**Devops:**

**To start learning DevOps as a beginner, you need a structured approach covering foundational topics and practical tools. Below is a step-by-step guide:**

**1. Linux: Bash:**

* + File system: ls, cp, mv, rm
  + permissions: chmod, chown
  + processes: ps, top, kill
  + packages: apt, yum

**2. Networking concepts: Wireshark:**

* + OSI and TCP/IP models
  + IP addressing and subnetting
  + DNS and DHCP
  + HTTP, HTTPS,FTP, and SSH
  + Firewalls and security groups
  + ping, traceroute, and netstat

**3. GIT:**

* + Clone, commit, push, and pull
  + Branching and merging
  + Resolving merge conflicts
  + working with remote repositories

**4. Programming language: Python**

* Python syntax and data structure
* Modules and packages
* write and execute Python scripts
* work with files
* handle errors
* write automation scripts

**5. Cloud Providers: AWS:**

* Launch, configure, and mange virtual servers
* store and manage data
* manage users, groups, and roles
* setup and manage isolated networks

**6. Containerization: Docker:**

* Create Docker images
* starting, stopping and managing containers
* write Docker files
* Run multi-container apps using Docker compose

**7.CI/CD: Jenkins:**

* create and manage Jenkins pipelines
* write Jekinsfiles
* Integrate automate tests
* Automate the build process
* Automate the deployment

**8. Orchestration and management: Kubernetes:**

* overall architecture
* key components
* Manage resources
* scale application
* Networking model in Kubernetes

**9. Networking and Infrastructure as services: Nginx:**

* Configure Nginx as a reverse proxy
* configure it to act as a forward proxy
* caching strategies
* configure firewalls and security groups

**10. Configuration management: Ansible:**

* write Ansible playbooks
* Use roles and modules
* Manage variables and templates

**11. Infrastructure as code (IAC): Terraform:**

* Basic Concepts
* Terraform Configuration files
* Terraform modules
* Advanced concepts

**12. Monitoring and Logging: Prometheus and Grafana**

* Architecture and data model
* collect metrics
* write queries
* set up alerts

**Linux**

**Introduction:**

The beginning to 1969 where Ken Thompson and Dennis Ritchie of Bell Laboratories developed the UNIX operating system.  It was later rewritten in C to make it more portable and eventually became a widely used operating system. Then in 1991, a young fellow named Linus Torvalds started developing the Linux kernel.

**Linux Distribution**

1. **Debian**: Debian is an operating system composed entirely of free and open-source software. It’s widely known and has been in development for over 20 years. If you want a good "core" operating system, this is the one for you.
2. **Arch Linux**: Arch is a lightweight and flexible Linux distribution driven 100% by the community. Similar to Debian, Arch uses a rolling release model so incremental updates eventually become the Stable release. There’s a bit of a learning curve, but for the hardcore Linux users, this is a great choice.
3. **Red Hat Enterprise Linux**: Red Hat Enterprise Linux commonly referred to as RHEL is developed by Red Hat. RHEL has strict rules to restrict free re-distribution although it still provides source code for free. RHEL-based operating systems will differ slightly from the Debian-based operating systems, most noticeably in package management. If you decide to go with RHEL it’s probably best if you know you’ll be working with it.
4. **Fedora**: Backed by Red Hat, the Fedora Project is community driven containing open-source and free software. Red Hat Enterprise Linux branches off Fedora, so think of Fedora as an upstream RHEL operating system. Eventually RHEL will get updates from Fedora after thorough testing and quality assurance. Fedora is great if you want a Red Hat based operating system without the price tag. Recommended for desktop and laptop.
5. **Ubuntu**: One of the most popular Linux distributions for personal machines is Ubuntu. Ubuntu also releases its own desktop environment manager Unity by default. Great for any platform, desktop, laptop and server.
6. **Linux Mint**: Linux Mint is based off of Ubuntu. It uses Ubuntu’s software repositories so the same packages are available on both distributions. If you prefer a lighter distro than Ubuntu, you may be interested in Linux Mint. Great user interface, great for beginners and less bloated than Ubuntu.
7. **Gentoo**: Gentoo offers ridiculous flexibility with the operating system at a price. It’s made for advanced users who don’t mind getting their hands dirty with the system. If you’re just getting started with Linux and want to take a more difficult path
8. **OpenSUSE**: openSUSE Linux is created by the openSUSE Project. A community that promotes the use of Linux everywhere, working together in an open, transparent and friendly manner as part of the worldwide Free and Open-source Software community. openSUSE includes everything you need to enjoy the Internet worry free of viruses/spy-ware and to live out your creativity, be it with your photos, videos, music or code.

**Command Line**

**Shell**: The shell is basically a program that takes your commands from the keyboard and sends them to the operating system to perform. The shell program uses bash (Bourne Again shell), almost all Linux distributions will default to the bash shell. There are other shells available such as ksh, zsh, tsch, etc. $date; $whoami

1. **Pwd (print working directory):** Everything in Linux is a file. Every file is organized in a

hierarchical directory tree. The first directory in the filesystem is aptly named the root directory. The root directory has many folders and files which you can store more folders and files, etc. Ex:

/  
|-- bin  
| |-- file1  
| |-- file2  
|-- etc  
| |-- file3  
| `-- directory1  
| |-- file4  
| `-- file5  
|-- home  
|-- var

The pwd to see where you are, you can use the pwd command, this command means “print working directory” and it just shows you which directory you are in. Ex: $pwd

1. **Cd (Change Directory):** There are two different ways to specify a path, with absolute and

relative paths.

* **Absolute path**: This is the path from the root directory. The root directory is commonly shown as a slash.  Every time your path starts with / it means you are starting from the root directory
* **Relative path**: This is the path from where you are currently in filesystem

**Shortcuts for cd to help out:**

* . (current directory). This is the directory you are currently in. ex: cd .
* .. (parent directory). Takes you to the directory above your current. Ex: cd ..
* ~ (home directory). This directory defaults to your “home directory”. Such as /home. ex: cd ~ (or)
* Only if we type cd it will redirect to home directory. ex: cd
* - (previous directory). This will take you to the previous directory you were just at. Ex: cd -

1. **Ls(List Directories**): The ls command will list directories and files in the current directory.

**ls -a** : Filenames that start with . are hidden, you can view them

**ls -l** :  -l for long, this shows a detailed list of files in a long format. This will show you detailed information, starting from the left: file permissions, number of links, owner name, owner group, file size, timestamp of last modification, and file/directory name.

total 40

drwxr-xr-x 3 raji raji 4096 Oct 25 15:49 Desktop

drwxr-xr-x 2 raji raji 4096 Oct 16 19:20 Documents

ls -la or ls -al

ls -R: recursively list directory contents

ls -r: reverse order while sorting

ls -t: sort by modification time, newest first

1. **Touch:**Touch allows you to the create new empty files.