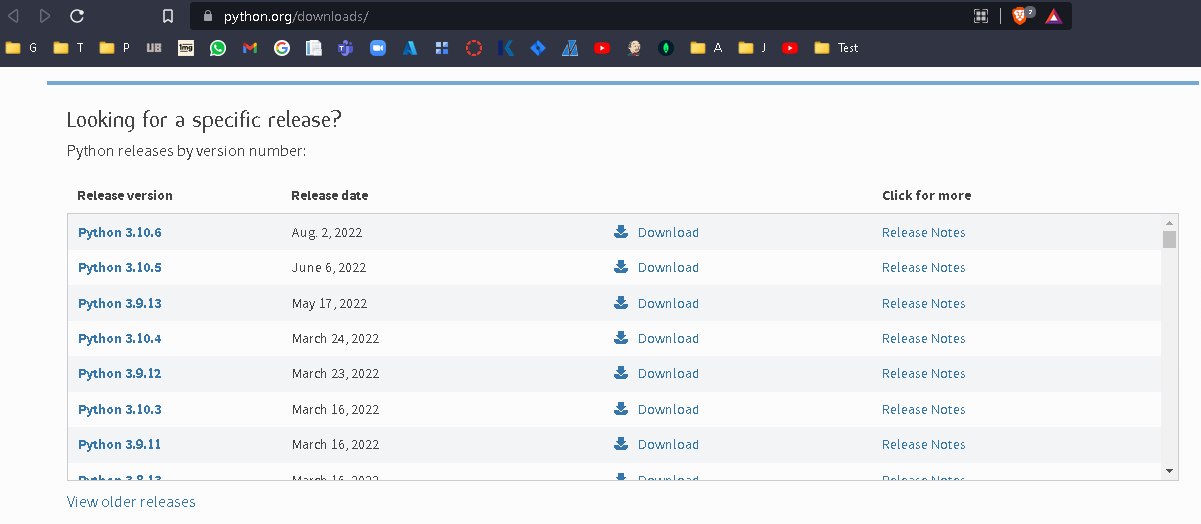
**Python 3 Installation on Windows**

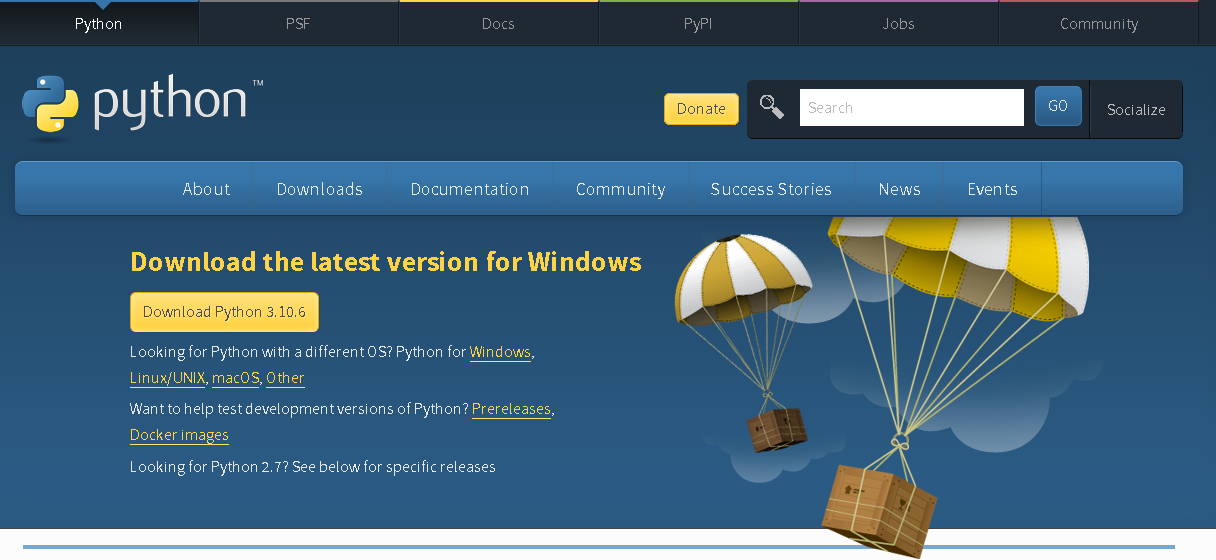
**Step 1: Select Version of Python to Install**

Goto <https://www.python.org/downloads/> and select the version of Python to install



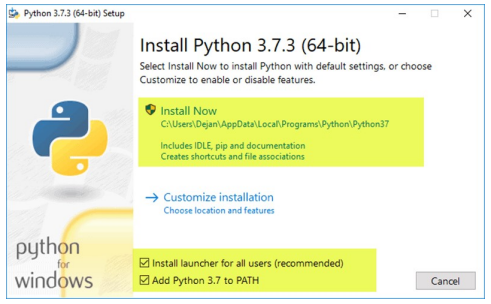
**Step 2: Download Python Executable Installer**

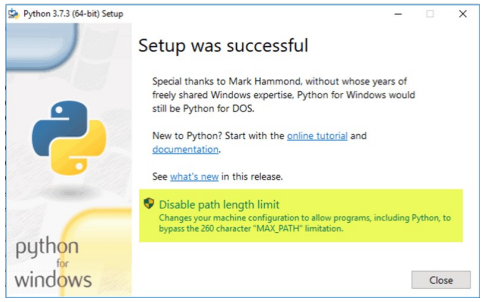
Goto <https://www.python.org/downloads/> and click on the Download Python button to download the Python executable installer



**Step 3: Run Executable Installer**

Double click on the downloaded file to start the installation of Python



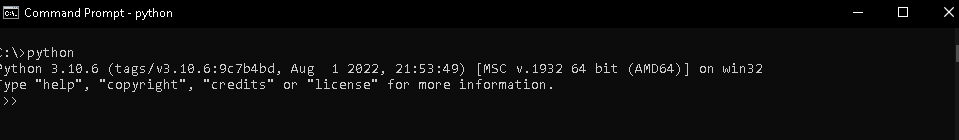


**Step 4: Verify Python Was Installed on Windows**

1. Press “Windows” + R keys (The “Run” dialog box opens)

2. Enter “cmd” in the text box and click on the “Ok” button. (The Command Shell opens)

3. At the Shell prompt enter “Python” and press the “Enter” key (Python interactive Shell is displayed)

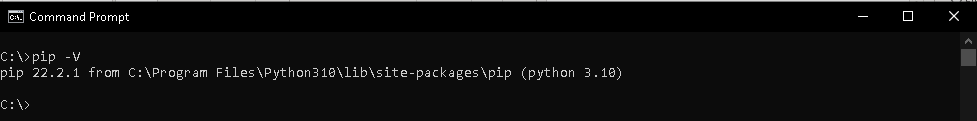


**Step 5: Verify Pip Was Installed**

1. Press “Windows” + R keys. (The “Run” dialog box opens)

2. Enter “cmd” in the text box and click on the “Ok” button. (The Command Shell opens)

3. At the Shell prompt enter “Pip -v” and press the “Enter” key (Pip version is displayed)



**Step 6: Add Python Path to Environment Variables (Optional)**

1. Press “Windows” + R keys. (The “Run” dialog box opens)

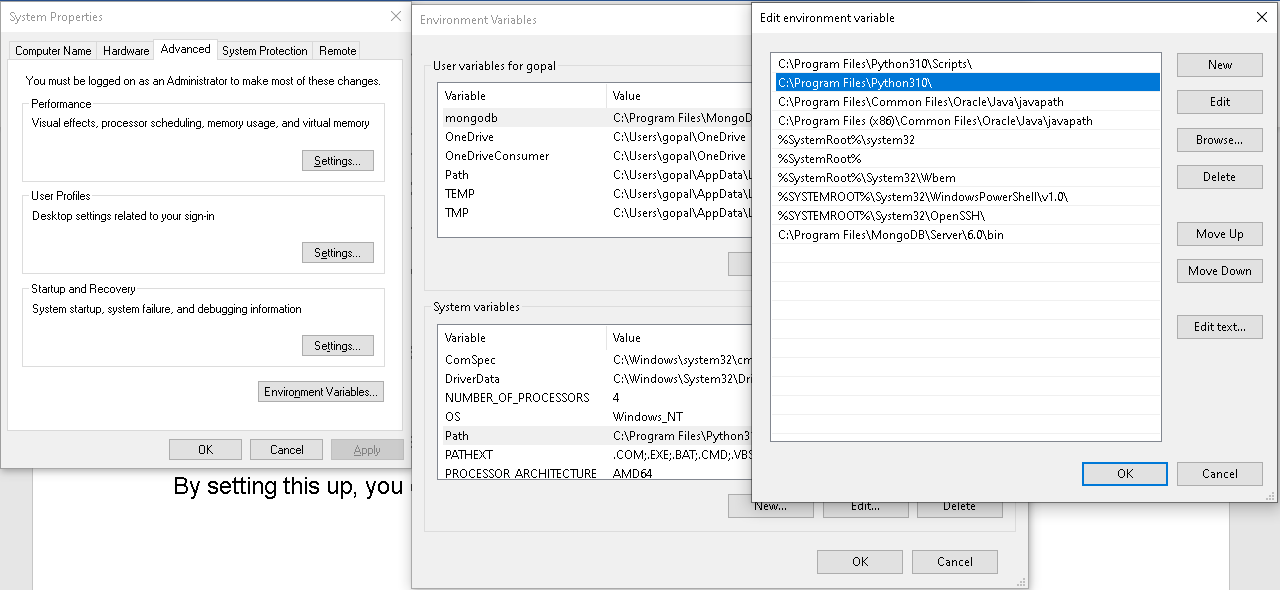
2. Enter “sysdm.cpl” in the text box and click on the “Ok” button. (The System Properties dialog box is displayed)

3. Click on the “Advanced” tab and later Click on the “Environmental variables” button.

4. The “Environmental variables” window is displayed.

5. Double Click on the “path” in the “System variables” box. (a new window pane is displayed)

6. Enter the Python directory path in it and click on the” ok” button. ()



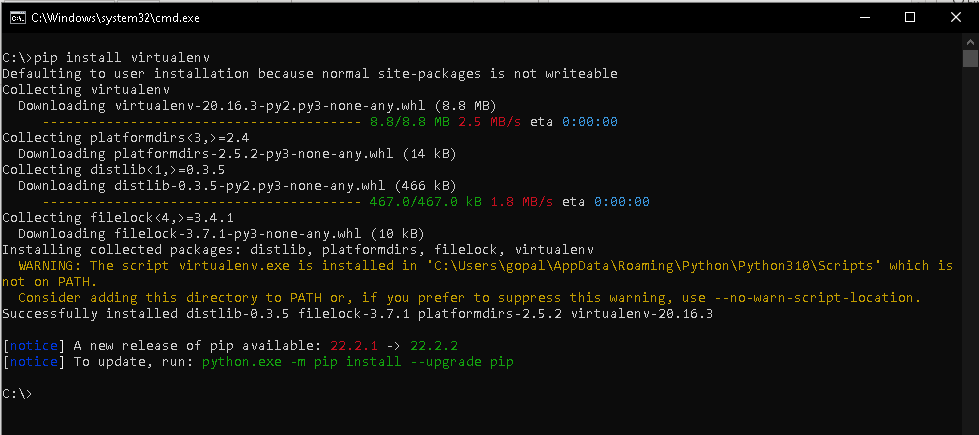
S**tep 7: Install virtualnv (Optional)**

To install virtualnv:

1. Press “Windows” + R keys. (The “Run” dialog box opens)

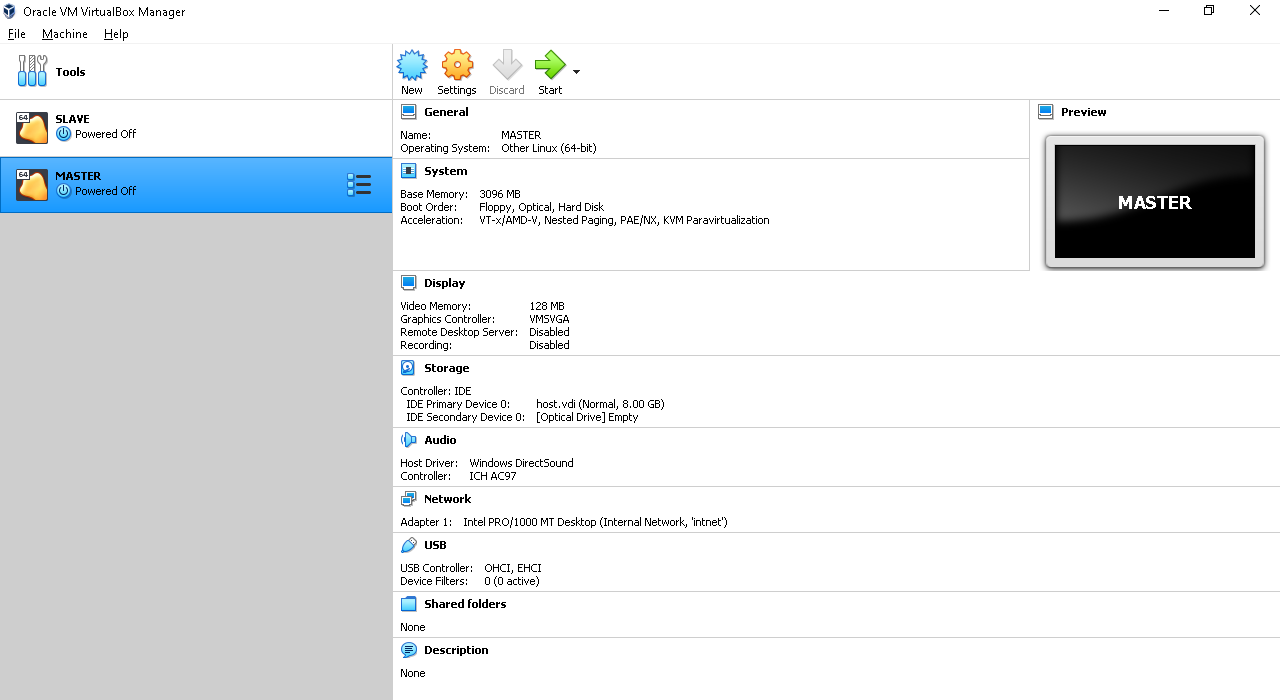
2. Enter “cmd” in the text box and click on the “Ok” button. (The Command Shell opens)

3. At the Shell prompt enter “pip install virtualenv” and press the “Enter” key (virtualnv is installed on your system).



**How to Install Ansible?**

Create and configure two ‘virtual machines’ in ‘Virtual Box’ hypervisor software, named Slave and Master with “Centos 7 Operating System”.



**Step 1: Update Your Control Node**

Execute the below command to update the existing packages in the OS.

yum update

**Step 2: Install EPEL Repository**

Execute the below command to update the install “Extra packages for Enterprise Linux”

yum install epel-release

**Step 3: Install Ansible**

Execute the below commands in sequence to update the ‘Yum’ repository and install Ansible.

Yum update

yum install ansible

**Step 4: Create a User for Ansible**

Execute the below commands in sequence to create a user for operating Ansible jobs.

useradd ansibleadmin

passwd ansibleadmin

\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

Configure the ‘ansibleadmin’ user as sudoer with root user privileges

Vi etc/sudoers (the ‘sudeors’ file is opened for editing, enter below command to update)

username ALL=(ALL) NOPASSWD:ALL (save the file by pressing ‘**Esc**’ key and **:wq**)

**Step 5: Configure Our ‘ansibleadmin’ User for SSH Access**

Create the SSH public and private key pair.

ssh-keygen

Copy the public key and paste it to our Managed node with the below command.

ssh-copy-id 192.168.100. 30

**Step 6: Create an Inventory**

Create and configure an inventory list of your managed nodes.

vim /home/ansibleadmin/inventory

add the hostname of the Managed Node.

192.168.100.30 ssh\_user=ansibleadmin ssh\_password=anisble (save the file by pressing ‘**Esc**’ key and **:wq**)

**Step 7: Create an Ansible Playbook for installing Nginx on the Managed Node.**

On the Master node, enter the following commands to configure the playbook

vim /home/simplilearn/install-nginx.yml

Enter the below code in the install-nginx.yml file.

--

- hosts: 192.168.100.30

become: yes

tasks:

- name: Installs nginx web server

Apt: name: nginx state: started update\_cache: true

notify:

- start nginx

handlers:

- name: start nginx

service:

name: nginx

state: started

**Step 8: Run the Playbook**

Execute the following commands to run the playbook

ansible-playbook -i /home/ansibleadmin/inventory /home/admin/install-nginx.yml

**Jenkins – Maven Integration**

1. Run Jenkins as a Standalone Application  
2. Login to the Jenkins  
3. Click on Manage Jenkins  
4. Click on Global Tool Configuration  
5. Under Maven, click on Add Maven  
6. Uncheck install automatically  
7. Provide the path for the Maven bin folder  
8. Click on Apply  
9. In the Jenkins Dashboard click on New Item  
Enter an Item Name  
Select Maven Project  
Click on apply and save  
In the configure page of the Maven, enter a description, and under  
Maven click on advanced.  
Choose a custom workspace. Custom workspace is the path where  
pom.xml is present  
Under build, set the goals and options to package  
Click on Save  
Click on Build Now which is present in New Item  
Wait till the build is a success

**Jenkins – Git Integration**

1. Run Jenkins as a Standalone Application

2. Login to the Jenkins

3. Go to Manage Jenkins

4. Go to Global Tool Configuration

5. In Git / Path to Git executable enter C:\<whatever the path is>\git.exe.

6. Click on Save.

7. Log in to the GitHub repository and select the repository created.

Click on the repository and make sure that the latest code is present

Now in Jenkins dashboard click on “New Item”

Type a project name and select Freestyle project from a list of jobs

enlisted

Select Git under source code management and provide the Github

repository URL

Provide the credentials

Provide the required build commands to run the code

Here we have used execute windows batch command: mvn package

Click on Apply and Save button

Click on Build Now

Find the result in the console output

https://www.javatpoint.com/jenkins-setup-build-jobs

**Jenkins – Selenium Integration**

1. Run Jenkins as a Standalone Application

2. Login to the Jenkins

3. Click on Manage Jenkins

4. Click on Global Tool Configuration

5. Under Maven, click on Add Maven

6. Uncheck install automatically

7. Provide the path for the Maven bin folder

8. Click on Apply

9. In the Jenkins Dashboard click on New Item

Enter an Item Name

Select Maven Project

Click on apply and save

In the configure page of the Maven, enter a description, and under

Maven click on advanced.

Choose a custom workspace. Custom workspace is the path where

pom.xml is present of the Maven Enabled Selenium Project created in

Eclipse

Under build, set the goals and options to clean compile test

Click on Save

Click on Build Now which is present in New Item

Wait till the build is a success