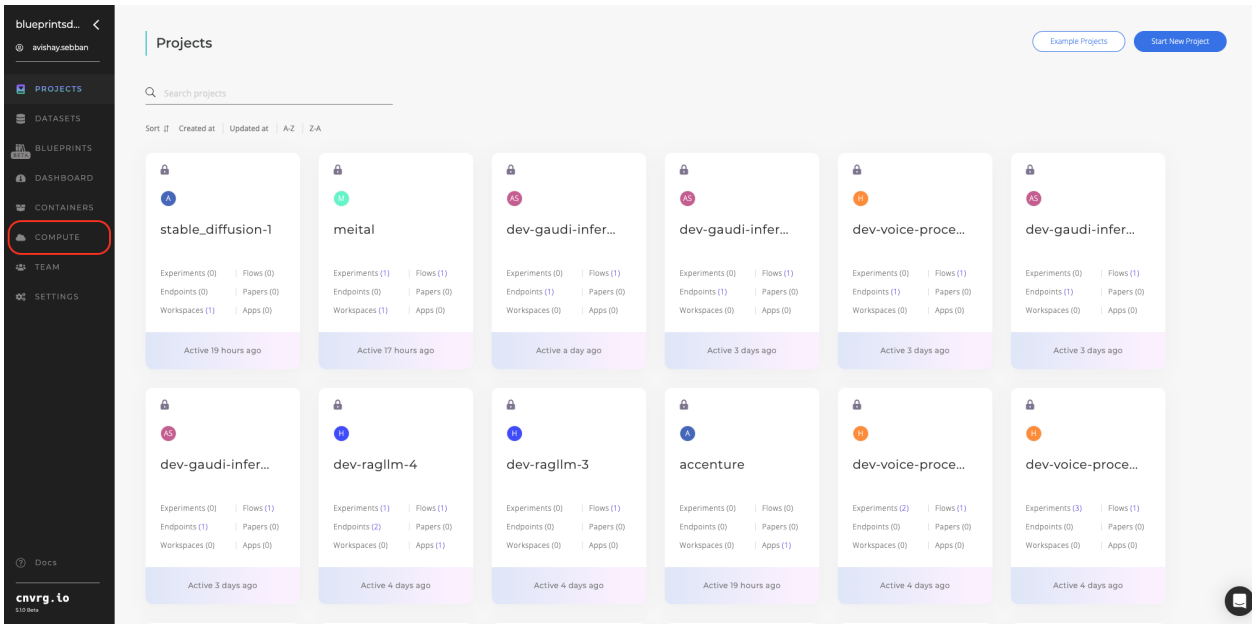
Workshop run stable diffusion in cnvrg.io: a step by step guide with Gaudi

cnvrg.io is a machine learning platform that streamlines the development and deployment of machine learning models. It provides tools for version control, collaboration, model training, and deployment, allowing data scientists and machine learning engineers to focus on building and improving models, rather than worrying about infrastructure and other technical details.

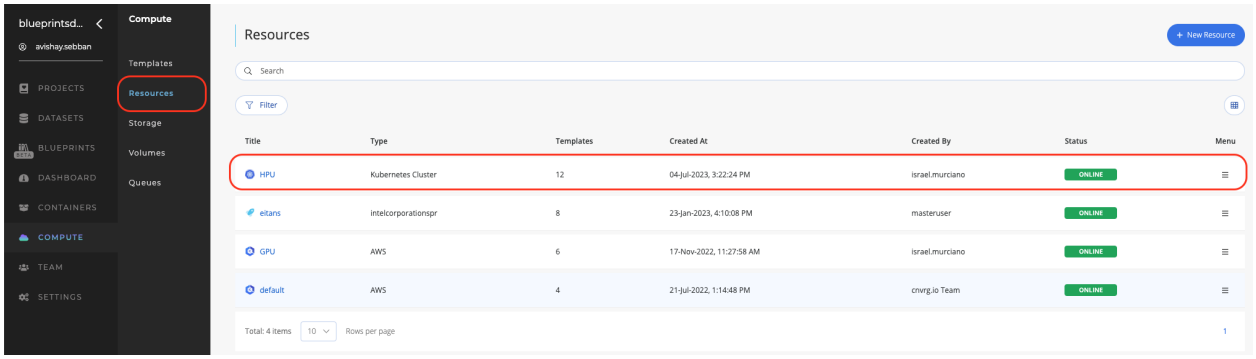
Using the cnvrg.io platform, it is easy to run a stable diffusion model and generate your own custom image.

1. Login to your account.

2. Once you've entered cnvrg.io, click on resource on the left side

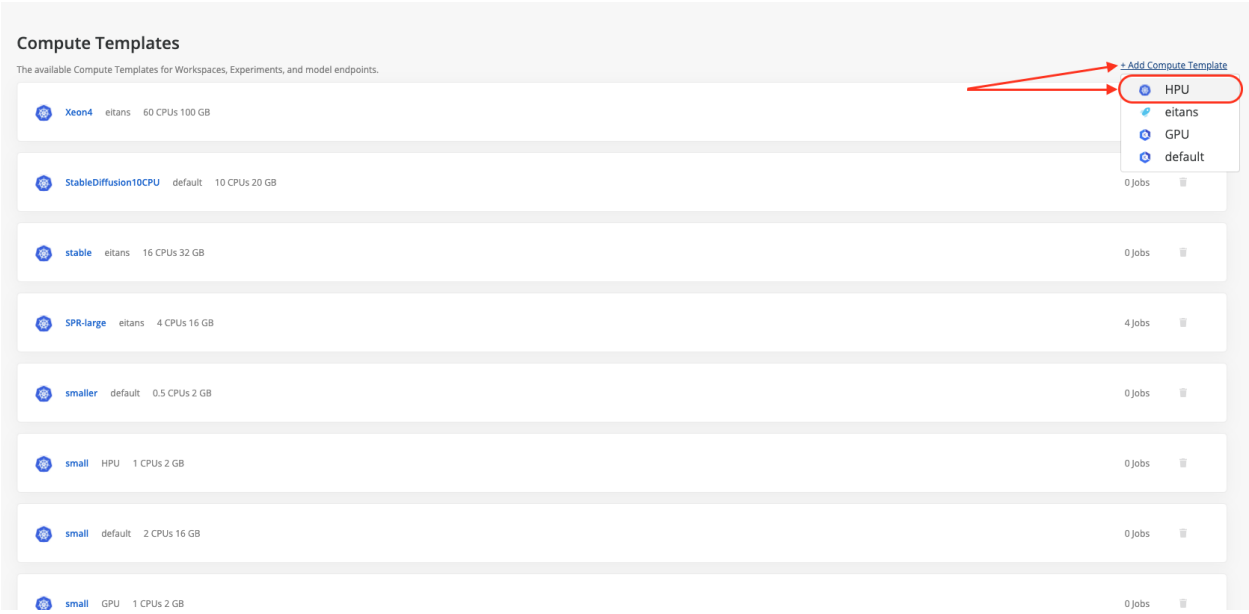


3. In cnvrg.io you are able to connect any kind of workload to the platform. Make sure you have Gaudi HPU integrated in your environment.

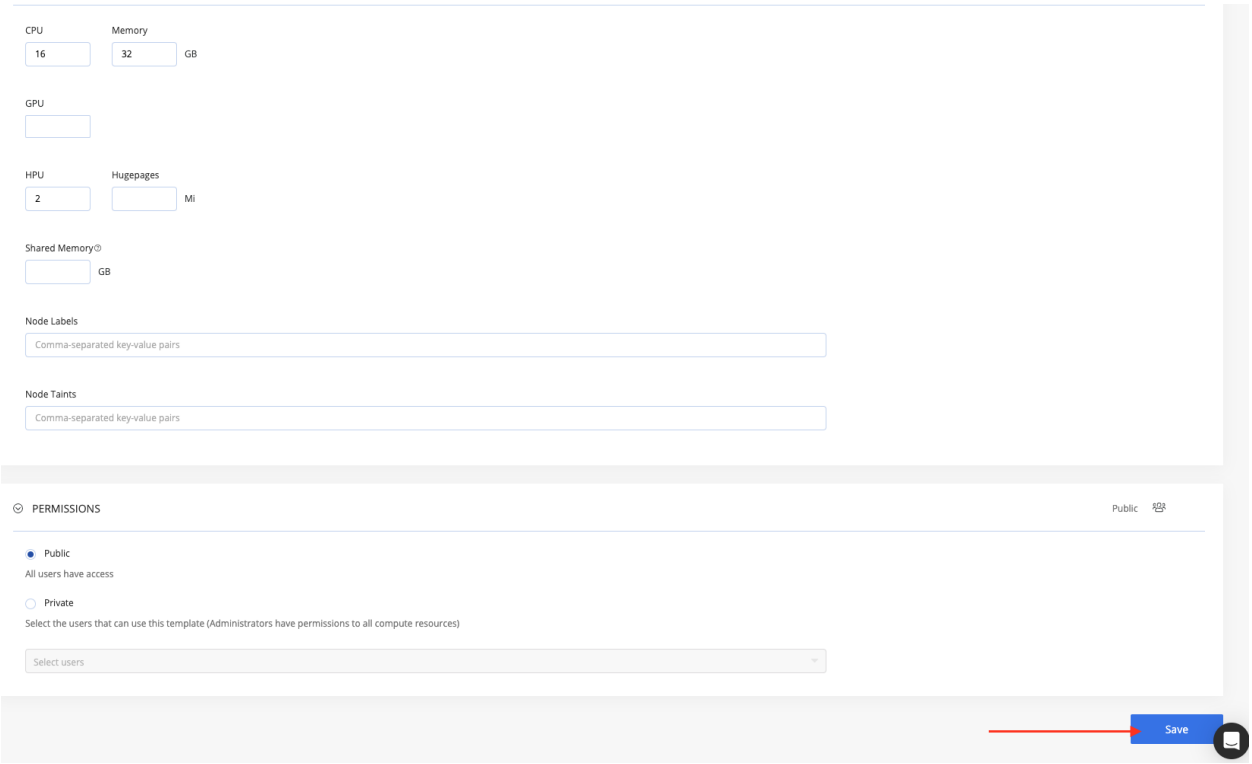


4. Create a template to assign the relevant number of CPU MEMO and HPU.

Make sure you choose the relevant resource to assign for.

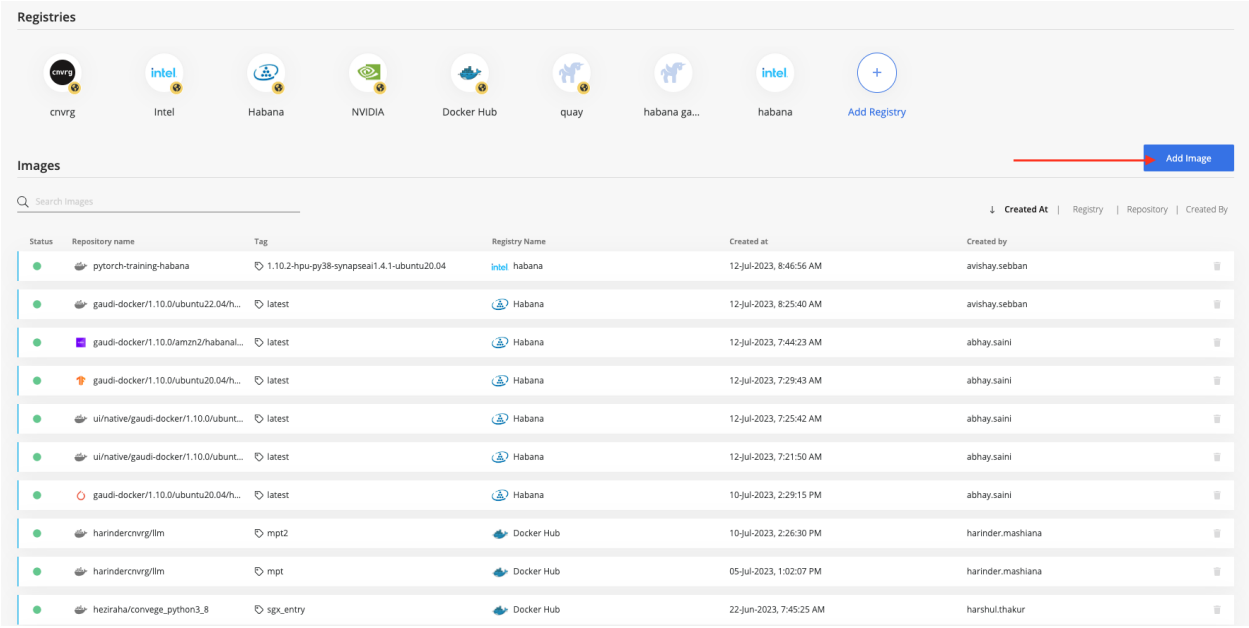


5. Provide input title name - Gaudi Small
Insert the provided details.
CPU: 16
Memory: 32
HPU: 2

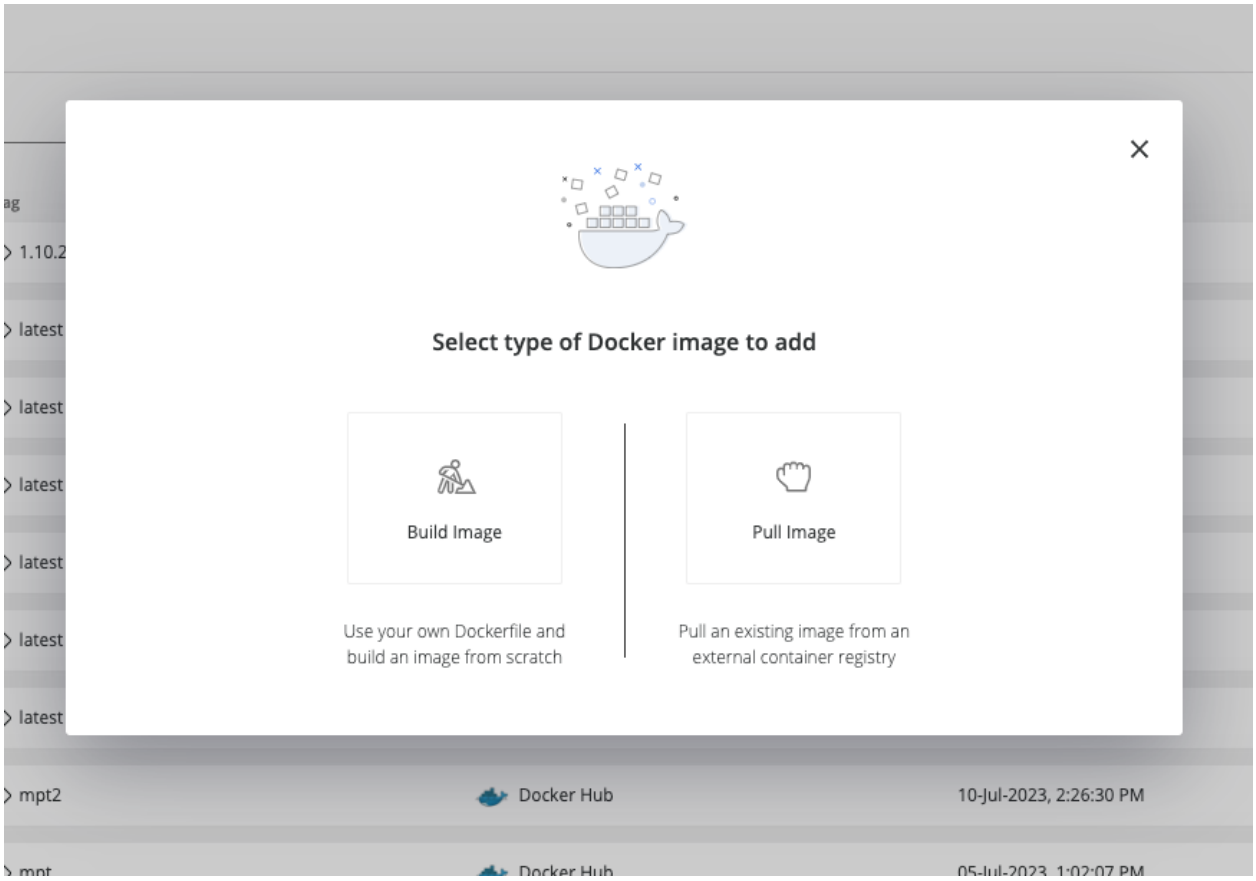


And click on save.

6. Bring your own container
Click on **Add Image**



And choose **Pull image**

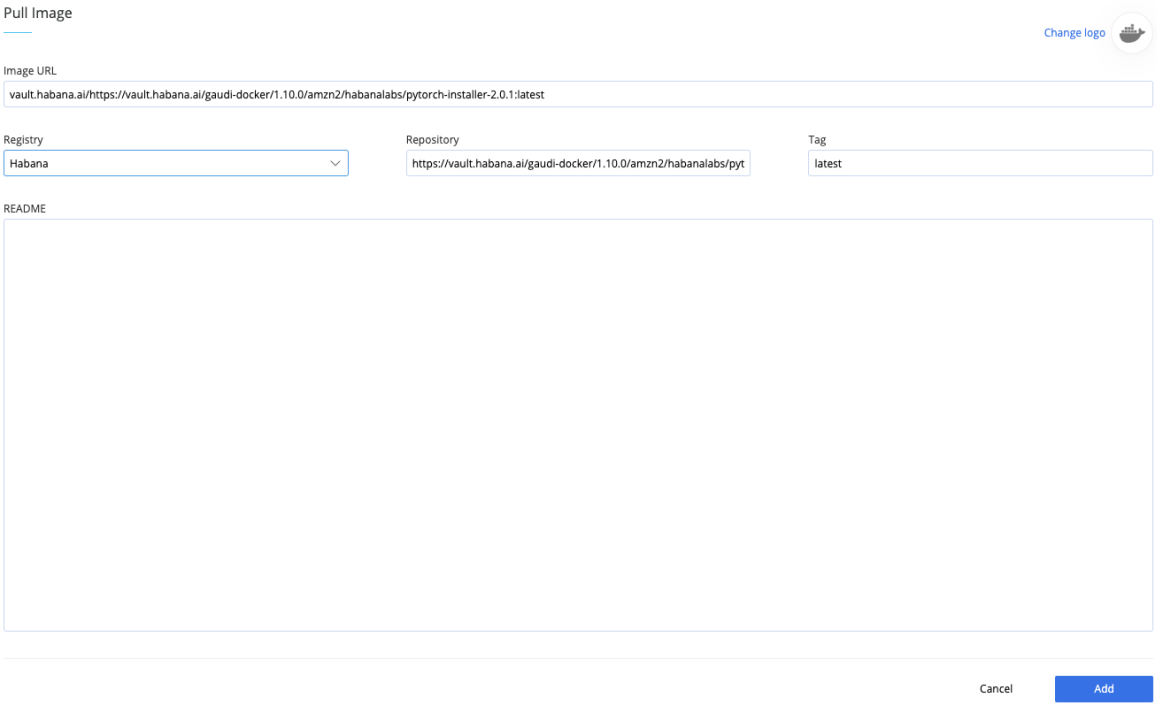


Provide the relevant information:

Image:

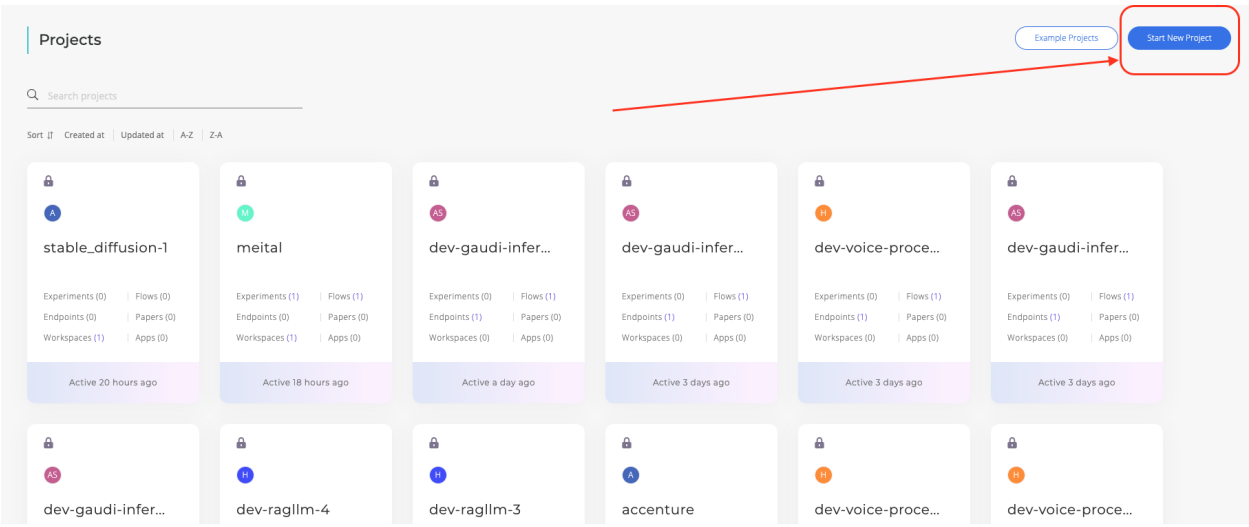
https://vault.habana.ai/gaudi-docker/1.10.0/amzn2/habanalabs/pytorch-installer-2.0.1:latest

choose the relevant **Registry**: Habana

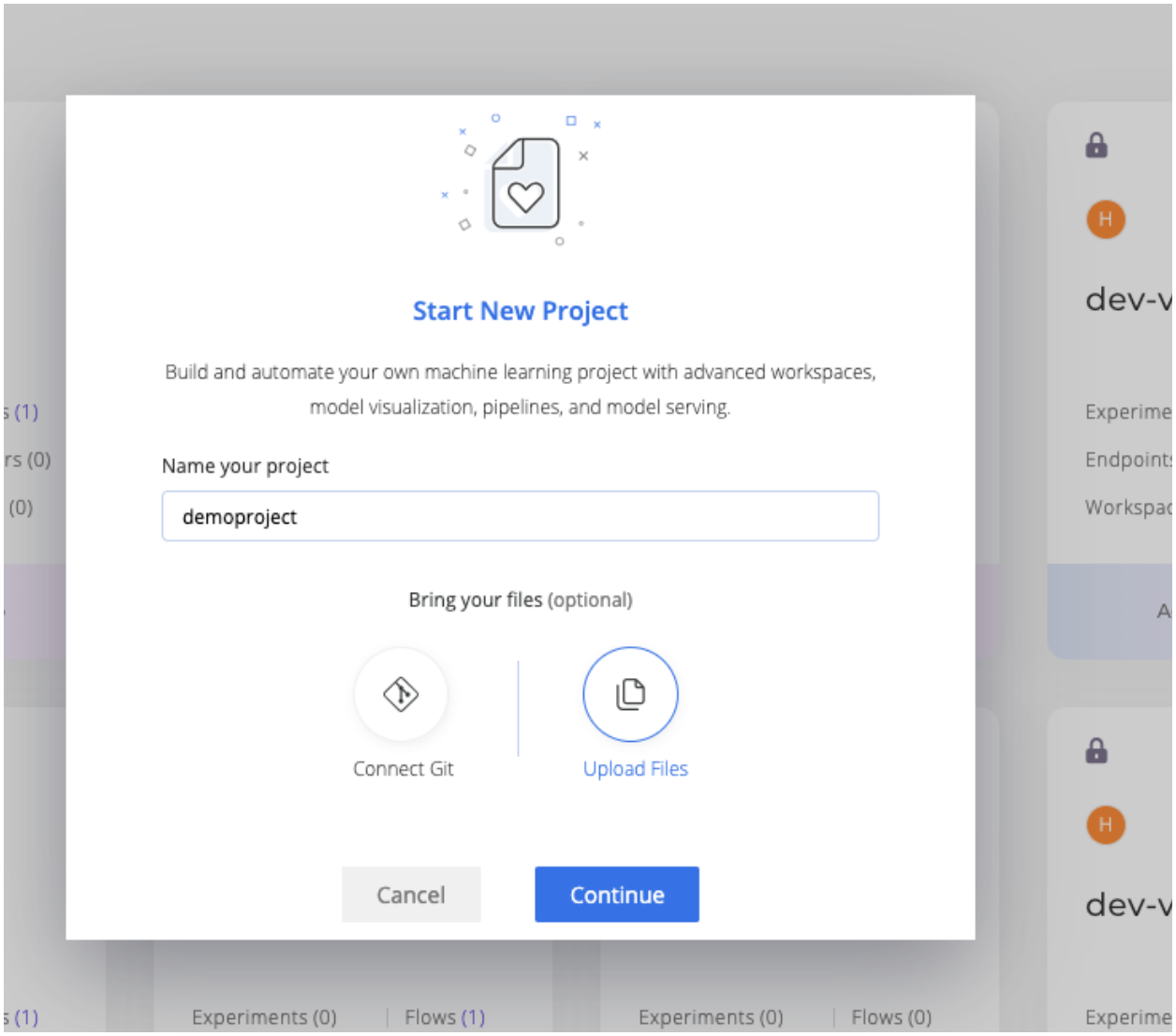


Click on the **Add** button**.**

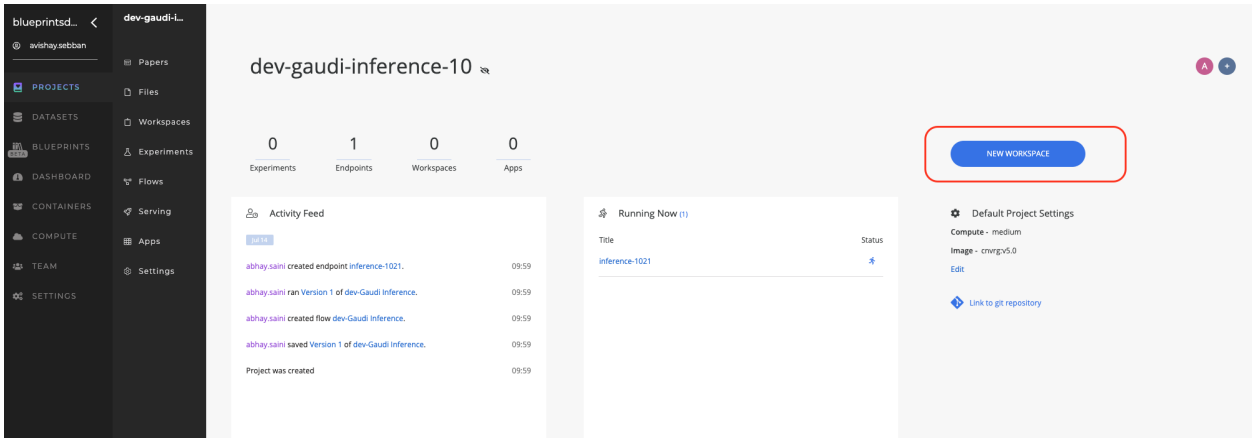
7. Start your project - let's bring the code into a working project.



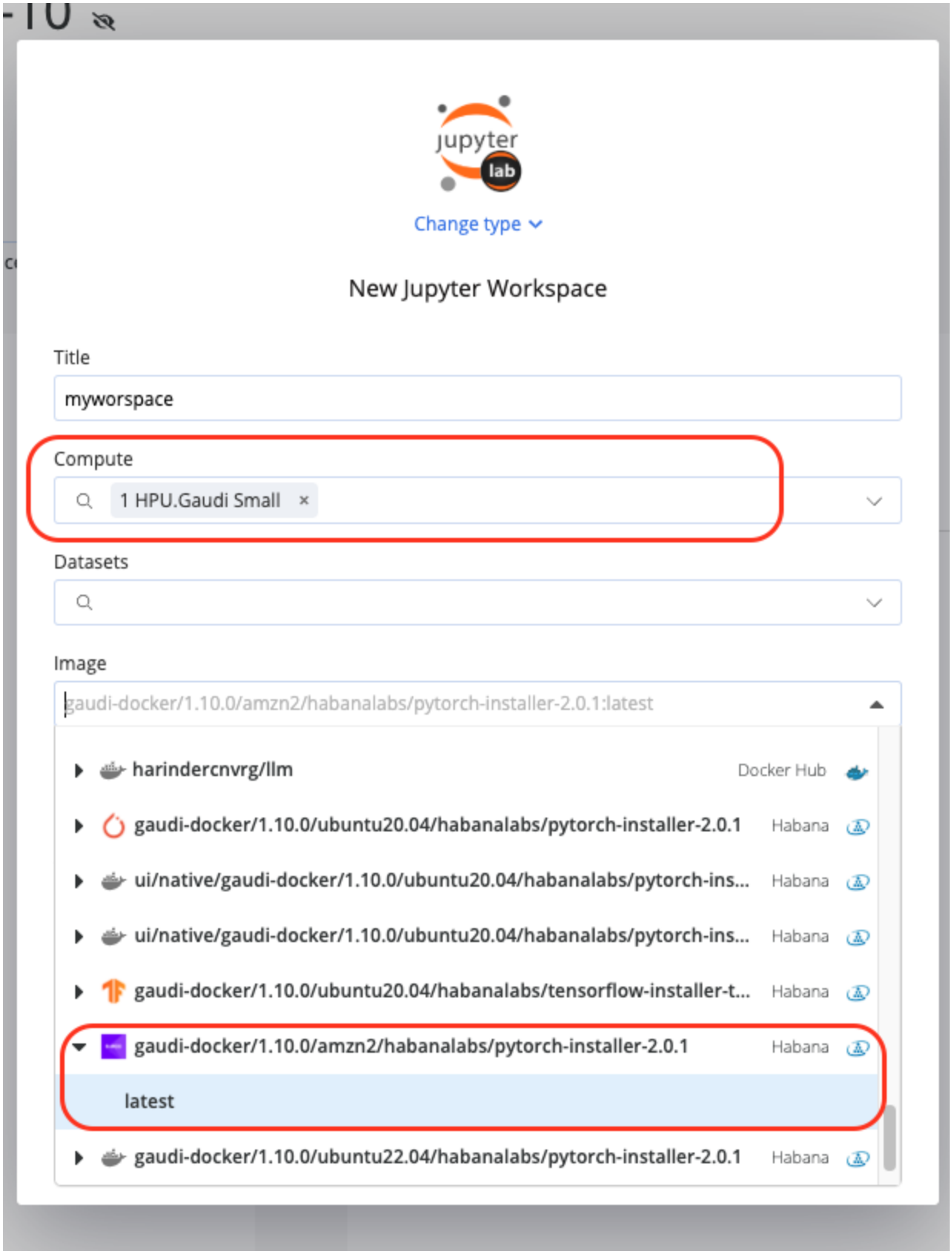
Click on the project on the left side, and “Start new project” Name your project(for example : **demoproject**) and click submit.



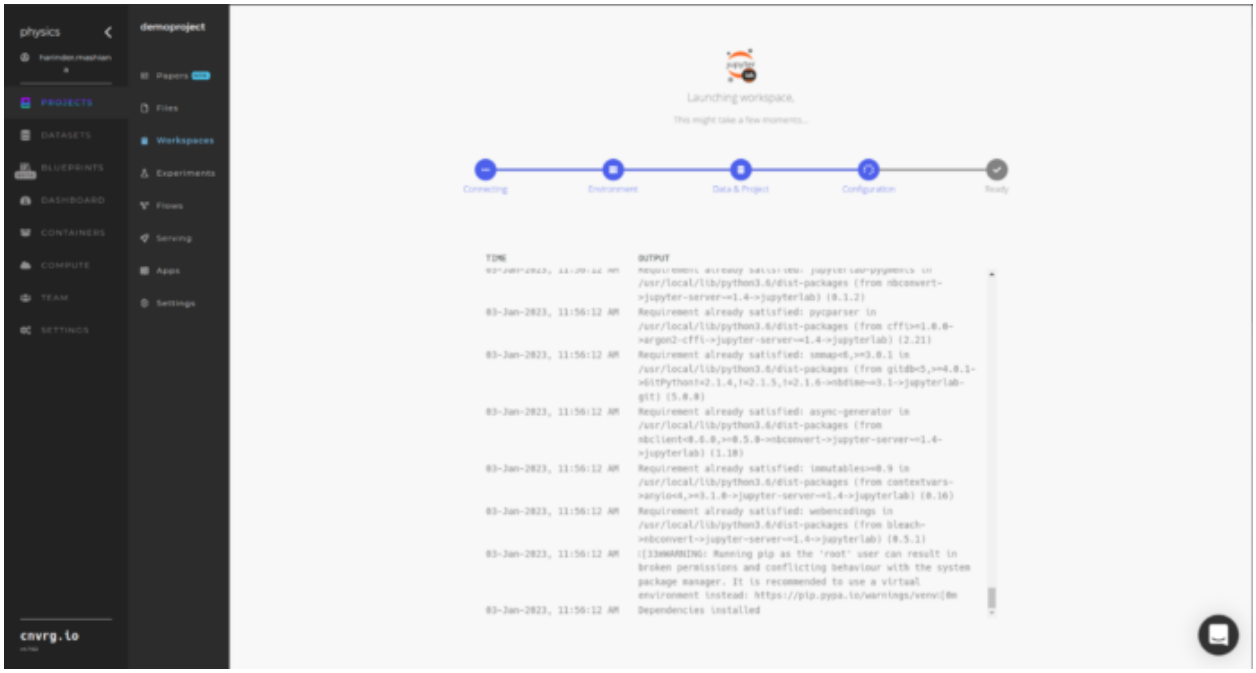
8. Click on the blue ‘new workspace’ button to start a jupyterlab workspace so we can easily create, edit and run code.



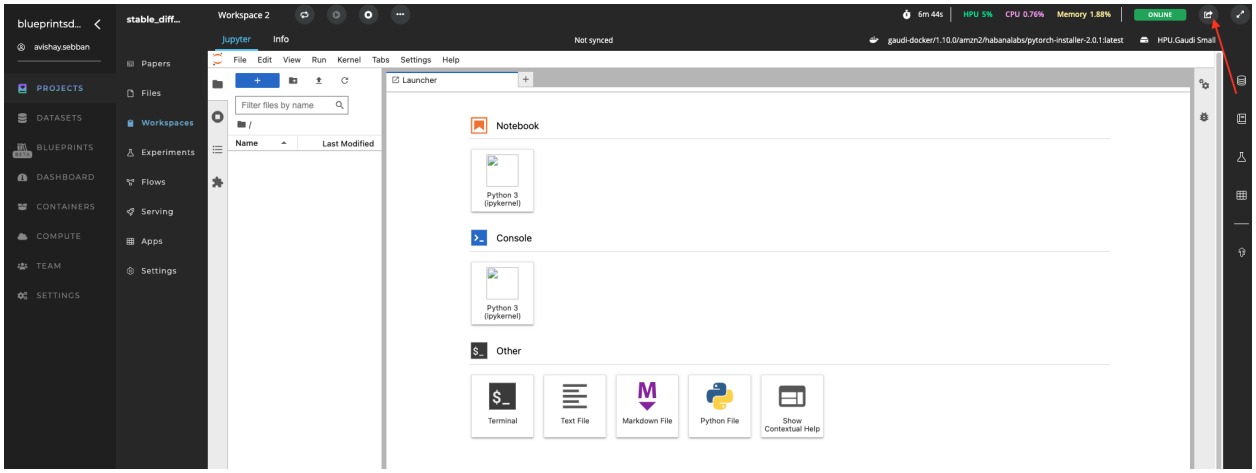
Make sure to choose the relevant **compute** and the right container **image**.


Change type
New Jupyter Workspace
Title
myworspace
Compute
1 HPU.Gaudi Small
Datasets
Image
gaudi-docker/1.10.0/amzn2/habanalabs/pytorch-installer-2.0.1:latest
harindercnvrg/llm
Docker Hub
gaudi-docker/1.10.0/ubuntu20.04/habanalabs/pytorch-installer-2.0.1
Habana
ui/native/gaudi-docker/1.10.0/ubuntu20.04/habanalabs/pytorch-ins...
Habana
ui/native/gaudi-docker/1.10.0/ubuntu20.04/habanalabs/pytorch-ins...
Habana
gaudi-docker/1.10.0/ubuntu20.04/habanalabs/tensorflow-installer-t...
Habana
gaudi-docker/1.10.0/amzn2/habanalabs/pytorch-installer-2.0.1
Habana
latest
gaudi-docker/1.10.0/ubuntu22.04/habanalabs/pytorch-installer-2.0.1
Habana

9. Next, the workspace will start loading and you will see a screen like this:



Once the workspace is up please click on "**open in a new tab**" icon



10. Now click on the **terminal** option to start a terminal so you can clone the git repository for stable diffusion.

The first thing you will want to do when running your script is to verify that your model actually runs on the Gaudi accelerator.
The Gaudi runtime environment includes the **hl-smi** tool which reports resource utilization of the Gaudi cores.

Run the following commands

> hl-smi

This is how is look like when you allocate 1 HPU

```
sh-4.2# hl-smi
+-----------------------------------------------------------------------------+
| HL-SMI Version:                              hl-1.10.0-fw-43.2.0.0          |
| Driver Version:                                     1.10.0-416d95e          |
|-------------------------------+----------------------+----------------------+
| AIP  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | AIP-Util  Compute M. |
|===============================+======================+======================|
|   0  HL-205              N/A  | 0000:90:1e.0     N/A |                    0 |
| N/A   56C   N/A   109W / 350W |    512MiB / 32768MiB |     5%           N/A |
|-------------------------------+----------------------+----------------------+
| Compute Processes:                                               AIP Memory |
|  AIP       PID   Type   Process name                             Usage      |
|=============================================================================|
|   0        N/A   N/A    N/A                                      N/A        |
+=============================================================================+
```

And this is how is look like when you allocate 2 HPU


```
sh-4.2# hl-smi
+-----------------------------------------------------------------------------+
| HL-SMI Version:                              hl-1.10.0-fw-43.2.0.0          |
| Driver Version:                                     1.10.0-416d95e          |
|-------------------------------+----------------------+----------------------+
| AIP  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | AIP-Util  Compute M. |
|===============================+======================+======================|
|   0  HL-205              N/A  | 0000:10:1d.0     N/A |                    0 |
| N/A   52C   N/A   102W / 350W |    512MiB / 32768MiB |     1%           N/A |
|-------------------------------+----------------------+----------------------+
|   1  HL-205              N/A  | 0000:a0:1d.0     N/A |                    0 |
| N/A   58C   N/A   108W / 350W |    512MiB / 32768MiB |     4%           N/A |
|-------------------------------+----------------------+----------------------+
| Compute Processes:                                               AIP Memory |
|  AIP       PID   Type   Process name                             Usage      |
|=============================================================================|
|   0        N/A   N/A    N/A                                      N/A        |
|   1        N/A   N/A    N/A                                      N/A        |
+=============================================================================+
```

For the subsequent steps, please refer to the detailed instructions provided in the following link.