IRIMASO MAURIE ASSIGNMENT PRESENTATION

Download and Install Windows 11 STEPS

### **Step 1: Check System Requirements**

Ensure your PC meets the minimum requirements: 1 GHz processor, 4 GB RAM, 64 GB storage, UEFI with Secure Boot, TPM 2.0, DirectX 12 graphics, and HD display. Verify compatibility using the PC Health Check tool from Microsoft.

### **Step 2: Backup Your Data**

Backup all important data to an external drive or cloud storage to prevent data loss during installation.

### **Step 3: Get Windows 11**

Obtain Windows 11 through Windows Update, the Windows 11 Installation Assistant, Media Creation Tool, or by downloading the ISO file from the Microsoft website.

### **Step 4: Use Windows 11 Installation Assistant**

Download and run the Installation Assistant, follow on-screen instructions to upgrade your current Windows version to Windows 11.

### **Step 5: Create Installation Media (if needed)**

Use the Media Creation Tool to create a bootable USB drive or DVD. Follow prompts to select language, edition, architecture, and create the media.

### **Step 6: Install Windows 11 from Installation Media**

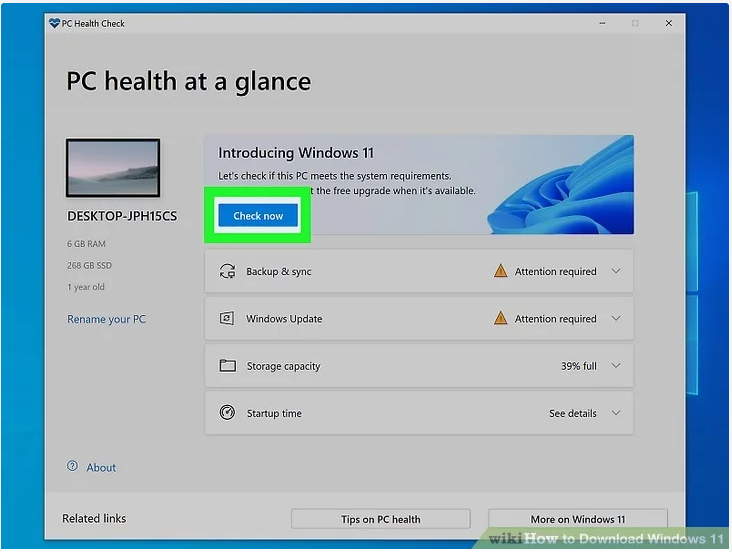
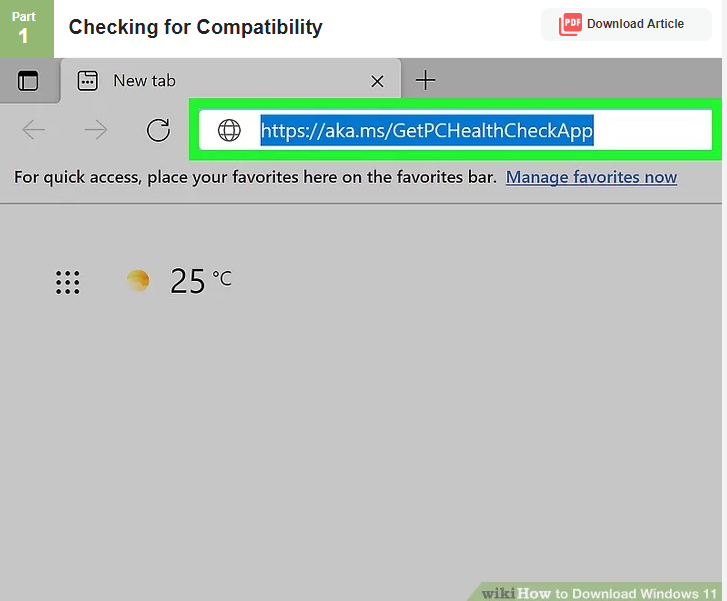
Insert the installation media, restart your PC, boot from the media, and follow on-screen instructions to start the installation. Choose between an upgrade or a fresh install.

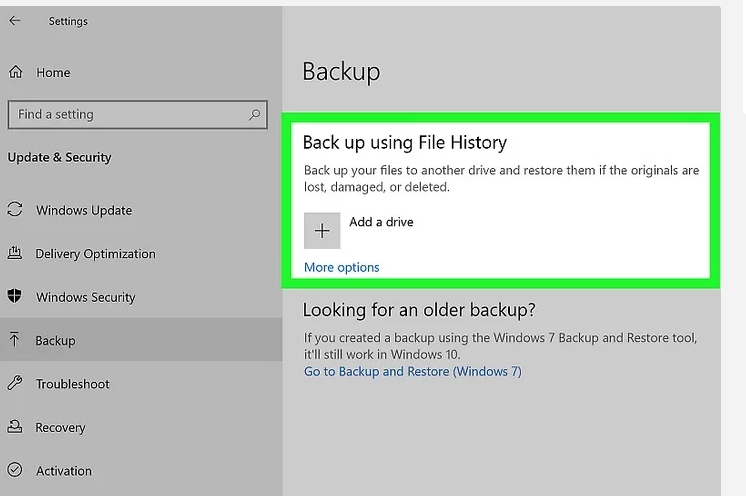
### **Step 7: Set Up Windows 11**

After installation, follow prompts to configure region, keyboard, network, and sign in with a Microsoft or local account. Customize your settings as needed.

### **Step 8: Install Updates and Drivers**

Go to Settings > Windows Update to check for updates. Install necessary drivers from your PC manufacturer’s website if they aren’t automatically updated.





2.INSTALLING VSCODE STUDIO STEPS

### **Step 1: Download VS Code**

Visit the [Visual Studio Code website](https://code.visualstudio.com/) and download the installer for your OS.

### **Step 2: Run the Installer**

Locate the downloaded file and double-click it to start the installation.

### **Step 3: Accept the License Agreement**

Accept the license terms and click "Next."

### **Step 4: Choose Install Location**

Choose or confirm the installation location and click "Next."

### **Step 5: Select Additional Tasks**

Select additional tasks like creating a desktop icon, then click "Next."

### **Step 6: Install VS Code**

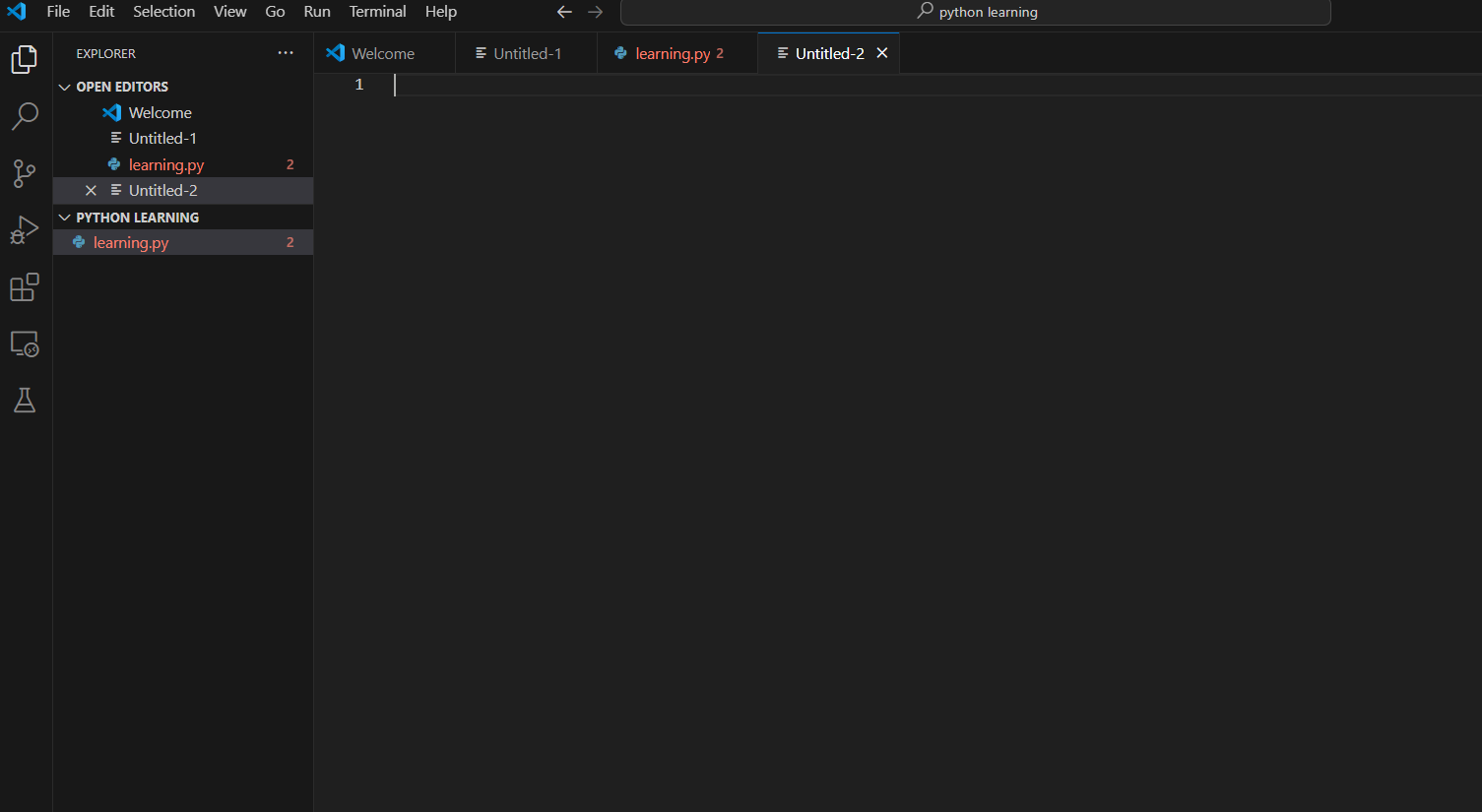
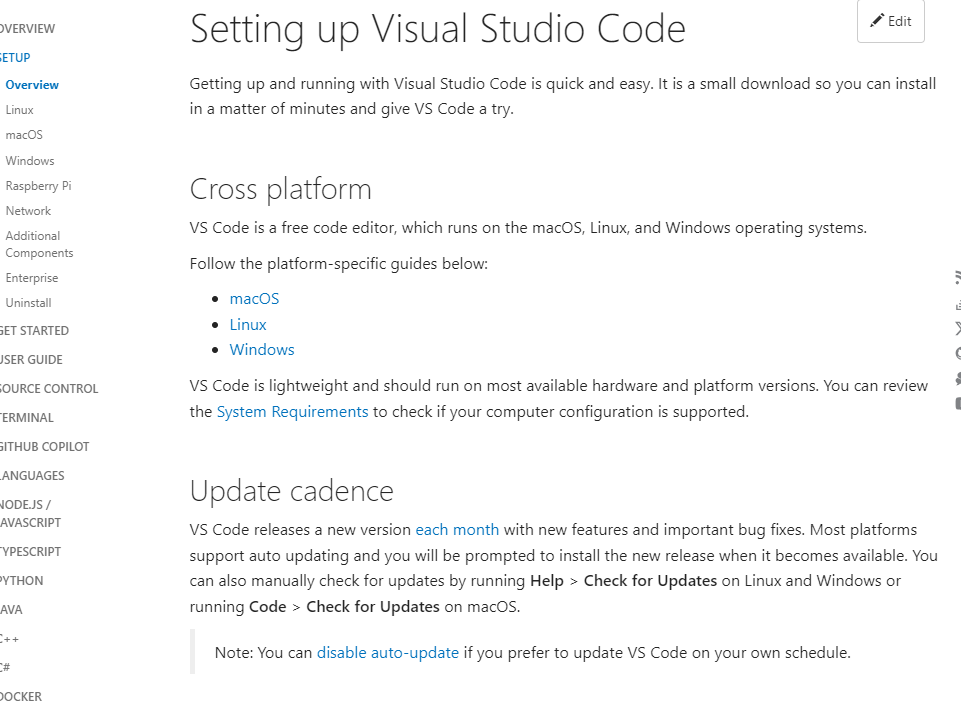
Click "Install" to begin the installation process.

### **Step 7: Launch VS Code**

Click "Finish" and open VS Code from the Start menu or Applications folder.

### **Step 8: Install Extensions (Optional)**

Open the Extensions view in VS Code and install any desired extensions.



**3 INSTALLING PYTHON STEPS**

### **Step 1: Download Python**

Visit the [Python website](https://www.python.org/downloads/) and download the installer for your OS.

### **Step 2: Run the Installer**

Locate the downloaded file and double-click it to start the installation.

### **Step 3: Customize Installation**

Check "Add Python to PATH" and click "Customize installation."

### **Step 4: Select Optional Features**

Select the optional features you want to install and click "Next."

### **Step 5: Advanced Options**

Choose any advanced options and click "Install."

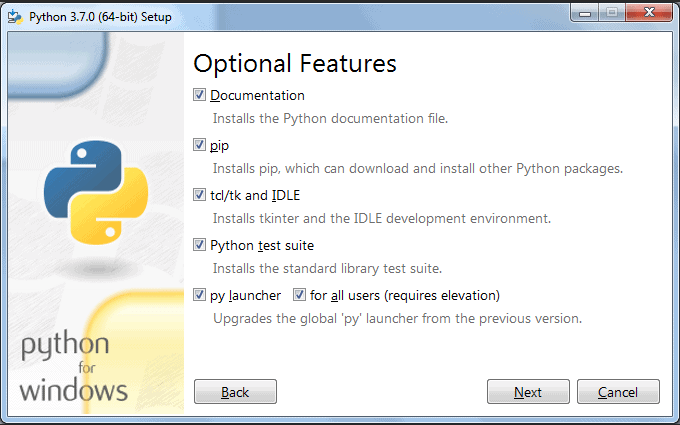
### **Step 6: Complete Installation**

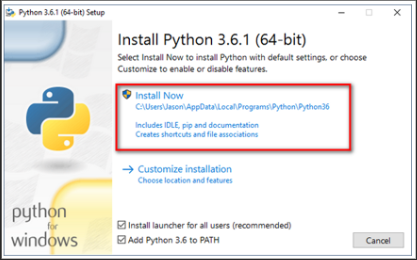
Wait for the installation to complete and click "Close."

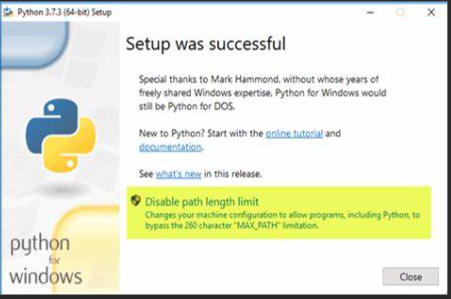
### **Step 7: Verify Installation**

Open Command Prompt or Terminal and type python --version to verify the installation.

By following these steps, you can successfully install Python on your computer.







**4.INSTALLING GITHUB &CONFIGURE IT & CREATE ACCOUNT FIRST COMMIT AND FIRST REPOSITORY**

#### **Step 1: Download Git**

Visit the [Git website](https://git-scm.com/) and download the installer for your OS.

#### **Step 2: Run the Installer**

Locate the downloaded file and double-click it to start the installation.

#### **Step 3: Choose Installation Options**

Follow the setup wizard, selecting your preferred options, and click "Next" until the installation begins.

#### **Step 4: Complete Installation**

Wait for the installation to complete and click "Finish."

#### **Step 5: Configure Git**

Open Command Prompt (Windows) or Terminal (macOS/Linux) and run:

**git config --global user.name "Your Name"**

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

### **Creating a Git Repository and Making the First Commit**

#### **Step 6: Create a New Directory**

Create a new directory for your project:

**mkdir my-project**

**cd my-project**

#### **Step 7: Initialize Git Repository**

Initialize a new Git repository:

**git init**

#### **Step 8: Create a File**

Create a new file (e.g., README.md) and add some content to it:

**echo "# My Project" > README.md**

#### **Step 9: Add File to Staging Area**

Add the file to the staging area:

**git add README.md**

#### **Step 10: Commit the Changes**

Commit the changes with a message:

**git commit -m "Initial commit"**

### **Creating a GitHub Account**

#### **Step 11: Visit GitHub**

Go to the [GitHub website](https://github.com/).

#### **Step 12: Sign Up**

Click "Sign up" and fill out the registration form with your details.

#### **Step 13: Verify Email**

Check your email for a verification message from GitHub and follow the instructions to verify your account.

### **Pushing the Repository to GitHub**

#### **Step 14: Create a New Repository on GitHub**

Click the "+" icon in the upper-right corner, select "New repository," name it, and click "Create repository."

#### **Step 15: Add Remote Repository**

Add the GitHub repository as a remote:

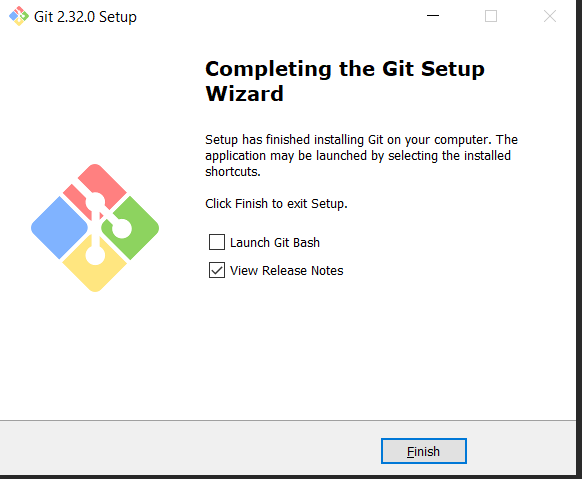
git remote add origin <https://github.com/your-username/my-project.git>

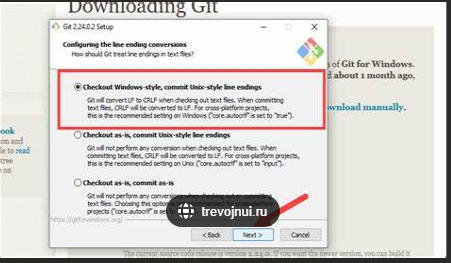
#### **Step 16: Push to GitHub**

Push your local repository to GitHub:

git push -u origin master

By following these steps, you can install and configure Git, create a repository, make your first commit, create a GitHub account, and push your repository to GitHub.





5**. Install Package Managers: If applicable, install package managers like pip (Python).**

### **Installing pip (Python)**

#### **Step 1: Check Python Installation**

Ensure Python is installed by running:

python --version

#### **Step 2: Download get-pip.py**

Download the get-pip.py script from the [official source](https://bootstrap.pypa.io/get-pip.py).

#### **Step 3: Run get-pip.py**

Run the downloaded script to install pip:

python get-pip.py

### **Verifying pip Installation**

#### **Step 4: Verify pip Installation**

Check the installation by running:

pip --version

### **Using pip to Install Packages**

#### **Step 5: Install a Package**

Use pip to install a package, for example, requests:

pip install requests

By following these steps, you can install pip, verify the installation, and use it to install Python packages.

**6. Configure a Database (MySQL): Download and install MySQL database.**

#### **Step 1: Download MySQL**

Visit the [MySQL website](https://dev.mysql.com/downloads/mysql/) and download the installer for your operating system.

#### **Step 2: Run the Installer**

Locate the downloaded file and double-click it to start the installation process.

#### **Step 3: Choose Setup Type**

Select the setup type (e.g., Developer Default, Server only) and click "Next."

#### **Step 4: Install MySQL**

Click "Execute" to download and install MySQL server and other selected products.

### **Configuring MySQL**

#### **Step 5: Configure MySQL Server**

Proceed to the configuration steps, setting up MySQL Server (e.g., type and networking).

#### **Step 6: Set Root Password**

Set a strong root password and optionally create additional user accounts.

#### **Step 7: Configure Windows Service**

Choose to run MySQL as a Windows service and configure the service name.

#### **Step 8: Apply Configuration**

Click "Execute" to apply the configuration settings.

### **Verifying MySQL Installation**

#### **Step 9: Verify Installation**

Open Command Prompt or Terminal and verify the installation by running:

mysql --version

#### **Step 10: Access MySQL Shell**

Access the MySQL shell using the root account:

mysql -u root -p

### **Creating a Database**

#### **Step 11: Create a Database**

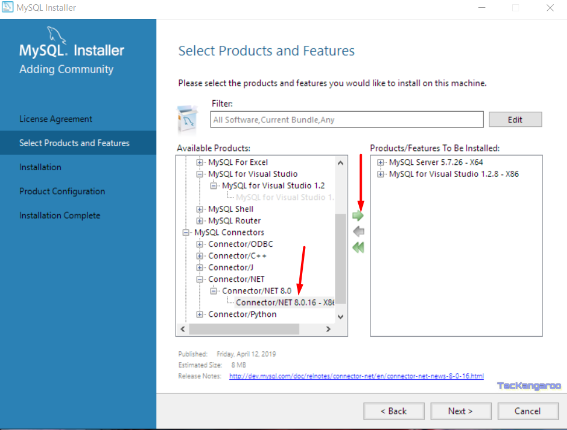
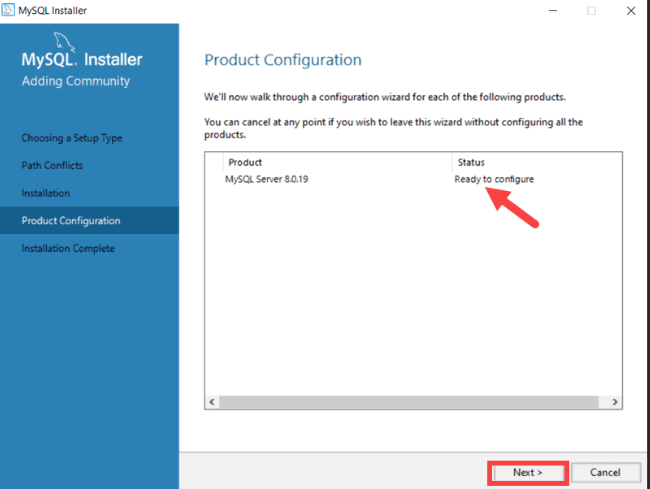
Create a new database within the MySQL shell:

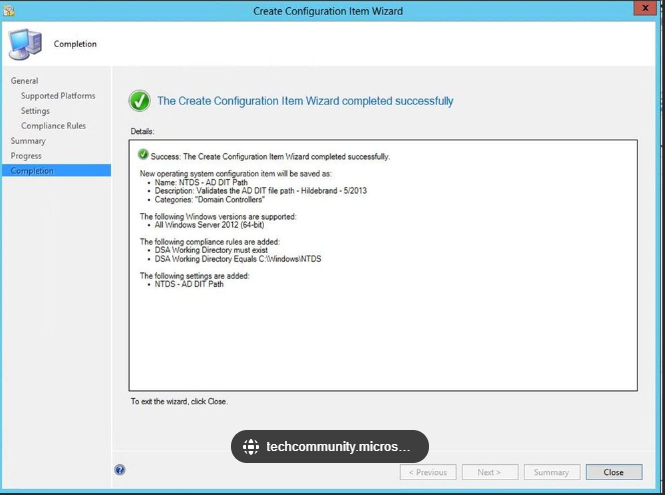
sql

Copy code

CREATE DATABASE mydatabase;

By following these steps, you can download, install, and configure a MySQL database, verify the installation, and create your first database.





Set Up Development Environments and Virtualization (Optional): Consider using virtualization tools like Docker or virtual machines to isolate project dependencies and ensure consistent environments across different machines.

#### **Step 1: Choose Virtualization Tool**

Decide whether to use Docker for containerization or virtual machines (e.g., VirtualBox, VMware) for isolation.

#### **Step 2: Install Virtualization Software**

Download and install Docker from Docker's official website or the preferred virtualization software.

#### **Step 3: Configure Virtualization Tool**

Follow the installation wizard instructions to complete the setup of Docker or virtual machine software.

#### **Step 4: Run Virtualization Software**

Start Docker Desktop or the virtual machine software to initialize the environment.

#### **Step 5: Pull Docker Images (if using Docker)**

If using Docker, pull relevant images for your development environment from Docker Hub:

docker pull image\_name:tag

#### **Step 6: Create Dockerfile (if customizing Docker)**

Create a Dockerfile to define the environment setup, including dependencies and configurations.

#### **Step 7: Build Docker Image (if using Docker)**

Build a Docker image based on your Dockerfile:

docker build -t myapp .

#### **Step 8: Run Docker Container (if using Docker)**

Run a Docker container based on your image, mapping ports and volumes as needed:

docker run -d -p 8080:80 --name mycontainer myapp

#### **Step 9: Set Up Virtual Machine (if using VMs)**

For virtual machines, install the operating system and required software within the VM environment.

#### **Step 10: Configure Networking and Sharing (if using VMs)**

Set up networking and shared folders to facilitate communication between the host and virtual machine.

### **Managing Development Environments**

#### **Step 11: Install Development Tools**

Install development tools and dependencies inside the Docker container or virtual machine.

#### **Step 12: Version Control Integration**

Integrate with version control systems (e.g., Git) to manage project changes and collaboration.

#### **Step 13: Document Environment Setup**

Document steps and configurations to replicate the environment across different machines or for future reference.

By following these steps, you can effectively set up development environments using virtualization tools like Docker or virtual machines, ensuring consistent and isolated environments for your projects.

**Explore Extensions and Plugins: Explore available extensions, plugins, and add-ons for your chosen text editor or IDE to enhance functionality, such as syntax highlighting, linting, code formatting, and version control integration.**

#### **Step 1: Open Your Text Editor or IDE**

Launch Visual Studio Code, IntelliJ IDEA, or your preferred IDE.

#### **Step 2: Access Extensions or Plugins Marketplace**

Navigate to the extensions marketplace by clicking on the Extensions icon in Visual Studio Code's sidebar.

#### **Step 3: Browse Categories or Search**

Explore categories like "Languages" or search for specific functionalities such as "Python" or "Git" to find relevant extensions.

#### **Step 4: Read Reviews and Ratings**

Review user ratings and read feedback to assess the quality and usefulness of extensions.

#### **Step 5: Install Selected Extension or Plugin**

Click "Install" next to an extension like "Python" by Microsoft to add it to your editor.

#### **Step 6: Configure Extension Settings (if necessary)**

Adjust settings in the extension's configuration page to suit your development preferences.

#### **Step 7: Explore Additional Extensions (Optional)**

Continue discovering new extensions for features like debugging or theme customization.

#### **Step 8: Update Installed Extensions**

Periodically check for updates to installed extensions to access new features and improvements.

#### **Step 9: Remove or Disable Extensions (if necessary)**

Disable or uninstall extensions causing conflicts or no longer needed to streamline your environment.

#### **Step 10: Document Installed Extensions**

Maintain a list documenting installed extensions and their purposes for future reference.

Following these steps ensures you can effectively enhance your development environment with the best tools available for your needs.

**Document Your Setup: Create a comprehensive document outlining the steps you've taken to set up your developer environment. Include any configurations, customizations, or troubleshooting steps encountered during the process.**

#### **Step 1: Outline Sections**

Create sections for different aspects like installation steps, configurations, customizations, and troubleshooting.

#### **Step 2: Introduction**

Provide an introduction explaining the purpose and scope of the document.

#### **Step 3: List Requirements**

List hardware and software requirements needed for the setup.

#### **Step 4: Installation Steps**

Detail installation steps for tools like IDEs, version control systems, and databases.

#### **Step 5: Configuration Details**

Document configurations such as environment variables, editor settings, and plugin installations.

#### **Step 6: Customizations**

Describe any custom scripts, themes, or workflows implemented.

#### **Step 7: Troubleshooting Steps**

Include troubleshooting steps for common issues encountered during setup.

#### **Step 8: Screenshots and Examples**

Add screenshots and examples to clarify complex configurations or setup procedures.

#### **Step 9: Final Checks and Testing**

Verify all steps are accurate and functional by re-following them.

#### **Step 10: Review and Update**

Regularly review and update the document to reflect changes or improvements in your setup.

#### **Step 11: Share and Collaborate**

Share the document with team members or on relevant platforms to collaborate and gather feedback.

#### **Step 12: Maintenance**

Maintain the document as a living resource, updating it with new configurations or tools as your environment evolves.

By following these steps, you can create a detailed document that serves as a reference for setting up and maintaining your developer environment effectively