Assessment Set 16

Important Instructions (Read Carefully)

- This assessment consists of **4 questions**:
  - **Q1 & Q2:** SDN and Network Automation (using **Mininet and Ryu Controller**)
  - **Q3 & Q4:** DevOps (Git, GitHub, Jenkins, Terraform, Ansible, Prometheus, Grafana)
- **Participants must complete at least ONE question from SDN/Network Automation (Q1 or Q2) AND ONE question from DevOps (Q3 or Q4).**
- **Participants may be asked to present one or both completed tasks during the assessment.**
- All tasks must be **demonstrated live** with clear explanation of design decisions and outcomes.

Q1. SDN – Traffic Prioritization Using Ryu (QoS-Based Flows)

Task:
Implement traffic prioritization in an SDN environment using **Mininet and Ryu**, based on traffic type or source host.

Requirements:

- Create a Mininet topology with:
  - At least **3 hosts and 1 OpenFlow switch**
- Use Ryu controller to:
  - Identify traffic based on **source IP or protocol**
  - Install flow rules that assign **different priorities**
- Ensure:
  - High-priority traffic is matched and forwarded first

Live demonstration must include:

- Mininet topology
- Ryu controller logic for priority-based flows
- Flow table showing different priorities
- Traffic verification using `ping` / `iperf`
- Explanation of **QoS concepts in SDN**

Q2. Network Automation – Automated Remediation Using Ansible

Task:
Use **Ansible** to automatically detect and remediate a network issue in a Mininet environment.

Requirements:

- Create a playbook to:
  - Detect failed connectivity between two hosts
  - Trigger corrective action (restart service, reconfigure link, or re-run Mininet command)
- Use:
  - Conditional logic (`when`)
  - Registered variables

Live demonstration must include:

- Playbook logic walkthrough
- Failure simulation
- Automated remediation execution
- Verification of restored connectivity
- Explanation of **closed-loop automation**

Q3. DevOps – Jenkins Pipeline for Infrastructure as Code Validation

Task:
Create a Jenkins pipeline to validate Infrastructure as Code (IaC).

Requirements:

- Pipeline must:
  - Pull code from a Git repository
  - Run:
    - `terraform validate` **or**
    - `ansible-lint` / syntax check
- Pipeline must fail on validation errors

Live demonstration must include:

- Jenkinsfile

- Pipeline execution (success and failure scenarios)
- Console output explanation
- Explanation of **shift-left validation**

Q4. DevOps – Alerting with Prometheus and Grafana

Task:
Configure alerting for infrastructure metrics using **Prometheus and Grafana**.

Requirements:

- Define an alert rule for:
  - High CPU usage **or**
  - Low available memory
- Configure Grafana to:
  - Display alert status on a dashboard

Live demonstration must include:

- Prometheus alert rules file
- Active alerts page
- Grafana dashboard with alert indicators
- Explanation of **proactive monitoring and alerting**