How to deploy a NIM model in PC-AI – for beginners.

In this document I will show how to deploy a model in PC-AI using MLIS.

For this example, will use the NVIDIA-NIM vista 3D model (https://build.nvidia.com/nvidia/vista-3d)

Requirements:

Create an API key to be able to use the service, for NVIDIA NIM you need an NGC API key, for Hugging face you need a HF API key and so forth.

Overall steps:

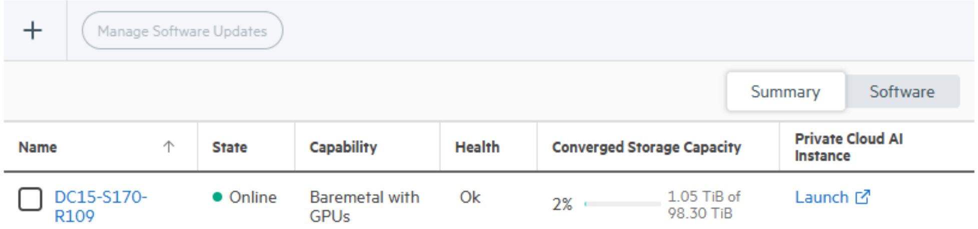
During the process we will complete the following steps – I will comment on each step below.

0) **MLIS** – access MLIS within Private Cloud - AI
1) **Registries** - if there is no registry for NVIDIA-NIM models, you need to create one. In this example, I will create fra-onboarding
2) **Packaged models** – need to package the model. In this example, I will package NVIDIA-NIM vista 3D model under the name vista-3d-fra
3) **Deployments** – need to create a model deployment – In this example, the model wil be deployed under fra-onboarding name
4) **API-tokens** – need to generate an API token. This is needed to be able to make post requests to the model – in the example the api token will be called fra-onboarding.

Detailed step – by - step guide:

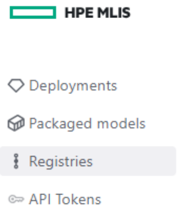
0) MLIS

- Go to **https://common.cloud.hpe.com/** and login.
- Choose a workspace
- Under **Featured Services** look for the **catalog** and find **Private Cloud A**I -> **Launch**
- In the dashboard you see a summary, the click on systems



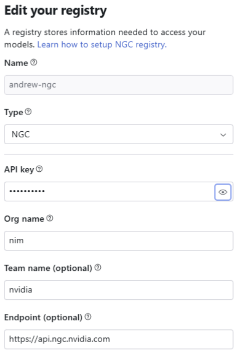
Name	State	Capability	Health	Converged Storage Capacity	Private Cloud AI Instance
☐ DC15-S170-R109	● Online	Baremetal with GPUs	Ok	2% 1.05 TiB of 98.30 TiB	Launch

- Click on Launch.
- On the left look for tool and frameworks then under data science Open HPE MLIS



1) Registries

Example of a NGC registry (zoom the images when needed)

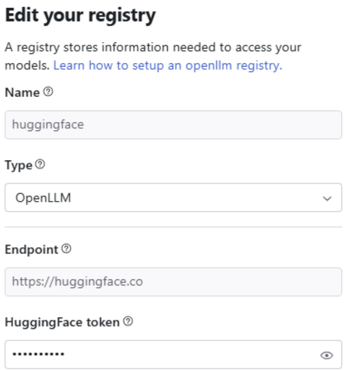


Edit your registry

A registry stores information needed to access your models. Learn how to setup NGC registry.

Name	andrew-ngc
Type	NGC
API key	••••••••••
Org name	nim
Team name (optional)	nvidia
Endpoint (optional)	https://api.ngc.nvidia.com

Example of an Hugging face registry (zoom the images when needed



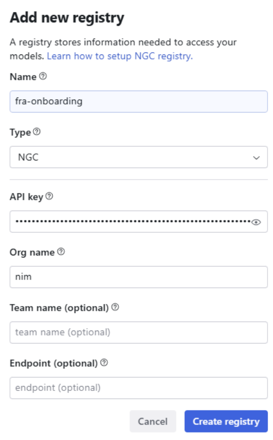
Edit your registry

A registry stores information needed to access your models. Learn how to setup an openllm registry.

Name	huggingface
Type	OpenLLM
Endpoint	https://huggingface.co
HuggingFace token	••••••••••

Create a registry

Click on `add new registry`



Add new registry

A registry stores information needed to access your models. Learn how to setup NGC registry.

Name: fra-onboarding

Type: NGC

API key: ••

Org name: nim

Team name (optional): team name (optional)

Endpoint (optional): endpoint (optional)

Name: your choice of name

Type: Since I am looking for models in NGC, In the registry type I chose NGC.

API key: The API KEY is your NGC API-KEY

Org name: I am looking for nim models, so I wrote **nim** as Org name (I noticed that if I used a different name I would not find nims model when I go to the next step – model package)

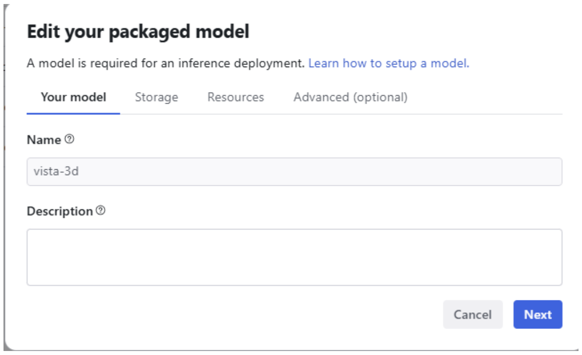
Team name **nvidia** and **Endpoint** since they are optional, they can be left empty and they will be automatically compiled.

Now that the registry is ready, I can package a model.

2) Packaged models

Click on **Add new model**

Your model: Chose the name and the description



Edit your packaged model

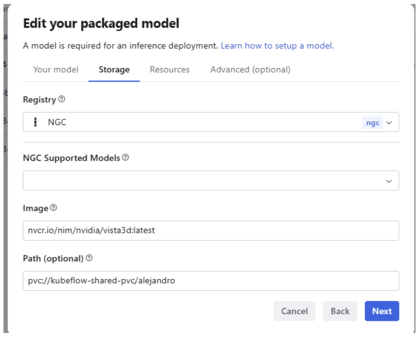
A model is required for an inference deployment. Learn how to setup a model.

Name: vista-3d

Description:

Storage: you should see the NIM model you are interested on in the drop down list. The image should self compile, **Path** is the field you can use if you'd like the model to be downloaded/cached once and stay in the persistent volume claim (PVC).

In this case if you type pvc://Kubeflow-shared-pvc/francesco it will create a folder there if it doesn't exist and the model will be cached there.



Edit your packaged model

A model is required for an inference deployment. Learn how to setup a model.

Registry: NGC

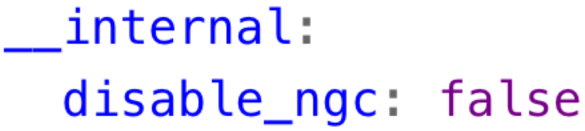
NGC Supported Models:

Image: nvcr.io/nim/nvidia/vista3d:latest

Path (optional): pvc://kubeflow-shared-pvc/alejandro

Note: If you don't see NVIDIA Vista among the models, you need to make sure that MLIS was deployed with disable_ngc set to false.

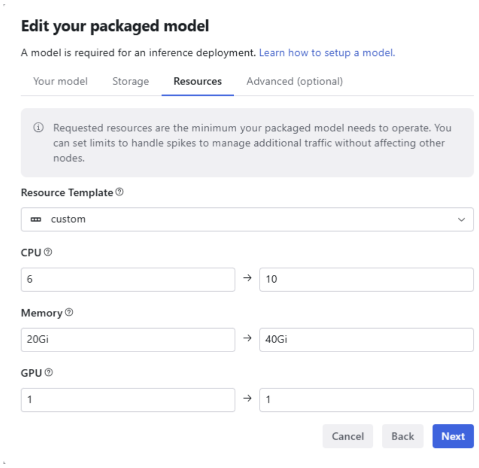
In AIE go to Tools and Frameworks, data science, MLIS config. At the very bottom



```
__internal:
  disable_ngc: false
```

Resources: This is an example for NVIDIA NIM VISTA3D, which is already available – zoom images as needed.

These are Alejandro's config for the model when it comes to deciding the resources



Edit your packaged model

A model is required for an inference deployment. Learn how to setup a model.

ⓘ Requested resources are the minimum your packaged model needs to operate. You can set limits to handle spikes to manage additional traffic without affecting other nodes.

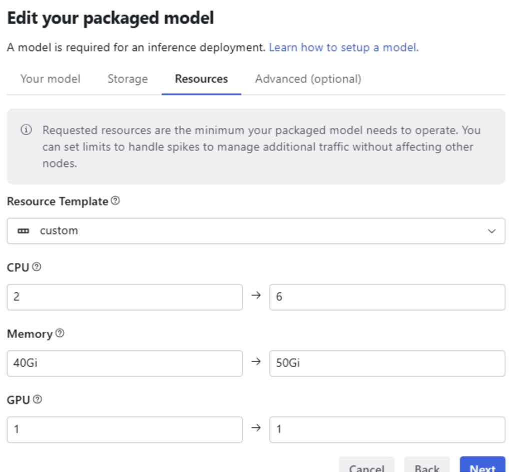
Resource Template: custom

CPU: 6 → 10

Memory: 20Gi → 40Gi

GPU: 1 → 1

These are Andrew’s:



Edit your packaged model

A model is required for an inference deployment. Learn how to setup a model.

ⓘ Requested resources are the minimum your packaged model needs to operate. You can set limits to handle spikes to manage additional traffic without affecting other nodes.

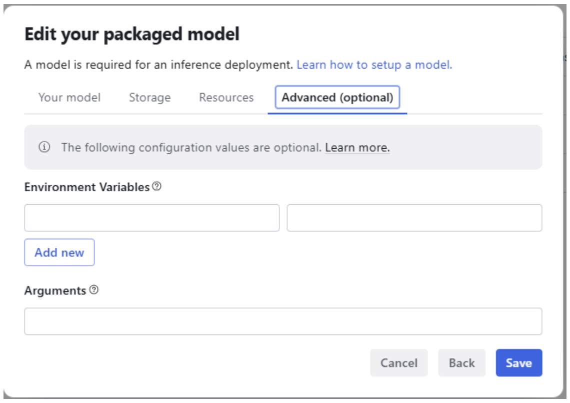
Resource Template: custom

CPU: 2 → 6

Memory: 40Gi → 50Gi

GPU: 1 → 1

ADVANCED (optional): Alejandro left it empty



Edit your packaged model

A model is required for an inference deployment. Learn how to setup a model.

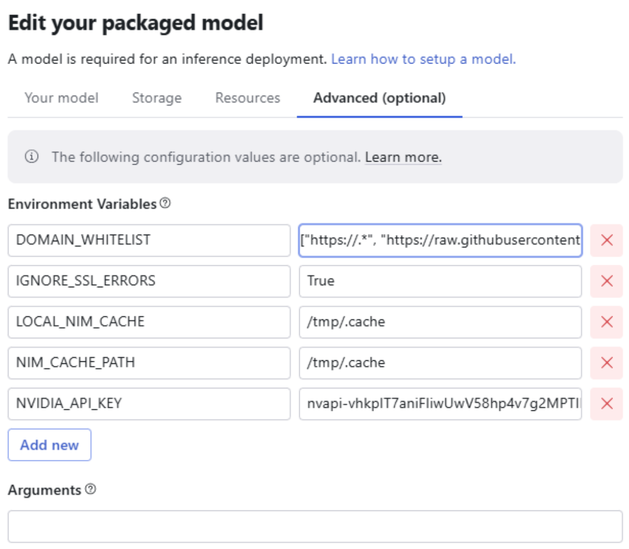
ⓘ The following configuration values are optional. Learn more.

Environment Variables

Arguments

while Andrew didn’t

And used variables from https://docs.nvidia.com/nim/medical/vista3d/latest/advanced-usage.html



Edit your packaged model

A model is required for an inference deployment. Learn how to setup a model.

ⓘ The following configuration values are optional. Learn more.

Environment Variables

DOMAIN_WHITELIST	["https://.*", "https://raw.githubusercontent
IGNORE_SSL_ERRORS	True
LOCAL_NIM_CACHE	/tmp/.cache
NIM_CACHE_PATH	/tmp/.cache
NVIDIA_API_KEY	nvapi-vhkplT7aniFliwUwV58hp4v7g2MPTI

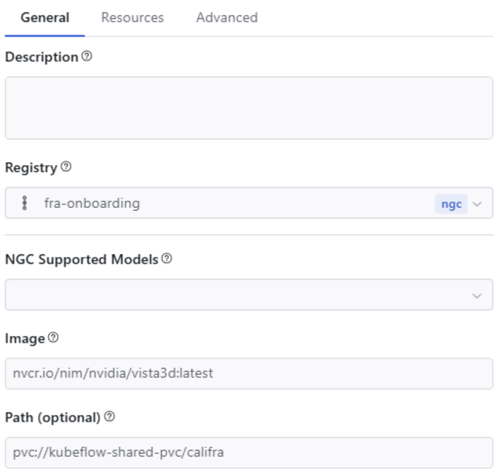
Arguments

I noticed that if I left https://.*.ingress.pcai0109.dc15.hpecolo.net/.* out of the DOMAIN_WHITELIST hence, using the default values I was not able to make post requests to the model, and do inference, so I**decided to use the DOMAIN_WHITELIST like Andrew and included** left https://.*.ingress.pcai0109.dc15.hpecolo.net/.*

DOMAIN_WHITELIST = ["https://.*", "https://raw.githubusercontent.com/NVIDIA/.*", "https://assets.ngc.nvidia.com/products/api-catalog/vista3d/.*", "https://storage.googleapis.com/.*", "https://.*.s3.amazonaws.com/.*", "https://.*.blob.core.windows.net/.*",**"https://.*.ingress.pcai0109.dc15.hpecolo.net/.*"**]

Below are my settings:

Response:



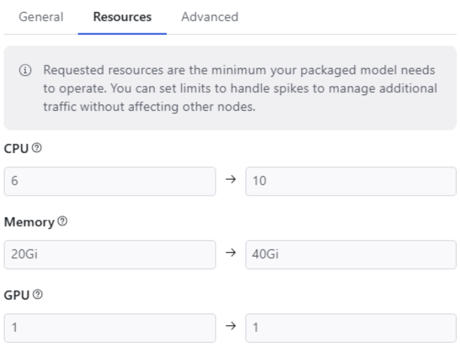
Description

Registry
fra-onboarding ngc

NGC Supported Models

Image
nvcr.io/nim/nvidia/vista3d:latest

Path (optional)
pvc://kubeflow-shared-pvc/califra

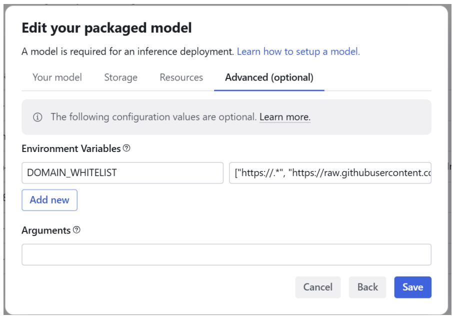


Requested resources are the minimum your packaged model needs to operate. You can set limits to handle spikes to manage additional traffic without affecting other nodes.

CPU
6 → 10

Memory
20Gi → 40Gi

GPU
1 → 1



Edit your packaged model

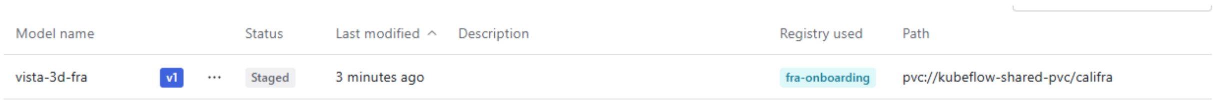
A model is required for an inference deployment. Learn how to setup a model.

The following configuration values are optional. Learn more.

Environment Variables
DOMAIN_WHITELIST ["https://.*", "https://raw.githubusercontent.cc

Arguments

Now that the model is added it will appear as staged



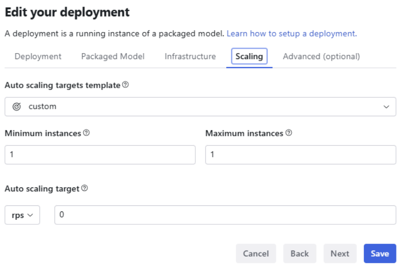
Model name			Status	Last modified ^	Description	Registry used	Path
vista-3d-fra	v1	⋯	Staged	3 minutes ago		fra-onboarding	pvc://kubeflow-shared-pvc/califra

Deployments

Click on **create a new deployment**.

- **Deployment:** Choose a deployment name and description

- **Packaged Model:** Select the packaged model you'd like to deploy, and its version
- **Infrastructure:** Leave endpoint security on under infrastructure
- **Scaling:** Select the auto scaling target template. Here I selected the same that Andrew had used:
  - Autoscaling targets template **custom,**
  - minimum instance **1**
  - maximum instances **1**
  - auto scaling target **rps 0**



Edit your deployment

A deployment is a running instance of a packaged model. Learn how to setup a deployment.

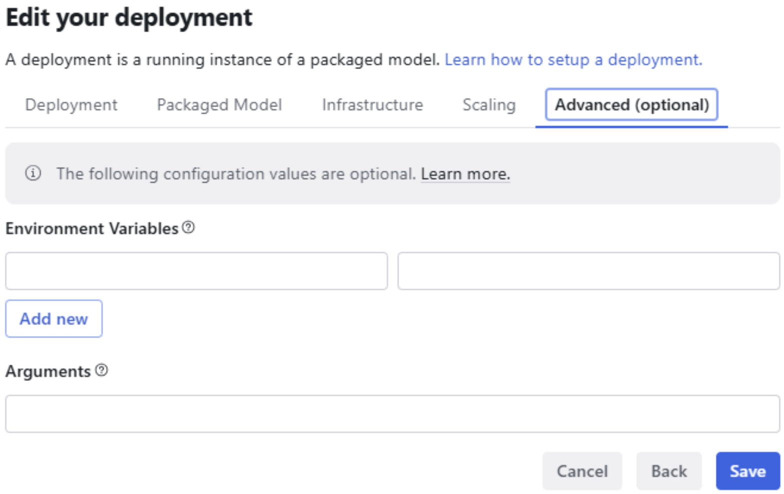
Auto scaling targets template: custom

Minimum instances: 1

Maximum instances: 1

Auto scaling target: rps 0

- **Advanced (optional) -** in this was left empty



Edit your deployment

A deployment is a running instance of a packaged model. Learn how to setup a deployment.

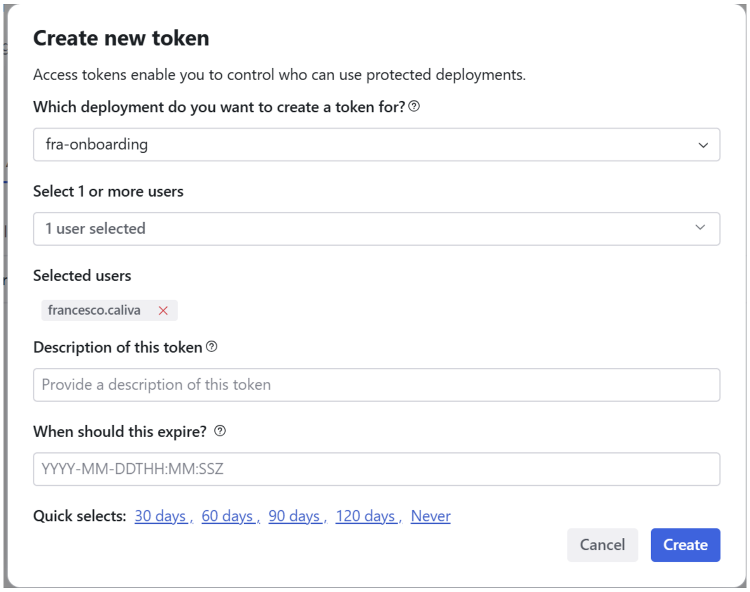
The following configuration values are optional. Learn more.

Environment Variables

Arguments

- **API Tokens**

  Create new API access token.



Create new token

Access tokens enable you to control who can use protected deployments.

Which deployment do you want to create a token for? ⓘ

fra-onboarding

Select 1 or more users

1 user selected

Selected users

francesco.caliva ✕

Description of this token ⓘ

Provide a description of this token

When should this expire? ⓘ

YYYY-MM-DDTHH:MM:SSZ

Quick selects: 30 days , 60 days , 90 days , 120 days , Never

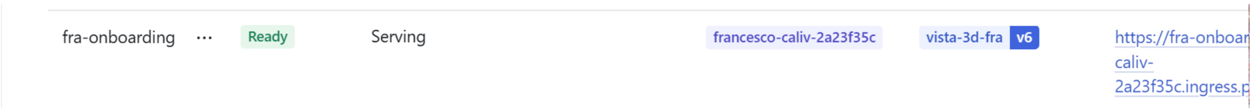
This API key is the key which you will use when sending post requests to the model for instance:

````
` ` `
base_url="https://fra-onboarding-predictor-francesco-caliv-2a23f35c.ingress.pcai0109.dc15.hpecolo.net"

mlis_token =
"eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE3NTcyODQwMTcsImlhdCI6M
Tc1NDY5MjAxOCwiaXNzIjoiYWlvbGlAaHBlLmNvbSIsInN1YiI6ImQ3MGZiYjU4LTdmY
mItNDcwNC1hZjRkLWFjZjAwMTYxYjhhMCIsInVzZXIiOiJhZG1pbiJ9.IuXF-
gZ5UiROspdBKPw1XZyb-9mO-
zPV13Cq6wnYjoKyu1ub5dVpeuthVVYQePRmaw8iV3sHAJkc3g3Dqx6jSkWTHZsGhl
wnnKK5lBtNm0L2ApHQAFuD7sQvbFigJ3eGf2Mi3Sm8NcNIQDTvCiERvRbQXYe6S8J
Q1GhfFv3I3cLU5xM8WnCtlBugRJeMp9_DPUfaZtdJj738FB0Pdnio8D19yVcHLXvOqO
3ordX8enLQs8Wq3sJXNC5ZLVG5TtUVh_qEJL7y9EloPbMZwGUL2Zq8Ytodvxz9N3qjR
8E0_utATJ38SMq_0ubC9nCC0juZGOLaaxQ9RtyfKE_BMCrAaA"

headers = {'Authorization': f'Bearer {mlis_token}'}"
response = requests.post(
        f'{base_url}/v1/vista3d/inference',
        json=data, headers=headers
)
````

` ` `

Ideally it is recommended to add the api-key in a .env file and then load it to memory using dotenv.load_dotenv() function.

Wait until it is ready and serving



fra-onboarding ··· Ready Serving francesco-caliv-2a23f35c vista-3d-fra v6 https://fra-onboar
caliv-
2a23f35c.ingress.p