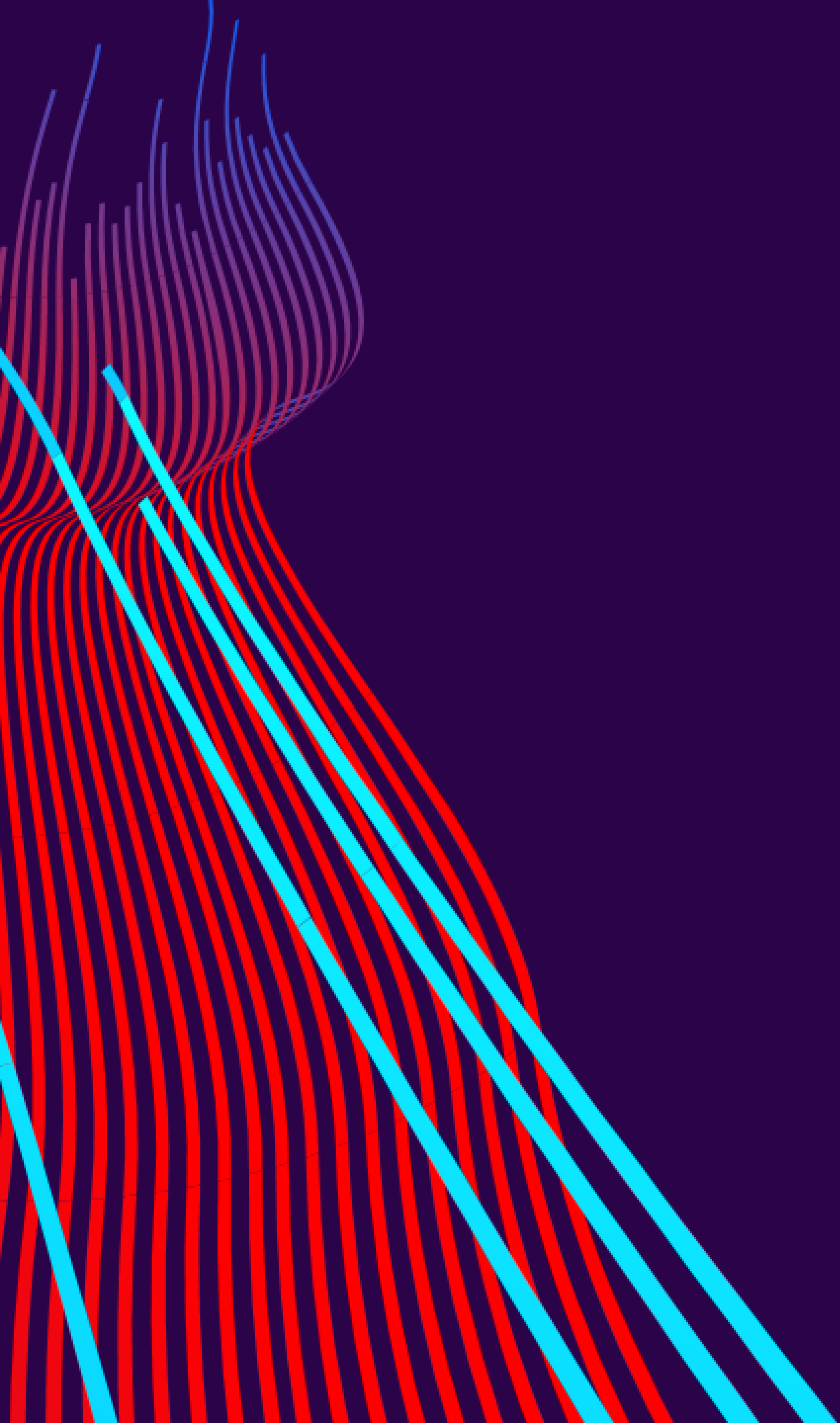
# L1: MODULE 3

HIMANSHU ARYA

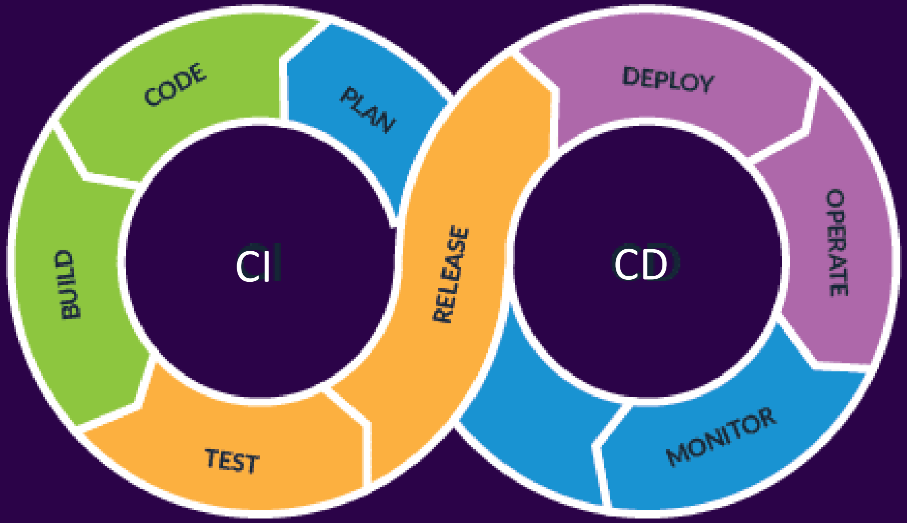


# AGENDA

1. CI/CD A GENTLE INTRODUCTION
2. CLOUD BASICS
3. CONFIGURATION MANAGEMENT
4. DEPLOYMENT STRATEGIES
5. WHAT IS DEVOPS ?
6. INFRA AS CODE BASICS
7. LINUX
8. LINUX CUSTOM SCRIPT TASKS

# CI/CD

## What is CI/CD?



### CONTINUOUS INTEGRATION

- A practice where developers integrate their code into a common repository, preferably several times a day.
- A series of scripts is run automatically to verify the changes that are pushed.

### CONTINUOUS DEPLOYMENT/DELIVERY

- A process that must happen after code is integrated for app changes to be delivered to users.

**Advantages:** Both the process are used for automating the building and deploying process.

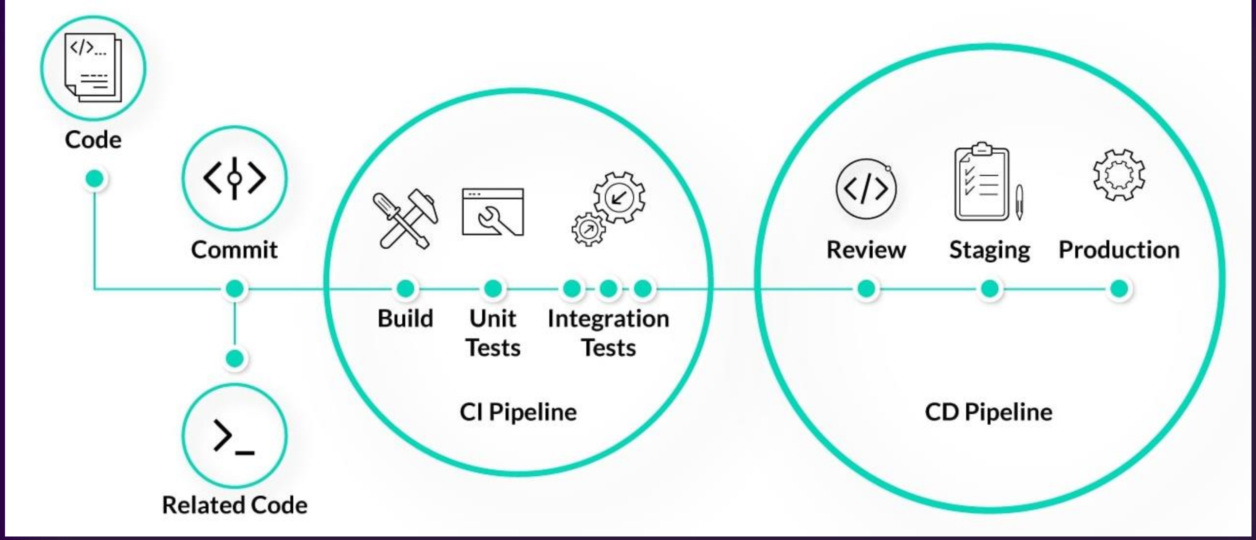
# CI/CD PIPELINE

**CI/CD relies on Tools to achieve optimal workflows :**

- Jenkins
- Spinnaker

**Benefits:**

- Smaller Code Changes.
- Continuous Testing.
- Early Fault Detections.
- Smaller Backlog.
- Customer Satisfaction.
- Easy Maintenance and Updates.

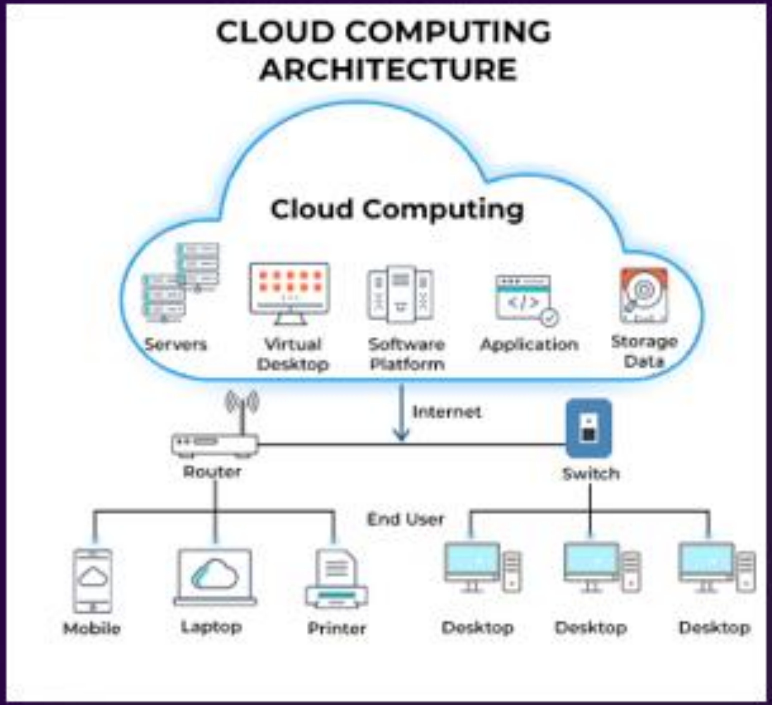


# CLOUD

**CLOUD COMPUTING** is the **on-demand delivery** of **computation power**, **database**, **storage**, applications, and other IT resources through a cloud services platform through the internet with pay-as-you-go pricing.

Advantages of cloud computing:

- Pay as you go
- Benefit from massive economies of scale
- Stop guessing capacity
- Increase speed and agility
- Realize cost savings
- Go global in minutes



# CLOUD

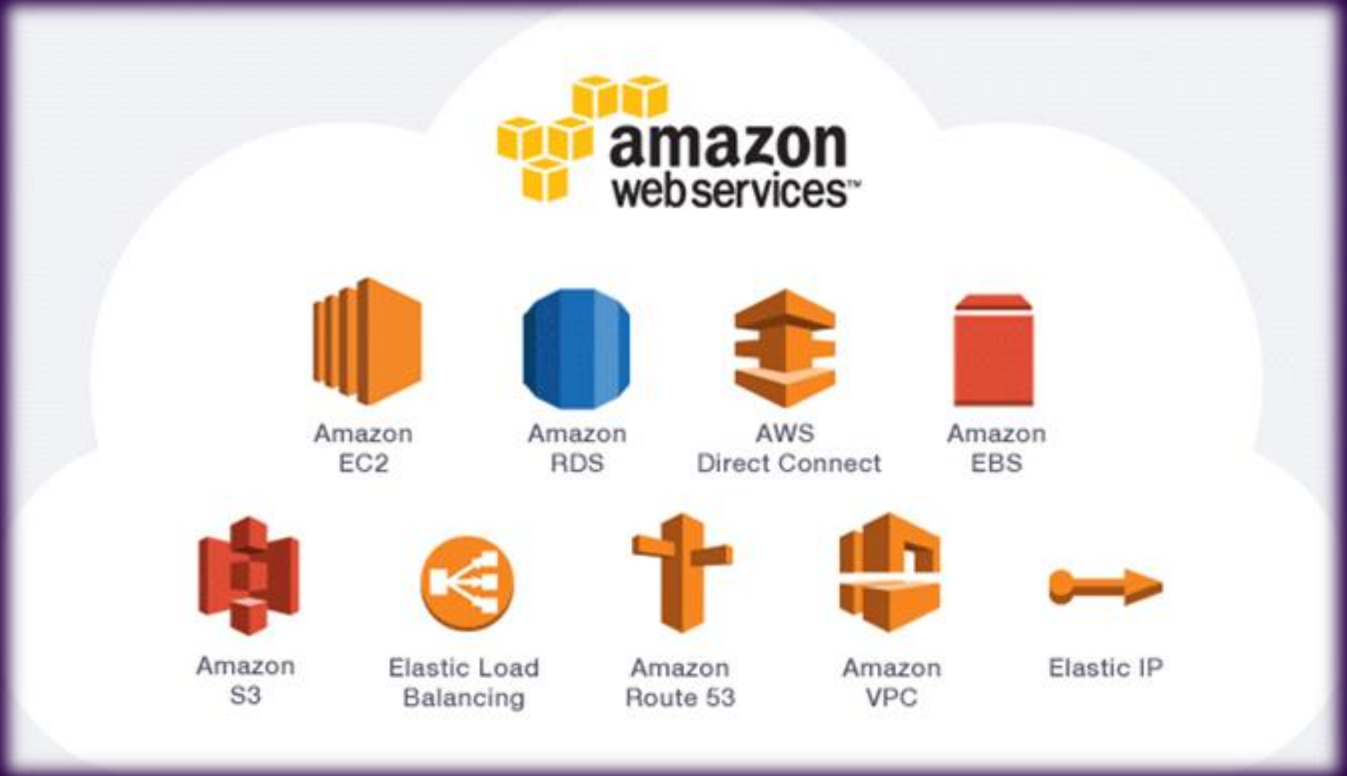
There are *different platform* which provides the cloud computing facilities.

These platforms are:

- Amazon Web Services (AWS)
- Google Cloud Platform
- Microsoft Azure

# AMAZON WEB SERVICES

The most popularly use amazon web services are:



# CONFIGURATION MANAGEMENT

- **Configuration Management** is a systems engineering process for establishing consistency of a product’s attributes throughout its life.
- With the help of ***tracking*** and ***monitors changes*** to a software systems configuration metadata.
- In software development, configuration management is commonly used alongside version control and CI/CD infrastructure.
- ***Git*** is a fantastic platform for managing configuration data.
- Version control also solves another configuration problem: ***unexpected breaking changes***.
- Managing unexpected changes using ***code review*** and ***version control*** helps to **minimize downtime**.

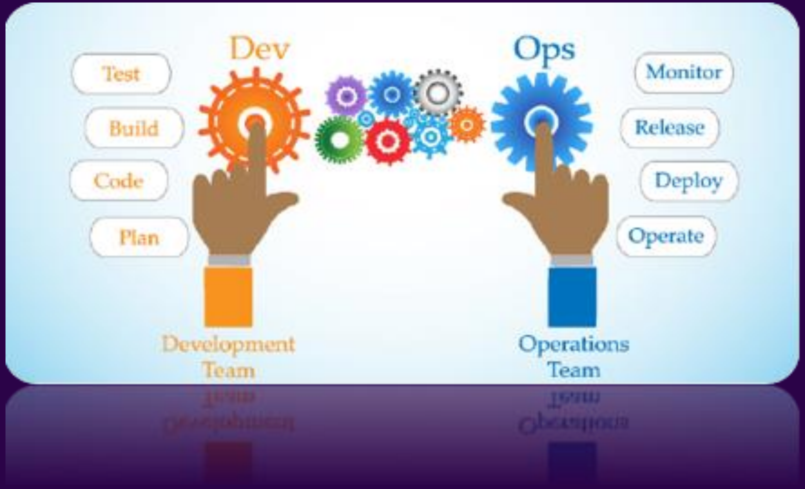
# DEPLOYMENT STRATEGIES

**There are 6 Deployment Strategies for Software Deployment we have:**

1. **Recreate**: Version A is terminated then version B is rolled out.
2. **Ramped** (*also known as rolling-update or incremental*): Version B is slowly rolled out and replacing version A.
3. **Blue/Green**: Version B is released alongside version A, then the traffic is switched to version B.
4. **Canary**: Version B is released to a subset of users, then proceed to a full rollout.
5. **A/B testing**: Version B is released to a subset of users under specific condition.
6. **Shadow**: Version B receives real-world traffic alongside version A and doesn’t impact the response.

# WHAT IS DEVOPS ?

**DevOps** is a process in which Modern software engineering Culture and Practices develop software where the *development teams* and *operation teams* work hand in hand as *one unit*, unlike the traditional ways; it is called the Agile Methodology.



Core Principle Of **DevOps:**

- Automating routine tasks.
- Enabling continuous delivery & improvement.
- Increase speed and efficiency of Software/Development.
- Enhancing product quality and customer satisfaction.

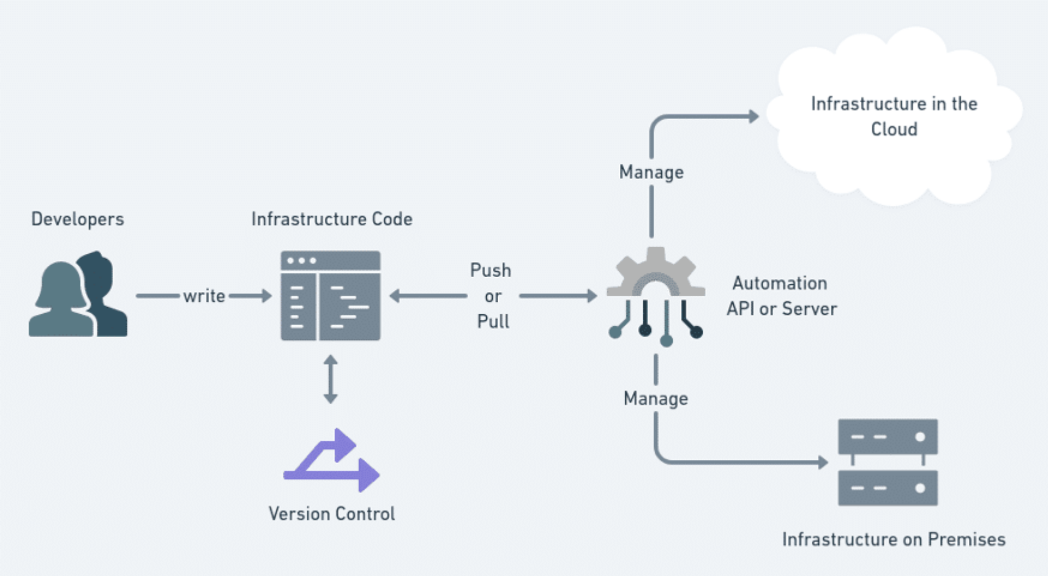
**6C's of DevOps Process:**

1. Continuous Integration
2. Continuous Delivery
3. Continuous Testing
4. Continuous Monitoring
5. Continuous Feedback
6. Continuous Improvement

# INFRA AS CODE BASICS

- **INFRASTRUCTURE AS CODE** is a higher-level abstraction used to write code/log in to a high-level language that utilizes modules to perform tasks on the machine level.
- Infrastructure as Code (IaC) allows IT operations teams to manage and provision *IT infrastructure automatically* through code, *removing the reliance on manual processes*. It is often described as “Programmable Infrastructure.”

- It includes the specification of **Virtual Machines**, **Storage**, **Network Configurations**, **Security features**, **User Accounts**, **Access Control Limits**, **Software Stacks** and so on.



# LINUX

- Just like *Windows, iOS, and Mac OS, Linux* is an operating system.
- Linux has *several different versions* to suit any type of user.
- These versions are called distributions (or, in the short form, “distros”).
- Nearly every distribution of Linux can be *downloaded for free*, burned onto disk (or USB thumb drive), and installed (on as many machines as you like).

Popular Linux distributions include:

- DEBIAN
- UBUNTU
- FEDORA

# LINUX

There are different CLI Command of Linux:

| LINUX COMMANDS | FUNCTIONS |
|---|---|
| ls | Displays information about files in the current directory. |
| pwd | Displays the current working directory. |
| mkdir | Creates a directory. |
| cd | To navigate between different folders. |
| rmdir | Removes empty directories from the directory lists. |
| cp | Copy files from one directory to another. |
| mv | Rename and Replace the files |
| rm | Delete files |
| uname | Command to get basic information about the OS |
| touch | Create empty files |
| ln | Create shortcuts to other files |

# LINUX

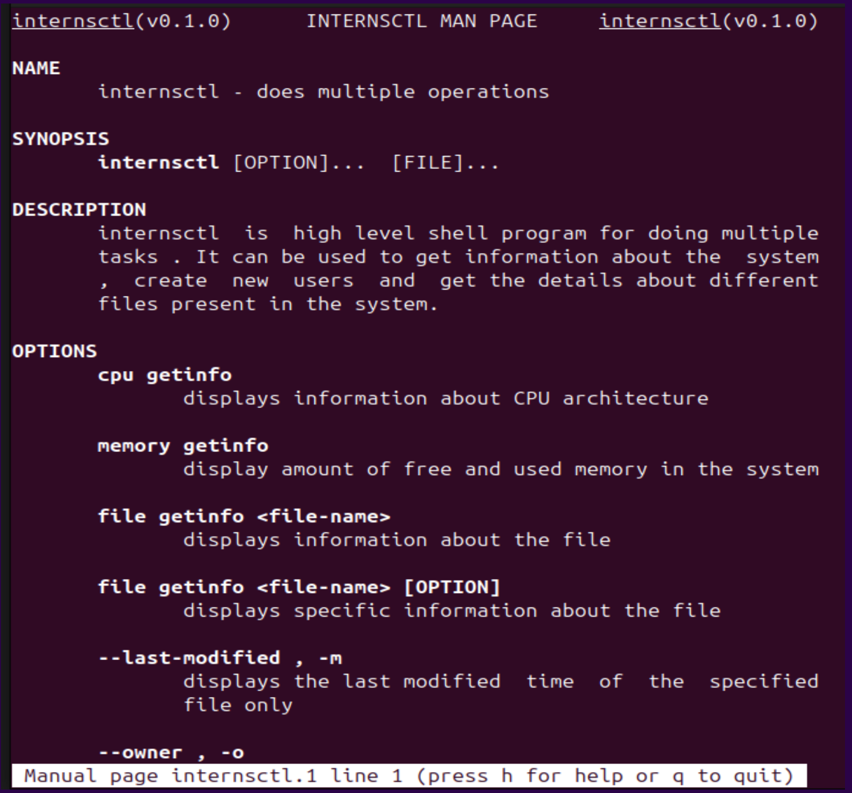
There are different CLI Command of Linux:

| LINUX COMMANDS | FUNCTIONS |
|---|---|
| cat | Display file contents on terminal. |
| clear | Clear terminal |
| man | Access manual for all Linux commands |
| grep | Search for a specific string in an output |
| echo | Print string or text to the terminal |
| wget | download files from the internet. |
| whoami | Displays the current username |
| sort | sort the file content |
| cal | View Calendar in terminal |
| whereis | View the exact location of any command typed after this command |
| wc | Check the lines, word count, and characters in a file using different options. |

# LINUX CUSTOM SCRIPT TASKS

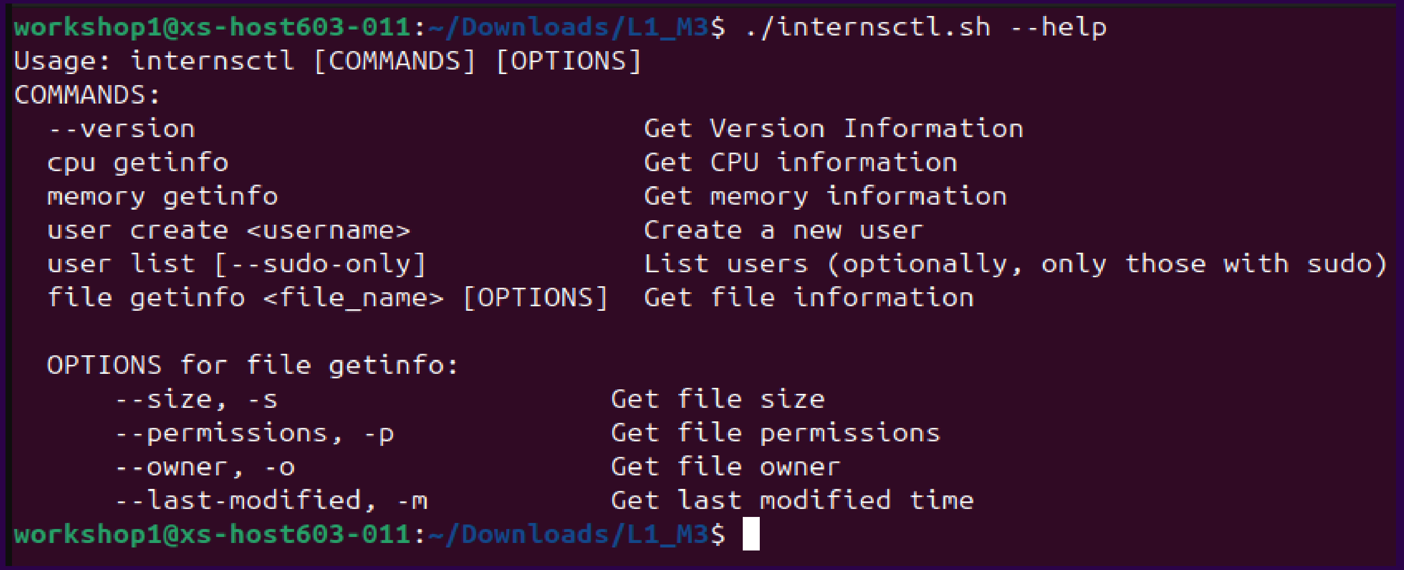
## Section A:

*$ man internsctl*



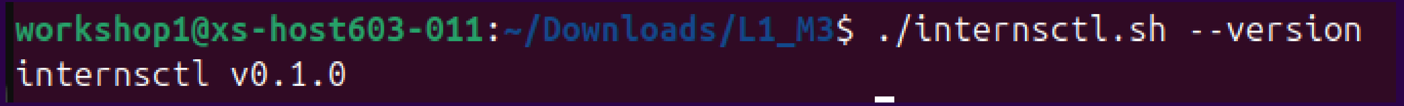
```
internsctl(v0.1.0)      INTERNSCTL MAN PAGE     internsctl(v0.1.0)

NAME
       internsctl - does multiple operations

SYNOPSIS
       internsctl [OPTION]...  [FILE]...

DESCRIPTION
       internsctl  is  high level shell program for doing multiple
       tasks . It can be used to get information about the  system
       ,  create  new  users  and  get the details about different
       files present in the system.

OPTIONS
       cpu getinfo
              displays information about CPU architecture

       memory getinfo
              display amount of free and used memory in the system

       file getinfo <file-name>
              displays information about the file

       file getinfo <file-name> [OPTION]
              displays specific information about the file

       --last-modified , -m
              displays the last modified  time  of  the  specified
              file only

       --owner , -o
 Manual page internsctl.1 line 1 (press h for help or q to quit)
```

*$ internsctl --help*



```
workshop1@xs-host603-011:~/Downloads/L1_M3$ ./internsctl.sh --help
Usage: internsctl [COMMANDS] [OPTIONS]
COMMANDS:
  --version                            Get Version Information
  cpu getinfo                          Get CPU information
  memory getinfo                       Get memory information
  user create <username>               Create a new user
  user list [--sudo-only]              List users (optionally, only those with sudo)
  file getinfo <file_name> [OPTIONS]   Get file information

  OPTIONS for file getinfo:
      --size, -s                       Get file size
      --permissions, -p                Get file permissions
      --owner, -o                      Get file owner
      --last-modified, -m              Get last modified time
workshop1@xs-host603-011:~/Downloads/L1_M3$
```

*$ internsctl --version*



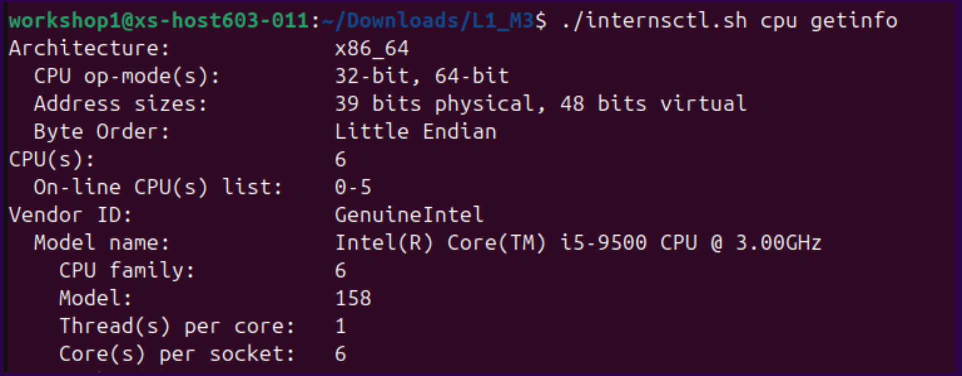
```
workshop1@xs-host603-011:~/Downloads/L1_M3$ ./internsctl.sh --version
internsctl v0.1.0
```

# LINUX CUSTOM SCRIPT TASKS

## Section B:

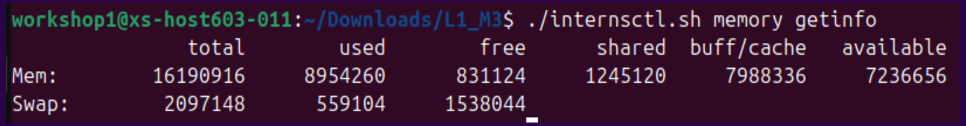
### Part1 | Level Easy

*$ internsctl cpu getinfo*



```
workshop1@xs-host603-011:~/Downloads/L1_M3$ ./internsctl.sh cpu getinfo
Architecture:                x86_64
  CPU op-mode(s):            32-bit, 64-bit
  Address sizes:             39 bits physical, 48 bits virtual
  Byte Order:                Little Endian
CPU(s):                      6
  On-line CPU(s) list:       0-5
Vendor ID:                   GenuineIntel
  Model name:                Intel(R) Core(TM) i5-9500 CPU @ 3.00GHz
    CPU family:              6
    Model:                   158
    Thread(s) per core:      1
    Core(s) per socket:      6
```

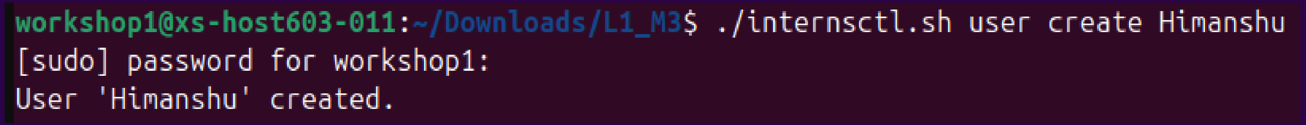
*$ internsctl memory getinfo*



```
workshop1@xs-host603-011:~/Downloads/L1_M3$ ./internsctl.sh memory getinfo
              total        used        free      shared  buff/cache   available
Mem:       16190916     8954260      831124     1245120     7988336     7236656
Swap:       2097148      559104     1538044
```

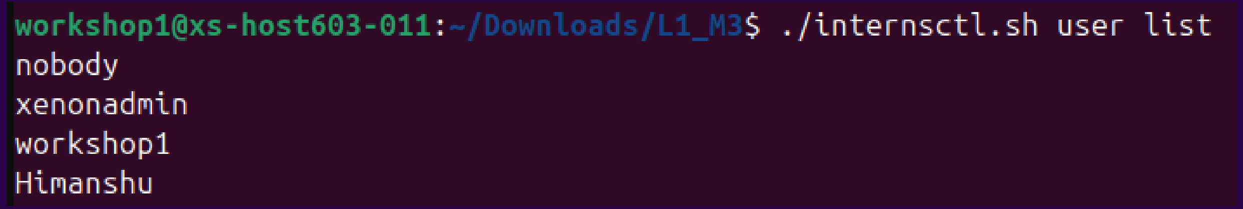
### Part2 | Level Intermediate

*$ internsctl user create <username>*



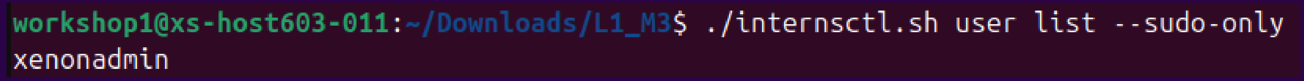
```
workshop1@xs-host603-011:~/Downloads/L1_M3$ ./internsctl.sh user create Himanshu
[sudo] password for workshop1:
User 'Himanshu' created.
```

*$ internsctl user list*



```
workshop1@xs-host603-011:~/Downloads/L1_M3$ ./internsctl.sh user list
nobody
xenonadmin
workshop1
Himanshu
```

*$ internsctl user list --sudo-only*



```
workshop1@xs-host603-011:~/Downloads/L1_M3$ ./internsctl.sh user list --sudo-only
xenonadmin
```

# LINUX CUSTOM SCRIPT TASKS

**Section B:**

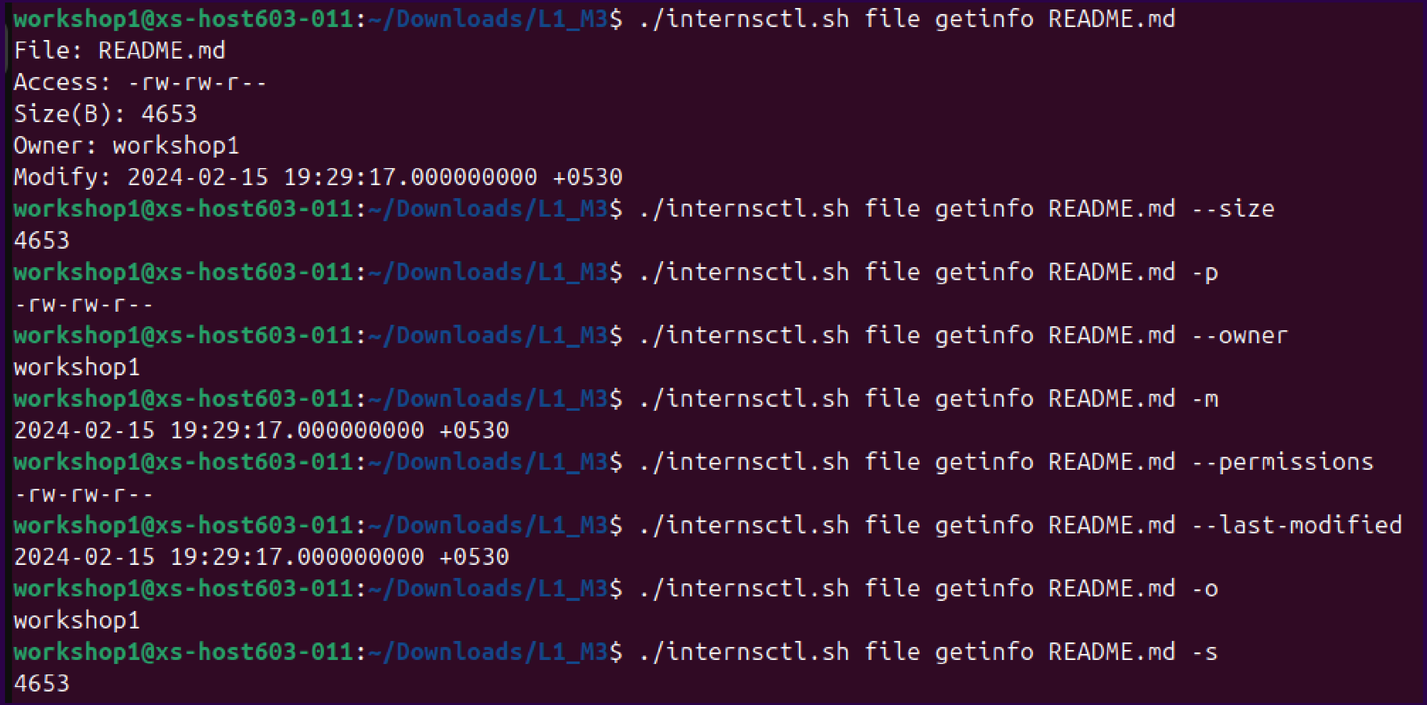
Part3 | Advanced Level

*$ internsctl file getinfo <file-name>*

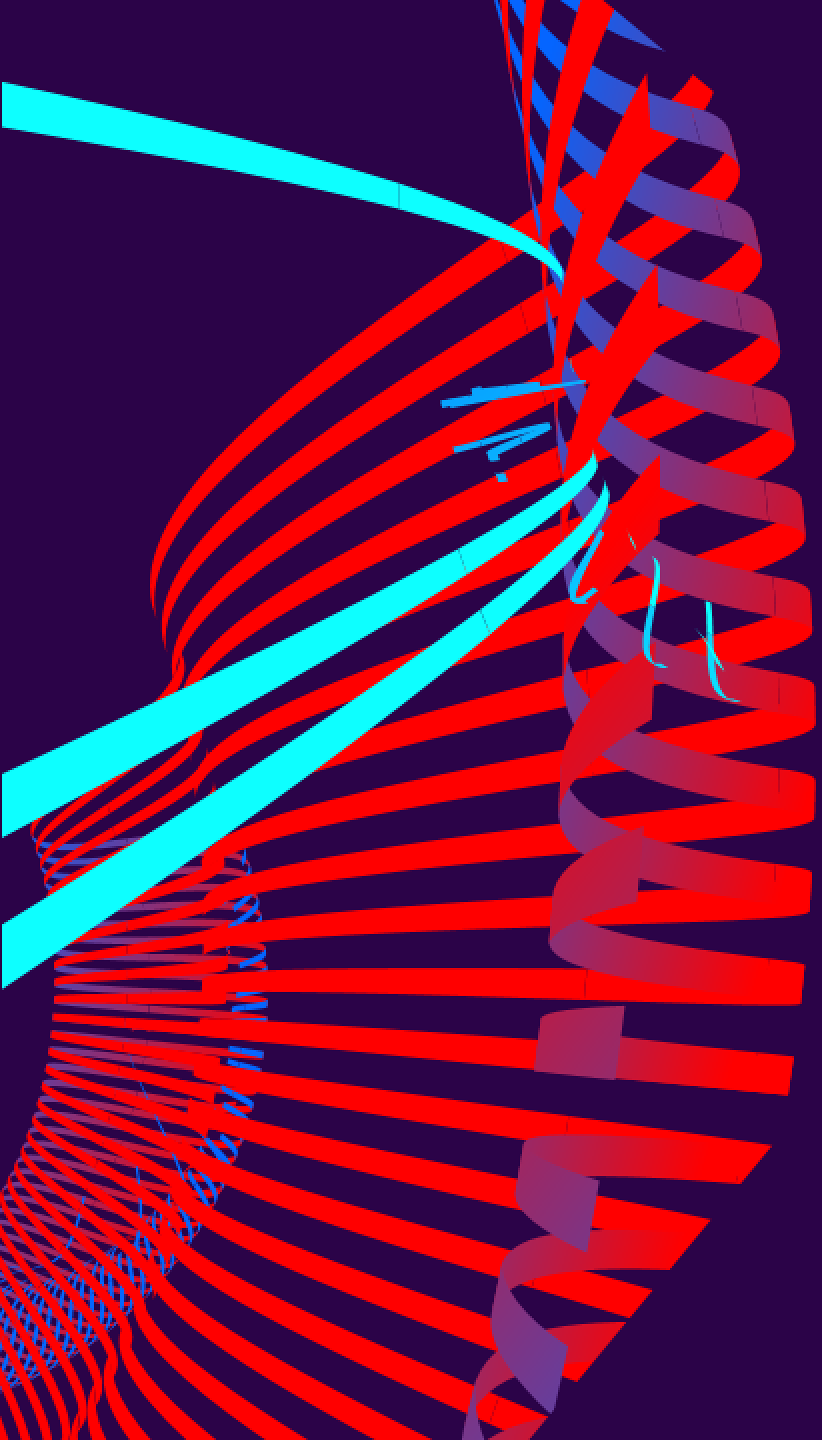
*$ internsctl file getinfo <file-name> [options]*

*[options]:*

*--size or –s*
*--permissions or –p*
*--owner or –o*
*--last-modified or -m*



```
workshop1@xs-host603-011:~/Downloads/L1_M3$ ./internsctl.sh file getinfo README.md
File: README.md
Access: -rw-rw-r--
Size(B): 4653
Owner: workshop1
Modify: 2024-02-15 19:29:17.000000000 +0530
workshop1@xs-host603-011:~/Downloads/L1_M3$ ./internsctl.sh file getinfo README.md --size
4653
workshop1@xs-host603-011:~/Downloads/L1_M3$ ./internsctl.sh file getinfo README.md -p
-rw-rw-r--
workshop1@xs-host603-011:~/Downloads/L1_M3$ ./internsctl.sh file getinfo README.md --owner
workshop1
workshop1@xs-host603-011:~/Downloads/L1_M3$ ./internsctl.sh file getinfo README.md -m
2024-02-15 19:29:17.000000000 +0530
workshop1@xs-host603-011:~/Downloads/L1_M3$ ./internsctl.sh file getinfo README.md --permissions
-rw-rw-r--
workshop1@xs-host603-011:~/Downloads/L1_M3$ ./internsctl.sh file getinfo README.md --last-modified
2024-02-15 19:29:17.000000000 +0530
workshop1@xs-host603-011:~/Downloads/L1_M3$ ./internsctl.sh file getinfo README.md -o
workshop1
workshop1@xs-host603-011:~/Downloads/L1_M3$ ./internsctl.sh file getinfo README.md -s
4653
```



# THANK YOU

ASSOCIATE SOFTWARE ENGINEER

HIMANSHU@XENONSTACK.COM