**Step-by-Step Guide: Day 1 Labs**

**Lab 1: Installing Git and Setting Up a Repository**

**Step 1: Install Git**

1. Download Git from the official website: <https://git-scm.com/downloads>
2. Run the installer and follow the default setup instructions.
3. Verify the installation by running the following command in the terminal:

git --version

**Step 2: Configure Git**

1. Set up your Git username and email:
2. git config --global user.name "Your Name"

git config --global user.email "your.email@example.com"

1. Verify your configuration:

git config --list

**Step 3: Create a Git Repository**

1. Navigate to a desired directory:

cd path/to/your/directory

1. Initialize a new Git repository:

git init

1. Create a sample file and add it to Git:
2. echo "# DevOps Project" > README.md
3. git add README.md

git commit -m "Initial commit"

**Lab 2: Working with Git - Branching and Code Reviews**

**Step 1: Create a New Branch**

1. Check the current branch:

git branch

1. Create and switch to a new branch:

git checkout -b feature-branch

**Step 2: Modify a File and Commit Changes**

1. Edit the README.md file and save changes.
2. Add and commit changes:
3. git add README.md

git commit -m "Updated README with project details"

**Step 3: Push the Branch to Remote Repository**

1. Connect to a remote repository (e.g., Azure DevOps):

git remote add origin https://dev.azure.com/your-organization/your-project/\_git/repository-name

1. Push the branch:

git push -u origin feature-branch

**Step 4: Create a Pull Request (PR) in Azure DevOps**

1. Navigate to Azure DevOps and open your repository.
2. Click **Branches** and find feature-branch.
3. Click **New Pull Request**.
4. Assign reviewers and add comments.
5. Click **Create** and wait for approvals.
6. Once approved, merge the PR into the main branch.

**Lab 3: Setting Up a CI/CD Pipeline Using Azure DevOps**

**Step 1: Create an Azure DevOps Project**

1. Go to [Azure DevOps](https://dev.azure.com/)
2. Click **Create Project**, provide a name, and choose **Private** visibility.
3. Click **Create**.

**Step 2: Create a CI/CD Pipeline**

1. Navigate to **Pipelines** > **New Pipeline**.
2. Select **Azure Repos Git** (or GitHub if using GitHub integration).
3. Choose **Starter Pipeline**.
4. Replace the default YAML with the following basic configuration:
5. trigger:
6. branches:
7. include:
8. - main
9. pool:
10. vmImage: 'ubuntu-latest'
11. steps:
12. - script: echo "Hello, DevOps!"

displayName: 'Run a sample script'

1. Click **Save and Run**.

**Step 3: Add Automated Testing**

1. Modify the YAML pipeline to include a test step:
2. steps:
3. - script: echo "Running tests..."

displayName: 'Run Tests'

1. Save and commit changes.

**Step 4: Deploy to a Test Environment**

1. Click **Releases** > **New Release Pipeline**.
2. Choose **Azure App Service Deployment**.
3. Select your application and configure the deployment stage.
4. Click **Save**, then **Create Release**.