

DevOps Learning Path: Beginner to Pro

Step 1: Understand What DevOps Is

- DevOps = Development + Operations
- Goal: Make software faster and better by working closely between developers and system admins.
- Key idea: Automate everything, so code moves fast and safely from developer to user.

Step 2: Learn Basic Linux Skills

- Most DevOps work is done on Linux servers.
- Learn these Linux commands and concepts:
 - File commands: ls, cd, mkdir, rm
 - File permissions: chmod, chown
 - Process commands: ps, top, kill
 - Network commands: netstat, ping
 - Package managers: yum, apt
 - Editing files: vim, nano
- Practice by using a Linux system or install Linux on a virtual machine (like VirtualBox).

Step 3: Learn Shell Scripting (Bash)

- Automate simple tasks with shell scripts.
- Learn variables, loops, conditions in Bash.

Example:

```
```bash
```

```
#!/bin/bash
```

```
for i in {1..5}
```

```
do
```

## DevOps Learning Path: Beginner to Pro

```
echo "Number $i"
```

```
done
```

```
...
```

- This will print numbers 1 to 5.

### Step 4: Learn Git (Version Control)

- DevOps needs Git for code version control.
- Learn:
  - git init, git add, git commit
  - git push and git pull
  - Branches: git branch, git checkout, git merge
- Use GitHub to store your code online.

### Step 5: Learn a Programming Language (Python)

- Python is simple and used a lot in DevOps for automation.
- Learn basics: variables, functions, loops, file handling.

Example Python code to read a file:

```
```python
```

```
with open('file.txt', 'r') as file:
```

```
    content = file.read()
```

```
    print(content)
```

```
```
```

### Step 6: Learn CI/CD Tools (Jenkins, GitHub Actions)

## **DevOps Learning Path: Beginner to Pro**

- Continuous Integration and Continuous Deployment help automatic testing and deployment.
- Learn Jenkins basics:
  - Install Jenkins on Linux.
  - Create a simple job to run a script.
- Learn GitHub Actions to automate builds in GitHub repos.

### **Step 7: Learn Containerization (Docker)**

- Docker helps to package apps with everything they need.
- Learn commands:
  - docker run, docker build, docker images, docker ps
- Make a Dockerfile and build an image.

### **Step 8: Learn Orchestration (Kubernetes)**

- Kubernetes helps manage many Docker containers.
- Learn basics:
  - Pods, Services, Deployments
  - Commands: kubectl get pods, kubectl apply -f deployment.yaml
- Use Minikube to run Kubernetes locally.

### **Step 9: Learn Cloud Platforms (AWS, Azure, GCP)**

- Most companies use cloud servers.
- Start with AWS Free Tier.
- Learn:
  - EC2 (virtual servers)

## DevOps Learning Path: Beginner to Pro

- S3 (storage)
- IAM (security users)
- Do simple projects like hosting a website on EC2.

### Step 10: Learn Infrastructure as Code (Terraform, Ansible)

- Terraform: write code to create servers and resources.
- Ansible: automate configuring servers.

Example Terraform:

```
``hcl

provider "aws" {

 region = "us-east-1"

}

resource "aws_instance" "example" {

 ami = "ami-123456"

 instance_type = "t2.micro"

}

...
```

- Learn basics, run commands, and create small projects.

### Step 11: Practice Projects and Build Portfolio

- Build small projects combining what you learned:
  - A website in Docker with CI/CD pipeline.
  - Automated server setup with Ansible.

## **DevOps Learning Path: Beginner to Pro**

- Kubernetes app deployment.
- Push your code on GitHub.

### **Step 12: Prepare for Interviews**

- Learn common DevOps interview questions:
  - What is CI/CD?
  - How does Docker work?
  - Explain Infrastructure as Code.
  - Basic Linux troubleshooting.
- Practice explaining your projects and scripts.

### **Step 13: Bonus Tips**

- Use free learning sites: YouTube, freeCodeCamp, Linux Journey, AWS Free Tier.
- Join DevOps communities on Reddit, Discord.
- Keep practicing daily, even 1 hour a day helps.
- Try to get an internship or small freelance jobs for experience.