbf1cf5026c467c51d6532a304acb35164d5aaee73d59e12def63095f4fe895f                                    0.0s
 => => extracting sha256:38966af6931dff98fc0ff3f63f490938a895c2739b20e819b60ad6024b6dbfe4                                    0.0s
 => => extracting sha256:c3ee70732c61e54665d4cd10d75c2962958b72d6dbefe015e76956109d9b5313                                    0.0s
 => => extracting sha256:7e2fd992447a7940a6090f3c4eb2dd92ad37ae1144d6a9285bf3eb08bbe9be6e                                    0.0s
 => => extracting sha256:76cbc9ea6abf200d8089d7fe3c6ad19d6f0ce9eb05199736fe1d62f711a3d507                                    0.0s
 => => extracting sha256:37f8bcf34db7931f3e1386852d3dde3d244cb54f28aabed22d4a69082078dc59                                    0.4s
 => [2/2] COPY . /usr/share/nginx/html                                                                                        0.1s
 => exporting to image                                                                                                        0.0s
 => => exporting layers                                                                                                       0.0s
```

```
 => => writing image sha256:06db0b1483660d2f27b333f4944e7113df3aee000dd816cc522bdf38e3aaedb8                                 0.0s
 => => naming to docker.io/library/static-web-app                                                                            0.0s

What's Next?
  View summary of image vulnerabilities and recommendations → docker scout quickview
PS C:\Users\rudra\desktop\containerdemo> docker run -d -p 8080:80 static-web-app
9cb5a41f78ce9281ee54af144dd8bff422876d5f4eb268171cc9afb9e77c70b0
PS C:\Users\rudra\desktop\containerdemo>
```

## CONCLUSION:

In summary, deploying a static web application on Docker in Windows 11 is a straightforward process. By installing Docker Desktop, creating a Dockerfile, building an image, and running a container, you can host your web app with ease. Managing containers and cleaning up resources is also manageable, making it an efficient and scalable solution for web application deployment.

**Deep Chavan**

**T11-15**

LAB ASSIGNMENT 9

**AIM:** Installation of nagios on ubuntu system.

**LAB OUTCOME:**
LO1, LO5 Mapped.

**THEORY:**
**Nagios** is an open-source monitoring and alerting system that is widely used to monitor the availability and health of IT infrastructure components, including servers, network devices, applications, and services. It helps organisations maintain the reliability and stability of their systems by providing real time visibility into the performance and status of various components in their environment.

**Key features and components of Nagios includes:**

1. **Host and Service Monitoring:** Nagios allows users to define hosts (e.g., servers, routers) and services (e.g., web services, email servers) to be monitored. It periodically checks these hosts and services to ensure they are functioning correctly.
2. **Alerting and Notification**: When Nagios detects a problem or a service outage, it can send notifications to designated administrators or teams through various methods, including email, SMS, and custom scripts. This enables timely responses to issues.
3. **Flexible Configuration:** Nagios is highly configurable and allows users to define custom checks, thresholds, and notification rules. This flexibility makes it suitable for a wide range of monitoring scenarios.
4. **Web Interface:** Nagios provides a web-based dashboard that offers a real-time view of the monitored infrastructure's status. Administrators can access this dashboard to see which services are up or down and view historical performance data.
5. **Plugin Architecture:** Nagios uses a plugin system that allows users to extend its monitoring capabilities. Many plugins are available for monitoring specific applications, devices, or protocols, and users can develop custom plugins as needed.
6. **Performance Graphs**: Nagios can collect performance data and display it in graphs and charts.

This helps in analysing historical trends and identifying potential issues before they become critical.

7. **Event Logging**: Nagios keeps a detailed log of monitoring events and notifications. This log can be useful for troubleshooting and auditing.
8. **Scheduled Downtime**: Administrators can schedule downtime for planned maintenance or upgrades to prevent unnecessary alerts during maintenance windows.
9. **Community and Support**: Nagios has an active user community, which provides resources, documentation, and support for users. There are also commercial versions and third-party tools built around Nagios for additional features and support.

Nagios is highly versatile and can be used in various IT environments, from small businesses to large enterprises. It plays a crucial role in ensuring the availability and performance of critical infrastructure components, helping organisations proactively address issues and minimise downtime. Additionally, Nagios can be integrated into larger IT management and monitoring solutions to provide comprehensive visibility into an organisation's technology stack.

**WORKING:**

Nagios works based on a simple yet effective principle: monitoring and alerting. It continuously checks the status and performance of various hosts (e.g., servers, network devices) and services (e.g., applications, websites) by running predefined checks, known as plugins.

1. **Configuration:** The Nagios administrator defines what needs to be monitored and how in a configuration file. This includes specifying hosts, services, notification settings, and alert thresholds. Users can configure checks to run at specific intervals, such as every minute, and set warning and critical thresholds for each service (e.g., response time should be under 200ms).

2. **Checks and Plugins**: Nagios periodically runs these checks based on the defined intervals and uses plugins to perform the checks. A plugin is a small script or executable that carries out a specific monitoring task, such as pinging a server, checking the HTTP response code of a website, or monitoring disk space usage.

3. **Status Data**: After running the checks, Nagios collects status data, including the results of the checks, timestamps, and performance metrics (if applicable). This data is stored internally.

4. **Alerting and Notifications**: If a check indicates a problem (e.g., a service is down, a server is unresponsive, a threshold is exceeded), Nagios triggers an alert. Alerts can be notifications sent to administrators or teams through various means, such as email, SMS, or custom scripts. These alerts inform relevant personnel about the issue, enabling them to take action.

5. **Dashboard and Reports**: Nagios provides a web-based dashboard where administrators can view the real-time status of monitored hosts and services. They can see which services are up, which are down, and view historical performance data in the form of graphs and charts. This dashboard is a central hub for monitoring and managing the infrastructure.

6. **Event Logging**: Nagios keeps a detailed log of all monitoring events, including check results, alerts, and notifications. This log serves as an audit trail and can be useful for troubleshooting and historical analysis.

7. **Scheduled Downtime**: Nagios allows administrators to schedule downtime for hosts and services during planned maintenance windows. This prevents Nagios from generating unnecessary alerts during maintenance activities.

8. **Recovery Notifications:** When a previously failed service or host returns to a healthy state, Nagios can send recovery notifications to inform administrators that the issue has been resolved.

9. **Escalation and Acknowledgements**: Nagios supports advanced features like escalation, where alerts can be escalated to higher-level teams if not acknowledged or resolved within a certain timeframe. Administrators can also acknowledge alerts, indicating that they are aware of the issue and are working on it.

10. **Performance Data:** Nagios can collect and display performance data, which helps in identifying trends, bottlenecks, and potential issues before they become critical. This data can be used for capacity planning and optimization.

Nagios operates continuously, providing real-time monitoring and alerting for IT infrastructure components. It helps organisations maintain the availability and performance of their systems and applications while enabling prompt responses to issues, ultimately reducing downtime and ensuring a more stable IT environment.

**INSTALLATION STEPS:**

Activities
Terminal
Sep 26 14:02
lab1002@lab1002-HP-280-G3-MT: ~
Fetched 9,678 kB in 4s (2,328 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
201 packages can be upgraded. Run 'apt list --upgradable' to see them.
lab1002@lab1002-HP-280-G3-MT: $ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following NEW packages will be installed:
firefox linux-headers-6.2.0-33-generic linux-hwe-6.2-headers-6.2.0-33 linux-image-6.2.0-33-generic linux-modules-6.2.0-33-generic
linux-modules-extra-6.2.0-33-generic
The following packages have been kept back:
gjs libgjs0g
The following packages will be upgraded:
alsa-ucm-conf apparmor apport apport-gtk apt apt-utils base-files bind9-dnsutils bind9-host bind9-libs cups cups-bsd cups-client
cups-common cups-core-drivers cups-daemon cups-ipp-utils cups-ppdc cups-server-common distro-info distro-info-data dpkg evince
evince-common file fwupd fwupd-signed gdm3 gir1.2-adw-1 gir1.2-gdm-1.0 gir1.2-gnomedesktop-3.0 gir1.2-gtk-4.0 gir1.2-mutter-10
gir1.2-pango-1.0 gnome-control-center gnome-control-center-data gnome-control-center-faces gnome-desktop3-data gnome-remote-desktop
gnome-session-canberra gnome-settings-daemon gnome-settings-daemon-common gnome-shell gnome-shell-common gnome-shell-extension-ubuntu-dock
im-config initramfs-tools initramfs-tools-bin initramfs-tools-core iptables isc-dhcp-client isc-dhcp-common libadwaita-1-0 libapparmor1
libapt-pkg6.0 libc-bin libc6 libc6-dbg libcanberra-gtk3-0 libcanberra-gtk3-module libcanberra-pulse libcanberra0 libcups2 libcupsimage2
libegl-mesa0 libevdocument3-4 libevview3-3 libflac8 libfprint-2-2 libfwupd2 libfwupdplugin5 libgbm1 libgdm1 libgl1-amber-dri
libgl1-mesa-dri libglapi-mesa libglx-mesa0 libgnome-bg-4-1 libgnome-desktop-3-19 libgnome-desktop-4-1 libgpgme11 libgpgmepp6
libgssapi-krb5-2 libgtk-4-1 libgtk-4-bin libgtk-4-common libidn12 libinput-bin libinput10 libip4tc2 libip6tc2 libk5crypto3 libkrb5-3
libkrb5support0 libldap-2.5-0 libldap-common libmagic-mgc libmagic1 libnbin-glib4 libnbin-proxy libnm-glib0 libnutter-10-0
libnautilus-extension1a libnetplan0 libnss-systemd libpam-systemd libpango-1.0-0 libpangocairo-1.0-0 libpangoft2-1.0-0 libpangoxft-1.0-0
libpcsclite1 libpulse-mainloop-glib0 libpulse0 libpulsedsp libqml-glib5 libqml-proxy libsasl2-2 libsasl2-modules libsasl2-modules-db
libsasl2-modules-gssapi-mit libsmbclient libsnmp-base libsnmp40 libspeechd2 libsystemd0 libudev1 libunwind8 libwbclient0 libwebp7
libwebpdemux2 libwebpmux3 libxatracker2 libxtables12 linux-firmware linux-generic-hwe-22.04 linux-headers-generic-hwe-22.04
linux-image-generic-hwe-22.04 locales mesa-vulkan-drivers modemmanager mokutil mutter-common nautilus nautilus-data netplan.io
openssh-client pulseaudio pulseaudio-module-bluetooth pulseaudio-utils python-apt-common python3-apport python3-apt python3-debian
python3-distro-info python3-distupgrade python3-macaroonbakery python3-problem-report python3-software-properties python3-speechd
python3-tz samba-libs software-properties-common software-properties-gtk speech-dispatcher speech-dispatcher-audio-plugins
speech-dispatcher-espeak-ng systemd systemd-hwe-hwdb systemd-oomd systemd-sysv systemd-timesyncd tcpdump thermald thunderbird
thunderbird-gnome-support thunderbird-locale-en thunderbird-locale-en-us tzdata ubuntu-advantage-tools ubuntu-desktop

Activities
Terminal
Sep 26 14:02
lab1002@lab1002-HP-280-G3-MT: ~
Fetched 9,678 kB in 4s (2,328 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
201 packages can be upgraded. Run 'apt list --upgradable' to see them.
lab1002@lab1002-HP-280-G3-MT: $ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following NEW packages will be installed:
firefox linux-headers-6.2.0-33-generic linux-hwe-6.2-headers-6.2.0-33 linux-image-6.2.0-33-generic linux-modules-6.2.0-33-generic
linux-modules-extra-6.2.0-33-generic
The following packages have been kept back:
gjs libgjs0g
The following packages will be upgraded:
alsa-ucm-conf apparmor apport apport-gtk apt apt-utils base-files bind9-dnsutils bind9-host bind9-libs cups cups-bsd cups-client
cups-common cups-core-drivers cups-daemon cups-ipp-utils cups-ppdc cups-server-common distro-info distro-info-data dpkg evince
evince-common file fwupd fwupd-signed gdm3 gir1.2-adw-1 gir1.2-gdm-1.0 gir1.2-gnomedesktop-3.0 gir1.2-gtk-4.0 gir1.2-mutter-10
gir1.2-pango-1.0 gnome-control-center gnome-control-center-data gnome-control-center-faces gnome-desktop3-data gnome-remote-desktop
gnome-session-canberra gnome-settings-daemon gnome-settings-daemon-common gnome-shell gnome-shell-common gnome-shell-extension-ubuntu-dock
im-config initramfs-tools initramfs-tools-bin initramfs-tools-core iptables isc-dhcp-client isc-dhcp-common libadwaita-1-0 libapparmor1
libapt-pkg6.0 libc-bin libc6 libc6-dbg libcanberra-gtk3-0 libcanberra-gtk3-module libcanberra-pulse libcanberra0 libcups2 libcupsimage2
libegl-mesa0 libevdocument3-4 libevview3-3 libflac8 libfprint-2-2 libfwupd2 libfwupdplugin5 libgbm1 libgdm1 libgl1-amber-dri
libgl1-mesa-dri libglapi-mesa libglx-mesa0 libgnome-bg-4-1 libgnome-desktop-3-19 libgnome-desktop-4-1 libgpgme11 libgpgmepp6
libgssapi-krb5-2 libgtk-4-1 libgtk-4-bin libgtk-4-common libidn12 libinput-bin libinput10 libip4tc2 libip6tc2 libk5crypto3 libkrb5-3
libkrb5support0 libldap-2.5-0 libldap-common libmagic-mgc libmagic1 libnbin-glib4 libnbin-proxy libnm-glib0 libnutter-10-0
libnautilus-extension1a libnetplan0 libnss-systemd libpam-systemd libpango-1.0-0 libpangocairo-1.0-0 libpangoft2-1.0-0 libpangoxft-1.0-0
libpcsclite1 libpulse-mainloop-glib0 libpulse0 libpulsedsp libqml-glib5 libqml-proxy libsasl2-2 libsasl2-modules libsasl2-modules-db
libsasl2-modules-gssapi-mit libsmbclient libsnmp-base libsnmp40 libspeechd2 libsystemd0 libudev1 libunwind8 libwbclient0 libwebp7
libwebpdemux2 libwebpmux3 libxatracker2 libxtables12 linux-firmware linux-generic-hwe-22.04 linux-headers-generic-hwe-22.04
linux-image-generic-hwe-22.04 locales mesa-vulkan-drivers modemmanager mokutil mutter-common nautilus nautilus-data netplan.io
openssh-client pulseaudio pulseaudio-module-bluetooth pulseaudio-utils python-apt-common python3-apport python3-apt python3-debian
python3-distro-info python3-distupgrade python3-macaroonbakery python3-problem-report python3-software-properties python3-speechd
python3-tz samba-libs software-properties-common software-properties-gtk speech-dispatcher speech-dispatcher-audio-plugins
speech-dispatcher-espeak-ng systemd systemd-hwe-hwdb systemd-oomd systemd-sysv systemd-timesyncd tcpdump thermald thunderbird
thunderbird-gnome-support thunderbird-locale-en thunderbird-locale-en-us tzdata ubuntu-advantage-tools ubuntu-desktop

```
Activities   Terminal                    Sep 26 14:05
lab1002@lab1002-HP-280-G3-MT: ~
update-alternatives: using /usr/bin/g++ to provide /usr/bin/c++ (c++) in auto mode
Setting up build-essential (12.9ubuntu3) ...
Setting up php (2:8.1+92ubuntu1) ...
Processing triggers for ufw (0.36.1-4ubuntu0.1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.3) ...
Processing triggers for php8.1-cli (8.1.2-1ubuntu2.14) ...
Processing triggers for libapache2-mod-php8.1 (8.1.2-1ubuntu2.14) ...
lab1002@lab1002-HP-280-G3-MT:~$ sudo useradd nagios
lab1002@lab1002-HP-280-G3-MT:~$ sudo groupadd nagcmd
lab1002@lab1002-HP-280-G3-MT:~$ sudo usermod -a -G nagcmd nagios
lab1002@lab1002-HP-280-G3-MT:~$ sudo usermod -a -G nagcmd www-data
lab1002@lab1002-HP-280-G3-MT:~$ wget https://github.com/NaglosEnterprises/naglosscore/archive/naglos-4.4.6.tar.gz
--2023-09-26 14:04:26--  https://github.com/NaglosEnterprises/naglosscore/archive/naglos-4.4.6.tar.gz
Resolving github.com (github.com)... 20.207.73.82
Connecting to github.com (github.com)|20.207.73.82|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://codeload.github.com/NaglosEnterprises/naglosscore/tar.gz/refs/tags/naglos-4.4.6 [following]
--2023-09-26 14:04:26--  https://codeload.github.com/NaglosEnterprises/naglosscore/tar.gz/refs/tags/naglos-4.4.6
Resolving codeload.github.com (codeload.github.com)... 20.207.73.88
Connecting to codeload.github.com (codeload.github.com)|20.207.73.88|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/x-gzip]
Saving to: 'naglos-4.4.6.tar.gz'

naglos-4.4.6.tar.gz                     [       <=>                                                  ]  10.81M  8.06MB/s    in 1.3s

2023-09-26 14:04:28 (8.06 MB/s) - 'naglos-4.4.6.tar.gz' saved [11333431]

lab1002@lab1002-HP-280-G3-MT:~$ tar -zxvf naglos-4.4.6.tar.gz
naglosscore-naglos-4.4.6/
naglosscore-naglos-4.4.6/.gitignore
naglosscore-naglos-4.4.6/.travis.yml
naglosscore-naglos-4.4.6/CONTRIBUTING.md
naglosscore-naglos-4.4.6/Changelog
naglosscore-naglos-4.4.6/INSTALLING
naglosscore-naglos-4.4.6/LEGAL
naglosscore-naglos-4.4.6/LICENSE
```

```
Activities   Terminal                    Sep 26 14:07
lab1002@lab1002-HP-280-G3-MT: ~/naglosscore-naglos-4.4.6
naglosscore-naglos-4.4.6/xdata/xodtemplate.c
naglosscore-naglos-4.4.6/xdata/xodtemplate.h
naglosscore-naglos-4.4.6/xdata/xpddefault.c
naglosscore-naglos-4.4.6/xdata/xpddefault.h
naglosscore-naglos-4.4.6/xdata/xrddefault.c
naglosscore-naglos-4.4.6/xdata/xrddefault.h
naglosscore-naglos-4.4.6/xdata/xsddefault.c
naglosscore-naglos-4.4.6/xdata/xsddefault.h
lab1002@lab1002-HP-280-G3-MT:~$ cd naglosscore -naglos-4.4.6
bash: cd: too many arguments
lab1002@lab1002-HP-280-G3-MT:~$ cd naglosscore-naglos-4.4.6
lab1002@lab1002-HP-280-G3-MT:~/naglosscore-naglos-4.4.6$ ./configfure --with-http-conf=/etc/apache2/sites-enabled
bash: ./configfure: No such file or directory
lab1002@lab1002-HP-280-G3-MT:~/naglosscore-naglos-4.4.6$ ./configure --with-http-conf=/etc/apache2/sites-enabled
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking whether make sets $(MAKE)... yes
checking whether ln -s works... yes
checking for strip... /usr/bin/strip
checking how to run the C preprocessor... gcc -E
checking for grep that handles long lines and -e... /usr/bin/grep
checking for egrep... /usr/bin/grep -E
checking for ANSI C header files... yes
checking whether time.h and sys/time.h may both be included... yes
checking for sys/wait.h that is POSIX.1 compatible... yes
checking for sys/types.h... yes
checking for sys/stat.h... yes
checking for stdlib.h... yes
```

```
Activities   Terminal                    Sep 26 14:07
lab1002@lab1002-HP-280-G3-MT: ~/nagioscore-nagios-4.4.6

 ----------------------
                HTML URL:  http://localhost/nagios/
                 CGI URL:  http://localhost/nagios/cgi-bin/
 Traceroute (used by WAP):


Review the options above for accuracy.  If they look okay,
type 'make all' to compile the main program and CGIs.

lab1002@lab1002-HP-280-G3-MT:~/nagioscore-nagios-4.4.6$ make all
cd ./base && make
make[1]: Entering directory '/home/lab1002/nagioscore-nagios-4.4.6/base'
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o nagios.o nagios.c
nagios.c: In function 'main':
nagios.c:611:25: warning: ignoring return value of 'asprintf' declared with attribute 'warn_unused_result' [-Wunused-result]
  611 |                         asprintf(&nac-xx[MACRO_PROCESSSTARTTIME], "%lld", (unsigned long long)program_start);
      |                         ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
nagios.c:841:25: warning: ignoring return value of 'asprintf' declared with attribute 'warn_unused_result' [-Wunused-result]
  841 |                         asprintf(&nac-xx[MACRO_EVENTSTARTTIME], "%lld", (unsigned long long)event_start);
      |                         ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
nagios.c: In function 'nagios_core_worker':
nagios.c:176:17: warning: ignoring return value of 'read' declared with attribute 'warn_unused_result' [-Wunused-result]
  176 |                 read(sd, response + 5, sizeof(response) - 4);
      |                 ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
nagios.c: In function 'test_path_access':
nagios.c:122:17: warning: ignoring return value of 'asprintf' declared with attribute 'warn_unused_result' [-Wunused-result]
  122 |                 asprintf(&path, "%s/%s", p, program);
      |                 ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE   -c -o broker.o broker.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE   -c -o nebmods.o nebmods.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o ../common/shared.o ../common/shared.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE   -c -o query-handler.o query-handler.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o workers.o workers.c
workers.c: In function 'handle_worker_result':
workers.c:801:25: warning: ignoring return value of 'asprintf' declared with attribute 'warn_unused_result' [-Wunused-result]
  801 |                         asprintf(&error_reason, "timed out after %.2fs", tv_delta_f(&wpres.start, &wpres.stop));
      |                         ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
workers.c:804:25: warning: ignoring return value of 'asprintf' declared with attribute 'warn_unused_result' [-Wunused-result]
```

```
Activities   Terminal                    Sep 26 14:09
lab1002@lab1002-HP-280-G3-MT: ~/nagioscore-nagios-4.4.6

make[1]: Leaving directory '/home/lab1002/nagioscore-nagios-4.4.6'
lab1002@lab1002-HP-280-G3-MT:~/nagioscore-nagios-4.4.6$ sudo make install-init
/usr/bin/install -c -m 755 -d -o root -g root /lib/systemd/system
/usr/bin/install -c -m 755 -o root -g root startup/default-service /lib/systemd/system/nagios.service
lab1002@lab1002-HP-280-G3-MT:~/nagioscore-nagios-4.4.6$ sudo make install-config
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc/objects
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/nagios.cfg /usr/local/nagios/etc/nagios.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/cgi.cfg /usr/local/nagios/etc/cgi.cfg
/usr/bin/install -c -b -m 660 -o nagios -g nagios sample-config/resource.cfg /usr/local/nagios/etc/resource.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/templates.cfg /usr/local/nagios/etc/objects/templates.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/commands.cfg /usr/local/nagios/etc/objects/commands.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/contacts.cfg /usr/local/nagios/etc/objects/contacts.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/timeperiods.cfg /usr/local/nagios/etc/objects/timeperiods.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/localhost.cfg /usr/local/nagios/etc/objects/localhost.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/windows.cfg /usr/local/nagios/etc/objects/windows.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/printer.cfg /usr/local/nagios/etc/objects/printer.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/switch.cfg /usr/local/nagios/etc/objects/switch.cfg

*** Config files installed ***

Remember, these are *SAMPLE* config files.  You'll need to read
the documentation for more information on how to actually define
services, hosts, etc. to fit your particular needs.

lab1002@lab1002-HP-280-G3-MT:~/nagioscore-nagios-4.4.6$ sudo make install-commandmode
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/var/rw
chmod g+s /usr/local/nagios/var/rw

*** External command directory configured ***

lab1002@lab1002-HP-280-G3-MT:~/nagioscore-nagios-4.4.6$ sudo make install-exfoliation

*** Exfoliation theme installed ***
NOTE: Use 'make install-classicui' to revert to classic Nagios theme

lab1002@lab1002-HP-280-G3-MT:~/nagioscore-nagios-4.4.6$
```

```
Activities   Terminal                    Sep 26 14:11
lab1002@lab1002-HP-280-G3-MT: ~/nagioscore-nagios-4.4.6
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/windows.cfg /usr/local/nagios/etc/objects/windows.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/printer.cfg /usr/local/nagios/etc/objects/printer.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/switch.cfg /usr/local/nagios/etc/objects/switch.cfg

*** Config files installed ***

Remember, these are *SAMPLE* config files.  You'll need to read
the documentation for more information on how to actually define
services, hosts, etc. to fit your particular needs.

lab1002@lab1002-HP-280-G3-MT:~/nagioscore-nagios-4.4.6$ sudo make install-commandmode
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/var/rw
chmod g+s /usr/local/nagios/var/rw

*** External command directory configured ***

lab1002@lab1002-HP-280-G3-MT:~/nagioscore-nagios-4.4.6$ sudo make install-exfoliation

*** Exfoliation theme installed ***
NOTE: Use 'make install-classicui' to revert to classic Nagios theme

lab1002@lab1002-HP-280-G3-MT:~/nagioscore-nagios-4.4.6$ sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/apache2/sites-available/nagios.conf
if [ 1 -eq 1 ]; then \
        ln -s /etc/apache2/sites-available/nagios.conf /etc/apache2/sites-enabled/nagios.conf; \
fi

*** Nagios/Apache conf file installed ***

lab1002@lab1002-HP-280-G3-MT:~/nagioscore-nagios-4.4.6$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
lab1002@lab1002-HP-280-G3-MT:~/nagioscore-nagios-4.4.6$ sudo a2enmod rewrite
\\Enabling module rewrite.
To activate the new configuration, you need to run:
  systemctl restart apache2
lab1002@lab1002-HP-280-G3-MT:~/nagioscore-nagios-4.4.6$
```

```
Activities   Terminal                    Sep 26 14:11
lab1002@lab1002-HP-280-G3-MT: ~/nagioscore-nagios-4.4.6
*** Config files installed ***

Remember, these are *SAMPLE* config files.  You'll need to read
the documentation for more information on how to actually define
services, hosts, etc. to fit your particular needs.

lab1002@lab1002-HP-280-G3-MT:~/nagioscore-nagios-4.4.6$ sudo make install-commandmode
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/var/rw
chmod g+s /usr/local/nagios/var/rw

*** External command directory configured ***

lab1002@lab1002-HP-280-G3-MT:~/nagioscore-nagios-4.4.6$ sudo make install-exfoliation

*** Exfoliation theme installed ***
NOTE: Use 'make install-classicui' to revert to classic Nagios theme

lab1002@lab1002-HP-280-G3-MT:~/nagioscore-nagios-4.4.6$ sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/apache2/sites-available/nagios.conf
if [ 1 -eq 1 ]; then \
        ln -s /etc/apache2/sites-available/nagios.conf /etc/apache2/sites-enabled/nagios.conf; \
fi

*** Nagios/Apache conf file installed ***

lab1002@lab1002-HP-280-G3-MT:~/nagioscore-nagios-4.4.6$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
lab1002@lab1002-HP-280-G3-MT:~/nagioscore-nagios-4.4.6$ sudo a2enmod rewrite
\\Enabling module rewrite.
To activate the new configuration, you need to run:
  systemctl restart apache2
lab1002@lab1002-HP-280-G3-MT:~/nagioscore-nagios-4.4.6$ sudo systemctl restart apache2
lab1002@lab1002-HP-280-G3-MT:~/nagioscore-nagios-4.4.6$ sudo systemctl start nagios
lab1002@lab1002-HP-280-G3-MT:~/nagioscore-nagios-4.4.6$ sudo systemctl enable nagios
Created symlink /etc/systemd/system/multi-user.target.wants/nagios.service → /lib/systemd/system/nagios.service.
lab1002@lab1002-HP-280-G3-MT:~/nagioscore-nagios-4.4.6$
```

**CONCLUSION:**

In this assignment, we successfully installed Nagios on an Ubuntu system, setting up a robust monitoring and alerting framework for efficient system management and proactive issue resolution.

**Deep Chavan**
**T11-15**

# LAB ASSIGNMENT 10

**AIM:** To study Puppet tools in Devops.

**LAB OUTCOME:**
LO1, LO6 Mapped.

**THEORY:**

**WHAT IS A PUPPET TOOL?**
Puppet is an open-source configuration management and automation tool used for deploying, configuring, and managing servers and infrastructure as code. It helps system administrators and DevOps teams automate repetitive tasks, enforce consistent configurations, and ensure that infrastructure is in the desired state, reducing manual effort and minimizing errors.

**Features of Puppet:**

- **Automation**: Puppet automates the configuration and management of servers and infrastructure, reducing manual tasks and errors.
- **Declarative Language**: It uses a declarative language to specify desired infrastructure states, making it easy to define what the system should look like.
- **Infrastructure as Code (IaC)**: Puppet treats infrastructure configurations as code, enabling versioning, testing, and collaboration like software code.
- **Resource Abstraction**: Puppet abstracts system resources into manageable "resources," simplifying configuration management.
- **Module-Based**: It organizes configurations into reusable modules, streamlining the management of common software and services across different parts of your infrastructure.

**WHAT CAN A PUPPET DO?**
The Puppet Server is responsible for:

- Compiling the Catalog File for hosts, based on system, configuration, manifest file, etc. Puppet prepares a Catalog File based on the manifest file, which is a Puppet program used to control the systems running the Puppet Agent. After processing the manifest file, the Puppet Server prepares the Catalog File based on the target platform.
- Sending the Catalog File to Agents when they query the Server.
- Storing information about the entire environment, such as host information, metadata such as authentication keys. ● Gathering reports from each Agent and then preparing the overall report.

**HOW DO PUPPETS WORK?**
Puppet uses an agent-server model for configuring systems, with agents and a server known as the Puppet Agent and Puppet Server, respectively. The key points are:

- Puppet Agent must be installed on each system for management.
- Agents connect securely to the Puppet Server to receive instructions in a file referred to as the Catalog.

- Agents execute instructions to reach the desired system state and report back to the server.
- Puppet employs a declarative Domain Specific Language (DSL) for configuration attributes, defined in manifests.
- Agents collect facts about themselves and send them to the Puppet Master.
- The Puppet Master compiles catalogs specifying how each node should be configured.
- Catalogs are sent to agents, which then apply the configurations, ensuring consistency. - The Puppet Master manages the entire infrastructure, including compiling catalogs, sending reports, and handling file transfers.
- Communication is secured through SSL/TLS protocols, encrypting traffic between nodes and the master.

**Puppet Blocks**

Puppet Blocks. Resources- Puppet has many built-in resources like file, user, package and service. The puppet language allows administrators to manage the system resources independently and ensure that the system is in the desired state.

**Puppet Resources**

Resources are one of the key fundamental units of Puppet used to design and build any particular infrastructure or a machine. They are mainly used for modeling and maintaining system configurations. Puppet has multiple types of resources, which can be used to define the system architecture or the user has the leverage to build and define a new resource.

**Puppet Classes**

Puppet classes are defined as a collection of resources, which are grouped together in order to get a target node or machine in a desired state. These classes are defined inside Puppet manifest files which are located inside Puppet modules.

**Puppet Modules**

Modules serve as the basic building blocks of Puppet and are reusable and shareable. Modules contain Puppet classes, defined types, tasks, task plans, functions, resource types and providers, and plug-ins such as custom types or facts. Modules must be installed in the Puppet modulepath.

**Puppet Manifest File**

A manifest is a file containing Puppet configuration language that describes how resources should be configured. The manifest is the closest thing to what one might consider a Puppet program. It declares resources that define state to be enforced on a node.

**PUPPET MANIFEST FILES:**

- In puppet, all the programs are written in Ruby programming language and added with an extension of .pp is known as manifests. The full form of .pp is the puppet program.
- Manifest files are puppet programs. This is used to manage the target host system. All the puppet programs follow the puppet coding style. We can use a set of different kinds of resources in any manifest, which is grouped by definition and class.
- Puppet manifest also supports the conditional statement. The default manifest file is available in the /etc/puppet/manifests/site.pp location.

**SYNTAX OF A MANIFEST FILE:**

A manifest file contains the following components:

1. Resource Declaration
2. Comments
3. Variables
4. Conditional Statements
5. Classes and Modules
6. Include Classes
7. Templates
8. Resource Ordering
9. Node Definitions

**EXAMPLE OF MANIFEST FILES:**

```
# This is a Puppet manifest file # Define a file
resource file { '/etc/myconfig': ensure => 'file',
owner => 'root', group => 'root', mode     =>
'0644', content => 'This is the configuration
content.',
# Set file content
}
# Define a package resource package
{ 'nginx':
  ensure => 'installed',
}
# Define a service resource service
{ 'nginx':
```

```
  ensure    =>
  'running', enable
            =>
  true, subscribe =>
  File['/etc/myconfig'],
}
#   Node   definition   node
'webserver.example.com'    {
include nginx_config
}
```

**WHY DO WE NEED PUPPET MANIFEST FILES?**

Puppet manifest files are crucial for:

- Automation: They automate configurations, reducing manual work.
- Consistency: Ensure a uniform state across servers.
- Version Control: Track changes and collaborate using code repositories.
- Modularity: Reuse configurations for efficiency.
- Customization: Adapt configurations using variables.
- Documentation: Clearly document infrastructure setups. ● Scalability: Scale configurations effortlessly.
- Compliance: Enforce security and compliance standards. ● Change Control: Manage changes systematically.

**BENEFITS OF PUPPET:**

1. Eliminates time consuming, complex and stressful manual configurations of the infrastructure
2. Automates the process of configurations, controlling and managing large numbers (over 100 servers) of servers and other infrastructure
3. Eliminates complex error-prone tasks of automating the infrastructure deployment and configuration
4. It is an inexpensive method of solving the configurations bottlenecks and latency in the speed
5. Puppet is used as a continuous delivery model to the software release cycle by automating the operations and deployment workflow

## CONCLUSION:

Through this assignment, I have learnt the concept of Puppet Tools in DevOps, its different components and working. I also understood the fundamentals of Manifest files in Puppet.

Deep Chavan
T11-15
DevOPs Lab

# WRITTEN ASSIGNMENT - 1

Q.1] What is test automation? What are it's advantages?

Ans: Test automation or automation testing refers to the use of specialised tools and scripts to automatically execute test cases, compare the actual outcomes with the expected outcomes, and report the results. It is primarily used to perform repetitive tasks and regression testing to ensure that software applications continue to work as expected after changes or updates.

Advantages of automation testing:

- Faster execution: automated tests can run much faster, ~~than~~ especially for repetitive and regression tasks, significantly reducing the time required for testing.

- Increased Test Coverage: Automation allows for the execution of a larger number of test cases, ensuring that more aspects of the application are tested and potential issues are identified.

- Improved Accuracy: Automation reduces the risk of human errors that might occur during manual testing, leading to more reliable test results.

- Cost effective: Although there is an initial investment in setting up automation, it becomes cost-effective over time by reducing the need for manual intervent, enabling frequent testing

Sundaram FOR EDUCATIONAL USE

- Reusability of Test Scripts: Once written, test scripts can be ~~resent~~ reused across different versions of the application or in different projects, enhancing efficiency.

- Continuous Integration and Continuous Deployement (CI/CD): Continuous Automation testing is crucial for CI/CD pipelines, although allowing tests to run automatically as part of the development process ensuring rapid feedback and deployement.

Q2] What is XPath? Explain the difference between single slash and double slash in XPath.

Ans. XPath is a query language used for selecting nodes from an XML document. In the context of web development, XPath is commonly used in conjunction with tools like Selenium for navigating and interacting with elements on a web page. XPath allows you to locate elements based on their tagnames, attributes, text content and heirarchical relationships within the documents.

- XPath Syntax and Structure:
It is essentially a path through an XML document's structure. It consists of a sequence steps, each representatives of a node in the document tree.
- Nodes: Elements in the document tree. (like tags in HTML)
- Attributes: Properties associated with nodes (like id, class).

Sundaram
FOR EDUCATIONAL USE

- Text: Content within a document.

Difference between Single slash (/) and Double Slash (//) in XPath.

1.) Single Slash:
- Absolute Path: The single slash is used to create an absolute path. It starts the path from the root element of the document.
- Direct Child Selection: When used within an expression it selects direct children of the current node.
- Example:
/html/body/div: selects the  element that is a direct child of <body>, which is in turn a direct child of <html>

2.) Double Slash:
- Relative Path: Creates a relative path, searching the entire document or current node subtree to find matching elements.
- Any descendant selection: It selects elements anywhere in the document or current subtree, not just direct children.
- Examples
//div : Selects all  elements in the document no matter, where they are located.

Sundaram

FOR EDUCATIONAL USE

Deep Chavan
711-15
DevOPs lab

# WRITTEN ASSIGNMENT - 2

Q1] What is selenium? What are Selenium suite elements?

Ans → Selenium:

Open source automation testing framework primarily used for automating web applications. It allows testers to write test scripts across browsers in various programming languages. It is widely used for functional testing, regression testing, and is also a key tool in continuous integration, continuous deployement (CI/CD) pipelines.

It component's are:

1) Selenium's Web Driver:

It is the core component of the selenium suite. It provides a programming interface to interact with web browsers by sending commands and retrieving results. Web Driver directly communicates with the browser, driving the browser's behaviour.

2) Selenium's IDE:

It is a browser extension that allows testers to record edit, and replay test cases in a user friendly interface. It is primarily used for quick prototyping of test cases and is ideal for beginners.

3) Selenium Grid:

It allows the execution of test scripts on multiple machines and browsers in parallel, effectively distributing the test load. It supports running tests on a distributed network of computers which is useful for cross browser testing and speeding up test execution.

Uniteram

4. Selenium RC (Remote Control) Deprecated:
It is used to write test cases in multiple languages and required starting a server before executing the test scripts, which made it slower and more complex than Web Driver.

Q.2] What makes Selenium such a widely used testing tool? Why is it advised to select Selenium as a testing tool for web applications or systems.

Ans • Open Source and Free: Selenium is completely free to use without licensing cost. This makes it an attractive option for organisations of all sizes.

• Cross Browser Compatibility: Selenium supports multiple browsers like Chrome, Firefox, allowing testers to perform cross-browser testing.

• Cross Platform Support: It can be used on various operating systems, ~~such as Windows~~, allowing testers to run their scripts on different platform.

• Integration with other tools: It integrates well with other tools and frameworks, such as Test NG, JUnit, Maven, making it a key component in automated testing ecosystem.

• Support multiple programming languages: Selenium supports a wide range of programming languages. This flexibility allows testers to write scripts in the language they are most comfortable in.

FOR EDUCATIONAL USE

Q3] Why select selenium for testing web applications?

Ans:

1.) Comprehensive Browser and OS support:
Selenium is ideal for testing web applications that need to work across multiple and operating systems ensuring broad compatibility.

2.) Cost effectiveness:
Since selenium is free and open-source, it significantly reduces the overall cost of test automation making it an economical choice for organizations.

3) Customization and flexibility:
Selenium provides the ability to customize and extend the testing framework, allowing testers to create highly tailored test cases that fit the specific needs of the application.

4.) Scalibility: With Selenium Grid:
Test cases can run in parallel across multiple environments making it scalable for large and complex projects.

5.) Integration Capabilities: Selenium can easily integrate with popular development, testing, and CI/CD tools, making it a seamless fit in the software development lifecycle.

6) Maturity and Well-established: Being one of the oldest and most mature automation testing

Sundaram
FOR EDUCATIONAL USE