Here's a **Hands-On Lab** for setting up a GitLab Server, creating a group and sample project, and adding users with different roles. This is ideal for a beginner-level session or training lab exercise.

**🧪 Hands-On Lab: GitLab Server – Group, Project, and Role Management**

**🧰 Prerequisites:**

* A working **GitLab Server** (Self-managed or GitLab.com).
* Admin/root access to GitLab or access to create groups and add members.
* A few test users (can be emails of teammates or dummy accounts).

**🎯 Objectives:**

1. Create a **Group**
2. Create a **Project** inside the group
3. Add members with different **Access Roles** (Guest, Reporter, Developer, Maintainer, Owner)
4. Verify access behavior for each role (optional if time permits)

**✅ Step-by-Step Instructions**

**🔹 Step 1: Login to GitLab**

* URL: http://<your-gitlab-server> or https://gitlab.com
* Log in with your credentials.

**🔹 Step 2: Create a Group**

1. Go to the **Menu** → Groups → Your Groups.
2. Click on **New Group**.
3. Choose **Create group**:
   * **Group name**: demo-group
   * **Group URL**: auto-fills
   * **Visibility Level**: Private or Internal (as required)
4. Click **Create group**.

**🔹 Step 3: Create a Project Inside the Group**

1. Navigate to your new group: Groups → demo-group.
2. Click on **New Project**.
3. Choose:
   * **Create blank project**
   * **Project name**: sample-project
   * **Project slug**: auto-fills
   * **Visibility Level**: Private
4. Click **Create project**.

**🔹 Step 4: Add Members to the Group**

1. Inside the demo-group, go to **Group Settings** → Members.
2. Click **Invite members**.
3. Enter the **email or username** of the user to invite.
4. Assign a **Role**:
   * **Guest** – Read-only access
   * **Reporter** – Can view and generate reports
   * **Developer** – Can push code, create branches
   * **Maintainer** – Can manage issues, pipelines, and settings
   * **Owner** – Full control (only at group level)
5. Click **Invite**.

📝 Repeat to add at least 3–4 members with different roles.

**🔹 Step 5: Verify Role-Based Access (Optional but Useful)**

Login as each invited user and try:

* Viewing the project
* Creating an issue
* Pushing code (for Developer+)
* Accessing settings (Maintainer+)

Use this table as reference:

| **Action** | **Guest** | **Reporter** | **Developer** | **Maintainer** | **Owner** |
| --- | --- | --- | --- | --- | --- |
| View project | ✅ | ✅ | ✅ | ✅ | ✅ |
| View issues | ✅ | ✅ | ✅ | ✅ | ✅ |
| Create/edit issues | ❌ | ✅ | ✅ | ✅ | ✅ |
| Push code | ❌ | ❌ | ✅ | ✅ | ✅ |
| Manage CI/CD | ❌ | ❌ | ✅ | ✅ | ✅ |
| Project settings | ❌ | ❌ | ❌ | ✅ | ✅ |
| Add/remove members | ❌ | ❌ | ❌ | ❌ | ✅ |

**🧼 Cleanup (Optional)**

* Remove test users from the group.
* Archive or delete the sample project and group.

**📝 Notes for Trainer or Facilitator:**

* Pre-create user accounts or simulate with multiple browser sessions.
* Discuss common use cases for each role.
* Ask learners to modify project files and observe permission enforcement.

Would you like a **PDF** or **PowerPoint version** of this lab?