Evaluation Warning: The document was created with Spire.Doc for Python.

Here’s a comprehensive **DevOps learning plan** based on **Linux, Jenkins, Ansible, Python, and Shell scripting**. This plan will be broken down into phases to progressively build your skills. The estimated completion time is 3-4 months (depending on prior knowledge and pace).

## **Phase 1: Linux Fundamentals (2-3 weeks)**

**Objective**: Gain a strong foundation in Linux as it's the backbone of most DevOps processes.

### Key Learning Areas:

* **Basic Commands**:
  + File and directory manipulation (ls, cp, mv, rm, etc.)
  + Viewing and editing files (cat, nano, vi, grep, etc.)
  + Process management (ps, top, kill)
  + Permissions and ownership (chmod, chown)
* **System Administration**:
  + Users and groups management
  + Package management (apt, yum, rpm)
  + Networking basics (IP configuration, ping, netstat, curl)
* **Filesystems and Disk Management**:
  + Mounting and unmounting drives
  + Disk partitioning and formatting (fdisk, mkfs)
* **Networking and Security**:
  + SSH setup and key management
  + Firewall basics (ufw, iptables)

### Hands-on Practice:

* Set up a **Linux VM** (using VirtualBox or Docker).
* Perform regular tasks like creating users, modifying file permissions, and configuring basic networking.

### Resources:

* [Linux Command Line Tutorial](https://linuxcommand.org)
* [Linux Journey](https://linuxjourney.com/)

## **Phase 2: Shell Scripting (1-2 weeks)**

**Objective**: Automate repetitive tasks and build simple workflows using shell scripts.

### Key Learning Areas:

* **Basic Scripting**:
  + Writing simple shell scripts
  + Using variables and environment variables
  + Control structures (if-else, loops)
* **Automation**:
  + Writing scripts to automate common system tasks
  + Scheduling tasks using cron
* **Advanced Scripting**:
  + Functions and debugging scripts
  + Handling errors and logging

### Hands-on Practice:

* Create a script that automates server setup (e.g., install packages, create users, configure firewall).
* Write scripts to back up data or log files from a server.

### Resources:

* [Bash Scripting Tutorial](https://ryanstutorials.net/bash-scripting-tutorial/)
* [Shell Scripting Crash Course](https://www.udemy.com/course/linux-shell-scripting-projects/)

## **Phase 3: Python for DevOps (2-3 weeks)**

**Objective**: Build automation scripts and applications in Python.

### Key Learning Areas:

* **Python Basics**:
  + Variables, data types, and operators
  + Control structures and loops
  + Functions, modules, and exception handling
* **File I/O and OS module**:
  + Reading from and writing to files
  + Using the os and subprocess modules for system interactions
* **Working with APIs**:
  + Using requests to interact with REST APIs
  + Automating data retrieval from APIs

### Hands-on Practice:

* Write a Python script to automate server monitoring (CPU, memory, disk usage).
* Use Python to interact with a cloud provider’s API (AWS, Azure, etc.).

### Resources:

* [Python for DevOps](https://www.oreilly.com/library/view/python-for-devops/9781492057685/)
* [Automate the Boring Stuff with Python](https://automatetheboringstuff.com/)

## **Phase 4: Jenkins for Continuous Integration/Continuous Deployment (CI/CD) (2-3 weeks)**

**Objective**: Learn how to automate build, test, and deployment pipelines using Jenkins.

### Key Learning Areas:

* **Introduction to Jenkins**:
  + Jenkins installation and setup
  + Overview of jobs, pipelines, and the Jenkins architecture
* **Creating CI/CD Pipelines**:
  + Freestyle jobs vs. pipeline jobs
  + Using Jenkins pipelines (Declarative and Scripted pipelines)
  + Automating builds and deployments
* **Integration with Version Control Systems**:
  + Setting up Jenkins with Git for automated builds
  + Triggering jobs from GitHub/GitLab
* **Plugins and Integrations**:
  + Configuring Jenkins plugins for extended functionality (e.g., Slack notifications)
  + Integrating Jenkins with Docker for containerized builds

### Hands-on Practice:

* Set up a Jenkins server on a Linux VM.
* Build a simple CI pipeline to pull code from GitHub, run tests, and deploy to a staging environment.

### Resources:

* [Jenkins Documentation](https://www.jenkins.io/doc/)
* [Jenkins CI/CD Pipeline Tutorials](https://www.udemy.com/course/jenkins-pipelines-step-by-step/)

## **Phase 5: Ansible for Configuration Management and Automation (2-3 weeks)**

**Objective**: Automate infrastructure provisioning, configuration, and orchestration using Ansible.

### Key Learning Areas:

* **Ansible Basics**:
  + Introduction to Infrastructure as Code (IaC)
  + Installing and configuring Ansible
  + Writing Playbooks and understanding YAML syntax
* **Inventory and Modules**:
  + Creating static and dynamic inventories
  + Using Ansible modules to perform tasks (package installation, service management, etc.)
* **Roles and Best Practices**:
  + Structuring playbooks with roles for reusability
  + Ansible Galaxy and community roles
* **Ansible Vault**:
  + Securing sensitive data using Ansible Vault
* **Advanced Ansible**:
  + Writing custom modules
  + Creating playbooks for cloud infrastructure (e.g., AWS, Azure)

### Hands-on Practice:

* Create Ansible playbooks to configure multiple Linux servers (e.g., set up web servers, database servers).
* Use Ansible to deploy a sample application to a cluster of VMs.

### Resources:

* [Ansible Documentation](https://docs.ansible.com/ansible/latest/index.html)
* [Ansible for DevOps](https://www.ansiblefordevops.com/)

## **Phase 6: DevOps Projects (4-6 weeks)**

**Objective**: Combine your knowledge of Linux, Jenkins, Ansible, Python, and shell scripting into real-world projects.

### Projects:

1. **Automated Server Setup**:
   * Use Ansible and Shell scripting to automate the provisioning of servers (install software, configure services).
2. **CI/CD Pipeline**:
   * Set up a Jenkins pipeline to automate building, testing, and deploying a Python or Java web application.
3. **Infrastructure Monitoring**:
   * Build a monitoring solution using Python (or a pre-built tool like Nagios) that sends alerts when servers/resources exceed certain thresholds.
4. **Multi-cloud Deployment**:
   * Write Ansible playbooks that can provision infrastructure on multiple cloud platforms (AWS, Azure) and deploy an application.

### Additional Resources:

* **Online Courses**:
  + [Linux Academy](https://linuxacademy.com/) (now part of A Cloud Guru)
  + [Udemy](https://www.udemy.com/) has multiple courses on Jenkins, Ansible, and Python for DevOps.
* **Books**:
  + *"The Phoenix Project"* by Gene Kim, Kevin Behr, and George Spafford (for DevOps culture and practices)
  + *"The DevOps Handbook"* by Gene Kim, Patrick Debois, and others

## **Outcome**:

By the end of this plan, you will have the skills to:

* Manage Linux systems efficiently.
* Write automation scripts in Shell and Python.
* Set up and manage Jenkins for CI/CD.
* Automate infrastructure configuration with Ansible.
* Build real-world DevOps pipelines and systems.

Good luck on your DevOps journey!