

LAUNCHING A BARE METAL INSTANCE WITH PYTHON

Introduction

In this tutorial I explain how to launch a bare metal instance on Chameleon with Python. For work in Chameleon with Python must have the *chi* library which is already present in Jupyter Lab Interface.

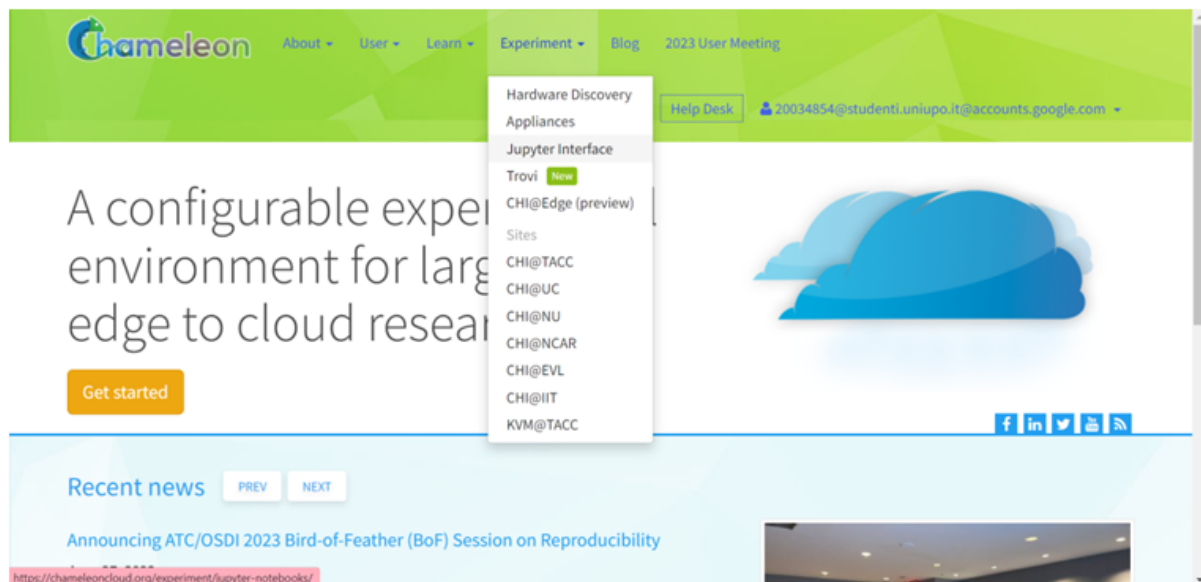
What is bare metal?

Bare metal is a physical server that the user rents in order to have exclusive access. In order the user must:

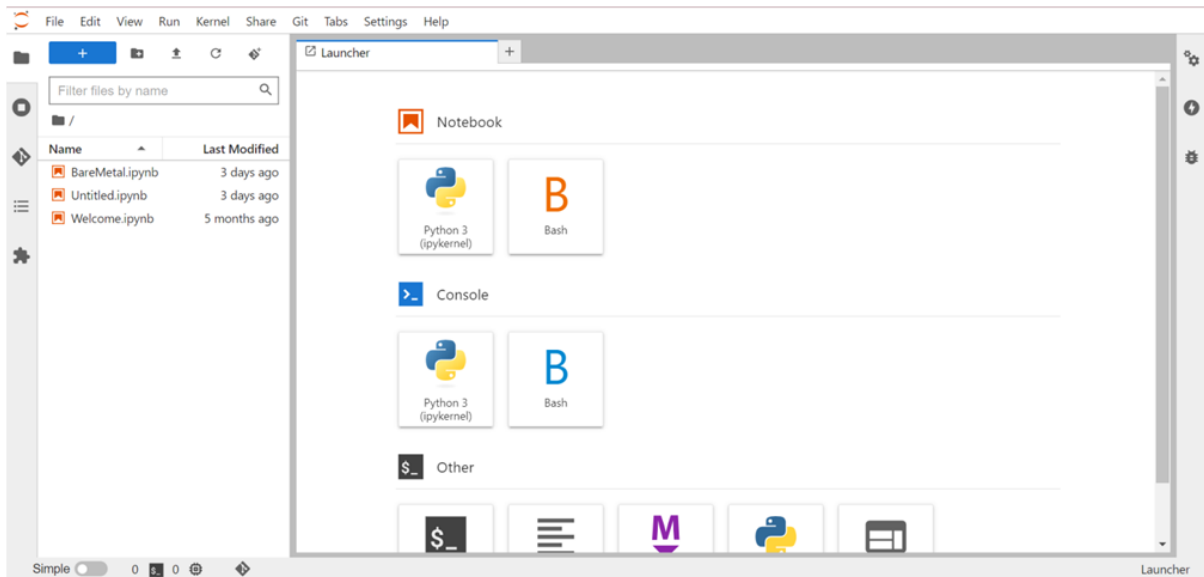
1. Make reservation for a node
2. Launch an instance
3. Associate floating IP to the instance

Environment

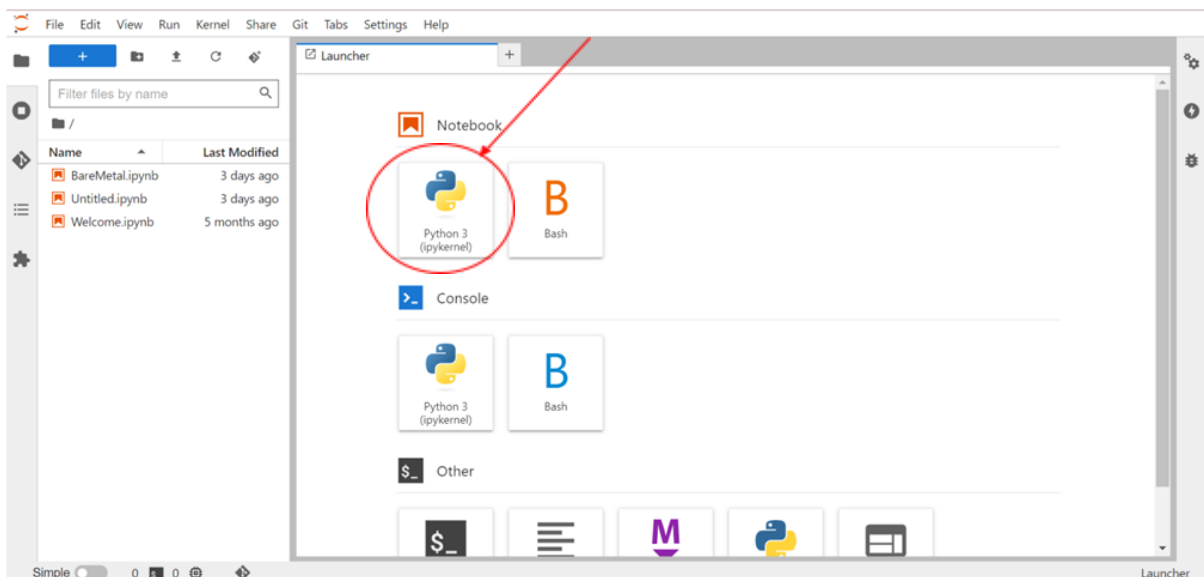
If you are a Chameleon user you can log in automatically to your Jupyter interface. Go to [Chameleon](#), click on “Experiment” -> “Jupyter Interface”



you will be redirected to an interface that looks like the following



In the Notebook session select Python3 (ipykernel)



Now that the working environment is set up you are ready to launch your first bare metal instance.

Follow the next commands.

Insert the follow commands in Python script for start to work with Chameleon:

```
import chi  
chi.use_site('CHI@UC')
```

Autore: Gloria Spina

You can use CHI@TACC if you want work with a different region

If you run the application with only this function you can see this

Now using CHI@UC:

URL: <https://chi.uc.chameleoncloud.org>

Location: Argonne National Laboratory, Lemont, Illinois, USA

Support contact: help@chameleoncloud.org

Click on the link to be redirected on the website that you choose



Make reservation for a node

Now is the time to create a lease. As first step you must have the correct import

```
from chi.lease import get_node_reservation
```

This import is very important for work with lease. Without this import you will have an error.

```
lease_name = "20034854_lease"

reservations = []

chi.lease.add_node_reservation(reservations, count=1)

chi.lease.add_fip_reservation(reservations, count=1)

start_date, end_date = chi.lease.lease_duration(days=1)

chi.lease.create_lease(lease_name, reservations, start_date=start_date,
end_date=end_date)

reservation_id = get_node_reservation(lease_name)
```

Since we work in a shared environment you must use your “registration number_lease” for the lease name.

To see your lease click on “Reservations” -> “Leases”



Now, you have submitted a request for the resources you need. Once a reservation has been accepted, users are guaranteed that resources will be available at the time they choose.

☐	20034854_lease	0abd0ecdfe980b1e2f0e5f81ecb6d9d8e6df5dbaa8da3b41c954c17d4ac8d9e9	2023-07-07 12:41 UTC	2023-07-08 12:40 UTC	PENDING	No	<button>Update Lease</button>
--------------------------	----------------	--	----------------------	----------------------	---------	----	-------------------------------

Autore: Gloria Spina

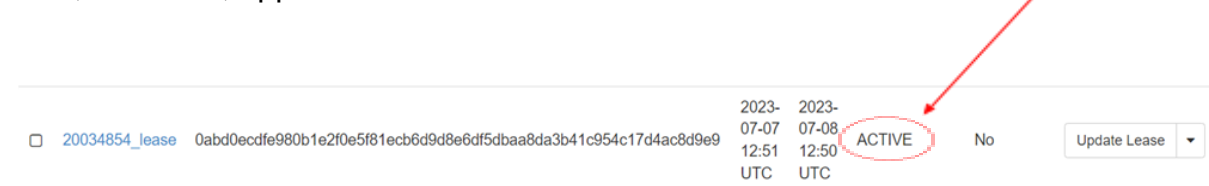
This is the lease just created. To notice that is reserved for one day, which is the time we requested it for.

Now, you have to wait for the lease to be created between the following commands.

ATTENTION: This command is very important because lets wait for that the lease switch from pending state to active state

```
chi.lease.wait_for_active(lease_name)
```

Now, the lease, appears as the follow



☐	20034854_lease	0abd0ecdfe980b1e2f0e5f81ecb6d9d8e6df5dbaa8da3b41c954c17d4ac8d9e9	2023-07-07 12:51 UTC	2023-07-08 12:50 UTC	ACTIVE	No	Update Lease ▼
--------------------------	----------------	--	----------------------	----------------------	--------	----	----------------

Launch an instance

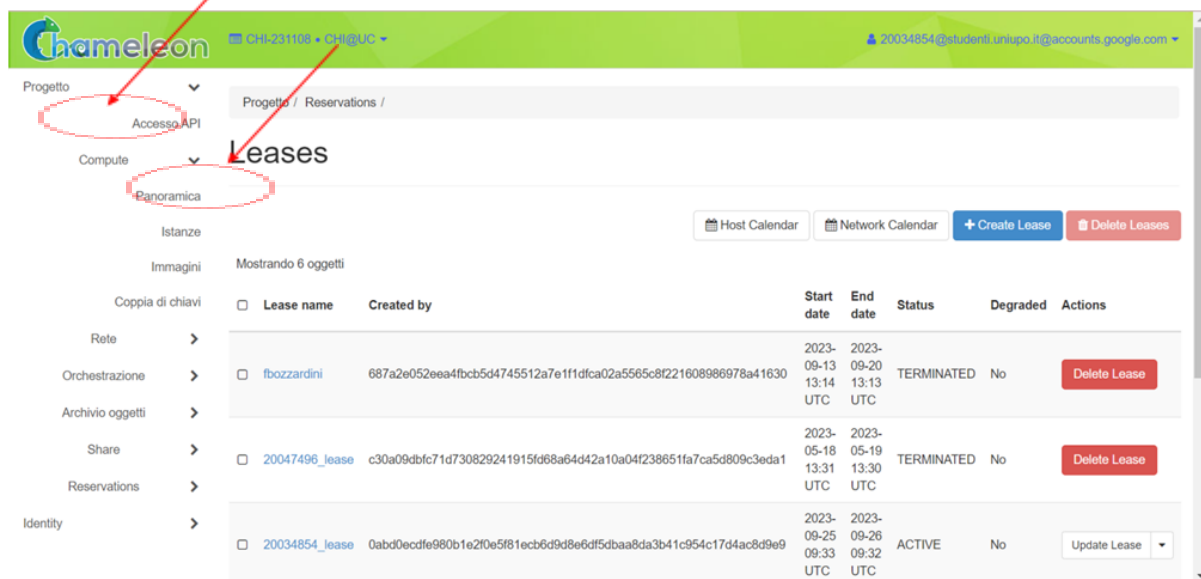
Once the lease has been activated, you can launch the last command that creates the bare metal instance. Before you must insert the follow import

```
from chi.server import create_server
```

```
server_name = "20034854_instance"
```

```
server = create_server(server_name, reservation_id=reservation_id)
```

To see the instance click on “Compute” -> “Istanze”



The instance appears as the next:

☐	20034854_4_instance	CC-U buntu 20.04	10.140.81.236	baremetal	0abd0ecdfe980b1e2f0e5f81ecb6d9d8e6df5dbaa8da3b41c954c17d4ac8d9e9	_20034854_studenti_uniupo_it_acc-jupyter	Attivo	None
--------------------------	---------------------	------------------	---------------	-----------	--	--	--------	------

ATTENTION: the creation of instance can take a long time

Associate floating IP to the instance

When the instance is ready you can associate a floating IP to it. Before proceeding, you need to make sure that the server is in the state “ACTIVE”. Insert the following code for this

```
chi.server.wait_for_active(server.id)
```

Finally, to associate the floating ip use

```
from chi.server import associate_floating_ip
```

```
ip = associate_floating_ip(server.id)
```

you can now view your instance with associated floating ip address

The screenshot shows the Chameleon web interface. On the left is a sidebar with navigation links: Progetto, Accesso API, Compute, Panoramica, **Istanze** (highlighted), Immagini, Coppia di chiavi, Rete, Orchestrazione, Archivio oggetti, Share, Reservations, and Identity. The main content area is titled 'Istanze' and shows a table with one instance. The table has columns: Instance Name, Image Name, IP Address, Flavor, Created By, Key Pair, and Stat. The instance listed is '2003485' with image 'CC-U buntu 20.04' and flavor 'baremetal'. The IP Address column shows '10.140.83.119' and '192.5.86.228'. A red circle is drawn around the floating IP address '192.5.86.228', and a red arrow points to it from the text above. The 'Key Pair' column shows '_20034854_studenti_uniupo_it_acc-jupyter' and the status is 'Attiv'.

Instance Name	Image Name	IP Address	Flavor	Created By	Key Pair	Stat
2003485 4_instance	CC-U buntu 20.04	10.140.83.119; 192.5.86.228	baremetal	0abd0ecdfe980b1e2f0e5f81ecb6d9d8e6df5dbaa8da3b41c954c17d4ac8d9e9	_20034854_studenti_uniupo_it_acc-jupyter	Attiv

The full Interactive Python Notebook file for the tutorial described above is available [here](#).