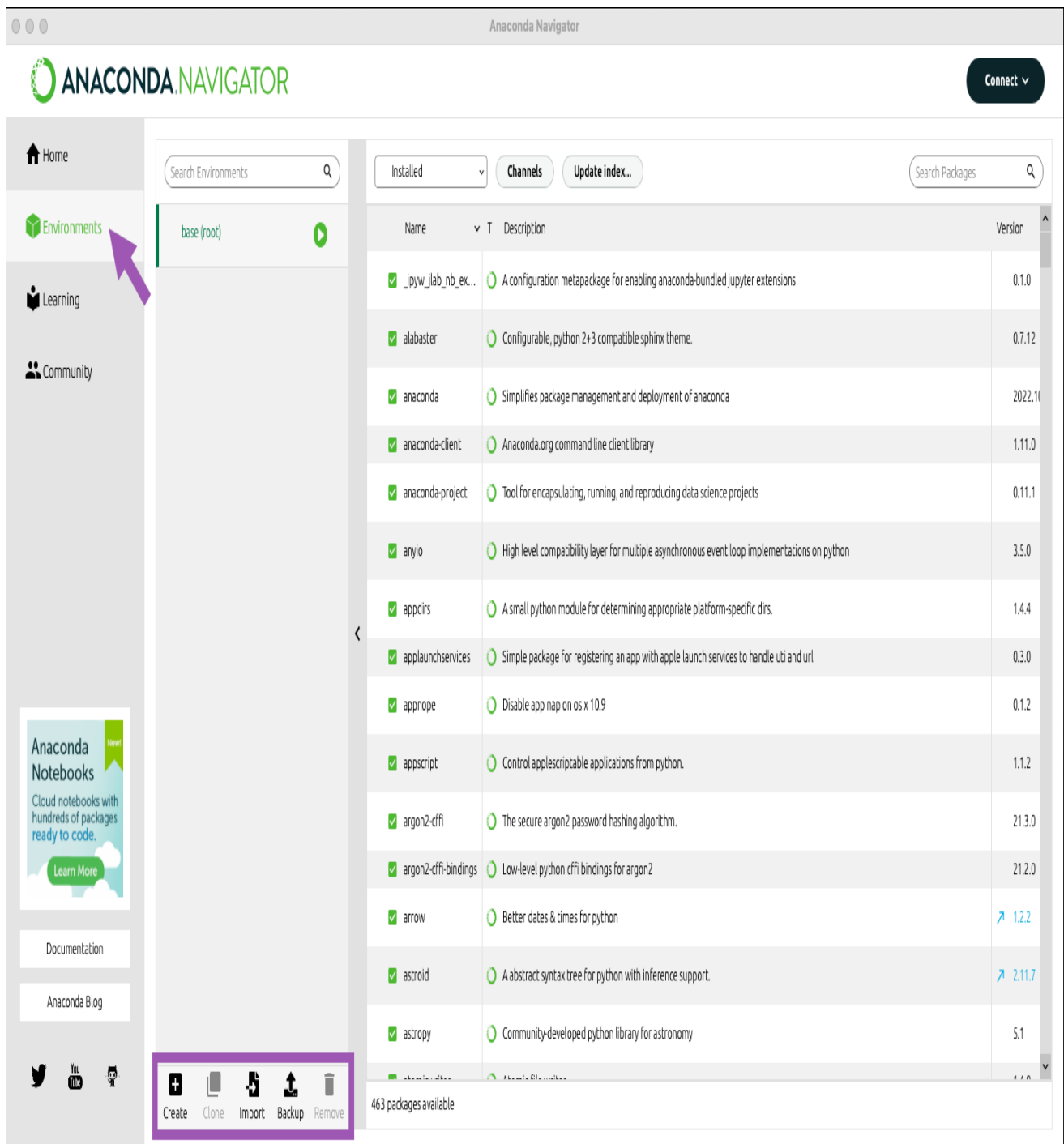


# How to create a New environment from Anaconda Navigator

On the **Environments** page, the left column displays your environments. At the bottom of the environments list are the **Create**, **Clone**, **Import**, **Backup**, and **Remove** buttons.



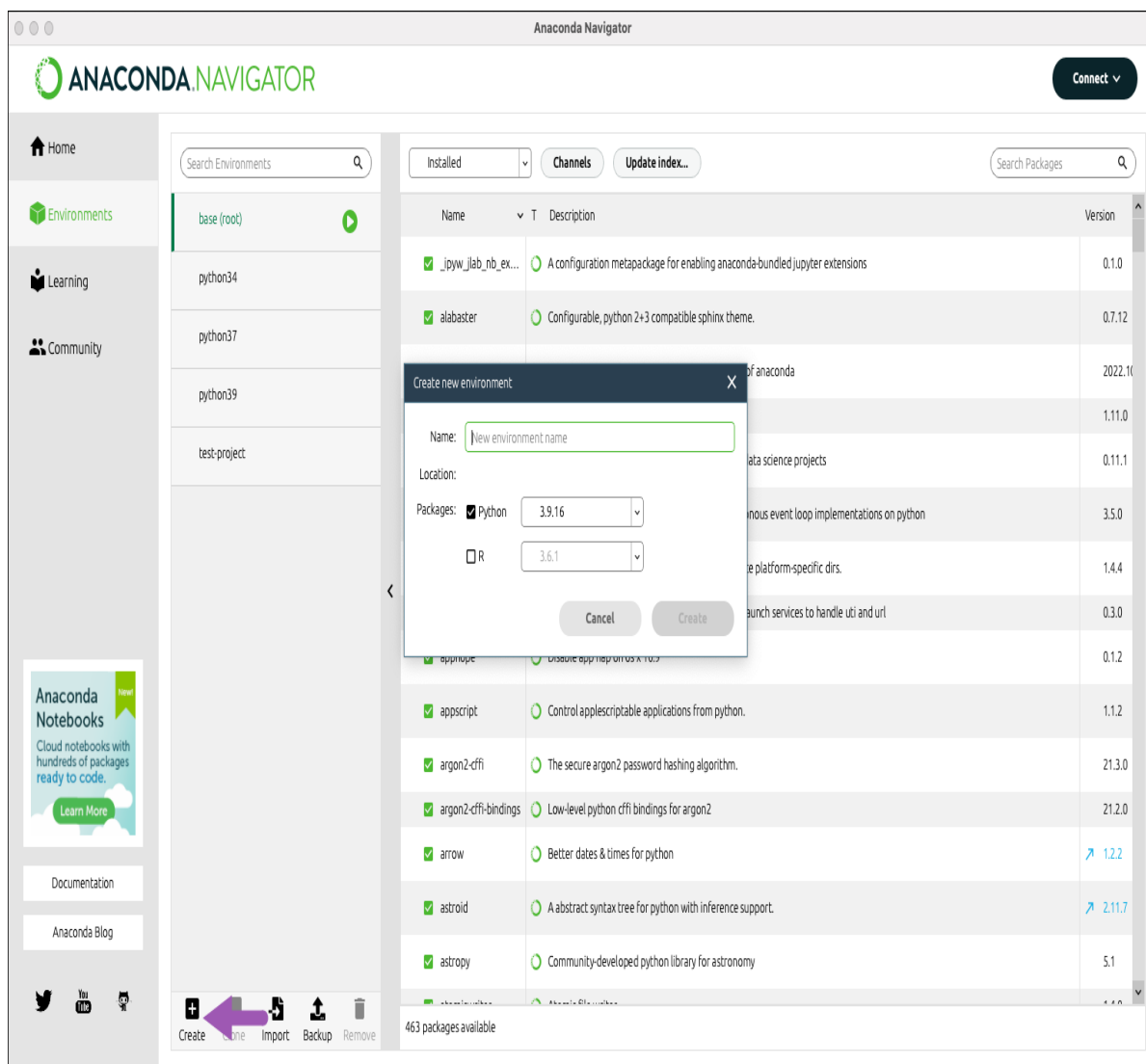
The screenshot shows the Anaconda Navigator interface. The sidebar on the left has a button labeled "Environments" which is highlighted with a purple arrow. The main content area is divided into two sections. The top section shows a search bar for environments and a list of installed packages. The bottom section shows a table of installed packages with columns for Name, Description, and Version. At the bottom of the environments list, there is a row of buttons: "Create", "Clone", "Import", "Backup", and "Remove".

Name	Description	Version
✓ jupyterlab_nb_ext...	A configuration metapackage for enabling anaconda-bundled jupyter extensions	0.1.0
✓ alabaster	Configurable, python 2+3 compatible sphinx theme.	0.7.12
✓ anaconda	Simplifies package management and deployment of anaconda	2022.10
✓ anaconda-client	Anaconda.org command line client library	1.11.0
✓ anaconda-project	Tool for encapsulating, running, and reproducing data science projects	0.11.1
✓ anyio	High level compatibility layer for multiple asynchronous event loop implementations on python	3.5.0
✓ appdirs	A small python module for determining appropriate platform-specific dirs.	1.4.4
✓ applaunchservices	Simple package for registering an app with apple launch services to handle uti and url	0.3.0
✓ appnope	Disable app nap on os x 10.9	0.1.2
✓ appscript	Control applescriptable applications from python.	1.1.2
✓ argon2-ffi	The secure argon2 password hashing algorithm.	21.3.0
✓ argon2-ffi-bindings	Low-level python ffi bindings for argon2	21.2.0
✓ arrow	Better dates & times for python	1.2.2
✓ astroid	A abstract syntax tree for python with inference support.	2.11.7
✓ astropy	Community-developed python library for astronomy	5.1

463 packages available

## Creating a new environment

1. At the bottom of the environments list, select **Create**.
2. In the Create new environment dialog, type a descriptive name for the new environment.



3. Select Python or R to set the package type for your environment.
4. Select a version for your Python or R installation.
5. Click **Create**.

# How to create a Python 3.5 environment from Anaconda prompt

## **Step 1: Check if conda is installed in your path.**

- Open up the anaconda command prompt.
- Type `conda -V` and press enter.
- If the conda is successfully installed in your system you should see a similar output.

`conda -V`

### **Output:**

```
nikhil@nikhil-Lenovo-V130-15IKB:~$ conda -V
conda 4.8.3
nikhil@nikhil-Lenovo-V130-15IKB:~$
```

## **Step 2: Update the conda environment**

- Enter the following in the anaconda prompt.
- `conda update conda`

## **Step 3: Set up the virtual environment**

- Type `conda search "^python$"` to see the list of available python versions.
- Now replace the envname with the name you want to give to your virtual environment and replace x.x with the python version you want to use.

`conda create -n envname python=x.x anaconda`

Let's create a virtual environment name Geeks for Python3.6

```
nikhil@nikhil-Lenovo-V130-15IKB: ~
File Edit View Search Terminal Help

nikhil@nikhil-Lenovo-V130-15IKB:~$ conda create -n Geeks python=3.6 anaconda
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

environment location: /home/nikhil/anaconda3/envs/Geeks

added / updated specs:
- anaconda
- python=3.6

The following packages will be downloaded:

package | build | size
-----|-----|-----
_anaconda_depends-2020.02 | py36_0 | 6 KB
alabaster-0.7.12 | py36_0 | 18 KB
anaconda-custom | py36_1 | 3 KB
anaconda-client-1.7.2 | py36_0 | 147 KB
anaconda-project-0.8.4 | py_0 | 210 KB
argh-0.26.2 | py36_0 | 36 KB
anaconda-1.2.0 | py36_0 | 164 KB
```

## **Step 4: Activating the virtual environment**

- To see the list of all the available environments use command `conda info -e`
- To activate the virtual environment, enter the given command and replace your given environment name with envname  
`conda activate envname`

```
nikhil@nikhil-Lenovo-V130-15IKB: ~  
File Edit View Search Terminal Help  
nikhil@nikhil-Lenovo-V130-15IKB:~$ conda activate Geeks  
(Geeks) nikhil@nikhil-Lenovo-V130-15IKB:~$
```

When conda environment is activated it modifies the PATH and shell variables points specifically to the isolated Python set-up you created.

## **Step 5: Installation of required packages to the virtual environment**

- Type the following command to install the additional packages to the environment and replace envname with the name of your environment.  
`conda install -n yourenvname package`

## **Step 6: Deactivating the virtual environment**

- To come out of the particular environment type the following command. The settings of the environment will remain as it is.  
`conda deactivate`

```
nikhil@nikhil-Lenovo-V130-15IKB: ~  
File Edit View Search Terminal Help  
(Geeks) nikhil@nikhil-Lenovo-V130-15IKB:~$ conda deactivate  
nikhil@nikhil-Lenovo-V130-15IKB:~$
```

## **Step 7: Deletion of virtual environment**

- If you no longer require a virtual environment. Delete it using the following command and replace your environment name with envname  
`conda remove -n envname -all`

nikhil@nikhil-Lenovo-V130-15IKB: ~

File Edit View Search Terminal Help

```
nikhil@nikhil-Lenovo-V130-15IKB:~$ conda remove -n Geeks --all
```

Remove all packages in environment /home/nikhil/anaconda3/envs/Geeks:

## Package Plan ##

environment location: /home/nikhil/anaconda3/envs/Geeks

The following packages will be REMOVED:

```
_anaconda_depends-2020.02-py36_0  
_libgcc_mutex-0.1-main  
alabaster-0.7.12-py36_0  
anaconda-custom-py36_1  
anaconda-client-1.7.2-py36_0  
anaconda-project-0.8.4-py_0  
argh-0.26.2-py36_0  
asn1crypto-1.3.0-py36_0  
astroid-2.4.0-py36_0  
astropy-4.0.1.post1-py36h7b6447c_1  
atomicwrites-1.4.0-py_0  
attrs-19.3.0-py_0
```