



# NGC Private Registry

## User Guide

# Table of Contents

Chapter 1. NGC Private Registry for Enterprise.....	1
Chapter 2. Getting Started.....	2
2.1. Obtaining a Private Registry.....	2
2.2. Activating a New NGC Account.....	5
2.2.1. Joining an NGC Org or Team with an Existing NVIDIA Account.....	6
2.2.2. Joining an Org or Team with a New NVIDIA Account.....	9
2.2.3. Joining an Org as Org Owner.....	16
2.2.4. Joining an Org or Team with an External SSO Company Account.....	23
2.2.5. Switching Orgs or Team After Logging into NGC.....	27
2.3. Generating Your NGC API Key.....	27
2.4. Managing Users and Teams in NGC.....	29
2.4.1. NGC Registry User Roles.....	29
2.4.2. Creating Teams.....	30
2.4.3. Creating Users.....	31
2.4.4. Adding a new User to a Team.....	32
2.4.5. Adding an Existing User to a Team.....	33
2.4.6. Changing User Roles.....	34
2.5. Introduction to the NGC Catalog and NGC CLIs.....	35
2.5.1. Installing NGC Registry CLI.....	36
2.5.2. Managing Users and Teams.....	36
2.5.2.1. Inviting users to the organization’s NGC account.....	37
2.5.2.2. Creating teams.....	37
2.5.2.3. Adding users to teams.....	37
2.5.2.4. Creating a team and adding a user in the same command.....	38
2.5.2.5. Creating a team and adding a user in the same command.....	38
Chapter 3. Docker Containers.....	41
3.1. What Is A Docker Container?.....	41
3.2. Why Use A Container?.....	41
3.3. Using NGC Container Registry from the Docker Command Line.....	42
3.3.1. Accessing the NGC Container Registry.....	42
3.3.2. Uploading an NVIDIA Container Image onto Your System.....	43
3.3.3. Tagging and Pushing a Container Image.....	44
3.4. Using the Container Registry.....	44
3.4.1. Viewing Container Image Information.....	44
3.4.2. Pulling a Container Image.....	45

3.4.3. Pushing a Container Image.....	45
3.4.4. Removing a Container Image.....	45
3.5. Updating Container Metadata.....	45
3.5.1. Updating Container Metadata via the NGC Website.....	46
3.5.2. Updating Container Metadata Using the NGC CLI.....	47
3.6. Multi-architecture Support for NGC Container Images.....	49
<b>Chapter 4. NGC Models.....</b>	<b>51</b>
4.1. Creating New NGC Models Using the NGC CLI.....	51
4.2. Creating a New Model Using the NGC Website.....	51
4.3. Uploading a New NGC Model Version Using the NGC CLI.....	54
4.4. Uploading an NGC Model Version Using the NGC Website.....	56
4.5. Editing NGC Model Information Using the NGC CLI.....	58
4.6. Editing NGC Model Information Using the NGC Website.....	59
<b>Chapter 5. NGC Resources.....</b>	<b>61</b>
5.1. Before You Begin.....	61
5.2. Uploading a Resource.....	61
5.3. Updating a Resource.....	62
5.4. Resource Commands.....	62
5.5. Deleting a Resource.....	64
<b>Chapter 6. NGC Helm Charts.....</b>	<b>66</b>
6.1. Introduction to NGC and Helm Charts.....	66
6.2. Creating and Packaging a Helm Chart.....	66
6.3. Manage Helm Charts Using the NGC Web UI.....	67
6.3.1. Viewing the List of Helm Charts and Getting Fetch Commands.....	67
6.3.2. Adding Helm Charts Using the NGC Web UI.....	71
6.3.3. Updating the Helm Chart Page From the Website.....	72
6.3.4. Removing Helm Charts from the Web UI.....	73
6.4. Manage Helm Charts Using the NGC CLI.....	75
6.4.1. Searching for Available Helm Charts in an Org.....	75
6.4.2. Fetching Helm Charts.....	76
6.4.3. Adding Helm Charts to a Private Registry.....	76
6.4.4. Getting Information about a Helm Chart.....	77
6.4.5. Pushing a Helm Chart.....	77
6.4.6. Listing Helm Chart Versions.....	78
6.4.7. Removing Helm Charts from a Private Registry.....	78
6.5. Manage Helm Charts Using the NGC API.....	79
6.5.1. Updating Information on the Helm Chart Page.....	79
6.5.2. Deleting Helm Charts Using the NGC API.....	80

6.6. Manage Helm Charts Using the Helm CLI.....	80
6.6.1. Setting Up an NGC Helm Repository.....	80
6.6.2. Searching for Available Helm Charts.....	81
6.6.3. Fetching Helm Charts.....	81
6.6.4. Adding Helm Charts to a Private NGC Org/Team.....	81
6.6.5. Removing Helm Charts from a Private NGC Org/Team.....	81
Chapter 7. Getting Support for NGC container registry.....	82
Chapter 8. Private Registry Size Limits.....	83

---

# Chapter 1. NGC Private Registry for Enterprise

This document describes how to use the NVIDIA® NGC Private Registry. This guide assumes the user is familiar with Linux and Docker and has access to an NVIDIA GPU-based computing solution, such as an NVIDIA DGX system or NVIDIA-Certified system configured for internet access and prepared for running NVIDIA GPU-accelerated Docker containers.

As data scientists build custom content, storing, sharing, and versioning this valuable intellectual property is critical to meeting their company's business needs. To address these needs, NVIDIA has developed the NGC Private Registry to provide a secure space to store and share custom containers, models, Jupyter notebooks, and Helm charts within your enterprise. The NGC Private Registry is available to DGX and NVIDIA AI Enterprise customers.

## Increased Collaboration

We all are used to working collaboratively using tools such as Slack or Microsoft Teams, to share our content and ensure that our colleagues are all aligned. The primary goal of NGC Private Registry is to enable sharing of artificial intelligence (AI) content such as containers, models, Helm charts within your organization. This feature empowers key stakeholders in your organization to collaborate without reinventing the wheel, increasing productivity, saving valuable resources, and bringing your products to market faster.

## Enterprise Ready

When sharing content across a large organization, it is essential to ensure that you can manage the users. The comprehensive user and team management in an NGC Private Registry allow administrators to control access to content stored in the registry.

With the power of the cloud, the content stored in the NGC Private Registry is always available with redundant storage that can be accessed from anywhere, making it extremely easy to get to your content.

---

# Chapter 2. Getting Started


## 2.1. Obtaining a Private Registry

This chapter provides instructions for DGX customers on obtaining a private registry.


After purchasing a support entitlement with NVIDIA, the end-customer will receive an NVIDIA Entitlement Certificate via email. The email will include all the pertinent instructions to register for technical support.


The following is an example of the NVIDIA Entitlement Certificate email.

**NVIDIA Entitlement Certificate - Ref 8246**

 noreply@nvidia.com(noreply@nvidia.com)  
To

Retention Policy: Never Delete (Never)

 The actual sender of this message is different than the normal sender. Click here to learn more.

 entitlement-8246 .pdf  
14 KB

**External email: Use caution opening links or attachments**

Thank you for your NVIDIA® DGX™ order!

This email provides important information from your sales order that you need to retain and to register for NVIDIA Enterprise Support.

- Please follow registration instructions in the attachment.
- The NGC Container Registry administrator is responsible for adding and managing new users and team for your organization in the NGC Container Registry.

NVIDIA Enterprise Support Registration is the **only way** to get access to the NGC Container Registry, receive notifications from support on new product updates and critical security patches, submit and prioritize support cases, check status on open cases, and get access to the knowledge base. Following registration, you will receive welcome emails with access information for the NVIDIA Enterprise Support Portal and the NGC Container Registry.

Helpful resources to get started on DGX systems can be found [here](#).

If your product requires a MAC address for installation, please refer to the attachment.

Sales order information:


PO Number	NVIDIA Sales Order	NVIDIA Delivery Number
630-- -- --	835	8246 .. --

Questions?

NVIDIA Enterprise Support contact information can be found here <https://www.NVIDIA.com/en-us/support/enterprise/>

Thank you!

The Entitlement Certificate itself is provided as a PDF attachment. The following is an example of an NVIDIA Entitlement Certificate.



NVIDIA Corporation  
2788 San Tomas Expressway  
SANTA CLARA CA 95051  
USA

**NVIDIA® Entitlement Certificate**  
This certificate serves as evidence that NVIDIA has entitled you for the following product(s).

<b>End Customer (9007862)</b>	<b>NVIDIA Delivery</b>	8246
END-Customer Name	<b>Entitlement Date</b>	07 FEB 2020
	<b>PO Number</b>	6300
	<b>NVIDIA Sales Order</b>	839


No	Entitlement Description	Quantity	Sales Type	Term	Start Date	End Date
1	DGX Station DL WS 4V100/256GB 32G Support  PAK ID: qjh2dxtbjx  Serial Number: 1234567	1 EA	Initial	3 Years	07 FEB 2020	08 MAR 2023
2	DGX Station DL WS 4V100/256GB 32G Support  PAK ID: qjh2dxtbjx  Serial Number: 1234568	1 EA	Initial	3 Years	07 FEB 2020	08 MAR 2023

Please follow the instructions provided in the following section to register your entitlements.

Thank You!

The PDF also includes instructions for using the certificate.





NVIDIA Corporation  
2788 San Tomas Expressway  
SANTA CLARA, CA 95051  
USA

**NOTICE**

**HOW TO USE THIS CERTIFICATE**

**Registration Instructions**

**Sales Type: Initial**

1. **Already have an account?** Please [Login](#)
2. **New User?** Please register your Primary Technical Contact for support and the NGC Container Registry using the [NVIDIA Enterprise Support Registration Form](#). The primary technical contact is the person who will be responsible for managing your DGX. The NGC Container Registry Administrator is responsible for adding and managing new users and teams for your organization in the NGC Container Registry.

NVIDIA Enterprise Support Registration is the only way to get access to the NGC Container Registry, receive notifications from support on new product updates and critical security patches, submit and prioritize support cases, check status on open cases, and get access to the knowledge base. Following registration, you will receive a welcome email with access information for the NVIDIA Enterprise Support Portal and the NGC Container Registry.

Helpful resources to get started on DGX Systems can be found [here](#)

**Questions?**  
NVIDIA Enterprise Support contact information can be found [here](#)

Rights and restrictions on the use, transfer and copying of the Support Services are set forth in NVIDIA's End User Terms and Conditions.

- ▶ If you already have an account, you can immediately log into the NVIDIA Enterprise Support portal.
- ▶ If you are a new user without an NGC Support account, click the NVIDIA Enterprise Support Registration Form link.

This link will have embedded information regarding your account. It is essential not to share this entitlement link outside of your organization.

Registration will provide an NGC private registry and NVIDIA Enterprise Support accounts. You'll receive a welcome email, at which time you can activate your NGC private registry account.

## 2.2. Activating a New NGC Account

Before using NGC, you must have an NGC account created by your organization owner or other administrators in your organization. You need an email address to set up an account. Choose one of the following processes depending on your situation for activating your NGC account.

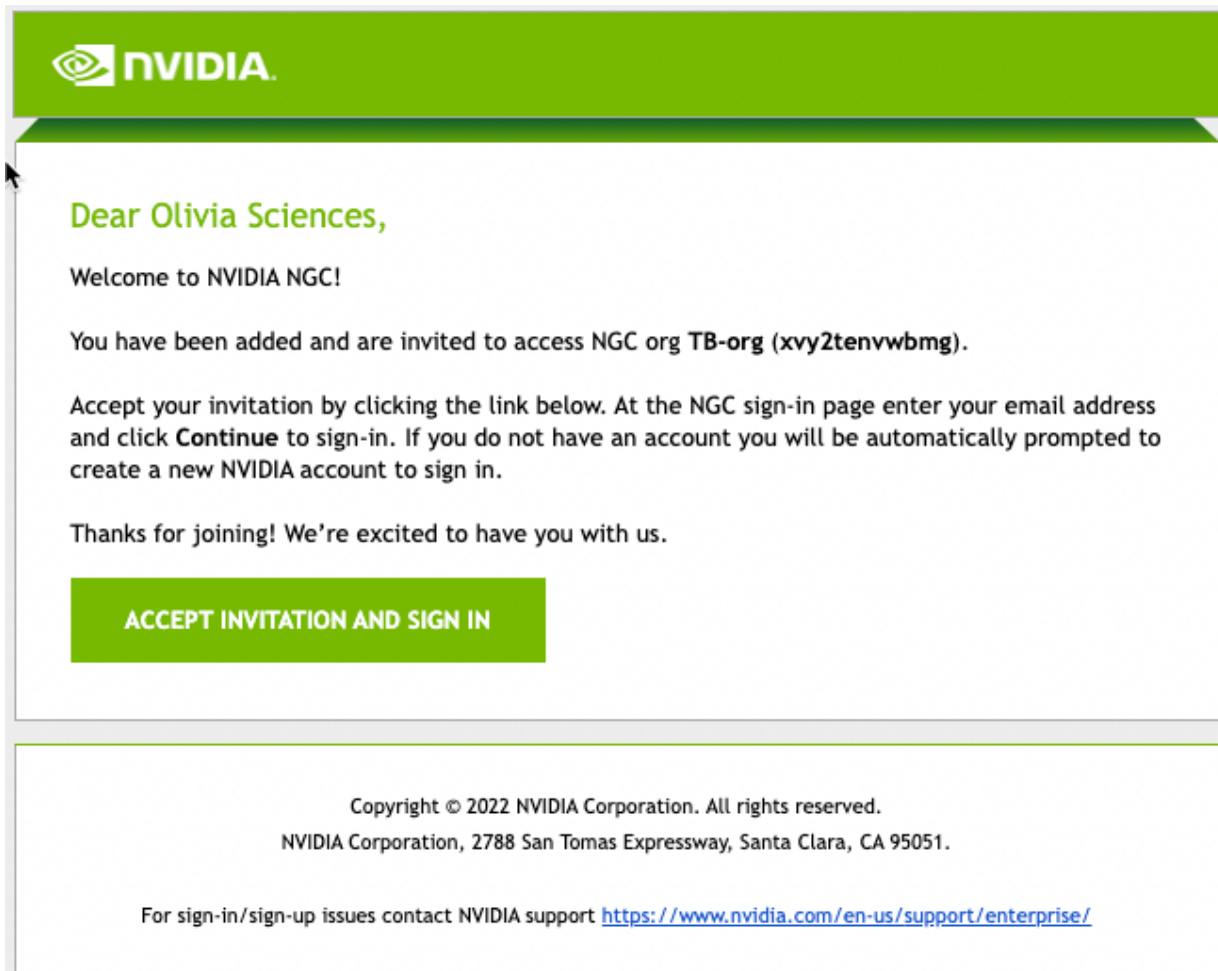
- ▶ [Joining an NGC Org or Team with an Existing NVIDIA Account](#)

- ▶ [Joining an Org or Team with a New NVIDIA Account](#)
- ▶ [Joining an Org as Org Owner](#)
- ▶ [Joining an Org or Team with an External SSO Company Account](#)
- ▶ [Switching Orgs or Team After Logging into NGC](#)

## 2.2.1. Joining an NGC Org or Team with an Existing NVIDIA Account

This section describes joining an org or team when your email address is already associated to an NVIDIA account.

After NVIDIA or your organization administrator adds you to a new org or team within an organization, you will receive a welcome email that invites you to continue the activation and sign in process.



1. Click the Accept Invitation and Sign In link to open the NGC sign-in page.

Enter your email address and sign in using your NVIDIA account credentials.

The image shows a login screen for NVIDIA NGC. It features the NVIDIA logo and the text "NVIDIA NGC". Below the logo, there is a welcome message: "Welcome to NVIDIA NGC - your portal to NVIDIA AI, Omniverse and high-performance computing (HPC).". The screen prompts the user to "Enter your email to sign in." and includes a text input field labeled "Email Address" containing the placeholder text "user@domain.com". A "Continue" button is positioned below the input field, and a link for "Use alternate method" is located at the bottom of the form area.

 **NVIDIA NGC**

Welcome to NVIDIA NGC - your portal to NVIDIA AI, Omniverse and high-performance computing (HPC).

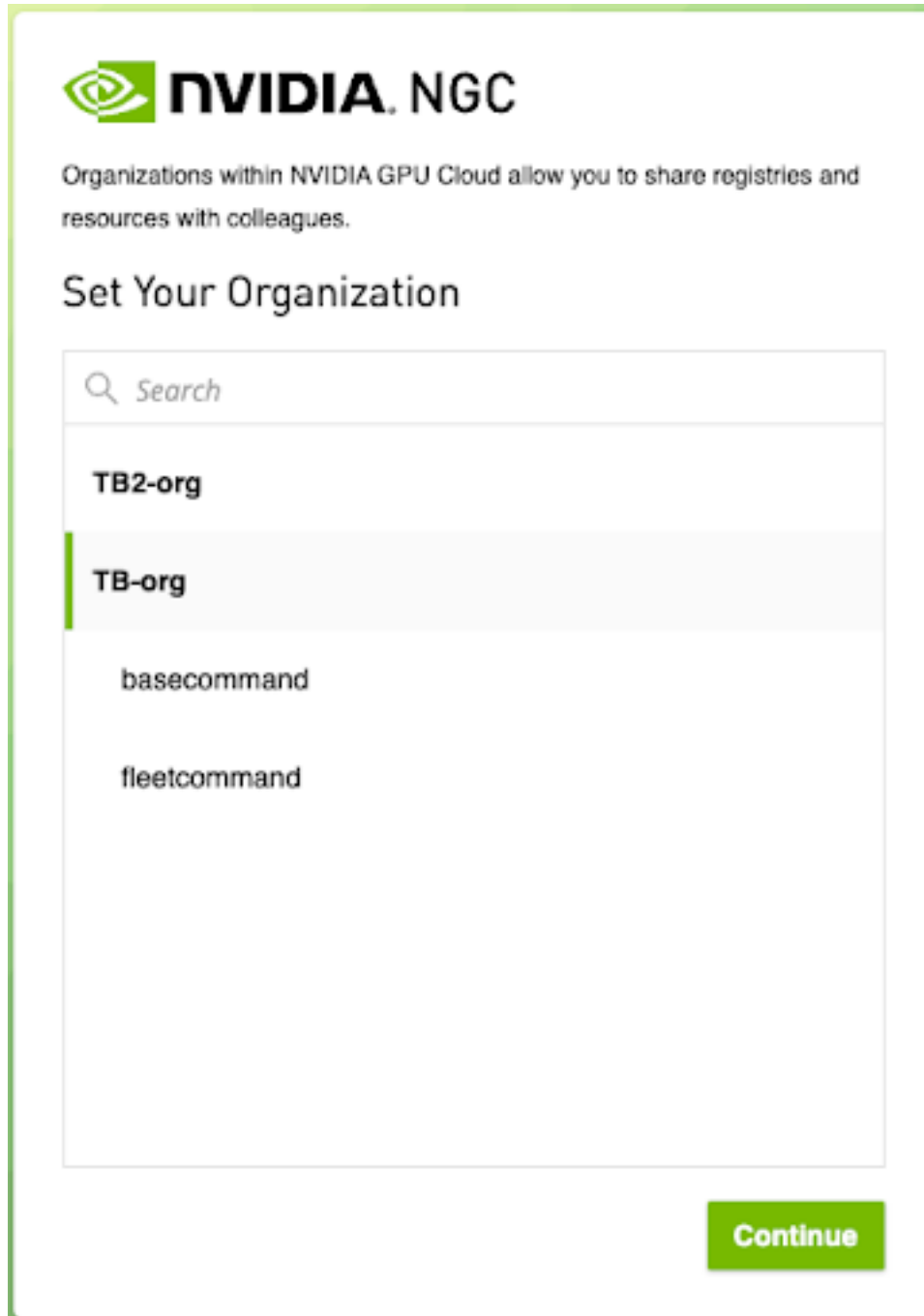
Enter your email to sign in.

**Email Address**

[Continue](#)

[Use alternate method](#)

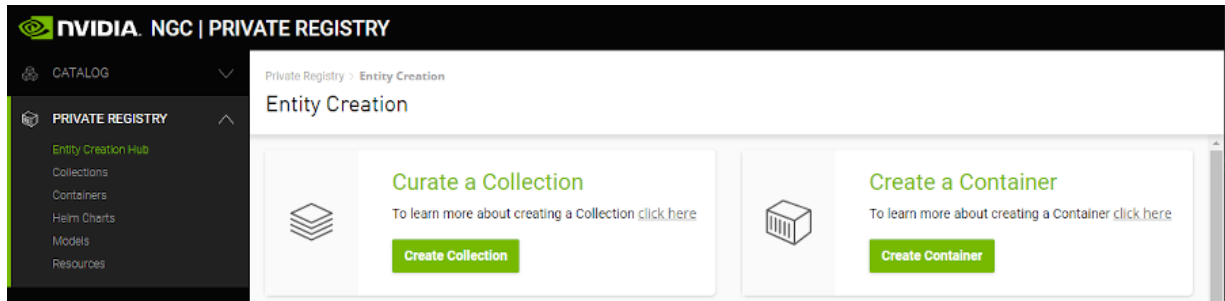
The Set Your Organization screen appears.



2. Select the new organization and team you have been invited to. Click Continue.  
You can always change to a different org or team that you are a member of after logging in. Refer to [Switching Orgs or Team After Logging into NGC](#) for more information.

The NGC web UI opens to the NGC Catalog landing page.

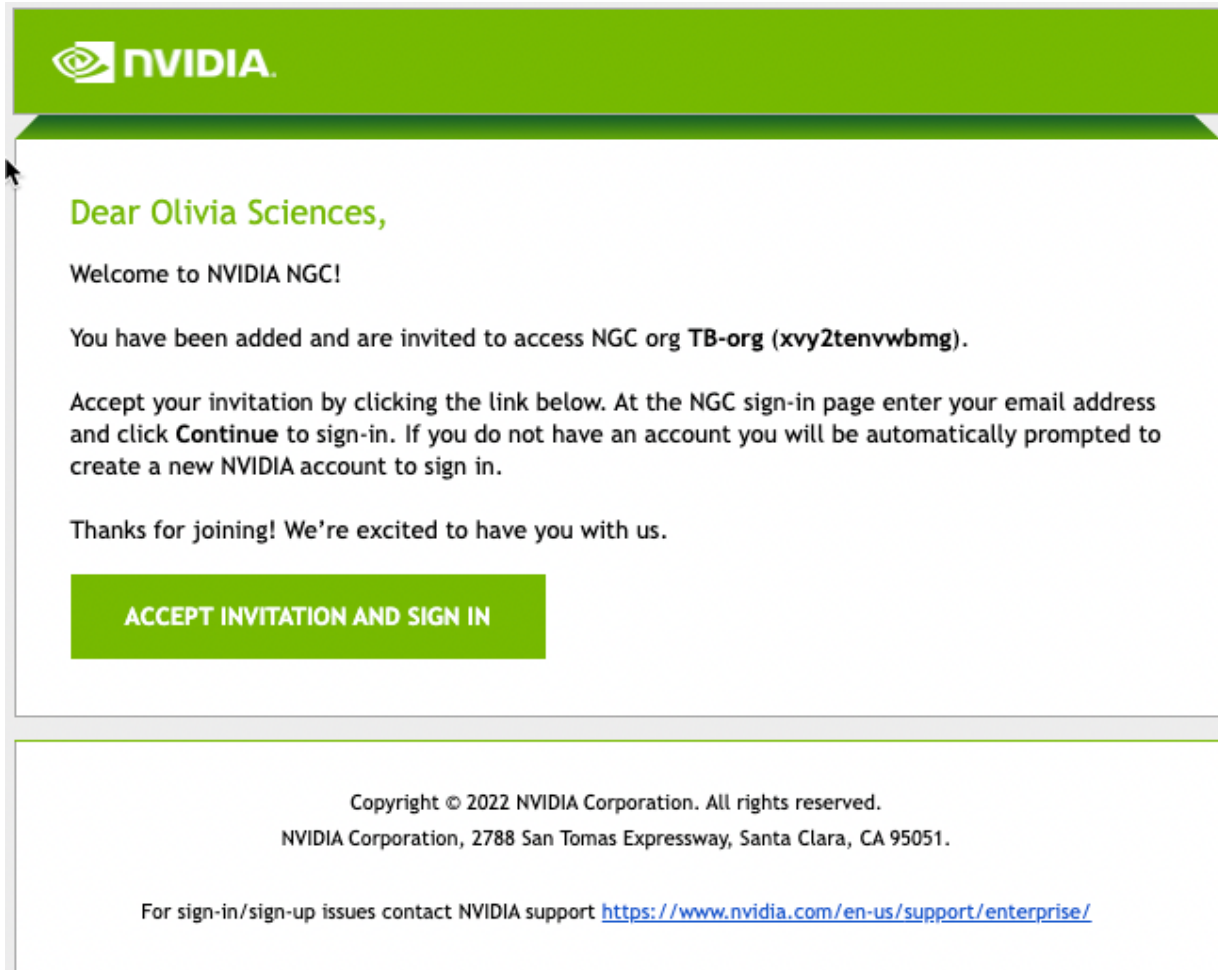
To view artifacts in your private registry, select Private Registry in the left navigation menu. You may select containers, Helm charts, models and resources as needed.



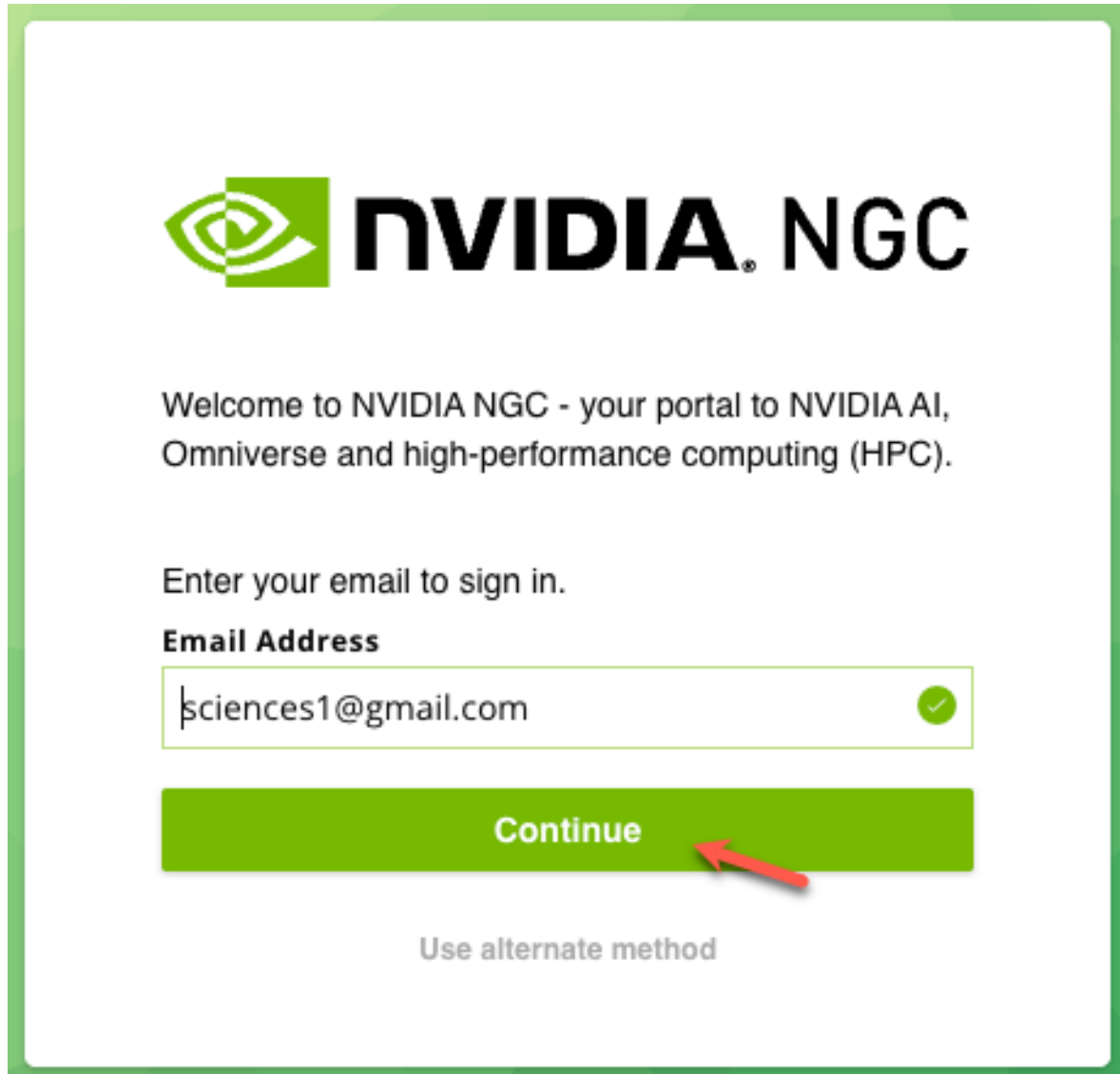
## 2.2.2. Joining an Org or Team with a New NVIDIA Account


This section describes activating a new account where the domain of your email address is not mapped to an organization's single sign-on.

After your organization administrator invites you to an org or team, you will receive a welcome email that invites you to continue the activation and login process.



1. Click the Accept Invitation and Sign In link to open the NGC sign-in dialog in your browser, or go to [NGC sign-in](#).





 **NVIDIA. NGC**

Welcome to NVIDIA NGC - your portal to NVIDIA AI, Omniverse and high-performance computing (HPC).

Enter your email to sign in.

**Email Address**




**Continue** 

[Use alternate method](#)


2. Type in your email address and click Continue. You will be automatically prompted to create a new NVIDIA account.

## Create Your Account

**Email**


**Password**  
 

**Good**

**Confirm password**  
 

Stay logged in [Log In With Security Device](#) ⓘ

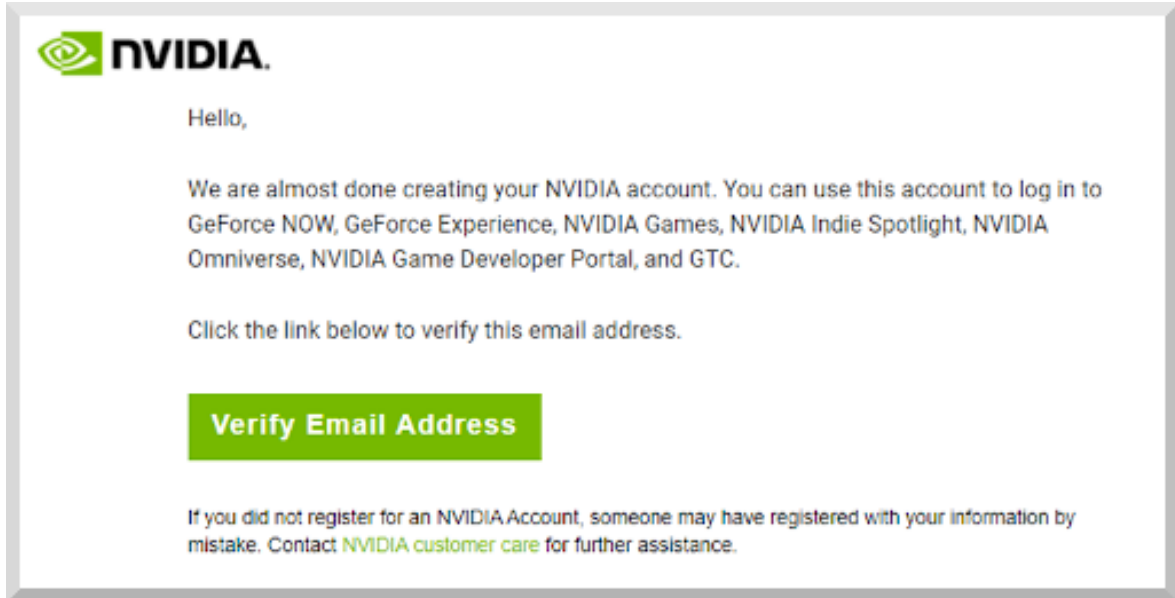
By proceeding, I agree to the [NVIDIA Account Terms Of Use](#) and [Privacy Policy](#).

**Create Account** 

[More Signup Options](#)

3. Fill in your information, create a password, agree to the Terms and Conditions, and click Create Account.  
An email is sent to you to verify your email address.
4. Open the email and click Verify Email Address.

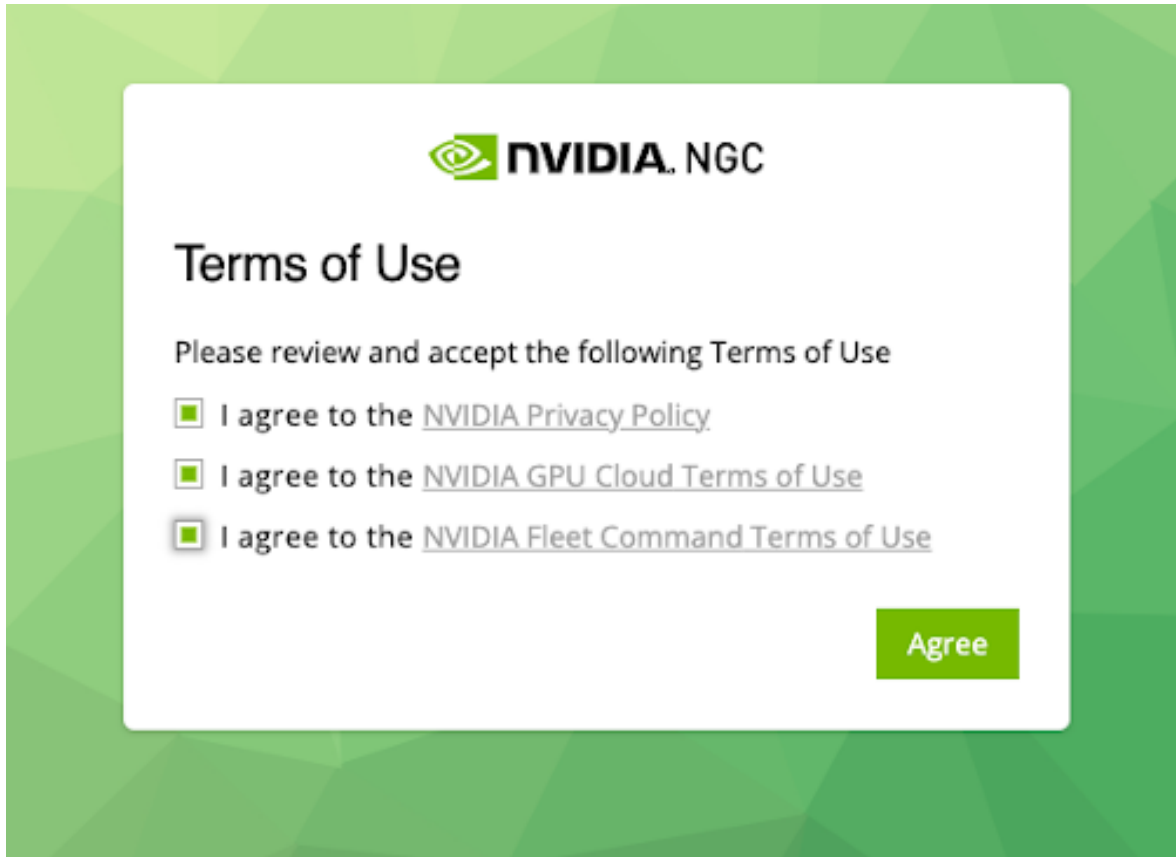





5. In the Almost done! dialog, select your communication preferences and then click Submit.

The screenshot shows a dialog box titled "Almost done!". Below the title is the instruction "Please confirm the information below to complete registration". There are two sections of settings, each with a checked checkbox: "Recommendation Settings" with the text "Yes, recommend content that I might enjoy based on how I engage with NVIDIA's websites, software, and events." and "Be the first to learn about new SDKs, developer tools and training" with the text "Send me the latest enterprise news, announcements, and more from NVIDIA. I can unsubscribe at any time." At the bottom, there is a privacy statement: "We promise to protect your privacy. We never sell your data. You can change your settings anytime at [privacy.nvidia.com](https://privacy.nvidia.com)." A green "SUBMIT" button is located at the bottom right.

6. In the NVIDIA Account Terms of Use dialog, select the desired options and click Agree.

A dialog box with a green geometric background. At the top center is the NVIDIA logo (an eye icon) followed by the text "NVIDIA. NGC". Below this is the heading "Terms of Use". Underneath the heading is the text "Please review and accept the following Terms of Use". There are three checkboxes, each followed by the text "I agree to the" and a link: "[NVIDIA Privacy Policy](#)", "[NVIDIA GPU Cloud Terms of Use](#)", and "[NVIDIA Fleet Command Terms of Use](#)". In the bottom right corner of the dialog box is a green button with the text "Agree" in white.

 **NVIDIA. NGC**

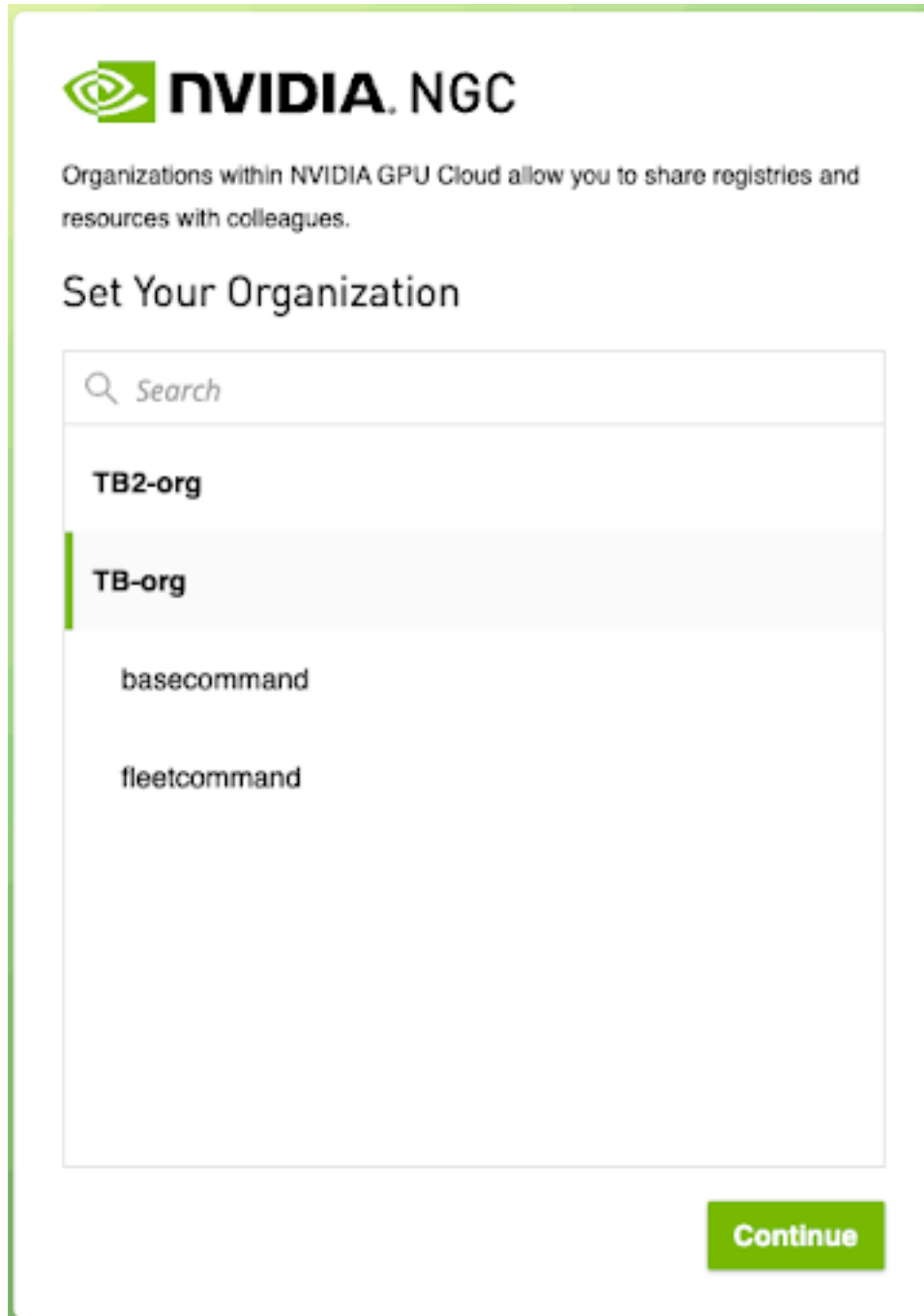
## Terms of Use

Please review and accept the following Terms of Use

- I agree to the [NVIDIA Privacy Policy](#)
- I agree to the [NVIDIA GPU Cloud Terms of Use](#)
- I agree to the [NVIDIA Fleet Command Terms of Use](#)

**Agree**

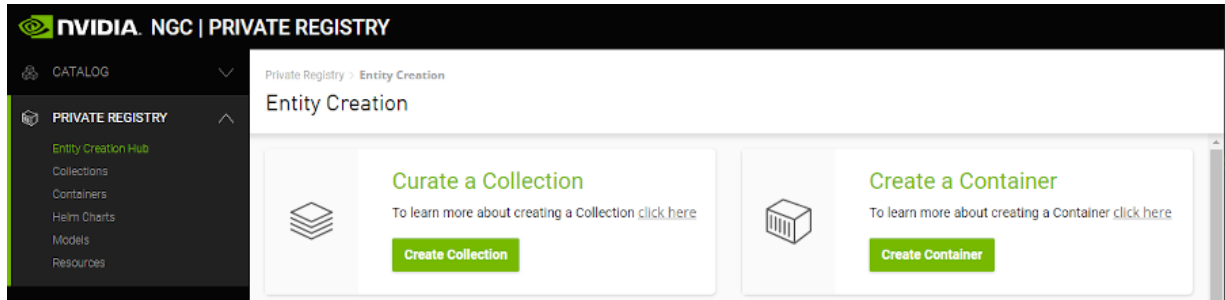
7. Select the organization and team you want to log in under and then click Continue.



You can always change to a different org or team that you are a member of after logging in. Refer to [Switching Orgs or Team After Logging into NGC](#) for more information.

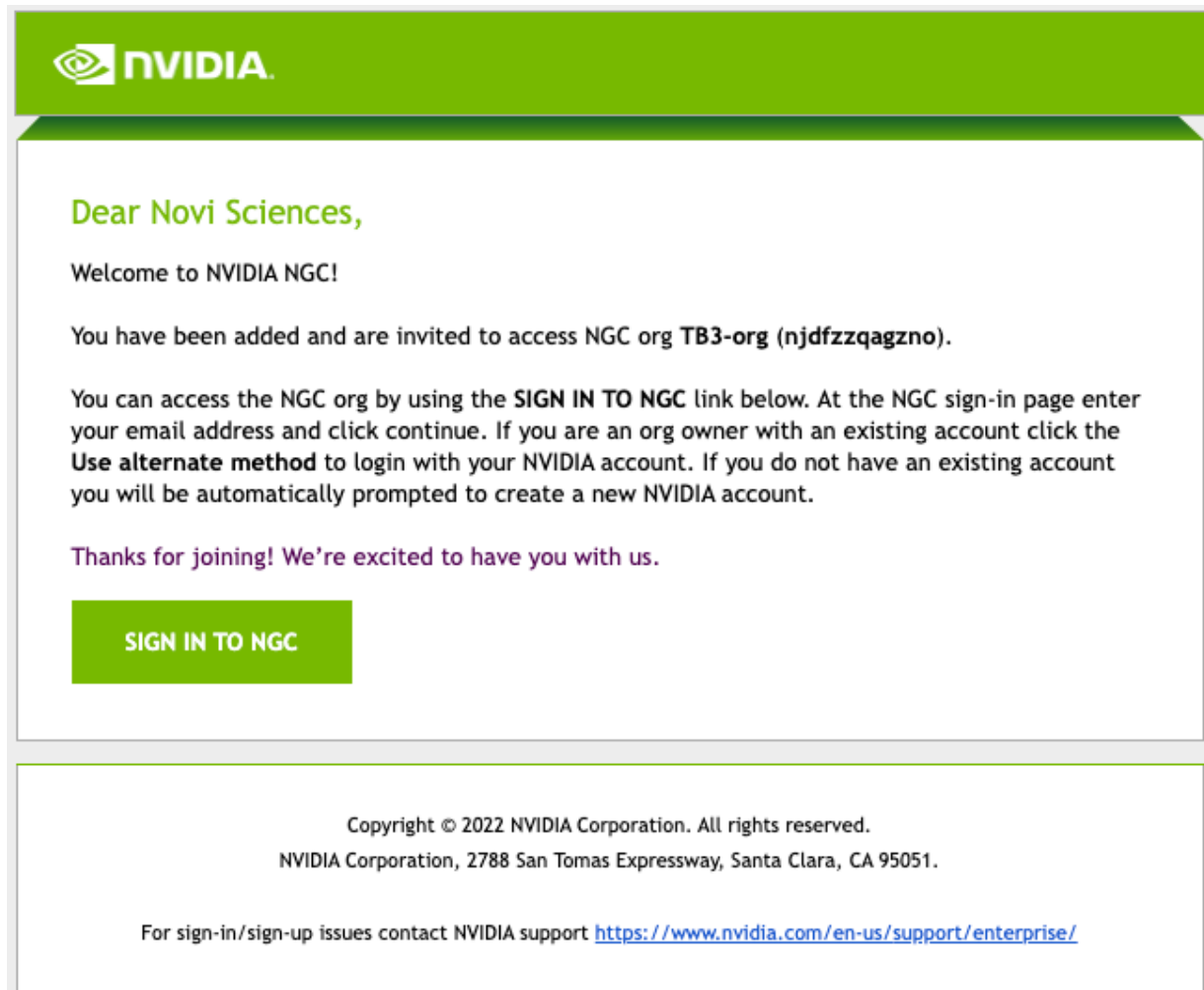
The NGC web UI opens to the NGC Catalog landing page.

To view artifacts in your private registry, select Private Registry in the left navigation menu. You may select containers, Helm charts, models and resources as needed.



### 2.2.3. Joining an Org as Org Owner

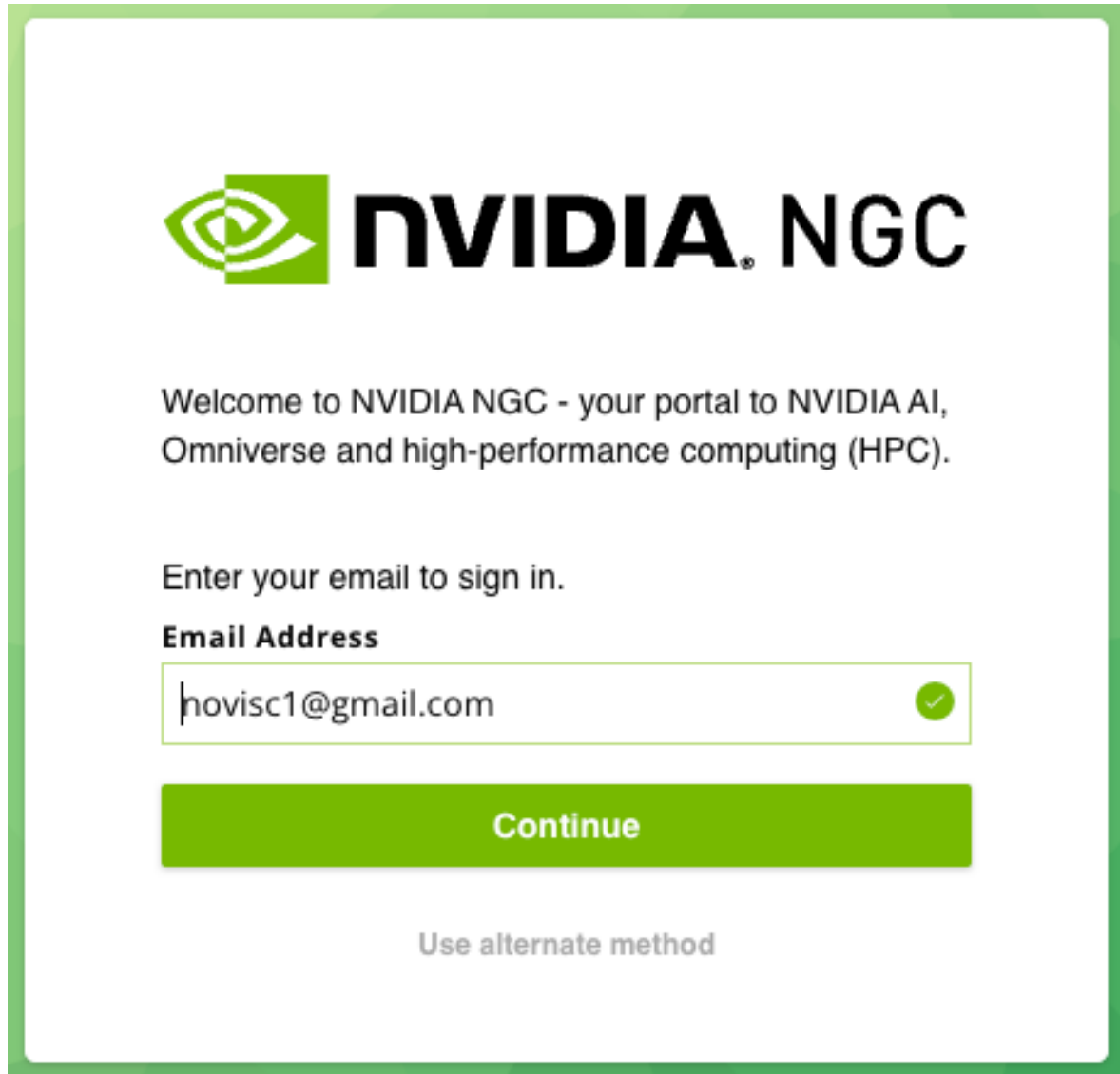
This section describes activating a new NGC org where you are joining as the org owner. After NVIDIA sets up your NGC org, you will receive a welcome email that invites you to continue the activation and login process.




In the example, "Novi Sciences" is the owner of the newly-created organization. The steps below assume Novi is new to NGC and explain how to create a new NVIDIA account and sign in as org owner. If you already have an NVIDIA account managing other NGC orgs as org owner, click on Use alternate method to sign in with your existing NVIDIA account and access your new org.

Take note of the following important information in the email.

- ▶ "TB3-org" is the display name for your org. The display name identifies your org in the NGC web UI.
  - ▶ "njdfzzqagzno" is the unique identifier for your org. This identifier represents your org namespace. You can use this identifier in some CLI commands.
1. Click Sign in to NGC, or using a browser, navigate to the [NGC sign in page](#). Type in your email address and click Continue.




 **NVIDIA NGC**

Welcome to NVIDIA NGC - your portal to NVIDIA AI, Omniverse and high-performance computing (HPC).

Enter your email to sign in.

**Email Address**

hovisc1@gmail.com 

**Continue**


[Use alternate method](#)

2. You will be presented with a create account screen.


Verify your email and create a password. Review the NVIDIA Account Terms of Use and Privacy Policy, and click Create Account.

## Create Your Account

**Email**

**Password**  
 

**Good**

**Confirm password**  
 

Stay logged in [Log In With Security Device](#) ⓘ

By proceeding, I agree to the [NVIDIA Account Terms Of Use](#) and [Privacy Policy](#).

[Create Account](#)

[More Signup Options](#)

- A verification email is sent.
3. Open the email and click Verify Email Address.



Hello,

You requested to use this email address to access your NVIDIA account.

Click the link below to verify this email address.

[Verify Email Address](#)

[Unsubscribe](#) | [Manage Preferences](#) | [Contact Us](#) | [Privacy Center](#)

© 2022 NVIDIA Corporation. All rights reserved.

NVIDIA Corporation, 2788 San Tomas Expressway, Santa Clara, CA 95051.

4. In the Almost done! dialog, select your communication preferences and then click Submit.

Almost done!

Please confirm the information below to complete registration

**Recommendation Settings**

Yes, recommend content that I might enjoy based on how I engage with NVIDIA's websites, software, and events.

**Be the first to learn about new SDKs, developer tools and training**

Send me the latest enterprise news, announcements, and more from NVIDIA. I can unsubscribe at any time.

*We promise to protect your privacy. We never sell your data. You can change your settings anytime at [privacy.nvidia.com](https://privacy.nvidia.com).*

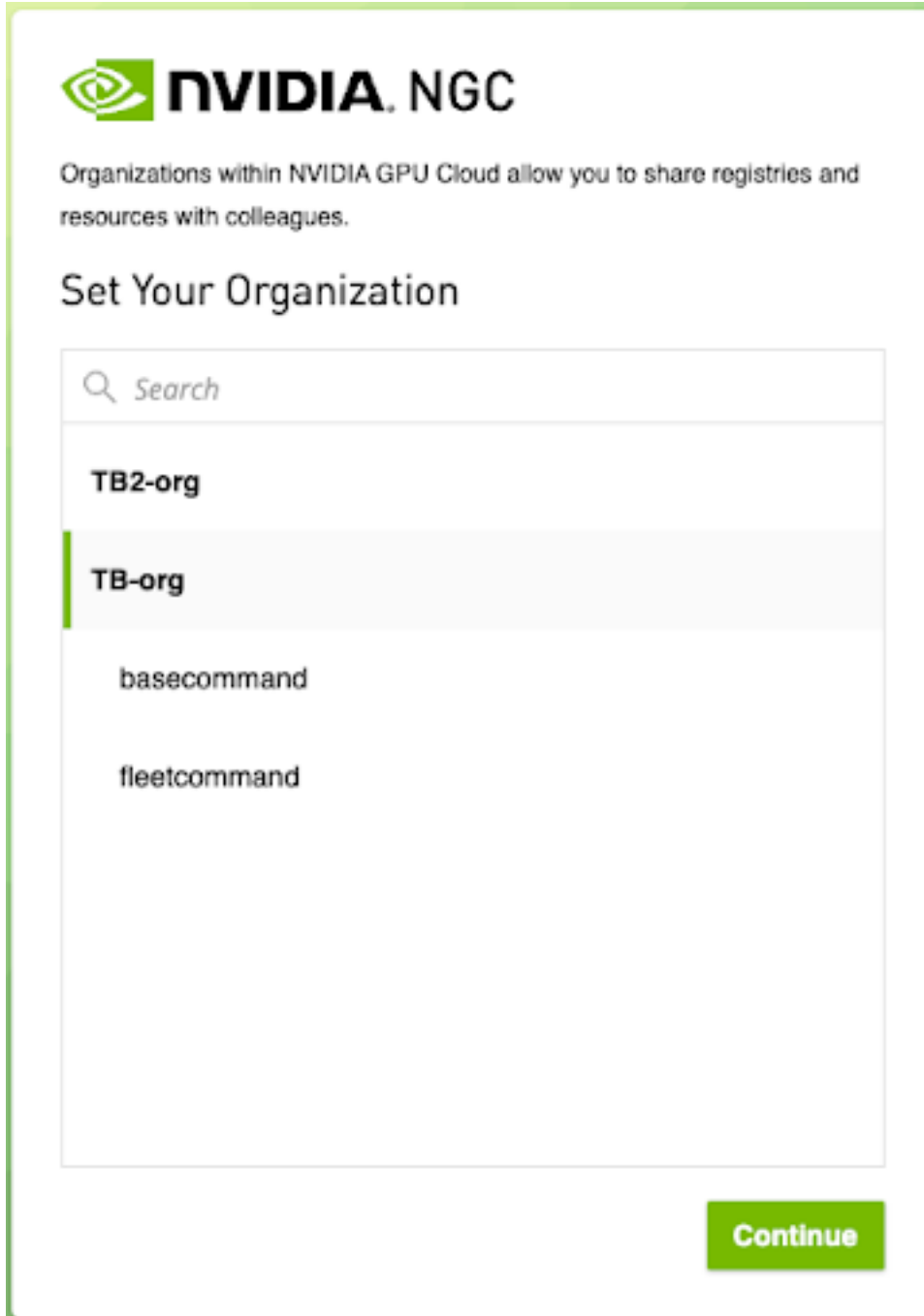
[SUBMIT](#)




5. In the NVIDIA Account Terms of Use dialog, select the desired options and click Agree.



6. Select the organization and team you want to log in under and then click Continue.



 **NVIDIA. NGC**

Organizations within NVIDIA GPU Cloud allow you to share registries and resources with colleagues.

## Set Your Organization

- TB2-org
- TB-org**
- basecommand
- fleetcommand

**Continue**

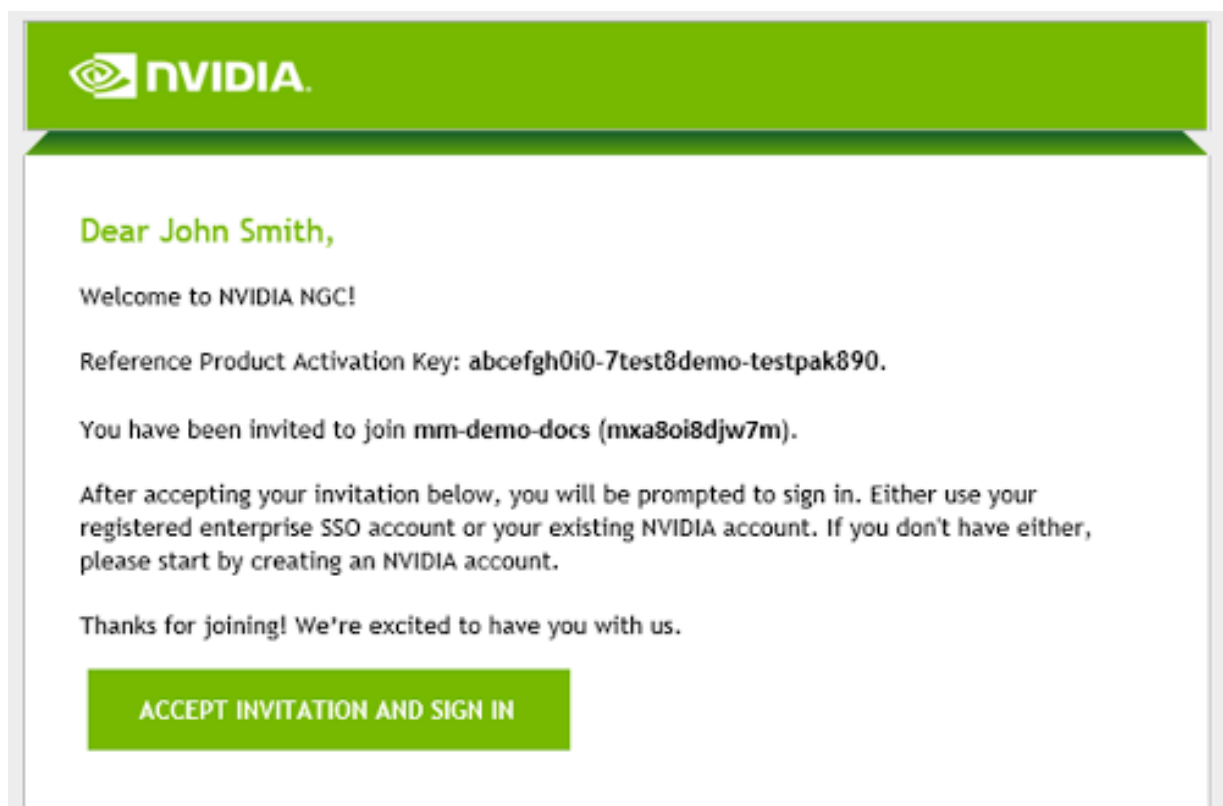
You can always change to a different org or team that you are a member of after logging in. Refer to [Switching Orgs or Team After Logging into NGC](#) for more information.

The NGC web UI opens to the NGC Catalog landing page.

## 2.2.4. Joining an Org or Team with an External SSO Company Account

This section describes joining an org or team that has been federated by your company to an external SSO/IdP authentication service and your email address domain requires NGC authentication against your company's single sign-on.

After your organization administrator adds you to a new org or team within the organization, you'll receive a welcome email that invites you to continue the activation and sign-in process.



1. Click the Accept Invitation and Sign In link to open the NGC sign-in page. Enter your email address and click Continue.

The image shows a login interface for NVIDIA NGC. It features the NVIDIA logo and the text "NVIDIA NGC" at the top. Below this, a welcome message reads: "Welcome to NVIDIA NGC - your portal to NVIDIA AI, Omniverse and high-performance computing (HPC)." The next instruction is "Enter your email to sign in." followed by the label "Email Address". A text input field contains the placeholder "user@domain.com". Below the input field is a grey "Continue" button. At the bottom of the form is a link that says "Use alternate method".

 **NVIDIA NGC**

Welcome to NVIDIA NGC - your portal to NVIDIA AI, Omniverse and high-performance computing (HPC).

Enter your email to sign in.

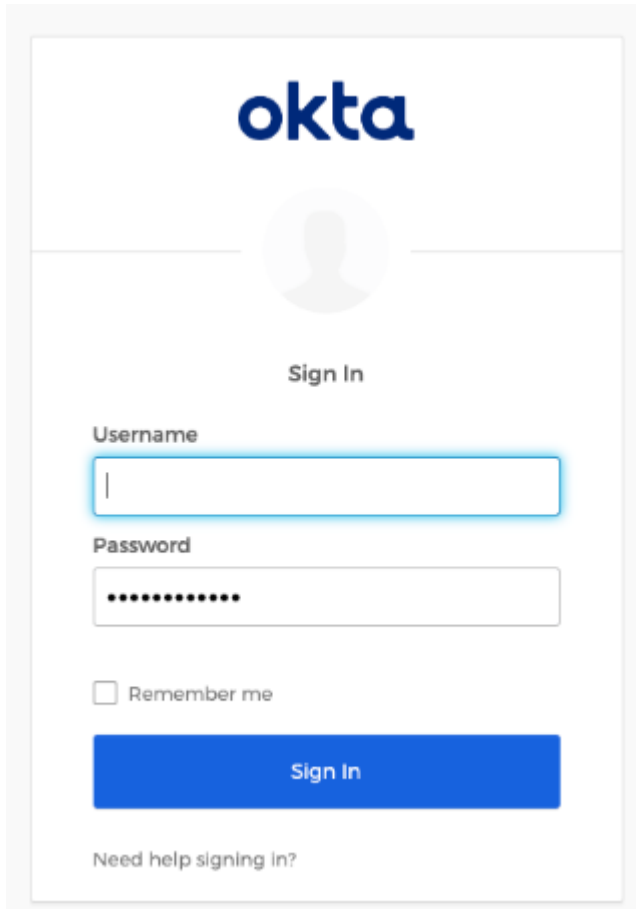
**Email Address**

[Continue](#)

[Use alternate method](#)

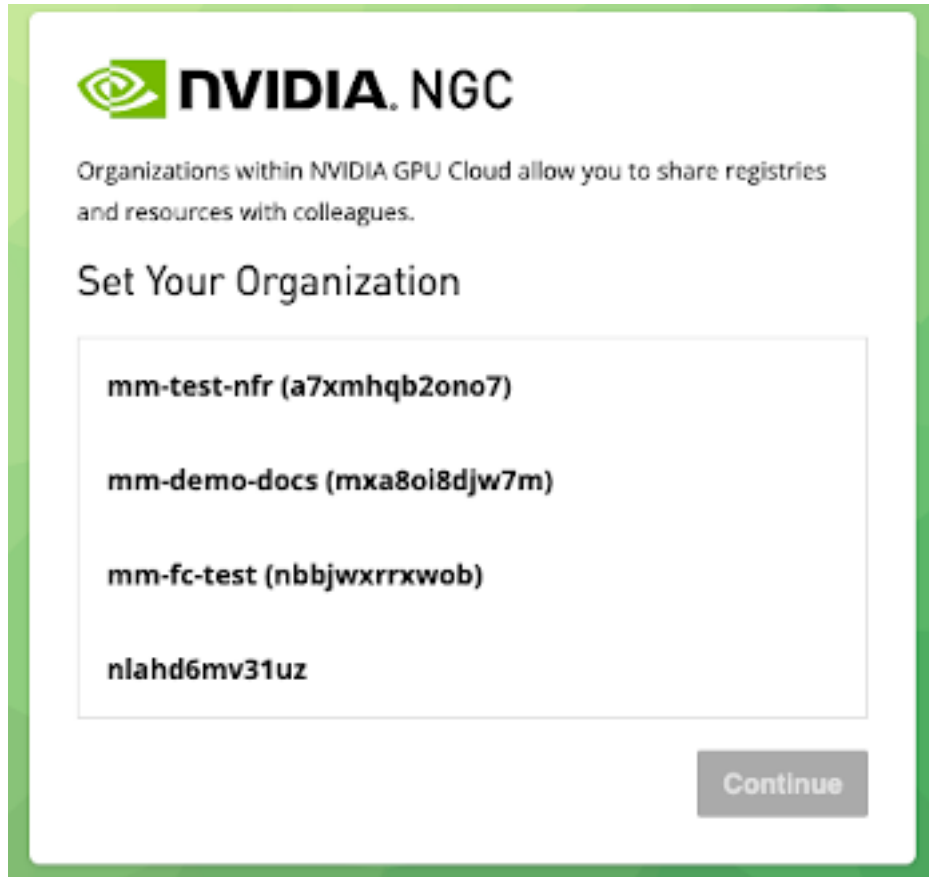
With your email address domain associated to an external SSO identity provider, you will be automatically redirected and prompted to authenticate against the external authentication method.

For example:

The image shows the Okta sign-in interface. At the top is the 'okta' logo in blue. Below it is a grey silhouette of a person's head and shoulders. Underneath the silhouette is the text 'Sign In'. The form contains two input fields: 'Username' with a blue border and a vertical cursor, and 'Password' with a grey border and ten black dots. Below the password field is a checkbox labeled 'Remember me'. At the bottom of the form is a blue button with the text 'Sign In'. Below the button is the text 'Need help signing in?'.

After successfully authenticating you will be redirected to NGC.

If you are a member of more than one NGC org, the Set Your Organization screen appears.

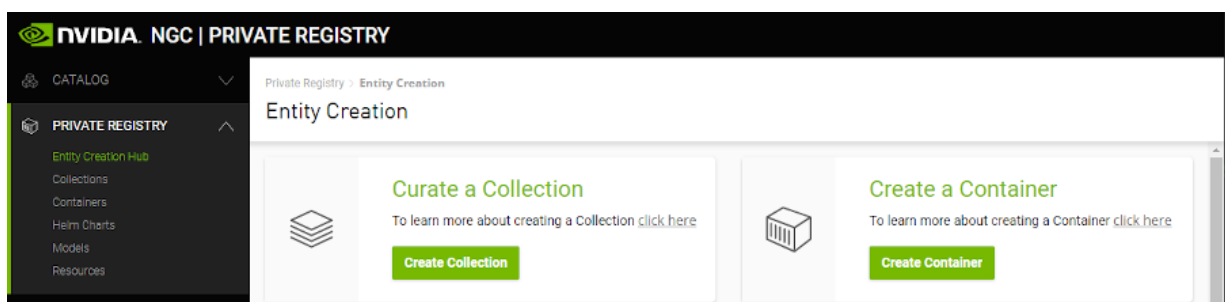


2. Select the new organization and team you have been invited to and Click Continue.

You can always change to a different org or team that you are a member of after logging in. Refer to [Switching Orgs or Team After Logging into NGC](#) for more information.

The NGC web UI opens to the NGC Catalog landing page.

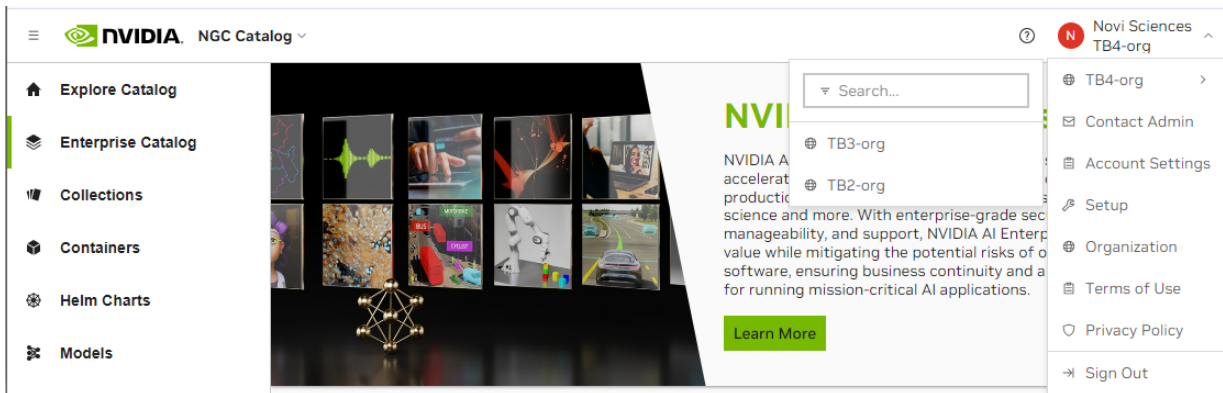
To view artifacts in your private registry, select Private Registry in the left navigation menu. You may select containers, Helm charts, models and resources as needed.



## 2.2.5. Switching Orgs or Team After Logging into NGC

This section describes switching to a different org or team after logging in.

In the top menu bar, click your user account icon. Then, select your org menu to expand the view to other available orgs. If you manage many orgs, you can use the search field to find the specific org you want to select. Select the desired org by clicking it once.



Depending on the org or team you select, your current page may also refresh.

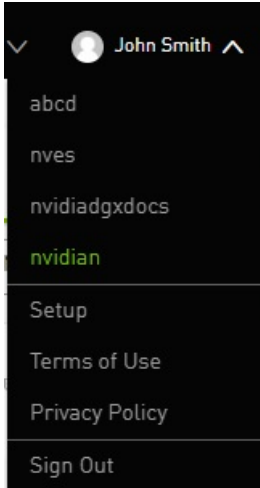
## 2.3. Generating Your NGC API Key

This section describes how to obtain an API key to access locked container images from the NGC Registry.

1. Sign in to the NGC website.

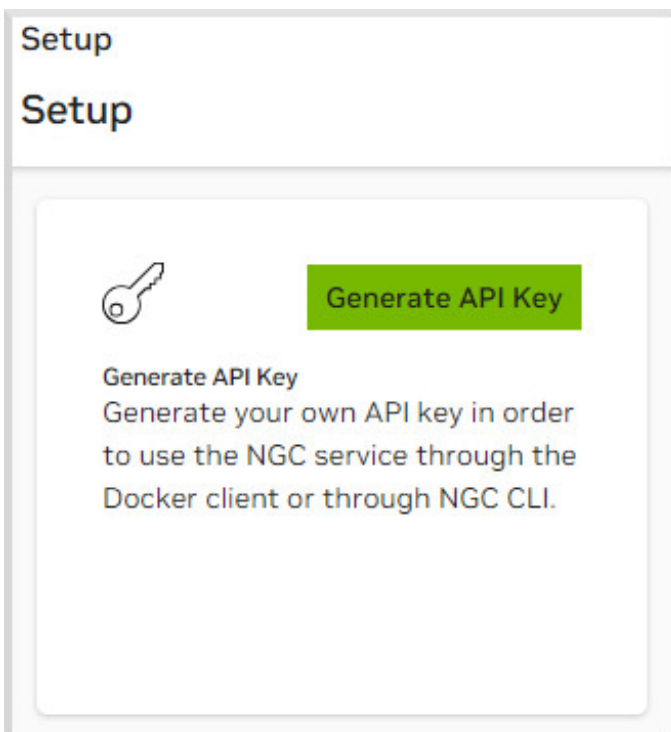
From a browser, go to <https://ngc.nvidia.com/signin/email> and then enter your email and password.

2. In the top right corner, click your user account icon and select Setup.



3. Click Get API key to open the Setup > API Key page.

The API Key is the mechanism used to authenticate your access to the NGC container registry.



4. Click Generate API Key to generate your API key.  
A warning message lets you know that your old API key will become invalid if you create a new key.
5. Click Confirm to generate the key.  
Your API key appears.



You only need to generate an API Key once. NGC does not save your key, so store it in a secure place.



Tip: You can copy your API Key to the clipboard by clicking the copy icon to the right of the API key.

If you lose your API Key, you can generate a new one from the NGC website. When you generate a new API Key, the old one is invalidated.

## 2.4. Managing Users and Teams in NGC

This chapter applies to organization and team administrators, and explains the tasks that an organization or team administrator can perform from the NGC website.

When the Organization was created, an Organization owner was created from the primary technical contact information provided during the sales process. This organization owner will receive an email from NGC. As the NGC Org owner for your organization, you can invite other users to join your organization's NGC account. Users can then be assigned as members of teams within your organization. Teams are useful for keeping custom work private within the organization. You can also create other administrators in the organization to share that responsibility.

The general workflow for building teams of users is as follows:

1. The organization admin invites users to the organization's NGC account.
2. The organization admin creates teams within the organization.
3. The organization admin adds users to appropriate teams, and typically assigns at least one user to be the team admin.
4. The organization or team admin can then add other users to the team.

### 2.4.1. NGC Registry User Roles

Prior to adding users and teams, familiarize yourself with the following definitions of each role.

The NGC container registry supports the following user roles.

#### Organizational and Team Level Roles

The following roles can be assigned to a user.

- ▶ **Org Owner** : This user is created at the time of Org creation. Only one user can be the Org Owner at a given moment. This user can download/upload, push/pull or delete, add/remove users and create teams within an organization.
- ▶ **Registry Admin** : This user can download/upload, push/pull or delete artifacts within an organization or team.
- ▶ **Registry User** : This user can download, upload, push/pull artifacts within an organization or team.

- ▶ Registry Read : This user can download and pull artifacts within an organization or team.
- ▶ User Admin : This user can view and invite other users and user admins within an organization. At the team level, the User Admin can view and invite other users and user admins to that team. A User Admin can only grant roles that they possess.
- ▶ User Read : This user can view details of an organization or team.



Note: A user must be a "Registry Read", "Registry User", and/or "User Admin" role to be a member of the organization or any team.

Capability	Registry Admin	Registry User	User Admin	Registry Read	User Read
Add teams	X	X	#	X	X
Add new users to orgs or teams	X	X	#	X	X
View users	#	X	#	X	X
Delete images	#	X	X	X	X
View/Edit all image information via UI and CLI	#	#	X	X	X
View all artifacts namely containers, model, resources	#	#	#	#	X
Download all artifacts namely containers, model, resources	#	#	X	#	X
Create and push/upload all artifacts namely containers, model, resources	#	#	X	X	X

## 2.4.2. Creating Teams

Creating teams is useful for allowing users to share images within a team while keeping them invisible to other teams in the same organization. Only organization administrators can create teams.

To create a team,

1. Log on to the [NGC website](#).
2. Select Organization > Teams from the left navigation menu, then click Create Team menu at the top of the screen.

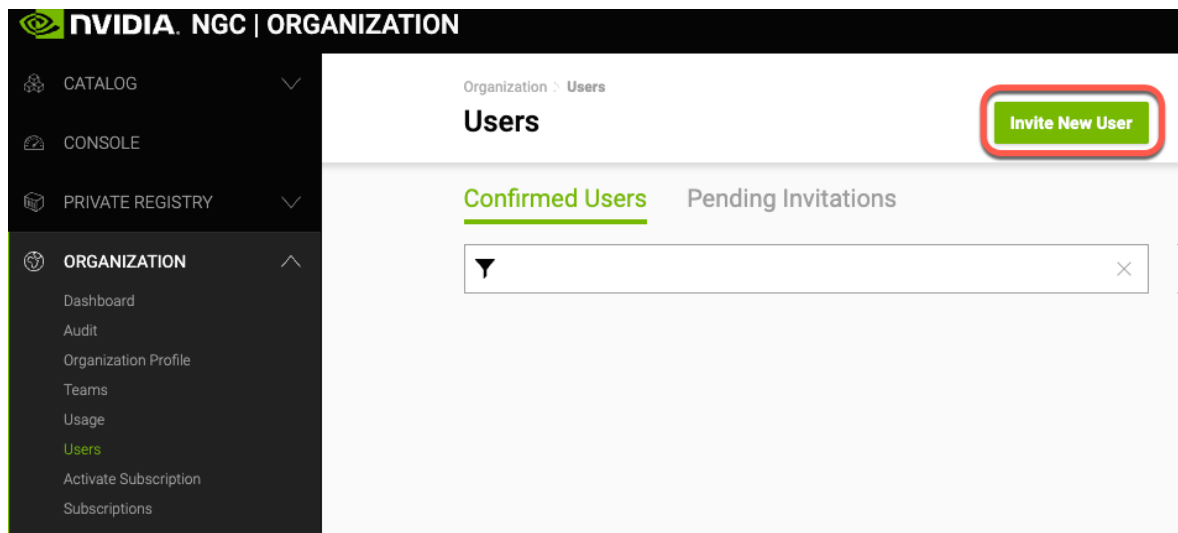


3. Enter a team name and description, then click Add Team. Team names must be all lowercase.

### 2.4.3. Creating Users

As the organization owner or user administrator, you must create user accounts to allow others to use the NGC container registry within the organization.

1. Log on to the [NGC website](#).
2. Click Organization > Users from the left navigation menu, then click Invite New User at the top right of the screen.



3. Fill out the Add User form for the new user as follows:

Organization > Users > Add User

## Add User

Create User

### User Information

Please complete this section with the user's information.

First Name \*  Last Name  Email \*

### Roles

Please select a context for role assignment.

Organization (ulhtfbtzcrrsh)  Team

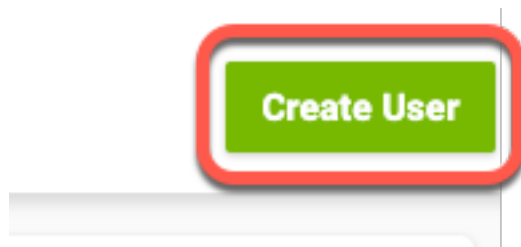
**NVIDIA AI Enterprise**  Viewer

**Private Registry**  Admin  User  Read

**Organization**  User Admin

Add Role

- Enter the display name and email where indicated.
- Select the organization or team to be assigned.
- Select the roles to assign to the user.
- Click Add Role and then click Create User when done.

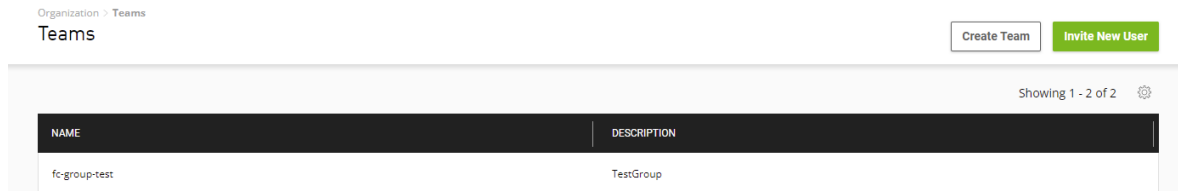


An invitation email is automatically sent to the user.

## 2.4.4. Adding a new User to a Team

Org owners or org level user administrators can add users to any team in the organization. Team user administrators can add users to their teams.

- Log on to the [NGC website](#).
- Click Organization > Teams from the left navigation menu, then select the team that you want to add a user.
- On the Teams page, click Invite New User.



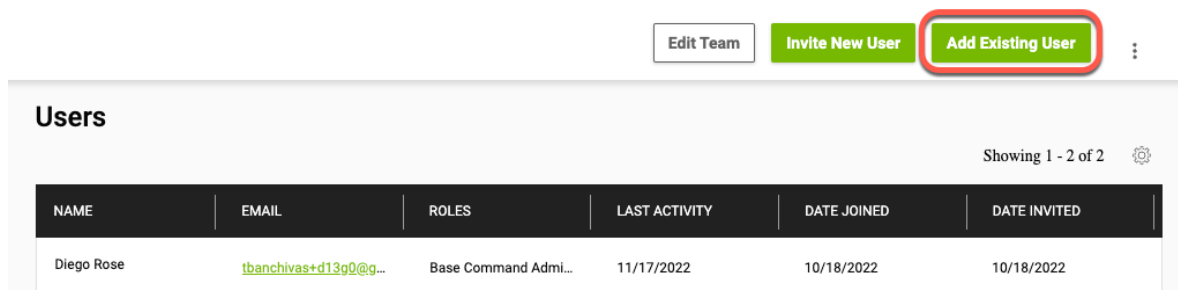
4. In the Add User dialog, follow the steps in section [Creating Users](#) to fill out the add user form and invite the new user to the team. Make sure the user is invited at the desired team context.

Users can be members of more than one team. To add a user to another team, repeat these steps for any additional teams.

## 2.4.5. Adding an Existing User to a Team

Org owners or org level user administrators can add users to any team in the organization. Team user administrators can add users to their teams.

1. Log on to the [NGC website](#).
2. Click Organization > Teams from the left navigation menu, then select the team that you want to add a user.
3. On the Users page, click Add Existing User.



4. In the Find Existing User dialog, enter the name of the user you want to add.



**Find Existing User** ✕

Search to find the existing user you would like to add to **basecommand**

**Find User**

Search | ▾

Cancel Edit User

5. Select the user and click Edit User.
6. On the Update User page, assign the user to the desired team and roles. Click Add Role to save your changes.

Organization > Users > Update User

### Update User

**User Information**

Please complete this section with the user's information.

**First Name \***  **Last Name**  **Email \***

**Roles** Add Role

Please select a context for role assignment.

Organization (xvy2tenvwbmj)

Team  X | ▾

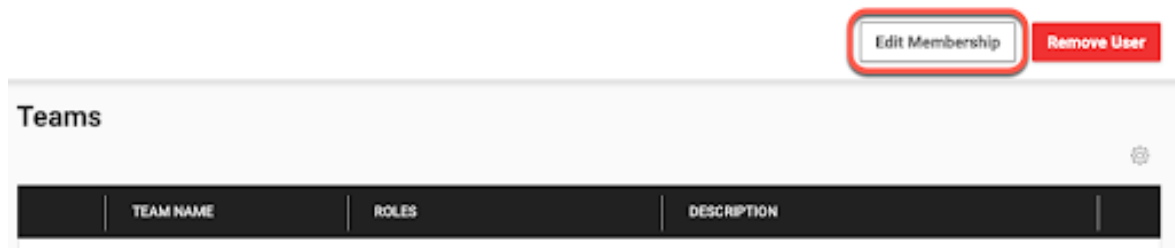
<p><b>Base Command Platform</b></p> <input type="checkbox"/> Admin <input type="checkbox"/> User <input type="checkbox"/> Viewer	<p><b>Fleet Command</b></p> <input type="checkbox"/> Admin <input checked="" type="checkbox"/> Operator <input type="checkbox"/> Viewer	<p><b>NVIDIA AI Enterprise</b></p> <input checked="" type="checkbox"/> Viewer	<p><b>Omniverse Cloud</b></p> <input type="checkbox"/> Admin <input type="checkbox"/> User <input type="checkbox"/> Read
<p><b>Private Registry</b></p> <input type="checkbox"/> Admin <input checked="" type="checkbox"/> User <input type="checkbox"/> Read	<p><b>Organization</b></p> <input type="checkbox"/> User Admin		

Users can be members of more than one team. To add a user to another team, repeat these steps for any additional teams.

## 2.4.6. Changing User Roles

You can change user assignments and roles for any users you create.

1. Log on to the [NGC website](#).
2. Select the org and team for which you want to change the user role.  
Click your user icon to select from the list of orgs, select an org, and if applicable, select a team.
3. Click Organization > Users from the left side menu.  
A list of all the users in the current registry space appears.
4. Select the user whose role you want to change.  
The User Information form appears.
5. Click Edit Membership.



A prompt appears for editing membership roles.

6. You can assign new roles, update and delete user [roles](#) and click Add Role when done.

## 2.5. Introduction to the NGC Catalog and NGC CLIs

### Unified NGC and Enterprise Catalog



Note: The Enterprise Catalog, which housed software supported by NVIDIA AI Enterprise (NVAIE), is now integrated into the NGC Catalog (public), providing users with a cohesive and comprehensive platform. NVAIE customers, with their active NVAIE entitlement, can access software and features exclusive to them from within the NGC Catalog.

The NGC Catalog aims to provide a centralized catalog of publicly available entities (e.g., containers, models, resources) alongside those that are part of products called *entitled* entities. This approach enables users to search and filter seamlessly across all entities, for a more efficient and improved user experience.

Users can view and download entitled entities by signing in to NGC. The NGC CLI is also available for downloading software using the API key. Access to all granted products will remain even when switching org/team context. Unauthenticated users will see a prompt to log in or gain access to the product when attempting to download gated features or entitled entities.

Publishers can publish and map entities to products. Access to entities is restricted by entity type, entity access type, user subscriptions, and roles, enhancing security and

control. For entitled entities, guest users are encouraged to convert to registered or subscribed status to access product-specific entities.

### Introduction to NGC CLIs

The NGC CLIs are command-line interfaces for managing content within the NGC Registry. The CLI operates within a shell and lets you use scripts to automate commands.

### NGC Catalog CLI

The NGC Catalog CLI is available to you if you have guest access to the NGC Registry, and with it, you can

- ▶ View a list of GPU-accelerated Docker container images, pre-trained deep-learning models, and scripts for creating deep-learning models.
- ▶ Download container images, models, and resources.

### NGC Registry CLI

The NGC Registry CLI is available to you if you are logged in with your own NGC account or with an NGC Private Registry account, and with it, you can

- ▶ View a list of GPU-accelerated Docker containers available and detailed information about each image.
- ▶ See a list of deep-learning models and resources and detailed information about them.
- ▶ Download container images, models, and resources.
- ▶ Upload container images, models, and resources.
- ▶ Create and manage users and teams (available to NGC Private Registry administrators).

For more details and best practices, visit the [NGC CLI documentation page](#).

## 2.5.1. Installing NGC Registry CLI

To install NGC Registry CLI,

1. Log in to your enterprise account on the NGC website (<https://ngc.nvidia.com>).
2. In the top right corner, click your user account icon and select Setup, then click Downloads under CLI from the Setup page.
3. From the CLI Install page, click the Windows, Linux, or macOS tab, according to the platform from which you will be running NGC Registry CLI.
4. Follow the instructions to install the CLI.
5. Verify the installation by entering `ngc --version`.  
The output should be `NGC CLI x.y.z` where x.y.z indicates the version.

## 2.5.2. Managing Users and Teams

This section applies to the organization and team administrators.



As the NGC administrator for your organization, you can invite other users to join your organization's NGC account. Users can then be assigned as members of teams within your organization. Teams are useful for keeping custom work private within the organization.

The general workflow for building teams of users is as follows:

1. The organization admin invites users to the organization's NGC account.
2. The organization admin creates teams within the organization.
3. The organization admin adds users to appropriate teams, and typically assigns at least one user to be the team admin.
4. The organization or team admin can then add other users to the team.

### 2.5.2.1. Inviting users to the organization's NGC account

Required Role: Org Admin (REGISTRY\_WRITE\_ADMIN\_ROLE)

Syntax

```
c:\> ngc org add-user <email> <name>
```

Example of adding John Smith (email: jsmith@example.com)

```
c:\> ngc org add-user jsmith@example.com "John Smith"
```

### 2.5.2.2. Creating teams

Required Role: Org Admin (REGISTRY\_WRITE\_ADMIN\_ROLE)

Syntax

```
c:\> ngc org add-team <name> <description>
```

Example of adding Team A

```
c:\> ngc org add-team team_a "Team A"
```

```
Team created.
-----
Team Information
Id: 363
Name: team-a
Description: Team A
Deleted: False
-----
```

### 2.5.2.3. Adding users to teams

Required Role: Org Admin (REGISTRY\_WRITE\_ADMIN\_ROLE) or Team Admin (REGISTRY\_WRITE\_TEAM\_ADMIN\_ROLE)

Syntax

```
c:\> ngc team add-user <email> <name>
```

Example of adding existing user John Smith to Team A as a regular user

```
c:\> ngc team add-user jsmith@example.com "John Smith" --team team-a --role
REGISTRY_WRITE_USER_ROLE
```

**Note:**

You do not need the --team argument if the target team is already set in your current NGC configuration.

## 2.5.2.4. Creating a team and adding a user in the same command

Required Role: Org Admin (REGISTRY\_WRITE\_ADMIN\_ROLE)

**Syntax**

```
c:\> ngc org add-user <email> <name> --team <name> --role <user-role>
```

Example of inviting new user John Smith to Team A as a team admin

```
c:\> ngc org add-user jsmith@example.com "John Smith" --team team-a --role
REGISTRY_WRITE_TEAM_ADMIN_ROLE
```

**Note:**

You do not need the --team argument if the target team is already set in your current NGC configuration.

## 2.5.2.5. Creating a team and adding a user in the same command

Role	Service	Access Levels
ADMIN	ACE	READ, ADMIN, WRITE
ADMIN	CONTAINER	READ, ADMIN, WRITE
ADMIN	DATASET	READ, ADMIN, WRITE
ADMIN	HELM	READ, ADMIN, WRITE
ADMIN	JOB	READ, ADMIN, WRITE
ADMIN	MODEL	READ, ADMIN, WRITE
ADMIN	MODELSCRIPT	READ, ADMIN, WRITE
ADMIN	ORG	READ, ADMIN, WRITE
ADMIN	TEAM	READ, ADMIN, WRITE
ADMIN	USER	READ, ADMIN, WRITE
ADMIN	WORKSPACE	READ, ADMIN, WRITE
EGX_ADMIN	EGX	READ, ADMIN, WRITE
EGX_ADMIN	ORG	READ, ADMIN, WRITE
EGX_ADMIN	TEAM	READ, ADMIN, WRITE
EGX_ADMIN	USER	READ, ADMIN, WRITE
EGX_READ	EGX	READ

Role	Service	Access Levels
EGX_READ	ORG	READ
EGX_READ	TEAM	READ
EGX_USER	EGX	READ, WRITE
EGX_USER	ORG	READ, WRITE
EGX_USER	TEAM	READ, WRITE
REGISTRY_READ	CONTAINER	READ
REGISTRY_READ	HELM	READ
REGISTRY_READ	MODEL	READ
REGISTRY_READ	MODELSRIPT	READ
REGISTRY_READ	ORG	READ
REGISTRY_READ	TEAM	READ
REGISTRY_ADMIN	CONTAINER	READ, ADMIN, WRITE
REGISTRY_ADMIN	HELM	READ, ADMIN, WRITE
REGISTRY_ADMIN	MODEL	READ, ADMIN, WRITE
REGISTRY_ADMIN	MODELSRIPT	READ, ADMIN, WRITE
REGISTRY_ADMIN	ORG	READ, ADMIN, WRITE
REGISTRY_ADMIN	TEAM	READ, ADMIN, WRITE
REGISTRY_ADMIN	USER	READ, ADMIN, WRITE
REGISTRY_USER	CONTAINER	READ, WRITE
REGISTRY_USER	HELM	READ, WRITE
REGISTRY_USER	MODEL	READ, WRITE
REGISTRY_USER	MODELSRIPT	READ, WRITE
REGISTRY_USER	ORG	READ, WRITE
REGISTRY_USER	TEAM	READ, WRITE
USER_ADMIN	CONTAINER	READ, ADMIN, WRITE
USER_ADMIN	HELM	READ, ADMIN, WRITE
USER_ADMIN	MODEL	READ, ADMIN, WRITE
USER_ADMIN	MODELSRIPT	READ, ADMIN, WRITE
USER_ADMIN	ORG	READ, ADMIN, WRITE
USER_ADMIN	TEAM	READ, ADMIN, WRITE
USER_ADMIN	USER	READ, ADMIN, WRITE
USER	ACE	READ, WRITE
USER	CONTAINER	READ, WRITE
USER	DATASET	READ, WRITE
USER	HELM	READ, WRITE
USER	JOB	READ, WRITE
USER	MODEL	READ, WRITE
USER	MODELSRIPT	READ, WRITE

<b>Role</b>	<b>Service</b>	<b>Access Levels</b>
USER	ORG	READ, WRITE
USER	TEAM	READ, WRITE
USER	WORKSPACE	READ, WRITE

---

# Chapter 3. Docker Containers

Over the last few years there has been a dramatic rise in the use of software containers for simplifying deployment of data center applications at scale. Containers encapsulate an application along with its libraries and other dependencies to provide reproducible and reliable execution of applications and services without the overhead of a full virtual machine.

GPU support within Docker containers enables GPU-based applications that are portable across multiple machines in a similar way to how Docker® enables CPU-based applications to be deployed across multiple machines.

## **Docker container**

A Docker container is an instance of a Docker image. A Docker container deploys a single application or service per container.

## **Docker image**

A Docker image is simply the software (including the filesystem and parameters) that you run within a nvidia-docker container.

## 3.1. What Is A Docker Container?

A Docker container is a mechanism for bundling a Linux application with all of its libraries, data files, and environment variables so that the execution environment is always the same, on whatever Linux system it runs and between instances on the same host.

Unlike a VM which has its own isolated kernel, containers use the host system kernel. Therefore, all kernel calls from the container are handled by the host system kernel. DGX™ systems uses Docker containers as the mechanism for deploying deep learning frameworks.

A Docker container is the running instance of a [Docker image](#).

## 3.2. Why Use A Container?

One of the many benefits to using containers is that you can install your application, dependencies and environment variables one time into the container image; rather than on each system you run on. In addition, the key benefits to using containers also include:

- ▶ Install your application, dependencies and environment variables one time into the container image; rather than on each system you run on.
- ▶ There is no risk of conflict with libraries that are installed by others.
- ▶ Containers allow use of multiple different deep learning frameworks, which may have conflicting software dependencies, on the same server.
- ▶ After you build your application into a container, you can run it on lots of other places, especially servers, without having to install any software.
- ▶ Legacy accelerated compute applications can be containerized and deployed on newer systems, on premise, or in the cloud.
- ▶ Specific GPU resources can be allocated to a container for isolation and better performance.
- ▶ You can easily share, collaborate, and test applications across different environments.
- ▶ Multiple instances of a given deep learning framework can be run concurrently with each having one or more specific GPUs assigned.
- ▶ Containers can be used to resolve network-port conflicts between applications by mapping container-ports to specific externally-visible ports when launching the container.

## 3.3. Using NGC Container Registry from the Docker Command Line

### 3.3.1. Accessing the NGC Container Registry

You can access the NGC container registry by running a Docker command from your client computer. You are not limited to using your NVIDIA DGX platform to access the NGC container registry. You can use any Linux computer with Internet access on which Docker is installed.

Before accessing the NGC container registry, ensure that the following prerequisites are met:

- ▶ Your NGC account is activated.
- ▶ You have an NGC API key for authenticating your access to NGC container registry. For more information, see [Generating Your NGC API Key](#).
- ▶ You are logged in to your client computer as an administrator user.

An alternate approach for enabling other users to run containers without giving them sudo privilege, and without having to type sudo before each docker command, is to add each user to the docker group, with the command:

```
sudo usermod -aG docker $USER
```

While this approach is more convenient and commonly used, it is less secure because any user who can send commands to the docker engine can escalate privilege and

run root level operations. If you choose to use this method, only add users to the docker group who you would trust with root privileges.

1. Log in to the NGC container registry.

```
sudo docker login nvcr.io
```

2. When prompted for your user name, enter the following text:

```
$oauthtoken
```

The `$oauthtoken` user name is a special user name that indicates that you will authenticate with an API key and not a user name and password.

3. When prompted for your password, enter your NGC API key as shown in the following example.

```
Username: $oauthtoken
Password: my-api-key
```



Tip: When you get your API key as explained in [Generating Your NGC API Key](#), copy it to the clipboard so that you can paste the API key into the command shell when you are prompted for your password.

### 3.3.2. Uploading an NVIDIA Container Image onto Your System

No container images are preloaded onto a DGX system. Instead, containers are available for download from the NGC container registry. NVIDIA has provided a number of containers for download from the NGC container registry. If your organization has provided you with access to any custom containers, you can download those as well.

Before loading an NGC container image, ensure that the following prerequisites are met:

- ▶ You have read access to the registry space that contains the container image.
- ▶ You are logged in to nvcr.io as explained in [Accessing the NGC Container Registry](#).



Tip: To browse the available containers in the NGC container registry, use a web browser to log in to your NGC account on the [NGC website \(http://ngc.nvidia.com/\)](http://ngc.nvidia.com/).

1. Run the command to download the container that you want from the registry.

```
sudo docker pull registry/registry-space/repository:tag
```

#### **registry**

The URL of the container registry, which for the NGC container registry is `nvcr.io`.

#### **registry-space**

The name of the space within the registry that contains the container. For example, `nvidia` is the registry space for containers provided by NVIDIA.

#### **repository**

Repositories are collections of containers of the same name, but distinguished from each other by their tags. Think of it as the main container name.

#### **tag**

A tag that identifies the version of the container.

- To confirm that the container was downloaded, list the Docker images on your system.

```
sudo docker images
```

The following are several examples of pulling container images.

- ▶ Example of pulling `tensorflow:18.06-py3` from the `nvidia` registry space.
- ▶ Example of pulling a custom container image tagged `v2.0` from the `acme` organization registry space.
- ▶ Example of pulling a custom container image tagged `v2.0` from the `acme/team` team registry space.

```
~$ sudo docker pull nvcr.io/nvidia/tensorflow:18.06-py3
```

```
~$ sudo docker pull nvcr.io/acme/custom-image:v2.0
```

```
~$ sudo docker pull nvcr.io/acme/zoom/custom-image:v2.0
```

### 3.3.3. Tagging and Pushing a Container Image

You can upload custom images to the registry if you have write access to the registry space. Uploading a container image involves first tagging the image and then pushing the image to the registry space.

In the following examples, the user is a member of the Acme (`mx80i8djw7m`) organization and the Zoom team within the Acme organization. Refer to [Joining an Org as Org Owner](#) on how to find your Org name.

- ▶ Tagging Example

This example tags a local container image `mycaffe` in the `mx80i8djw7m/zoom` team space with `v1.5`.

```
~$ sudo docker tag mycaffe nvcr.io/mx80i8djw7m/zoom/mycaffe:v1.5
```

- ▶ Pushing Example

This example pushes version `v1.5` of the `mycaffe` local container image to the `mx80i8djw7m/zoom` team space:

```
~$ sudo docker push nvcr.io/mx80i8djw7m/zoom/mycaffe:v1.5
```

## 3.4. Using the Container Registry

The `ngc registry image` commands let you access ready-to-use GPU-accelerated container images from the registry.

### 3.4.1. Viewing Container Image Information

There are several commands for viewing information about available container images.

To list container images:

```
C:\>ngc registry image list
+-----+-----+-----+-----+-----+-----+
| Name           | Repository | Latest Tag | Image Size | Updated | Permission |
+-----+-----+-----+-----+-----+-----+

```



				Date	
BigDFT	hpc/bigdft	cuda10-ubun tu1804-ompi 4-mkl	2.37 GB	Oct 18, 2019	unlocked
CANDLE	hpc/candle	20180326	1.52 GB	Oct 18, 2019	unlocked
...					

"Unlocked" permissions indicate images that do not require an API key to access.

To view detailed information about a specific image, specify the image and the tag.

Example:

```
C:\>ngc registry image info nvidia/caffe:19.02-py2
-----
Image Information
Name: nvidia/caffe:19.02-py2
Architecture: amd64
Schema Version: 1
-----
```

### 3.4.2. Pulling a Container Image

With the NGC Registry CLI you can pull (download) images to your system.

To pull an image to your registry space, specify the image and, optionally, the tag.

```
C:\>ngc registry image pull <image-name>[:<tag>]
```

If a tag is not specified, then the tag 'latest' will be used.

### 3.4.3. Pushing a Container Image

With the NGC Registry CLI you can push (upload) images to your registry space.

To push an image to your registry space, specify the image and, optionally, the tag.

```
C:\>ngc registry image push <image-name>[:<tag>]
```

If a tag is not specified, then the tag 'latest' will be used.

### 3.4.4. Removing a Container Image

With the NGC Registry CLI you can removed images that are no longer needed from your registry space.

To remove all versions of an image, specify the image.

```
C:\>ngc registry image remove <image-name>
```

To remove a specific image version, specify the image and tag.

```
C:\>ngc registry image remove <image-name>:<tag>
```

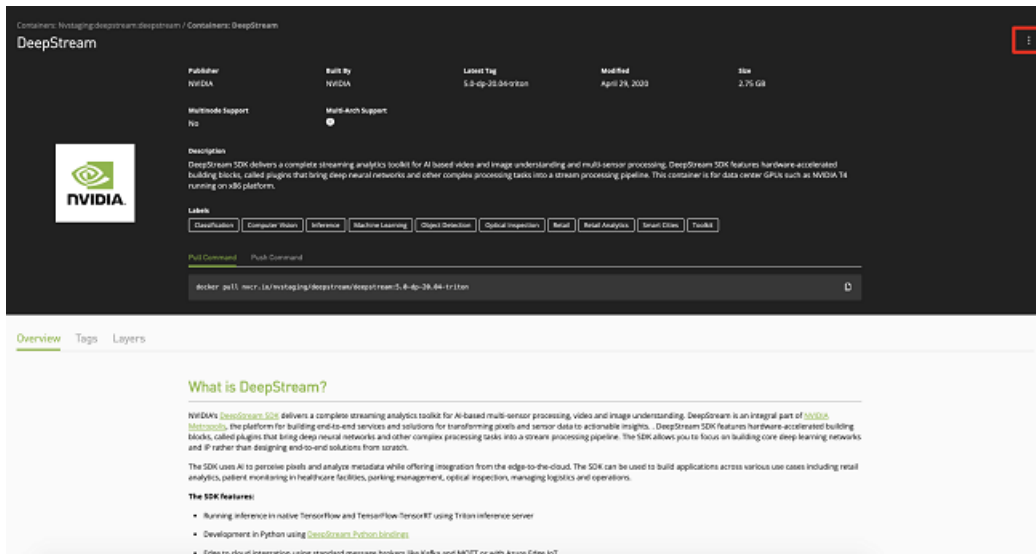
## 3.5. Updating Container Metadata

You can find best practices on how to fill out the metadata for your container in the Product Page Guidelines.

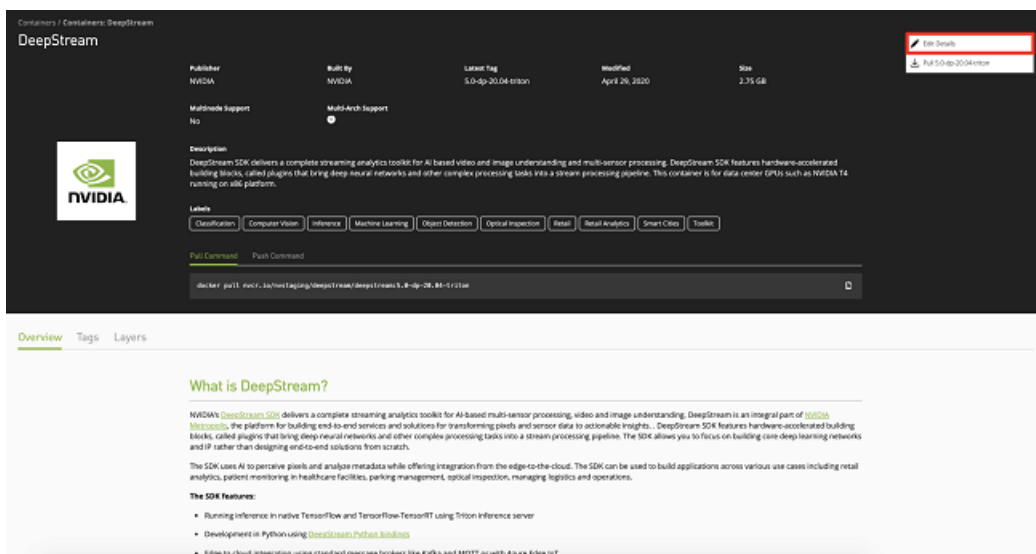
### 3.5.1. Updating Container Metadata via the NGC Website

Perform the following instructions to update the container metadata using the NGC website.

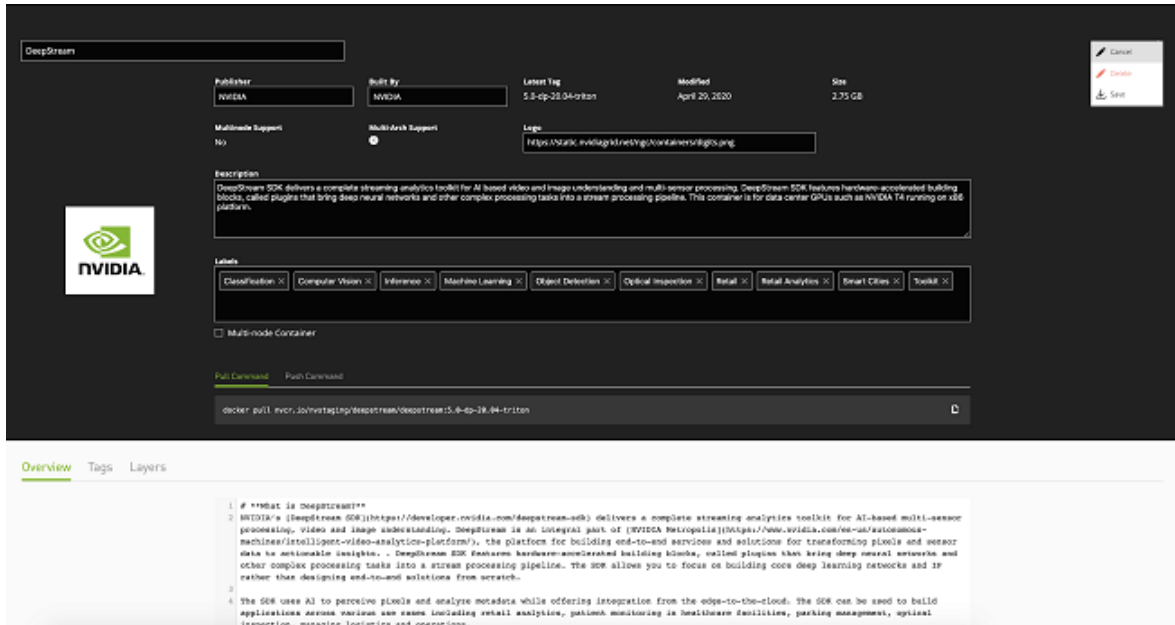
1. Click on the vertical ellipsis in the right upper corner of your container product page to reveal the entity action menu.



2. Select Edit Details from the entity action menu.



3. Update the container description and all other container metadata as needed.
4. To save your changes, click again on the vertical ellipsis to reveal the entity action menu and select Save.



### 3.5.2. Updating Container Metadata Using the NGC CLI

With the NGC Registry CLI you can update the container description and all the other container metadata.

With the NGC Registry CLI you can update the container description and all the other container metadata.

To update container metadata, use the following command.

```
ngc registry image update [--ace <name>] [--built-by <name>] [--debug]
                        [--desc <desc>] [--format_type <fmt>]
                        [--label <label>] [--logo <url>] [--org <name>]
                        [--overview <file.md>] [--publisher <publisher>]
                        [--team <name>] [-h]
                        <image>[:<tag>]
```

Specify a named argument (field that will be updated, and values to update field) as well as a positional argument (name of the container image and, optionally, tag).

Positional Arguments:

**<image>[:<tag>]**

Name of the image repository or tagged image, <image>[:<tag>]

Named Arguments

**--debug**

Enable debug mode.

**--format\_type**

Possible choices: ascii, csv, json. Specify the output format type. Supported formats are: ['ascii', 'csv', 'json']. Only commands that produce tabular data support csv format. Default: ascii.

**--org**

Specify the organization name. Use "--org no-org" to override other sources and specify no org. Default: current configuration.

**--ace**

Specify the ACE name. Use "--ace no-ace" to override other sources and specify no ACE. Default: current configuration.

**--team**

Specify the team name. Use "--team no-team" to override other sources and specify no team. Default: current configuration.

**--desc**

Description for the target image.

**--overview**

Documentation (text or markdown file) for the image.

**--label**

A label to describe the repository. Can be used multiple times.

**--logo**

A URL pointing to the logo for the repository.

**--publisher**

The person or entity publishing the image.

**--built-by**

The person who built the container image.

Specify the image and, optionally, the tag.

Example: Changing description of a container image

To view the existing container metadata use the following command.

```
$ ngc registry image info nvidia/testcontainer
-----
Image Repository Information
Name: testcontainer
Short Description: Test description.
Built By: Kristina
Publisher: NVIDIA
Logo: www.logo.com/logo.png
Labels: Machine Learning, Classification, Retail
Public: No
Last Updated: May 8, 2020
Latest Image Size: 60.27 MB
```

```
Latest Tag: 3.0
Tags:
  3.0
  2.0
  1.0
```

Update the container description with the following command.

```
$ ngc registry image update --desc "A test container image with useful tools."
  nvidia/testcontainer
```

```
Updating repository metadata
Repository metadata updated.
```

To check if the update was successful, run the info command again

```
$ ngc registry image info nvidia/testcontainer
```

```
-----
Image Repository Information
Name: testcontainer
Short Description: A test container image with useful tools.
Built By: Kristina
Publisher: NVIDIA
Logo: www.logo.com/logo.png
  Labels: Machine Learning, Classification, Retail
Public: No
Last Updated: May 8, 2020
Latest Image Size: 60.27 MB
Latest Tag: 3.0
Tags:
  3.0
  2.0
  1.0
-----
```

```
$ ngc registry image update Command overview
```

## 3.6. Multi-architecture Support for NGC Container Images

When running an image, docker will automatically select an image variant which matches your OS and architecture.

NGC Container Registry now allows users to leverage [docker multi-architecture](#). It can support multiple architectures, which means that a single image may contain variants for different architectures like ARM, x86, Power and others; and sometimes for different operating systems, such as Windows.

### Manifest Lists and Tags

NGC Container Registry now supports the manifest list schema now application/vnd.docker.distribution.manifest.list.v2+json providing the ability to assign multiple tags per image. For inspection of manifest list read instructions [here](#).

NGC UI allows you to navigate through the supported architecture.

Overview **Tags** Layers

Multi-Arch Size

**multi**  
03/09/2020 10:38AM 48.05 MB

Manifest

DIGEST	OS/ARCH	COMPRESSED SIZE	CREATED
sha256:3f26741aa0e01b0edfd4e08a411e43756ea7b9d649b3b272cea7fff981a4519b	linux/amd64	48.05 MB	03/09/2020
sha256:6d86b91bfb22c74a1ddc5c2ac8604f3de5621bd4e016a5b3866ec0cf655c712	linux/arm64	46.89 MB	03/09/2020

---

# Chapter 4. NGC Models

The NGC private registry lets you upload and access deep-learning models.

## 4.1. Creating New NGC Models Using the NGC CLI

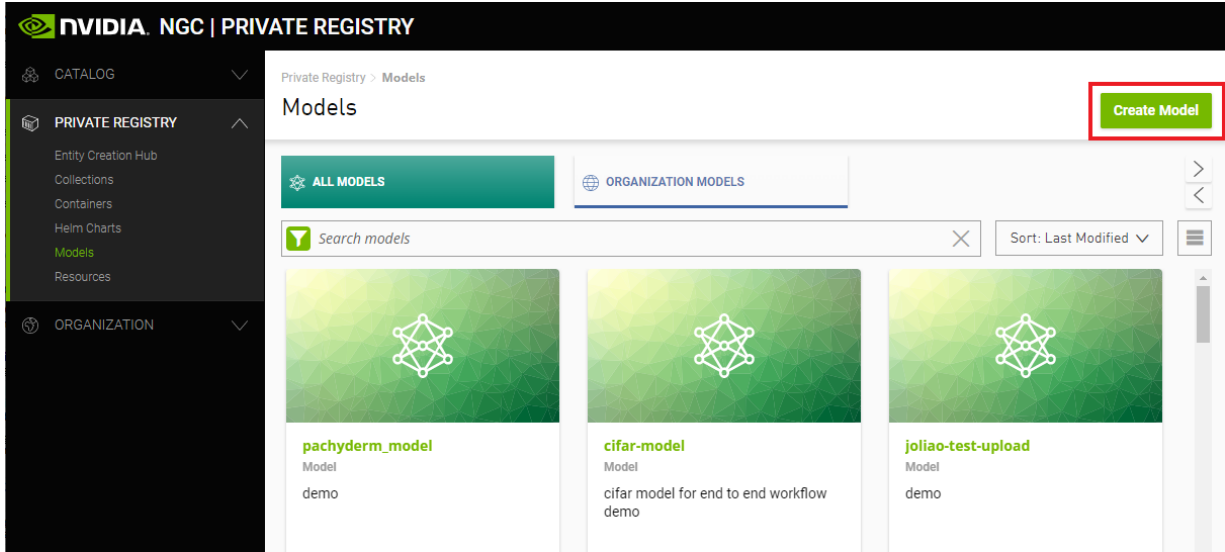
Issue `ngc registry model create -h` to see a description of available options and command descriptions for creating as model.

This example creates a new model called "Final Review Model" with all required and optional arguments used:

```
$ ngc registry model create \  
--application OBJECT_DETECTION \  
--format "cpkt" \  
--framework TensorFlow \  
--precision FP16 \  
--short-desc "A model for object detection using TensorFlow" \  
--built-by "My Name" \  
--display-name "Final Review Model" \  
--label "fast" --label "sparkly" \  
--owner-name "MyTeam" \  
--overview-filename /path/to/my/overview/file.md \  
--publisher "NVIDIA MyTeam" \  
--public-dataset-license <license> \  
--public-dataset-link "www.example.com" \  
--public-dataset-name "200_10x200_images" \  
nvidia/myteam/final_review_model
```

## 4.2. Creating a New Model Using the NGC Website

To create a new model asset, select Private Registry > Models from the left navigation pane. Click Create Model on the top right of the page.



The Create Model page walks you through the process of creating a new model asset.



### Create Model

Cancel Create

 **Create Model or Add Version**  
To learn more about Models or how to get started click [here](#)

Create Model Add New Version

#### Basic Information

Please complete this section to describe your resource.

**Name\***  
Ex: fine\_tuning\_flowtron

**Publisher** Ex: NVIDIA **Display Name\*** Ex: Fine-tuning Flowtron **Use Case** Select a use case for the model.

**Framework\*** Select a framework for the model. **Precision** Select a precision for the model. **Model Format** Select a format for the model.

**Description\***  
Enter a brief description of your model.

**Logo**  
Enter a URL to upload a logo.

#### Overview (optional)

Enter your resource overview in Markdown format.

1

#### Labels

For increased discoverability, we highly recommend you select one of the predefined labels below.

<b>Technology</b> <input type="checkbox"/> AI <input type="checkbox"/> Deep Learning	<b>Application</b> <input type="checkbox"/> Annotation <input type="checkbox"/> AR/VR <input type="checkbox"/> Automatic Speech Recognition <input type="checkbox"/> Autonomous Machines <input type="checkbox"/> Autonomous Vehicles	<b>Industry</b> <input type="checkbox"/> Aerospace <input type="checkbox"/> Automotive <input type="checkbox"/> Cloud Services <input type="checkbox"/> Energy <input type="checkbox"/> Financial Services	<b>Other</b> <input type="checkbox"/> AMP <input type="checkbox"/> Caffe <input type="checkbox"/> FP16 <input type="checkbox"/> FP32 <input type="checkbox"/> HDF5
--	--	---	---

You may also add your own custom labels.  
*Click to add custom labels. Press enter to confirm.*

Label Sets Create Label Set

#### Quick Deploy (optional)

Use this feature to allow users to a deploy this entity to an AI platform on a specified cloud service provider. This will be displayed as a "Deploy" button on the details page.

Enable Quick Deploy

Once this form has been completed and submitted you will have the option of creating a model version. This step can be skipped and completed later. See “Uploading A New Version” for more information.

Field	Validation	Description	Options
Name	String	The name of the model	-
Publisher	String	The name of the individual who owns the asset (dropdown)	-
Description	String	Short description of the model	-
Overview	Markdown (String)	A place to share more details/usage instructions for the model	-
Labels	String (List)	Tags to make the asset more discoverable	-
Use Case	String	Intended use case	Annotation, Automatic Speech Recongition, Image Classification, Image Segmentation, Image Synthesis, Natural Language Processing, Object Detection, Translation
Framework	String	Deep learning framework used to build the model	Caffe, Clara, NeMo/PyTorch, PyTorch, TensorFlow, TensorRT, Transfer Learning Toolkit
Model Format	String	Output format of the weights file	caffemodel, HDF5, ONNX, protobuf, PyTorch PTH, SavedModel, TensorFlow CKPT, TensorRT Plan, TLT
Precision	String	Training precision used	AMP, FP16, FP32, INT8

## 4.3. Uploading a New NGC Model Version Using the NGC CLI

Issue `ngc registry model upload-version -h` to see a description of available options and command descriptions for uploading a model version.

An example using all required and optional arguments to create model version 1 for the model created in the previous section.

```
$ ngc registry model upload-version \
  --accuracy-reached 96.5 \
  --batch-size 2000 \
  --gpu-model "V100" \
  --memory-footprint 4GB \
  --num-epochs 100 \
  --desc "A new and exciting version: 1" \
  --link "www.example.com/model/v1" \
  --link-type Other \
  --owner-name "My Name" \
  --source path/to/my/model/version/dir \
  nvidia/myteam/final_review_model:1
```

### Adding Custom Metrics

You can also upload custom metrics tables for each model version. Each table can hold up to twelve key-value attribute pairs. Three tables maximum per model version.

Metrics tables are defined as JSON tables - one table per file. You can add the table to the upload with `--metrics-file`.

Some example metrics files:

#### *zeppelin\_table.json*

```
{
  "name": "ZeppelinTable",
  "attributes": [
    {"key": "Robert", "value": "Plant"},
    {"key": "Jimmy", "value": "Page"},
    {"key": "John", "value": "Bonham"},
    {"key": "John", "value": "Paul Jones"}
  ]
}
```

#### *rhcp\_table.json*

```
{
  "name": "RHCPTable",
  "attributes": [
    {"key": "Anthony", "value": "Keidis"},
    {"key": "Michael", "value": "Balzary"},
    {"key": "John", "value": "Frusciante"},
    {"key": "Chad", "value": "Smith"}
  ]
}
```

The above example with a custom metrics tables included:

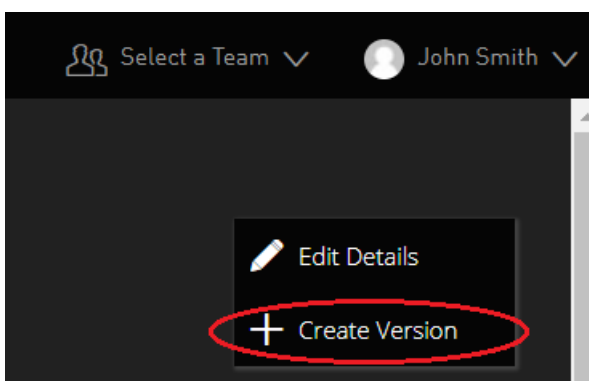
```
$ ngc registry model upload-version \
  --accuracy-reached 95.5 \
  --batch-size 2000 \
  --gpu-model "SomeGPUModel" \
  --memory-footprint 4GB \
  --num-epochs 100 \
  --desc "A new and exciting version: 1" \
  --link "www.example.com/model/v1" \
  --link-type Other \
  --owner-name "My Name" \link-type
  --metrics-file zeppelin_table.json \
```

```
--metrics-file rhcp_table.json \  
--source path/to/my/model/version/dir \  
nvidia/myteam/final_review_model:1
```

## 4.4. Uploading an NGC Model Version Using the NGC Website

There are two ways to upload a new version of a model via the NGC Website.


- ▶ Via the Model Creation page discussed above
- ▶ From the Model Details page for any model



From the version creation page, shown below, you can specify all the relevant information about the specific version that you are uploading. You can also upload files directly from your browser.

### Add Model Version

Cancel Create


Create Model or Add Version  
To learn more about Models or how to get started click [here](#)

Create Model Add New Version

**Identify Base Model**  
 Select the base Model you would like to add a Version.

**Base Model\***

**Basic Information**  
 Please complete this section to describe your model version.

**Version\***

Set this as the latest version

<b>Number of Epochs</b> Number of epochs trained	<b>Batch Size</b> Batch size for model	<b>GPU Model</b> GPU model and memory
<b>Accuracy Reached</b> Accuracy this model reached	<b>Memory Footprint</b> Memory	

**Additional Resources**  
 Add any additional resources for the model.

<b>Link Text</b> NGC, GitHub, Other...	<b>URL</b> URL to model script or additional toolsets for the model
---	--

**Model Credentials (Optional)**  
 Add up to 36 credentials. Maximum 3 tables.

[+ Add Custom Table](#)

Once you've completed the form, and uploaded any relevant files, submitting will publish the new version of the content.

### Adding Custom Metrics

As deep learning models evolve we're aware that you might also want to convey different information to distinguish between different versions. Using the NGC Model Registry, you can specify up to 36 different metrics to help people find the right versions.

When creating your version, simply "add custom metrics" to create the tables.

Custom Metrics (Optional)  
Add up to 36 metrics. Maximum 3 tables.

+ Add Custom Table

**Custom Table**

Table Title

Key Name (Column Label)	Value (Column Content)
<input type="text" value="Key"/>	<input type="text" value="Value"/>
<input type="text" value="Key"/>	<input type="text" value="Value"/>
<input type="text" value="Key"/>	<input type="text" value="Value"/>
<input type="text" value="Key"/>	<input type="text" value="Value"/>
<input type="text" value="Key"/>	<input type="text" value="Value"/>
<input type="text" value="Key"/>	<input type="text" value="Value"/>
<input type="text" value="Key"/>	<input type="text" value="Value"/>
<input type="text" value="Key"/>	<input type="text" value="Value"/>
<input type="text" value="Key"/>	<input type="text" value="Value"/>
<input type="text" value="Key"/>	<input type="text" value="Value"/>
<input type="text" value="Key"/>	<input type="text" value="Value"/>
<input type="text" value="Key"/>	<input type="text" value="Value"/>
<input type="text" value="Key"/>	<input type="text" value="Value"/>
<input type="text" value="Key"/>	<input type="text" value="Value"/>
<input type="text" value="Key"/>	<input type="text" value="Value"/>

NGC DOCS CUSTOM TABLE

KEY	VALUE
AWESOMENESS	10000000

- ▶ Model Name - String - The name of the model you for which you wish to upload a version
- ▶ Owner - String - The name of the individual who owns the asset (dropdown)
- ▶ Version - String - A way of identifying that version (we recommend SemVer)
- ▶ Overview - Markdown - A place to share more details/usage instructions for the model (shared across all versions)
- ▶ Number of Epochs - String - Number of Epochs trained (or N/A)
- ▶ Batch Size - String - Training Batch Size (or N/A)
- ▶ GPU Model - Drop Down - GPU family used for training
- ▶ Accuracy Reached - String - Accuracy of the model (or N/A)
- ▶ Memory Footprint - String - Memory Footprint used by the model
- ▶ Related Resources - You can optionally specify additional resources for your model
  - ▶ Link Text - Drop Down - The text to display for additional resources, such as containers or code samples, to accompany your version.
  - ▶ URL - String - The URL of the additional resource

Once you've entered the key/value pairs, select "add table".

## 4.5. Editing NGC Model Information Using the NGC CLI

Issue `ngc registry model update -h` to see a description of available options and command descriptions for editing a model or model version.

An example updating a model's overview file for a model.

```
$ ngc registry model update \
  --overview-filename "path/to/my/updated/overview/file.md" \
  nvidia/myteam/final_review_model
```

An example updating a model-version's accuracy reached and memory footprint.

```
$ ngc registry model update \
  --accuracy-reached 96.5 \
  --memory-footprint 16GB \
  nvidia/myteam/final_review_model:1
```

### **ngc registry model info nvidia/model-name**

show information about a model

`ngc registry model info nvidia/model-name:version`

show information about a model version

`ngc registry model list`

list available models

`ngc registry model download-version nvidia/model-name:version`

download the specified model-version

`ngc registry model remove nvidia/model-name:version`

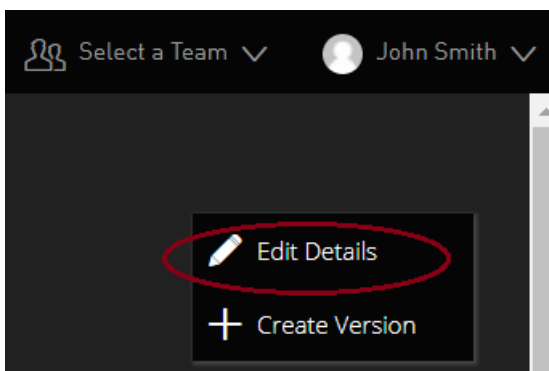
remove a model-version

`ngc registry model remove nvidia/model-name`

remove a model

## 4.6. Editing NGC Model Information Using the NGC Website

To edit a Model's metadata or overview tab, simply select "edit" from the top right of the model details page.



You can then edit any of the model's details, or even delete the model if you wish.

The screenshot displays the NGC Models interface. At the top, a search bar contains the text "NGC DOCS TEST1". To the right are buttons for "Cancel", "Delete", and "Save". Below the search bar, a metadata table lists the following information:

Field	Value
Publisher	dl@icodh3j
Application	Object Detection
Version	-
Modified	February 18, 2020
Size	0 B
Framework	TensorFlow
Model Format	SavedModel
Precision	AMP
GPU Model	-

Below the metadata is a teal neural network icon. A "Description" field contains the text "This is a description". A "Labels" section shows a tag for "TensorFlow". At the bottom, a navigation bar includes "Overview", "Version History", and "File Browser". The "Overview" tab is active, showing a "Markdown" editor area.



---

# Chapter 5. NGC Resources

The NGC private registry lets you upload and access resources for deep-learning models.

## 5.1. Before You Begin

With the NGC Registry CLI you can update the container description and all the other container metadata.

Be sure you know your context, or which org and team you are logged into. This determines which registry space your model will be uploaded. You can do this by entering the following:

```
$ ngc config current
```

If you intend to upload a model to a different registry space, or if no team is reported and you intend to upload to a team space, then you can either

- ▶ Use `ngc config set` to switch to another org or team.

```
$ ngc config set [--org <new org>][--team <new team>]
```

or

- ▶ Set the context at each command, using the same `--org` or `--team` options.

## 5.2. Uploading a Resource

The following is the general process for uploading a resource to the model script registry.

1. Create a resource in the registry.

This is a placeholder for your model and contains metadata about the resource.

Example of creating resource "cmr\_gnmt".

```
$ ngc registry resource create nvidia/cmr_gnmt
```

To see a complete list of required and optional arguments, enter the following.

```
$ ngc registry resource create -h
```

2. Upload your resource files.

Each time you upload files to the same resource, the upload becomes a unique version of the resource. You can specify the version when you upload, or let the CLI increment the version automatically.

Example: Uploading version 1 of the resource 'cmr\_gnmt' (required arguments omitted for simplicity).

```
$ ngc registry resource upload-version nvidia/cmr_gnmt:first-upload [--
source .<directory or file path for the model contents>]
-----
Transfer id: cmr_gnmt[version=first-upload] Upload status: Completed.
Uploaded local path: C:\resource
Total files uploaded: 26
Total uploaded size: 134.48 KB
Started at: 2019-03-15 17:18:09.083000
Completed at: 2019-03-15 17:18:21.698000
Duration taken: 12s seconds
-----
```

## 5.3. Updating a Resource

You can update or revise information for a resource or resource version.

The following is the basic command.

```
$ ngc registry resource update <org>/[<team>/]<resource-name[:version]>
```

To update information, use the optional arguments to specify the information to change. To see the list of arguments, run

```
$ ngc registry resource update -h
```

## 5.4. Resource Commands

The full list of optional commands for NGC resources can be seen here.

**--accuracy-reached <accuracy>**

Accuracy reached with target version.

**--ace <name>**

Specify the ACE name. Use "--ace no-ace" to override other sources and specify no ACE. Default: current configuration

**--advanced-filename <path>**

Advanced guide. Provide the path to a file that contains the "Advanced Guide" for the resource.

**--application <app>**

Target model application. Allowed values: CLASSIFICATION, OBJECT\_DETECTION, SEGMENTATION, TRANSLATION, TEXT\_TO\_SPEECH, RECOMMENDER, SENTIMENT, NLP, KUBEFLOW\_PIPELINE, OTHER.

**--batch-size <size>**  
The batch size of the target version.

**--built-by <name>**  
Builder of the target model.

**--debug**  
Enable debug mode.

**--desc <desc>**  
Full description of target version.

**--display-name <name>**  
Display name.

**--format <fmt>**  
Format of the target model.

**--format\_type <fmt>**  
Specify the output format type. Supported formats are: ascii, csv, json. Only commands that produce tabular data support csv format. Default: ascii

**--framework <fwk>**  
Framework used to train the target model. Allowed values: TensorFlow, Caffe2, CNTK, Torch, PyTorch, MXNet, Keras, Other.

**--gpu-model <model>**  
The GPU used to train the target version.

**--label <label>**  
Label for the resource. To specify more than one label, use multiple --label arguments.

**--logo <url>**  
URL for the resource logo image.

**--memory-footprint <footprint>**  
The memory footprint of the target version.

**--num-epochs <num>**  
The number of epochs for the target version.

**--org <name>**  
Specify the organization name. Use "--org no-org" to override other sources and specify no org. Default: current configuration

**--overview-filename <path>**  
Overview. Provide the path to a file that contains the overview for the resource.

**--performance-filename <path>**

Performance data. Provide the path to a file that contains the performance data for the resource.

**--precision <prec>**

Precision the target model was trained with. Allowed Values: FP16, FP32, INT8, FPBOTH, OTHER.

**--public-dataset-license <lcs>**

License for public dataset used in the target model.

**--public-dataset-link <url>**

Link to public dataset used in the target model.

**--public-dataset-name <name>**

Name of public dataset used in the target model.

**--publisher <name>**

Publisher of the target model.

**--quick-start-guide-filename <path>**

Quick start information. Provide the path to a file that contains the "Quick Start Guide" information for the resource.

**--release-notes-filename <path>**

Release notes. Provide the path to a file that contains the release notes for the resource.

**--setup-filename <path>**

Setup instructions. Provide the path to a file that contains the setup instructions for the resource.

**--short-desc <desc>**

Short description.

**--team <name>**

Specify the team name. Use "--team no-team" to override other sources and specify no team. Default: current configuration

## 5.5. Deleting a Resource

Only admins and creators of the model can delete a model.

Be sure the context is set appropriately for the resource you want to delete. For example, if you want to delete a model that you created in the team\_A space, then be sure to set the context as --team team\_A.

To remove the resource, including all versions of the resource, enter the following.

```
$ ngc registry resource remove <org>/[<team>/]<resource>
```

To remove only a specific version of the resource, enter the following.

```
$ ngc registry resource remove <org>/[<team>/]<resource:version>
```

---

# Chapter 6. NGC Helm Charts

This document describes how to use the NGC registry to manage Helm charts.

## 6.1. Introduction to NGC and Helm Charts

Helm is an application package manager running on top of Kubernetes. It lets you create Helm charts where you can define, install, and upgrade Kubernetes applications.

This document describes how to share Helm charts with others in your org or team using the NGC registry.

### Prerequisites

These instructions assume the following prerequisites are met.

- ▶ Helm v 2.x or 3.x installed

This is only required if you are creating or packaging Helm charts yourself. It is not needed otherwise.

- ▶ NGC organization account

See the section [Getting Started](#) for instructions.



Note: The asset `ngcdocstest` referenced below was created for example purposes only. It is intended merely as a guide and is not a requirement for publishing Helm assets to NGC.

## 6.2. Creating and Packaging a Helm Chart

This section describes how to package a Helm chart for publishing to NGC.

There is no need to “deploy” the Helm chart in order to publish your chart to NGC. A `.tgz` file of the chart can be published to an org in NGC without being deployed first to the GPU infrastructure.

1. Create a Helm chart template by issuing the following.

```
$ helm create <chart-name>
```

Where `<chart-name>` is the name of your choosing.

Example:

```
$ helm create ngcdocstest
```

2. Modify the contents of the template with your Helm chart data.
3. Package the Helm chart by issuing the following.

```
$ helm package <chart-name>
```

Example:

```
$ helm package ngcdocstest
```

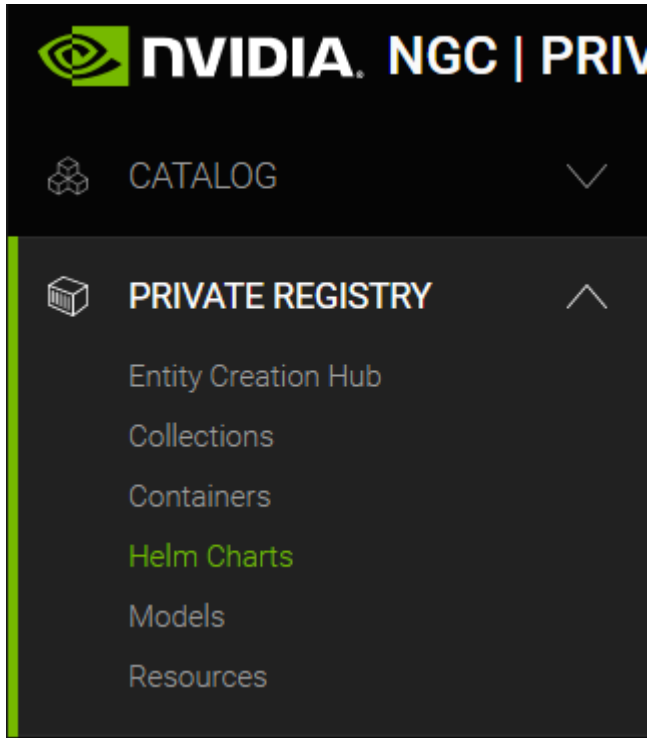
This example creates the tar package `ngcdocstest-0.1.0.tgz`.

## 6.3. Manage Helm Charts Using the NGC Web UI

### 6.3.1. Viewing the List of Helm Charts and Getting Fetch Commands

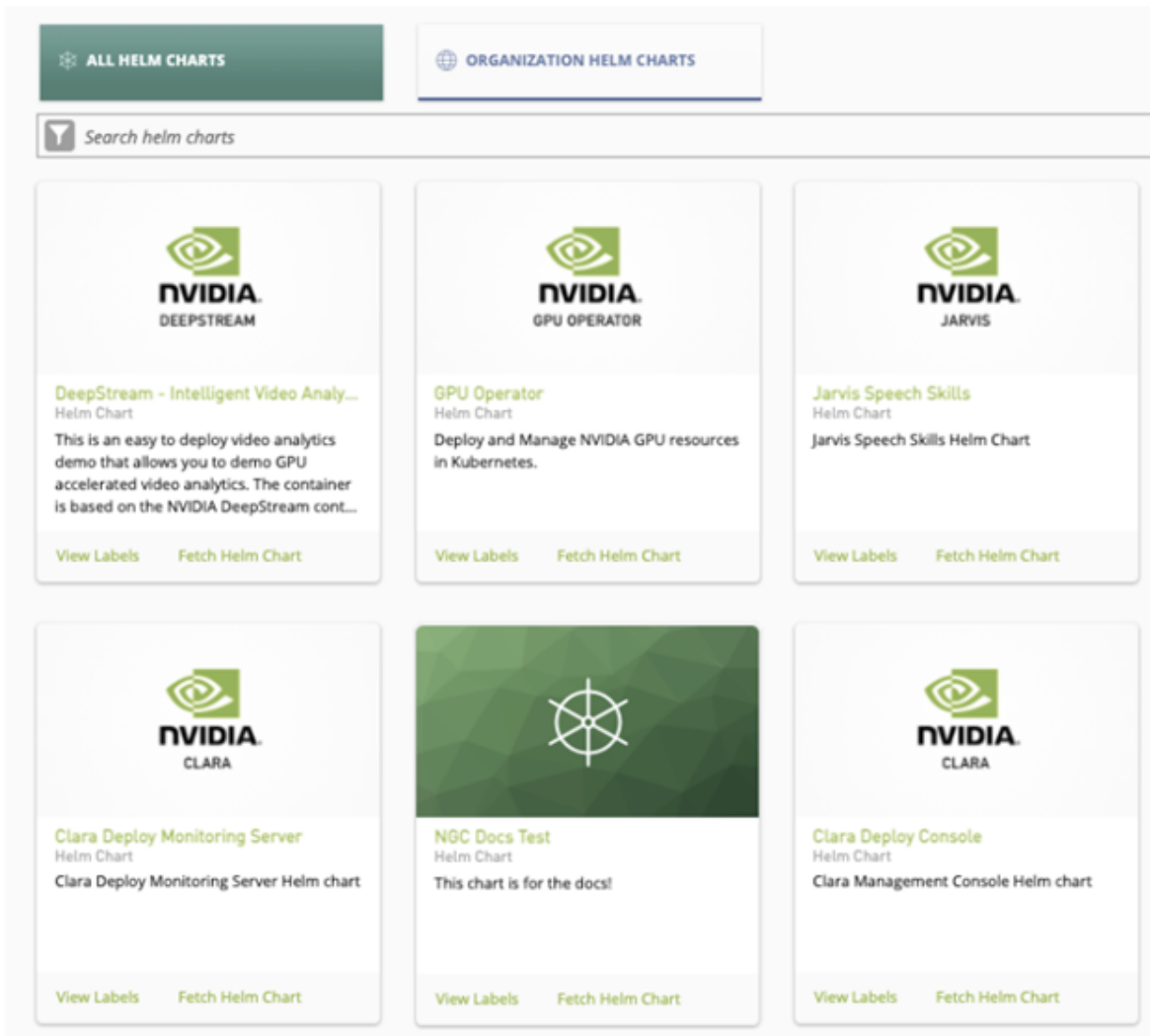
From the NGC website you can

- ▶ View the contents of the Helm chart repository.
  - ▶ Get the push command for a specific Helm chart in the repository.
1. From a browser, log in to <https://ngc.nvidia.com>.
  2. If you are a member of more than one org, select the one that contains the Helm charts that you are interested in, then click Sign In.
  3. Click Helm Charts from the left-side navigation pane.



The page presents cards for each available Helm chart.





4. Select one of the Helm chart cards.  
The page for each Helm chart provides information about the chart.

Helm Charts / Helm Charts: demo\_chart

## NGC Docs Test

Fetch 1.0.0

Publisher	Version	Size	Modified	Created
NVIDIA	0.1.0	10.52 KB	March 31, 20...	December 1...

**Description**  
This chart is for the docs!

**Labels**

Helm Chart Documentation

**Fetch Command**

```
$ helm fetch https://helm.ngc.nvidia.com/nvstaging/ngc_product/charts/demo_chart-0.1.0.tgz -
--username='$oauth_token' --password=<YOUR_API_KEY>
```

Overview File Browser Related Collections

The Fetch Command section shows the command to use for downloading the Helm chart package.

Click either the Fetch download button from the upper right corner or the copy icon next to the fetch command to copy the fetch command to the clipboard.

The File Browser tab lets you see the file content of the Helm chart package.

Overview **File Browser**

0.1.0

Filter files

FILE	SIZE	CREATED
ngcdocstest		
templates		
tests		
test-connection.yaml	395 B	8 days ago
_helpers.tpl	2 kB	8 days ago
deployment.yaml	2 kB	8 days ago
ingress.yaml	1 kB	8 days ago
NOTES.txt	2 kB	8 days ago
service.yaml	373 B	8 days ago
serviceaccount.yaml	215 B	8 days ago
.helmignore	342 B	8 days ago
Chart.yaml	108 B	8 days ago
values.yaml	421 B	8 days ago

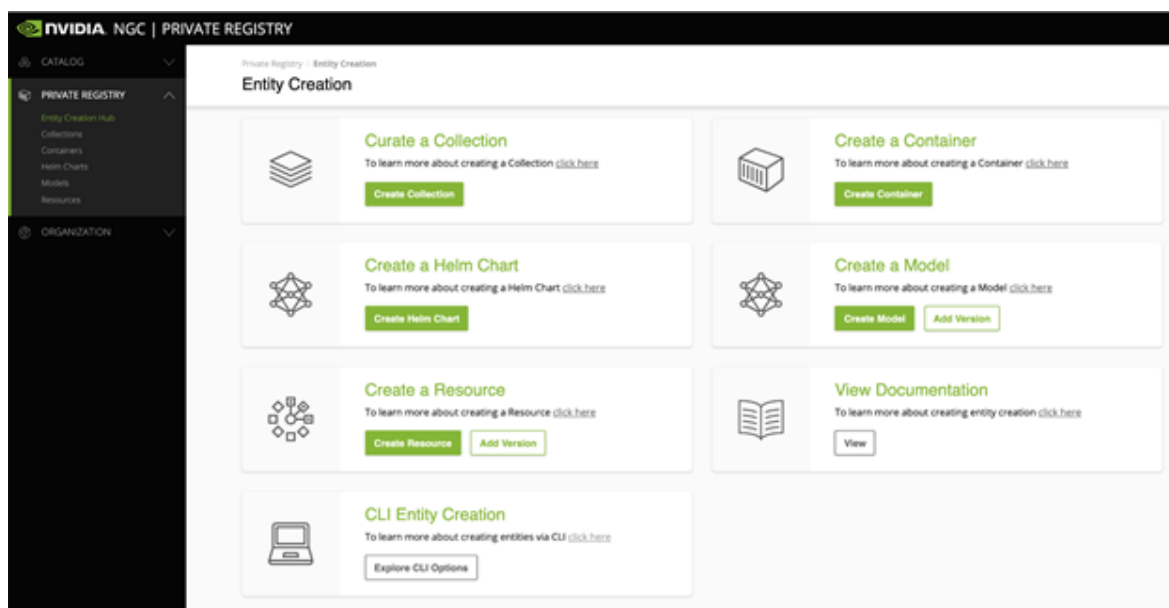
## 6.3.2. Adding Helm Charts Using the NGC Web UI



Note: Make sure you have the right permissions to create Helm Charts in your organization and/or team. You need to have the user role “Registry User” or “Registry Admin”. For details refer to [NGC Registry User Roles](#).

Before a chart can be uploaded to your organization’s registry, you must first create a record containing the basic information about the chart.

1. Click Entity Creation Hub under Private Registry section of the left side menu.



2. Click Create Helm Chart.
3. Fill in information about your Helm Chart.

4. Click “Create Helm Chart” .
5. To push (upload) a Helm chart to your org space, use the NGC CLI.

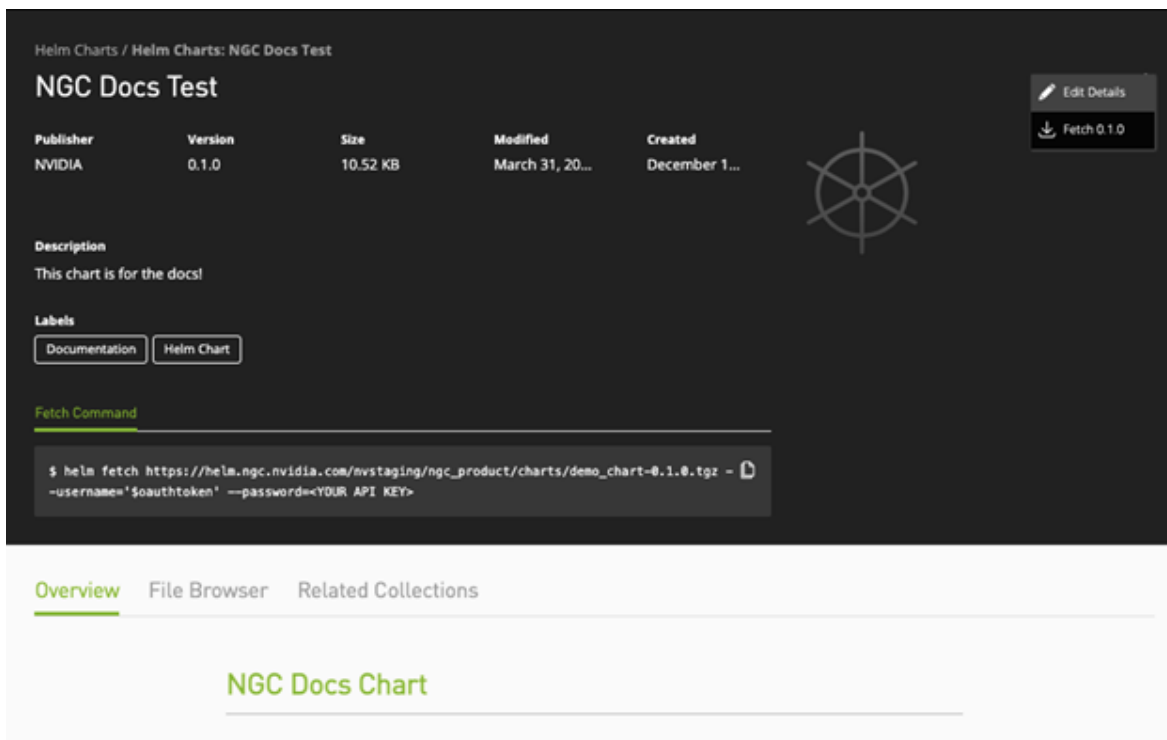
Example:

```
$ ngc registry chart push nvidian/ngdocstest:0.1.0
```

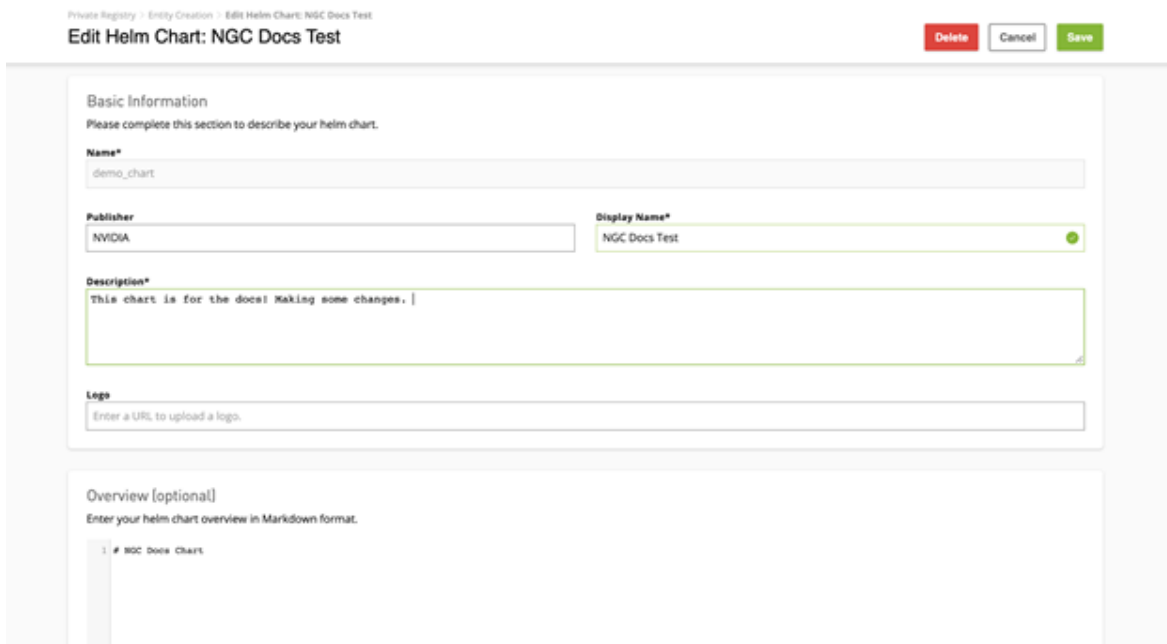
See [Pushing a Helm Chart](#) for details.

### 6.3.3. Updating the Helm Chart Page From the Website

1. To update the fields in the NGC Helm Chart page for a specific Helm chart, click Edit Details.



2. Edit each field as needed, then click Save.



## 6.3.4. Removing Helm Charts from the Web UI



Note: Make sure you have the right permissions to create Helm Charts in your organization and/or team. You need to have the user role “Registry Admin”. For details refer to [NGC Registry User Roles](#).

1. To delete a Helm chart, click Edit Details from the details page of the Helm chart to delete.

The screenshot shows the details page for a Helm chart named 'NGC Docs Test'. The page has a dark theme. At the top, it says 'Helm Charts / Helm Charts: NGC Docs Test'. Below that is the chart title 'NGC Docs Test'. There are two buttons in the top right: 'Edit Details' and 'Fetch 0.1.0'. A table lists chart details:

Publisher	Version	Size	Modified	Created
NVIDIA	0.1.0	10.52 KB	March 31, 20...	December 1...

Below the table is a description: 'This chart is for the docs!'. There are two labels: 'Documentation' and 'Helm Chart'. A 'Fetch Command' section shows a terminal command: `$ helm fetch https://helm.ngc.nvidia.com/nvstaging/ngc_product/charts/demo_chart-0.1.0.tgz --username='$oauthToken' --password='<YOUR API KEY>'`. At the bottom, there are tabs for 'Overview', 'File Browser', and 'Related Collections'. The 'Overview' tab is selected, and the title 'NGC Docs Chart' is displayed below it.

2. Click Delete to remove the Helm chart.

Private Registry > Entity Creation > Edit Helm Chart: NGC Docs Test

### Edit Helm Chart: NGC Docs Test

[Delete](#) [Cancel](#) [Save](#)

**Basic Information**  
Please complete this section to describe your helm chart.

**Name\***  
demo\_chart

**Publisher**  
NVIDIA

**Display Name\***  
NGC Docs Test

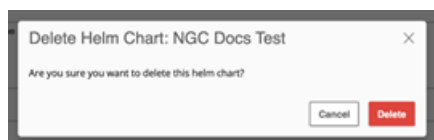
**Description\***  
This chart is for the docs!

**Logo**  
Enter a URL to upload a logo.

**Overview (optional)**  
Enter your helm chart overview in Markdown format.

# NGC Docs Chart

3. Click Delete at the confirmation dialog.



## 6.4. Manage Helm Charts Using the NGC CLI

### 6.4.1. Searching for Available Helm Charts in an Org

The NGC CLI supports wildcard searches, using standard Unix shell-style wildcards. For example, to see a list of all available Chart packages in your org, run the following command.

```
$ ngc registry chart list *<org_name>*
```

Example:

```
$ ngc registry chart list *nvidian*
```

That will return all charts with 'nvidian' anywhere in the name.

```

+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+
| Name          | Repository      | Version | Size      | Created By      |
| Description   | Created Date   | Last Modified |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
| fluentd-elasti| nvidian/fluen  | 4.8.1   | 245.61 KB | stg-3emmf14t83v | Changed
| sho          | Nov 15, 2019  | Dec 17, 2019 |
| csearch      | ntd-elasticse |         |           | d0s5v81qasfi479 | rt
| descript     |                |         |           | |
|              | arch          |         |           |
|              |               |         |           |
| clara        | nvidian/repo1 | 0.0.1   | 65.66 KB  | stg-p6urlvepnjb |
| Feb 07, 2020 | Mar 12, 2021  |         |           |
|              | /clara        |         |           | q06qfis2815m6a4 |
|              |               |         |           |

```

## 6.4.2. Fetching Helm Charts

To download (or "pull") a Chart package, run the following command. Note: if no version is specified, the most recent version will be pulled.

```
$ ngc registry chart pull org/[team/]chart[:version]
```

Example:

```
$ ngc registry chart pull nvidian/nginx-ingress:1.2.3 (pulls version 1.2.3)
```

```
$ ngc registry chart pull nvidian/nginx-ingress (pulls the latest version)
```

## 6.4.3. Adding Helm Charts to a Private Registry



Note: Make sure you have the right permissions to create Helm Charts in your organization and/or team. You need to have the user role "Registry User" or "Registry Admin". For details, refer to [NGC Registry User Roles](#).

### Creating a Chart

Before a chart can be uploaded to your organization's registry, you must first create a record containing the basic information about the chart. There are several values you can specify (issue `ngc registry chart create --help` to view all of them), but you must at least provide a short description of the chart.

```
$ ngc registry chart create <org>/[<team>/]<chart_name> --short-desc <description>
```

Example:

```
$ ngc registry chart create nvidian/ngcdocstest --short-desc "Doc testing chart"
```

```
Successfully created chart 'nvidian/ngcdocstest'.
```

```

-----
Chart Information
Name: ngcdocstest
Short Description: Doc testing chart
Display Name:
Team:
Publisher:
Built By:
Labels:

```



```

Logo:
Created Date: 2021-03-22 18:48:36 UTC
Updated Date: 2021-03-22 18:48:36 UTC
Read Only: False Latest Version ID:
Latest Version Size (bytes):
Overview:
-----

```

## Updating a Chart

You can update the metadata about a chart after it has been created with the `update` command.

```
$ ngc registry chart update <org>/[<team>/]<chart_name> --<property> <value>
```

### Example:

```
$ ngc registry chart update nvidian/ngcdocstest --publisher "test account" --
display-name "Helm Demo Chart" --built-by "my team"
```

```
Successfully updated chart 'nvidian/ngcdocstest'.
-----
```

```

Chart Information
Name: ngcdocstest
Short Description: Doc testing chart
Display Name: Helm Demo Chart
Team:
Publisher: test account
Built By: my team
Labels:
Logo:
Created Date: 2021-03-22 18:48:36 UTC
Updated Date: 2021-03-22 18:52:01 UTC
Read Only: False
Latest Version ID: 0.1.0
Latest Version Size (bytes): 10664
Overview:
-----

```

## 6.4.4. Getting Information about a Helm Chart

You can see the information about a chart at any time by running the `info` command:

### Example:

```
$ ngc registry chart info nvidia/ngcdocstest
```

```

-----
Chart Information
Name: ngcdocstest
Short Description: Doc testing chart
Display Name: Helm Demo Chart
Team: Publisher: test account
Built By: my team
Labels:
Logo:
Created Date: 2021-03-22 18:48:36 UTC
Updated Date: 2021-03-22 18:54:44 UTC
Read Only: False
Latest Version ID: 0.1.0
Latest Version Size (bytes): 10664
Overview: -----

```

## 6.4.5. Pushing a Helm Chart

To push (upload) a Helm chart to your org space, issue the following.

```
$ ngc registry chart push <org>/[<team>/]<chart_name>:<version>
```

Example:

```
$ ngc registry chart push nvidian/ngcdocstest:0.1.0

Successfully pushed chart version 'ngcdocstest:0.1.0'.
-----
Chart Version Information
Created Date: 2021-03-22 18:54:44 UTC
Updated Date: 2021-03-22 18:54:44 UTC
Version ID: 0.1.0
Total File Count: 11
Total Size: 10.41 KB
Status: UPLOAD_COMPLETE
-----
```

## 6.4.6. Listing Helm Chart Versions

To see a list of all available versions for a chart, specify the chart name, and use the wildcard '\*' for the version.

Example:

```
$ ngc registry chart list nvidian/nginx-ingress:*
+-----+-----+-----+-----+
| Version | File Count | File Size | Created Date |
+-----+-----+-----+-----+
| 0.8.0   | 27         | 181.94 KB | Mar 12, 2021 |
| 1.0.0   | 25         | 149.51 KB | Oct 02, 2020 |
| 0.0.6   | 25         | 149.51 KB | Oct 02, 2020 |
| 0.0.5   | 25         | 149.51 KB | Oct 02, 2020 |
| 0.6.0   | 25         | 149.51 KB | Sep 17, 2020 |
| 0.6.1   | 25         | 149.51 KB | Sep 17, 2020 |
| 1.26.2  | 68         | 109.19 KB | Feb 08, 2020 |
+-----+-----+-----+-----+
```

## 6.4.7. Removing Helm Charts from a Private Registry



Note: Make sure you have the right permissions to create Helm Charts in your organization and/or team. You need to have the user role “Registry Admin”. For details, refer to [NGC Registry User Roles](#).

If you are an admin, you can delete a specific version of a chart running the following command:

```
$ ngc registry chart remove <org>/[<team>/]<chart_name>:<version>
```

The following example removes just version 0.1.0:

```
$ ngc registry chart remove nvidian/ngcdocstest:0.1.0
```

The following example removes all versions and data about the chart:

```
$ ngc registry chart remove nvidian/ngcdocstest
```

If you do not specify a version, every version of the chart, as well as the chart metadata, will be deleted.

Example:

```
$ ngc registry chart remove nvidia/ngcdocstest
Are you sure you would like to remove nvidia/ngcdocstest? [y/n]y
Successfully removed chart version 'nvidia/ngcdocstest:0.1.0'.
Successfully removed chart 'nvidia/ngcdocstest'.
```

## 6.5. Manage Helm Charts Using the NGC API

### 6.5.1. Updating Information on the Helm Chart Page

The NGC API lets you specify information about your Helm chart. Use the NGC API Explorer page (URL: <https://docs.ngc.nvidia.com/models/index.html#!/Artifacts/updateArtifactInOrgUsingPATCH>). You can use the page to build the JSON file for use in a CURL command.

The following page elements can be edited.

Page Element	JSON Field	Description
Helm Chart name	displayName	The name of the Helm chart appearing in the title on the tile and Helm chart page
Publisher	publisher	The organization/entity responsible for creating the asset
Logo	logo	URL of the image to use as the logo for the asset
Description	shortDescription	A short description for the Helm chart
Labels	labels	Tags to enhance search results
Overview tab	description	Content of the “Overview” tab which can provide publishers to convey additional

The JSON column shows the corresponding JSON fields to use when updating the page using the NGC API.

The following shows the relevant fields in the JSON file.

```
{ "attributes": [
  {
    "key": "string",
    "value": "string"
  }
],
"builtBy": "string",
"description": "string",
"displayName": "string",
```

```
"labels": [
  "string" ],
"logo": "string",
"publisher": "string",
"shortDescription": "string"
}
```

## Example

The following shows example JSON values.

```
{ "builtBy": "NVIDIA",
  "description": "#NGC Docs Chart",
  "displayName": "NGC DOCS CHART TEST",
  "labels": [
    "Helm Chart", "Documentation"
  ],
  "shortDescription": "This charts is for the docs!"
}
```

The following is an example CURL command.

```
curl -X PATCH --header 'Content-Type: application/json' --header 'Accept:
application/json' --header 'Authorization: Bearer <<BEARER_TOKEN>>' -d '{ "builtBy":
"NVIDIA", "description": "#Le Chart", "displayName": "NGC DOCS TEST", "labels":
[ "Helm Chart", "Documentation" ], "shortDescription": "This chart is for the
docs&#33;" }' 'https://api.ngc.nvidia.com/v2/org/nvidian/helm-charts/ngcdocstest'
```

## 6.5.2. Deleting Helm Charts Using the NGC API

To remove Helm charts from your org or team, you must use the NGC API. Refer to <https://docs.ngc.nvidia.com/api/index.html#!/Model/proxyDeleteUsingDELETE> for a description of the relevant API.

To delete a Helm chart from an org space, issue the following:

```
$ curl -X DELETE --header 'Accept: application/json' --header 'Authorization: Bearer <Bearer
Token>' 'https://api.ngc.nvidia.com/v2/org/<org-name>/helm-charts/<chart-name>'
```

To delete a Helm chart from a Team space, issue the following:

```
$ curl -X DELETE --header 'Accept: application/json' --header 'Authorization: Bearer <Bearer
Token>' 'https://api.ngc.nvidia.com/v2/org/<org-name>/team/<team-name>/helm-charts/<chart-
name>'
```

## 6.6. Manage Helm Charts Using the Helm CLI

### 6.6.1. Setting Up an NGC Helm Repository

1. Obtain an NGC API Key.

See [Generating Your NGC API Key](#) for instructions.

2. Export the API Key for use in commands.

```
$ export NGC_API_KEY=<your-api-key>
```

3. Add the NGC org to your Helm repository.

```
$ helm repo add <repo-name> https://helm.ngc.nvidia.com/<org-name> --username=\  
$oauthtoken --password=$NGC_API_KEY
```

Where `<repo-name>` is a name of your choosing by which you will reference the repository.

## 6.6.2. Searching for Available Helm Charts

To view a list of available Chart packages in your org, issue the following.

```
$ helm search <repo-name>
```

## 6.6.3. Fetching Helm Charts

To download (or "fetch") a Helm chart package from the repo, issue the following.

```
$ helm fetch <repo-name>/<chart-name>
```

## 6.6.4. Adding Helm Charts to a Private NGC Org/ Team

These instructions assume the Helm push plugin is installed. To install the plugin, issue the following.

```
$ helm plugin install https://github.com/chartmuseum/helm-push
```

To push (upload) a Helm chart to your org space, issue the following.

```
$ helm push <chart-name>.tgz <repo-name>
```

## 6.6.5. Removing Helm Charts from a Private NGC Org/Team

To remove Helm charts from your org or team, you must use the NGC CLI or NGC API.

---

## Chapter 7. Getting Support for NGC container registry

For additional information on using the NGC container registry and for getting help if you encounter issues, send an email to [enterprisesupport@nvidia.com](mailto:enterprisesupport@nvidia.com) with a description of your issue and a ticket will be created for you.

---

# Chapter 8. Private Registry Size Limits

To maintain optimal performance and service quality, the following size limits are applied to the Private Registry:

Description	Limit	Note
Single image layer size	10 GB	Size limit per layer for Docker images (recommended).
Total image size	1 TB	Size limit for all Docker images stored in the registry (recommended).
Total model/resource size	5 TB	Size limit for all models or resources stored in the registry (enforced).





## Trademarks

NVIDIA and the NVIDIA logo are trademarks and/or registered trademarks of NVIDIA Corporation in the United States and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

## Copyright

© 2020-2024 NVIDIA CORPORATION & AFFILIATES. All rights reserved.

