**TERMINOLOGIES AND FUNDAMENTALS**

**1. Linux and Scripting**

* **Linux Fundamentals**: File system, permissions, networking, process management.
* **Shell Scripting**: Bash (essential), plus maybe some Python for automation.

**2. Version Control**

* **Git**: Branching, merging, rebasing, pull requests, and collaboration via GitHub or GitLab.

**3. Continuous Integration/Continuous Deployment (CI/CD)**

* **Concepts**: Pipelines, build automation, test automation, deployment strategies.
* **Tools**: Jenkins, GitHub Actions, GitLab CI, CircleCI.

**4. Configuration Management & Infrastructure as Code**

* **Tools**:
  + Ansible (agentless configuration)
  + Terraform (infrastructure provisioning)
* **Concepts**: Idempotency, declarative vs imperative approach.

**5. Containers & Orchestration**

* **Docker**: Build, run, network, and manage containers.
* **Kubernetes**: Pods, services, deployments, scaling, and Helm charts.

**6. Cloud Platforms**

* **Learn One Cloud Provider Well**:
  + **AWS** (most popular in industry)
  + Others: Azure, Google Cloud (GCP)
* **Core Services**: EC2, S3, IAM, VPC, RDS, CloudWatch, Lambda.

**7. Monitoring & Logging**

* **Concepts**: Observability, metrics, logs, traces.
* **Tools**: Prometheus, Grafana, ELK stack, Datadog, New Relic.

**8. Networking Basics**

* **Topics**: DNS, HTTP/HTTPS, TCP/IP, firewalls, VPNs, load balancers.

**9. Security Fundamentals**

* IAM (Identity and Access Management)
* Secrets management (e.g., HashiCorp Vault)
* Secure CI/CD practices

**10. Soft Skills & Collaboration**

* Agile/Scrum basics
* Communication with developers, testers, and operations
* Documentation and knowledge sharing

### HOW TO PROPERLY PROMTS AS DEV OPS ENG.. ✅ 1. ****Linux & Scripting****

**Bad prompt**:

How do I fix a permission issue?

**Good prompt**:

I'm trying to run a Bash script on Ubuntu 22.04 and it fails with Permission denied. I’ve used chmod +x script.sh but still can’t execute it. The script is in /usr/local/bin. What could be causing this?

### ✅ 2. ****Git & Version Control****

**Bad prompt**:

Git merge isn’t working.

**Good prompt**:

I'm trying to merge a feature branch into main, but I'm getting merge conflicts in config.yml. What’s the best way to resolve these conflicts and keep the version from the feature branch?

### ✅ 3. ****CI/CD (Jenkins, GitHub Actions, etc.)****

**Bad prompt**:

My pipeline is failing. Help?

**Good prompt**:

I'm using GitHub Actions to deploy a Node.js app. The workflow fails at the npm install step with EACCES error. I tried adding --unsafe-perm but it didn’t work. Any idea how to fix this in the GitHub runner?

### ✅ 4. ****Terraform / Infrastructure as Code****

**Bad prompt**:

Terraform apply isn’t working.

**Good prompt**:

I’m using Terraform to create an AWS EC2 instance. When I run terraform apply, I get an error: InvalidAMIID.NotFound. I double-checked the AMI ID for my region. Is there a better way to reference the latest AMI dynamically?

### ✅ 5. ****Docker****

**Bad prompt**:

My container is not working.

**Good prompt**:

I'm running a Docker container with a Python app using docker run -p 5000:5000 myapp, but when I go to localhost:5000 it doesn’t connect. The app runs on port 5000 inside the container. How can I debug this?

### ✅ 6. ****Kubernetes****

**Bad prompt**:

My pod won't start.

**Good prompt**:

I deployed a pod using a YAML file, but it’s stuck in CrashLoopBackOff. When I run kubectl logs <pod>, I see Connection refused on port 80. The container uses NGINX. How do I troubleshoot this startup failure?

### ✅ 7. ****Cloud (AWS Example)****

**Bad prompt**:

EC2 isn’t connecting.

**Good prompt**:

I launched an EC2 instance with a security group allowing port 22 from my IP. When I try to SSH, I get Connection timed out. I checked my IP and it matches. Could this be a VPC or routing table issue?

Git is for: Version Control in 2 words.

Git from Remote to local Git Clone.

Git hub website like Facebook.  
  
  
  
Git pushing from local to Remote using git commands\*\*\*\*\*\*\*

0. git pull - para ma-update yung main branch(assuming na nasa main branch).

1. git branch -c feature/test-study origin/main - creating new branch for new projects or file path and para maiwasan magupload sa ibang branch.

2. git switch branchName

3. git branch - to check the existing branch or if the new branch are already in the system.

4. git status - to check if there are changes in branch.

5. echo. > filename.txt - since there are no changes, "hard changes" is applied. (not best practice).

6. git status - to check if again if there are changes in the file

7. git add filename.txt - para nasave yung changes sa specific file.

8. git commit -m "Message be specific and reasonable"

9. git push origin HEAD