**System Administration Bootcamp: Managing Server and Optimizing IT Infrastructure**

**Day 1: User Management, System Configuration, and Security for Lab Usage**

**Session 1: Introduction to System Administration & User Management (Linux & Windows)**  
**Duration:** 2 hours

**Linux User Management:**

* **Creating Users in Linux:**
  + Introduction to user management commands like useradd, passwd, and usermod.
  + Assigning default directories, setting passwords, and specifying user permissions.
* **Deleting Users in Linux:**
* **Changing Ownership in Linux:**

**Hands-on:**

* Create a user for a student, assign a password, and change ownership of files to the user.

**Windows User Management:**

* **Creating Users in Windows:**
  + How to create user accounts in Windows through Control Panel or PowerShell.
  + **Using PowerShell** to automate account creation:
    - New-LocalUser -Name "student1" -Password (ConvertTo-SecureString "password" -AsPlainText -Force)
  + Adding users to groups: Add-LocalGroupMember -Group "Users" -Member "student1"
* **Deleting Users in Windows:**
  + How to delete users through Control Panel or PowerShell.
    - PowerShell command: Remove-LocalUser -Name "student1"
* **Changing Ownership and Permissions in Windows:**
  + How to modify file/folder ownership and permissions through GUI and command line.
  + **Command Example:**
    - takeown /f "file\_path" /r /d Y
    - icacls "file\_path" /grant student1:(F)
* **Hands-on:**
  + Create a user in Windows and assign proper file/folder permissions to that user.

**Session 2: Managing Common Storage for Students & Security Configuration**  
**Duration:** 2 hours

**Intranet-Based File Storage:**

* **Overview:**
  + How to set up shared folders on the **local network (Intranet)**, accessible by all students with unique logins.
  + **Linux:** Configure **Samba** for file sharing on the intranet.
    - **Basic Samba Configuration:**

**Windows:** Setting up shared folders using Windows File Sharing.

* + Example: Right-click on a folder > Properties > Sharing > Share with specific people.
  + Assign unique logins to students and grant them access to the shared folder.

 **Hands-on:**

* Set up a shared directory accessible to student users on both Linux (Samba) and Windows.

**Security for Exam Settings:**

* **Setting Up Security for Exams:**
  + **Linux:**
    - Limit access to the shared folder and restrict write permissions during exams.
    - Use **AppArmor** or **SELinux** for enhanced security.
    - Lock down student accounts and prevent system modifications during exam times.
  + **Windows:**
    - Use **Group Policy** to restrict user access to critical system settings and resources.
    - Set up user restrictions for the exam environment (e.g., prevent launching external applications).
* **Intranet Usage for Exam Storage:**
  + Demonstrate how student data will be securely stored on the shared network, with proper access control during exams.
  + **Security Measures:**
    - Encryption of exam files and folders.
    - Limiting the ability to copy or modify files during exam periods.
* **Hands-on:**
  + Demonstrate setting file/folder permissions during exam time using both Linux and Windows.

**Session 3: Lab Setup and Troubleshooting Common Issues**  
**Duration:** 2 hours

* **Managing Multiple Users for Lab Usage:**
  + How to set up unique user accounts for students in computer labs.
  + Scheduling tasks with **cron** on Linux to backup student data at regular intervals.
  + **Windows Task Scheduler** for automated tasks.
  + Managing user quotas on a shared network storage system.
* **Troubleshooting:**
  + Common issues with user permissions and login failures.
  + How to resolve access issues and permissions errors for both Linux and Windows.
  + **Logs to check:**
    - Linux: /var/log/auth.log
    - Windows: Event Viewer logs.
* **Hands-on:**
  + Troubleshooting user login issues, file access issues, and restoring user data.

**Afternoon Session: Networking Basics**

* **IP Addressing, Subnetting, and DNS Configuration**
  + Understanding IPs, subnet masks, and their roles in a network.
  + Setting up DHCP and DNS servers for local networking.
* **Hands-On Lab:**
  + Setting up a simple local network using multiple machines.
  + Troubleshooting common network issues.

**Day 2: Advanced Server Configuration and Virtualization**

**Morning Session: Installing and Configuring Servers**

* **Web Server Setup (Apache/Nginx)**
  + Installation and configuration for serving educational content (e.g., websites, files).
  + Practical exercises in setting up virtual hosts and managing server logs.
* **Database Server Installation (MySQL/PostgreSQL)**
  + Installing and configuring database servers for student projects.
  + Creating and managing user accounts and databases.

**Afternoon Session: Virtualization and Containerization**

* **Virtualization with VirtualBox/VMware**
  + Installing and configuring virtual machines for running isolated environments.
  + Using snapshots and cloning for backup and disaster recovery.
* **Introduction to Docker: Containerization**
  + Setting up Docker for isolated software environments.
  + Hands-on practice: Deploying a simple web application in a Docker container.

**Day 3: JupyterHub User Management, GPU Integration, and Efficient Software Management**

**Morning Session: JupyterHub Advanced Configuration**

1. **Introduction to JupyterHub**
   * Multi-user setup for accessing Jupyter notebooks.
   * Practical applications in teaching and research (e.g., programming courses, data science).
2. **Creating and Managing Users in JupyterHub**
   * **Adding Users**
     + Manual user addition via Linux commands
     + Bulk user creation using a script:
   * **Removing Users**
     + Deleting users and their home directories:
3. **GPU Integration in JupyterHub**

**Enabling GPU Access for JupyterHub**

* Install NVIDIA drivers and nvidia-docker:

**Afternoon Session: Efficient Software Installation and Management**

1. **Smart Techniques for Installing and Removing Software**
   * **Package Management (Linux)**
2. **Using Snap and Flatpak for Isolation**
   * **Isolate software dependencies using Snap and Flatpak:**
3. **Docker for Efficient Software Management**
4. **Best Practices in Software Management**