YAMLForge Project Progress Summary

■ Project Goal Build a mini Infrastructure-as-Code lab that demonstrates YAML's power through Ansible automation — covering network segmentation, service deployment, and security hardening — in a 3-zone architecture (DMZ, Internal, Secure).

■ So Far Completed

1. Project Setup Created complete folder structure: YAMLForge/ ■■■ inventories/lab.yml ■■■ group_vars/all.yml ■■■ playbooks/ ■ ■■■ 01-setup-network.yml ■ ■■■ 02-deploy-servers.yml ■ ■■■ (next → 03-security-hardening.yml) ■■■ roles/ ■■■ templates/

Defined the inventory (lab.yml) with hosts for: - Firewall / Switch - Web Server (192.168.10.10) - App Server (192.168.20.10) - DB Server (192.168.30.10)

Defined common variables (group_vars/all.yml) for VLANs and firewall rules.

2. Network Simulation Ran and verified: ansible-playbook -i inventories/lab.yml playbooks/01-setup-network.yml Output: VLANs and firewall rules displayed successfully. Demonstrated YAML-driven configuration parsing.

3. Web Server Automation Ran: ansible-playbook -i inventories/lab.yml playbooks/02-deploy-servers.yml Result: - Apache installed - Custom index page deployed - Apache restarted automatically Demonstrated end-to-end automation and handlers.

Concepts Already Demonstrated YAML syntax & hierarchy — lab.yml, all.yml Infrastructure as Code — Ansible playbooks Declarative automation — Tasks, handlers Multi-zone architecture — DMZ/Internal/Secure Service provisioning — Apache deployment Idempotence — Re-running playbooks safely

■ Next Step: Phase 4 – Security Hardening

Now we'll move into the security automation layer. This step will: - Apply SSH hardening across all hosts - Configure UFW firewall rules - Create secure admin user - Disable password & root login - Validate security posture