**Step-by-Step Guide to Setting Up Git in VDI**

**Table of Contents:**

1. **Introduction**
2. **Prerequisites**
3. **Installation Steps**
4. **Configuration**
5. **Generating SSH Keys**
6. **Cloning a Repository**
7. **Adding Git Authentication Code to Windows Generic**
8. **Creating a Key on Repository and Setting Its Duration**
9. **Troubleshooting**
10. **Conclusion**

**1. Introduction  
This guide provides a comprehensive, step-by-step process for setting up Git in a Virtual Desktop Infrastructure (VDI) environment. Whether you're a developer, system administrator, or student, this guide will help you install, configure, and use Git efficiently within your VDI.**

**2. Prerequisites  
Before you begin, ensure the following:**

* **VDI Access: Credentials to log into your VDI environment.**
* **Permissions: Administrative privileges may be required to install software.**
* **Internet Connection: Required for downloading Git and cloning repositories.**
* **GitHub/GitLab Account: For repository access (optional but recommended).**

**3. Installation Steps**

**Step 1: Log into the VDI.  
Open your VDI client and enter your credentials to access the virtual desktop environment.**

**Step 2: Open Terminal (Linux/Mac) or Command Prompt (Windows).**

**Step 3: Install Git.**

* **For Linux (Ubuntu/Debian):**

**sudo apt update**

**sudo apt install git**

* **For Windows:  
  Download Git for Windows from** [**https://git-scm.com/download/win**](https://git-scm.com/download/win) **and follow the installation wizard.**

**Step 4: Verify the Installation.**

**git --version**

**If Git is installed successfully, this command will display the installed version.**

**4. Configuration  
After installing Git, configure your user details. Currently, the account** [**tec\_ppadmin6@abc.com**](mailto:tec_ppadmin6@abc.com) **has Git repository access, so we will be setting this account globally:**

**git config --global user.name "tec\_ppadmin6"**

**git config --global user.email "tec\_ppadmin6@abc.com"**

**Verify the configuration:**

**git config --list**

**This will list all the configuration settings, confirming that your details are saved.**

**5. Generating SSH Keys  
For secure communication with remote repositories:**

**Step 1: Generate an SSH key:**

**ssh-keygen -t rsa -b 4096 -C "tec\_ppadmin6@abc.com"**

**Press Enter to accept the default file location and set a passphrase if desired.**

**Step 2: Add the SSH key to your GitHub/GitLab account.  
Copy the SSH key:**

**cat ~/.ssh/id\_rsa.pub**

**Then paste it into the SSH key section of your GitHub/GitLab account settings.**

**6. Cloning a Repository  
To clone a repository:**

**git clone https://github.com/tec\_ppadmin6/repository.git**

**Replace repository with the appropriate repository details.**

**For SSH:**

**git clone git@github.com:tec\_ppadmin6/repository.git**

**7. Adding Git Authentication Code to Windows Generic  
If you're using Git with HTTPS, you may need to store authentication credentials securely in Windows:**

**Step 1: Open Windows Credential Manager.**

* **Press Win + S, type Credential Manager, and press Enter.**

**Step 2: Click on Windows Credentials.**

**Step 3: Select Add a generic credential.**

**Step 4: Enter the following details:**

* **Internet or network address: git:https://github.com (or your Git server URL)**
* **Username: tec\_ppadmin6**
* **Password: Your Git personal access token or password**

**Step 5: Click OK to save.**

**This will allow Git to authenticate seamlessly without prompting for credentials each time.**

**8. Creating a Key on Repository and Setting Its Duration  
To enhance security, you can create a deploy key or personal access token with an expiration date:**

**Step 1: Log into your GitHub/GitLab account.**

**Step 2: Navigate to Settings > Developer Settings > Personal Access Tokens (or Deploy Keys).**

**Step 3: Click Generate new token (or Add new key).**

**Step 4: Provide a descriptive name for the key (e.g., tec\_ppadmin6 access key) and select the necessary scopes/permissions.**

**Step 5: Set an expiration date (e.g., 30 days, 90 days) to enhance security.**

**Step 6: Click Generate Token and copy the token securely. Use this token in place of your password for Git operations.**

**9. Troubleshooting**

* **Permission Denied (Public Key): Ensure your SSH key is added to your GitHub/GitLab account.**
* **SSL Certificate Issues: Use:**

**git config --global http.sslVerify false**

***(Only as a temporary solution for trusted environments.)***

* **Command Not Found: Ensure Git is installed correctly and added to your system's PATH.**

**10. Conclusion  
You have successfully installed and configured Git in your VDI environment. This setup enables you to manage your code repositories efficiently. For advanced Git commands and features, refer to the** [**official Git documentation**](https://git-scm.com/doc)**.**

**Additional Resources:**

* [**Git Documentation**](https://git-scm.com/doc)
* [**GitHub Guides**](https://guides.github.com/)
* [**GitLab Documentation**](https://docs.gitlab.com/)