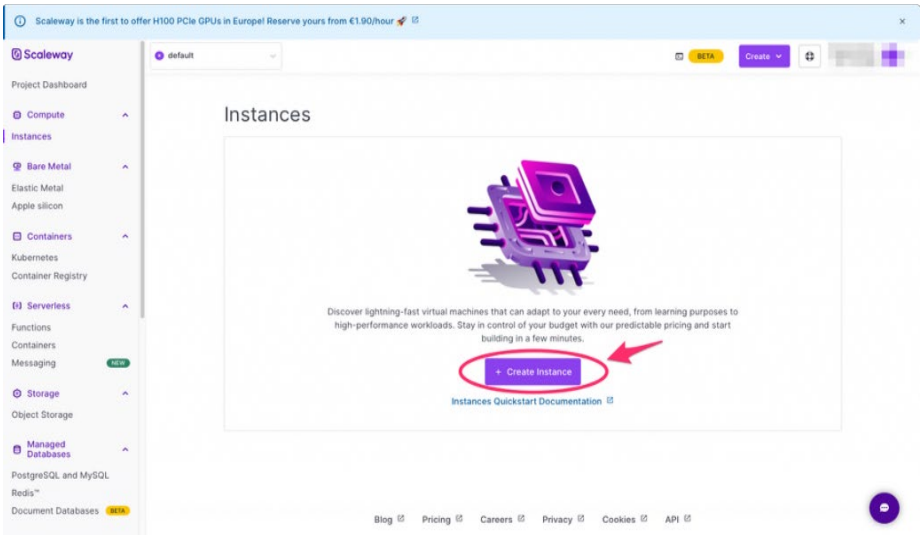
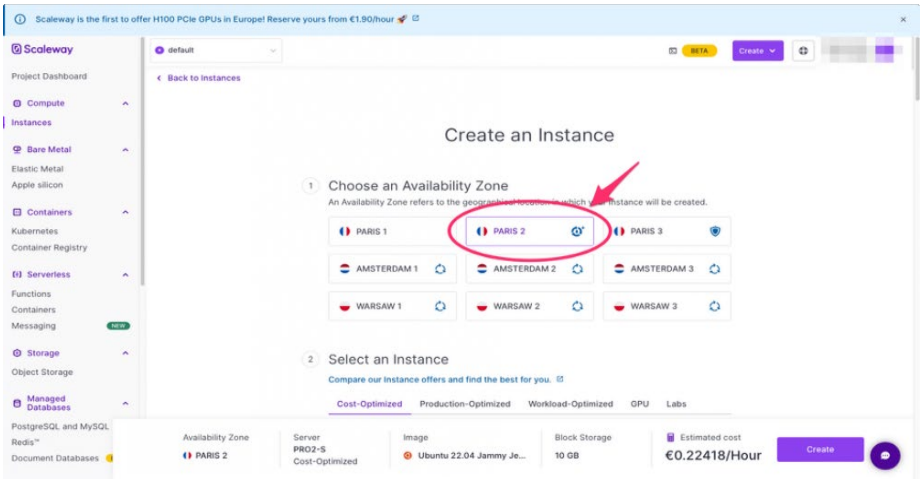
# Setting up Ampere Optimized AI Frameworks on Scaleway's COP-ARM Virtual Machines

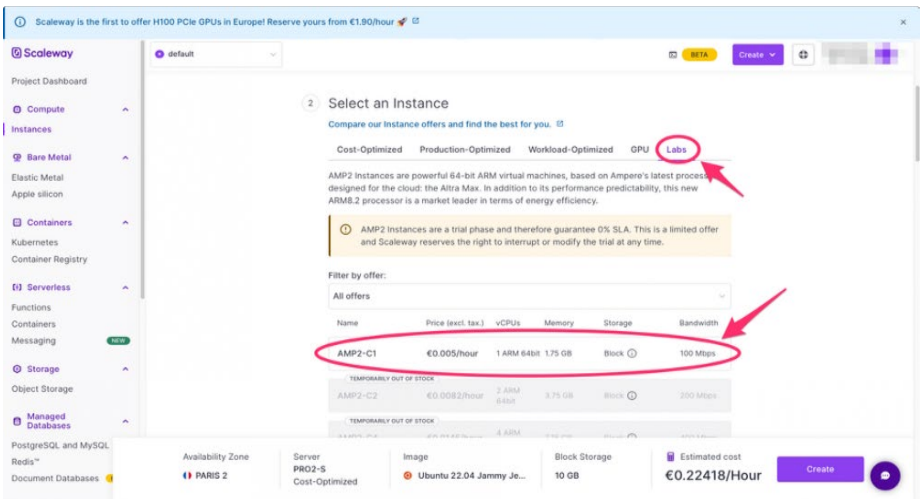
1. Go to scaleway.com and sign-in/sign-up. As a next step, click on **Instances**, then on **Create Instance**.



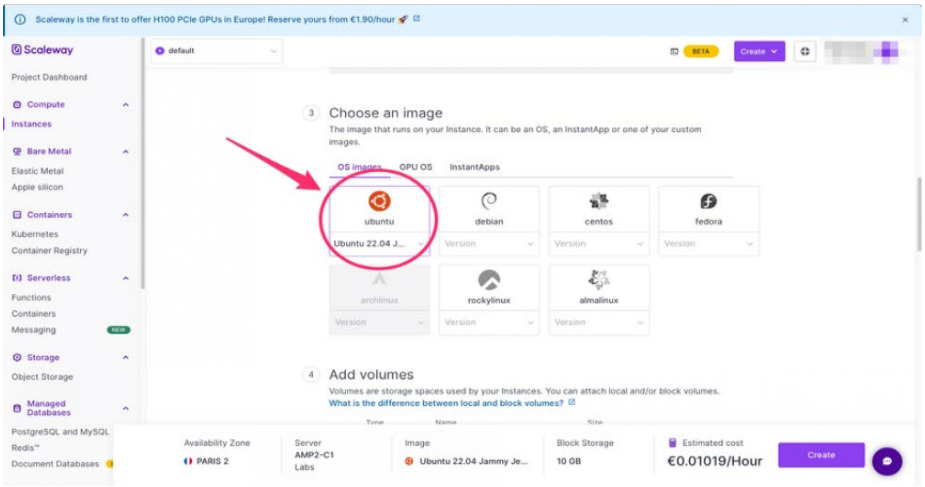

2. Select your **Availability Zone** - as of November 3rd, 2023, COP-ARM instances are available in **PARIS 2** zone only.



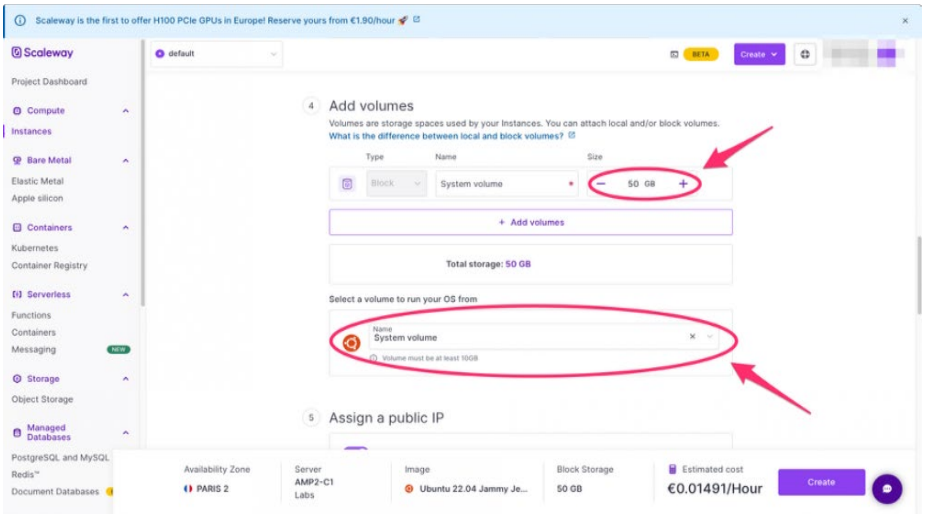

3. Click on **Labs** and select your **COP-ARM** offer accordingly to your requirements.



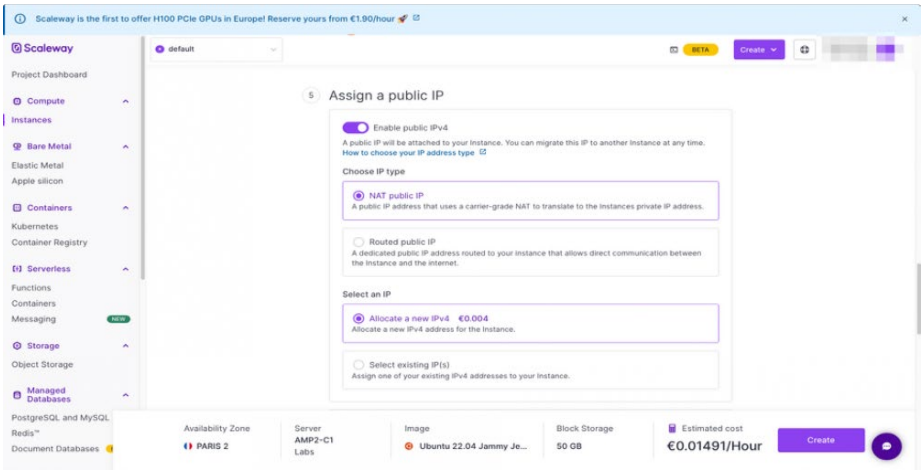
4. In **OS images** section, select **Ubuntu 22.04 Jelly Fish**.



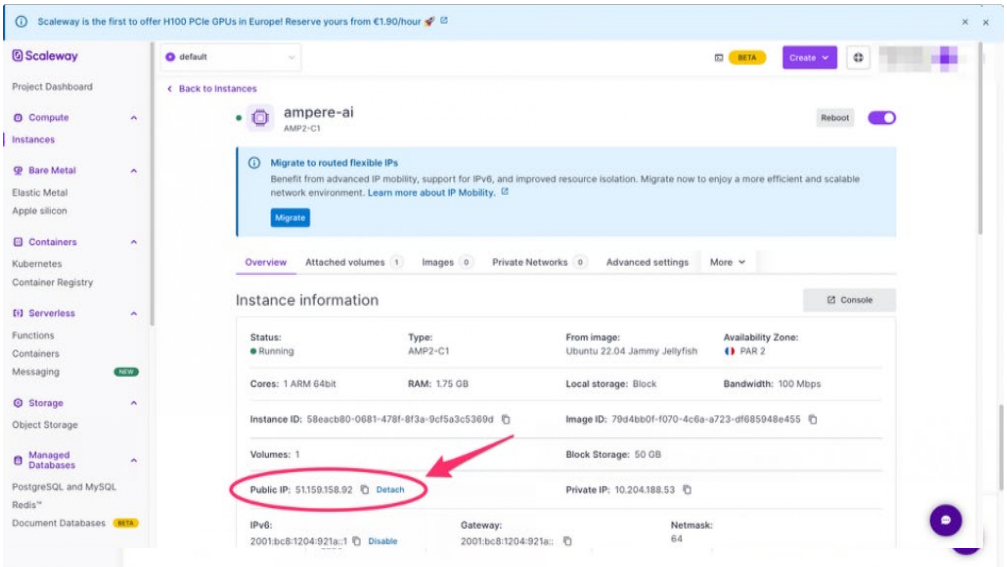

5. Adjust the size of your **System volume** and pick this volume as a **Volume to run your OS from**.



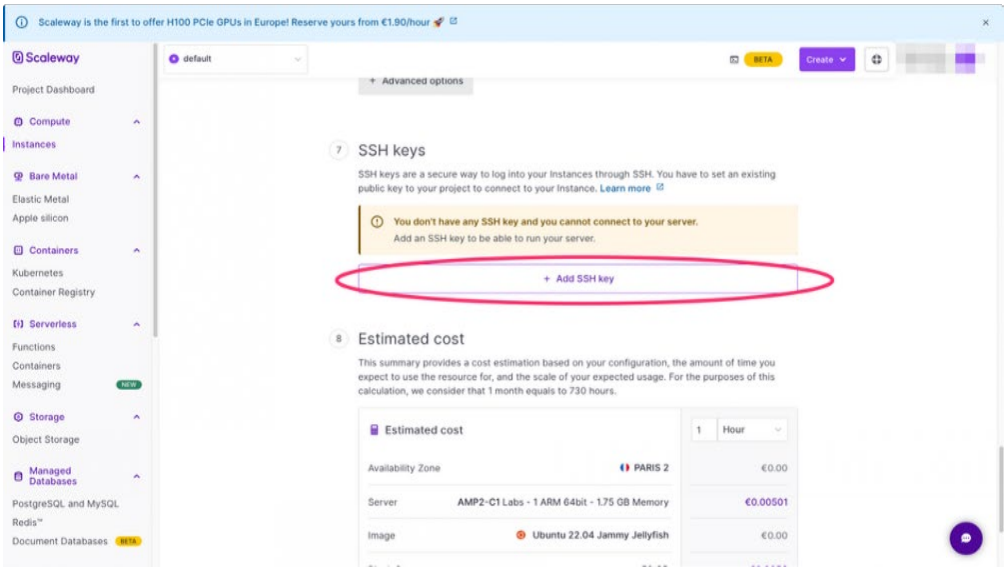

6. Leave IP settings in their default state - assign a public IPv4 address to the Instance.



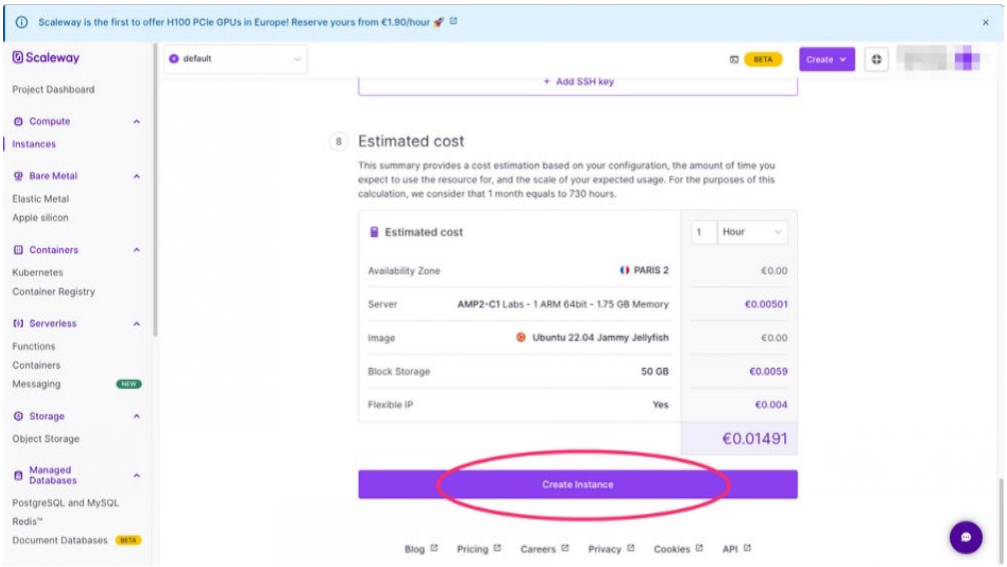
7. Name your Instance arbitrarily.




8. Click on **Add SSH key** and paste your pre-generated SSH key.




9. Validate cost estimation and click on **Create Instance**.




10. Wait for the instance to start and then copy its **Public IP** address.

11. SSH into the system by specifying user **ubuntu** @ public IP address of your Instance.

```
Last login: Fri Nov  3 17:19:08 on ttys002
[redacted] ~ % ssh ubuntu@51.159.158.92
```

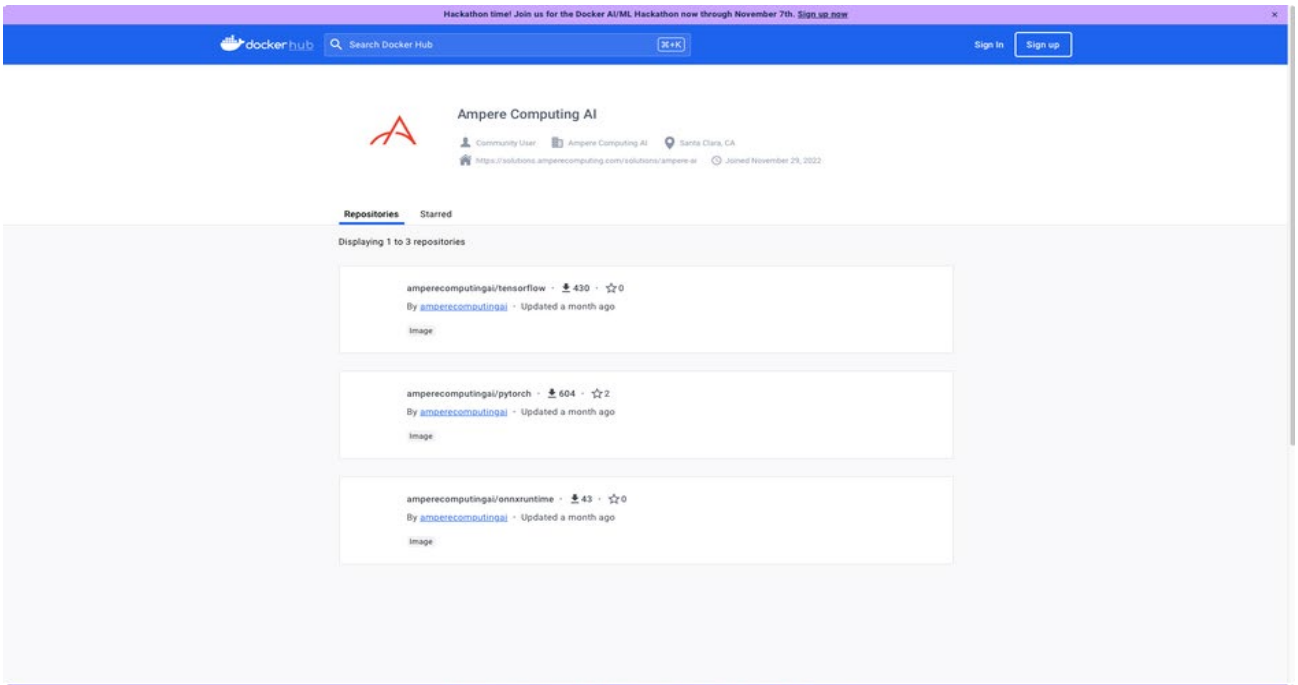
12. After accessing the Instance, run:

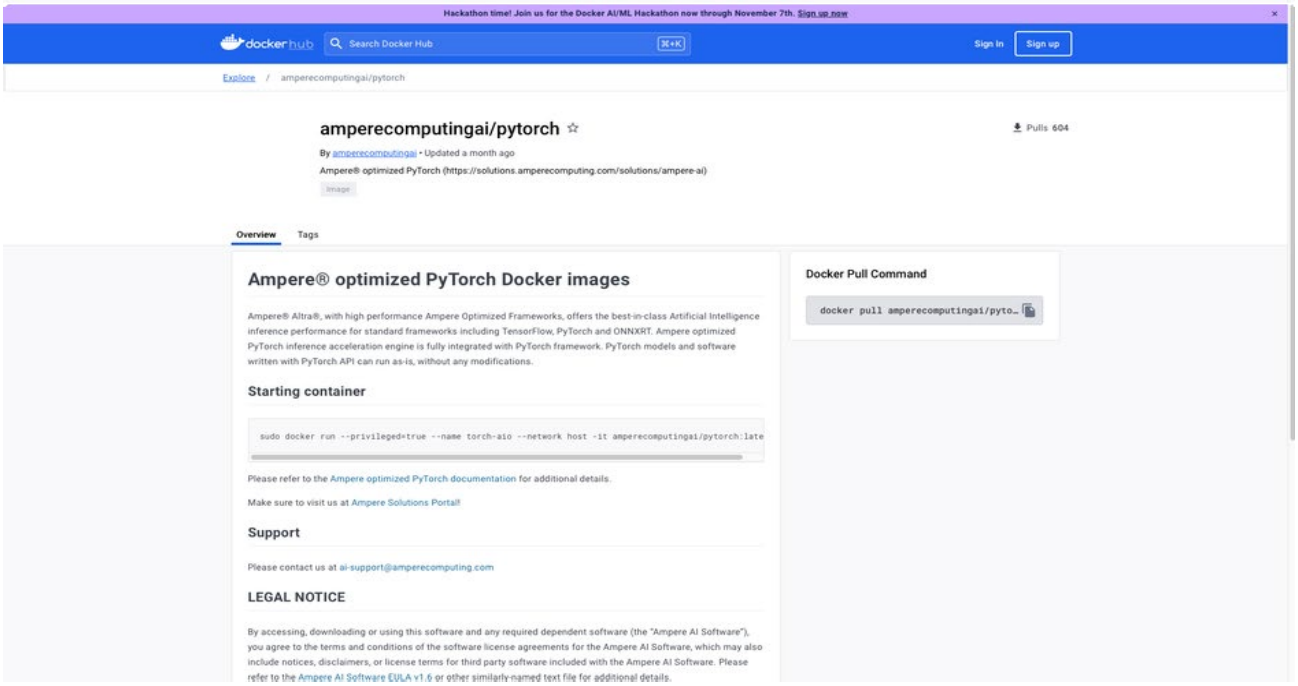
```
sudo apt-get update && sudo apt-get install -y docker.io
```

```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ampere-ai:~$ lscpu
Architecture:            aarch64
  CPU op-mode(s):        32-bit, 64-bit
  Byte Order:            Little Endian
CPU(s):                  1
  On-line CPU(s) list:   0
Vendor ID:               ARM
  Model name:            Neoverse-N1
    Model:               1
    Thread(s) per core:  1
    Core(s) per socket:  1
    Socket(s):           1
    Stepping:            r3p1
    BogoMIPS:            50.00
    Flags:               fp asimd evtstrm aes pmull sha1 sha2 crc32 atomics fphp asimdhp cpuid asimdrdm lrcpc dcpop asimddp ssbs
NUMA:
  NUMA node(s):          1
  NUMA node0 CPU(s):     0
Vulnerabilities:
  Itlb multihit:         Not affected
  L1tf:                  Not affected
  Mds:                   Not affected
  Meltdown:              Not affected
  Mmio stale data:       Not affected
  Retbleed:              Not affected
  Spec store bypass:     Mitigation; Speculative Store Bypass disabled via prctl
  Spectre v1:            Mitigation; __user pointer sanitization
  Spectre v2:            Mitigation; CSV2, BHB
  Srbds:                 Not affected
  Tsx async abort:       Not affected
ubuntu@ampere-ai:~$ sudo apt-get update && sudo apt-get install -y docker.io
```

13. Allow the docker daemon to install. In the meantime go to Ampere AI DockerHub and select your **Ampere optimized framework** of choice.



14. For example, to run Ampere optimized PyTorch image, execute the following snippet in your Instance's shell:

```
sudo docker run --privileged=true --name ampere-torch --network host -it amperecomputingai/pytorch:latest  # o
```

```
Selecting previously unselected package ubuntu-fan.
Preparing to unpack .../7-ubuntu-fan_0.12.16_all.deb ...
Unpacking ubuntu-fan (0.12.16) ...
Setting up dnsmasq-base (2.86-1.1ubuntu0.3) ...
Setting up runc (1.1.7-0ubuntu1~22.04.1) ...
Setting up dns-root-data (2021011101) ...
Setting up bridge-utils (1.7-1ubuntu3) ...
Setting up pigz (2.6-1) ...
Setting up containerd (1.7.2-0ubuntu1~22.04.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /lib/systemd/system/containerd.service.
Setting up ubuntu-fan (0.12.16) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ubuntu-fan.service → /lib/systemd/system/ubuntu-fan.service.
Setting up docker.io (24.0.5-0ubuntu1~22.04.1) ...
Adding group `docker' (GID 121) ...
Done.
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /lib/systemd/system/docker.socket.
Processing triggers for dbus (1.12.20-2ubuntu4.1) ...
Processing triggers for man-db (2.10.2-1) ...
Scanning processes...
Scanning processor microcode...
Scanning linux images...

Running kernel seems to be up-to-date.

Failed to check for processor microcode upgrades.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ampere-ai:~$ sudo docker run --privileged=true --name torch-aio --network host -it amperecomputingai/pytorch:latest
```

15. The image will get pulled automatically, followed by initialization of a docker container. You should see Ampere AI ASCII art. To verify the success, try to import your AI library of choice in python like this:

```
python3 -c "import torch"  # or 'onnxruntime', or 'tensorflow'
```

```
c81ba129bfa8: Pull complete
48547d582fea: Pull complete
b6ee8912bdd5: Pull complete
876c32dc7fd1: Pull complete
e165b5bdc616: Pull complete
464d98940764: Pull complete
5091cb71e717: Pull complete
758e390adbc6: Pull complete
cc28376d26f8: Pull complete
Digest: sha256:7d8e83b1cec60b106120dbdb1d657d5bb1aedc0c247573d7ad0396b432341d5a
Status: Downloaded newer image for amperecomputingai/pytorch:latest

Thank you for choosing AIO!
Please visit us at https://solutions.amperecomputing.com/solutions/ampere-ai

For quick start please run:
 cd /workspace/aio-examples/; bash start_notebook.sh
root@ampere-ai:/workspace# python3 -c "import torch"
```

16. Congratulations - you have successfully deployed Ampere optimized framework on your COP-ARM instance! In case of any issues please reach out to us via email: ai-support@amperecomputing.com