



# NGC REGISTRY CLI

DU-09433-001 \_v04 | April 2020

## User Guide



# TABLE OF CONTENTS

<b>Chapter 1. Introduction.....</b>	<b>1</b>
<b>Chapter 2. Installing NGC Registry CLI.....</b>	<b>2</b>
<b>Chapter 3. Getting Started with NGC CLI.....</b>	<b>3</b>
3.1. Obtaining Your API Key.....	3
3.2. User Role Capabilities.....	4
3.3. Specifying Your Org or Team When Running Commands.....	4
3.4. Getting Help Using NGC CLI.....	5
<b>Chapter 4. Running the Diagnostics.....</b>	<b>6</b>
<b>Chapter 5. Using the Container Registry.....</b>	<b>7</b>
5.1. Viewing Container Image Information.....	7
5.2. Pulling a Container Image.....	8
5.3. Pushing a Container Image.....	8
5.4. Removing a Container Image.....	8
<b>Chapter 6. Using the Model Registry.....</b>	<b>9</b>
6.1. Models.....	9
6.1.1. Creating and Uploading a Model.....	9
6.1.1.1. Before you begin.....	9
6.1.1.2. Creating a Model.....	9
6.1.1.3. Uploading a Model Version.....	10
6.1.1.4. Uploading Custom Metrics Tables.....	10
6.1.2. Sharing Models.....	11
6.1.2.1. Sharing a Model Outside of Your Org or Team.....	11
6.1.3. Viewing Model Information.....	12
6.1.4. Updating Model Information.....	12
6.1.5. Downloading a Model.....	13
6.1.6. Deleting a Model.....	13
6.2. Model-scripts.....	14
6.2.1. Creating a Model-script.....	14
6.2.2. Uploading a Model-script Version.....	14
6.2.3. Viewing Model-script Information.....	15
6.2.4. Updating a Model-script or Model-script Version.....	15
6.2.5. Downloading a Model-script.....	16
6.2.6. Deleting a Model-script.....	16
<b>Chapter 7. Managing Users and Teams.....</b>	<b>18</b>
7.1. Inviting users to the organization’s NGC account.....	18
7.2. Creating teams.....	18
7.3. Adding users to teams.....	19
7.4. Creating a team and adding a user in the same command.....	19

# Chapter 1.

## INTRODUCTION

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

With NGC Registry CLI, you can

- ▶ See a list of GPU-accelerated Docker containers available to you as well as detailed information about each container image.
- ▶ See a list of deep-learning models and model-scripts as well as detailed information about them.
- ▶ Download container images, models, and model-scripts.
- ▶ Upload container images, models, and model-scripts
- ▶ Admins can create and manage users and teams

If you will be pushing or pulling containers to or from the registry, you will need Docker installed on your local machine. To install Docker on your client machine, follow one of these instructions:

- ▶ For Ubuntu Linux:  
<https://docs.docker.com/engine/installation/linux/ubuntu/linux/>
- ▶ For Windows:  
<https://docs.docker.com/engine/installation/windows/#/docker-for-windows>
- ▶ For MacOS X:  
<https://docs.docker.com/engine/installation/mac/>

This document provides an introduction to using the NGC Registry CLI. For a complete list of commands and options, use the `-h` option as explained in the section [Getting Help Using NGC CLI](#).

# Chapter 2.

## INSTALLING NGC REGISTRY CLI

To install NGC Registry CLI,

1. Log in to your enterprise account on NGC website (<https://ngc.nvidia.com>).
2. In the top right corner, click your user account icon and select **Setup**, then click **Downloads** under **Install NGC 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 Registry CLI x.y.z**” where x.y.z indicates the version.

# Chapter 3.

## GETTING STARTED WITH NGC CLI

Now that you have installed NGC CLI, you can begin using it as follows:

### 3.1. Obtaining Your API Key

The API Key is the mechanism used to authenticate your access to the NGC Registry. When you generate an API Key, it authenticates your account so that you can use NGC from the command line.

Your API key also determines the organization and teams of which you are a member, as well as your user role. If you did not enter an API key when you configured the CLI, you will need to do so now.

To get your API key,

1. Log in to your enterprise account on the NGC website <https://ngc.nvidia.com>.
2. Click **Setup** from the side menu, then click **API Key** from the Setup page.
3. On the API Key page, click **Generate API Key**.
4. In response to the warning that your old API Key will become invalid, click **CONTINUE** to generate the key.

Your API key is displayed with examples of how to use it.

5. Click **Continue** to generate the key.

Your API key appears.

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

You only need to generate an API Key once, so save the key and store it in a secure place. Should you lose your API Key, you can generate a new one from the NGC website. As soon as you generate a new API Key the old one is invalidated.

Run `ngc config set` and then enter your API key, org, and team at the prompts.

## 3.2. User Role Capabilities

To determine your user role, run `ngc user who`.

The features available to you depend on your user role.

Capability	REGISTRY_WRITE_TEAM_ADMIN_ROLE	REGISTRY_WRITE_ADMIN_ROLE	REGISTRY_WRITE_USER_ROLE
Add teams		X	
Add new users		X	
Add users to teams	X	X	
Delete images	X	X	X
View all image information	X	X	X
View all model information	X	X	X
View all model-script information	X	X	X
Create and upload models	X	X	X
Download models	X	X	X
Download model-scripts	X	X	X
Delete models	X	X	X (if model creator)

## 3.3. Specifying Your Org or Team When Running Commands

When you configured NGC CLI for the first time, you selected the default organization to associate with your username. If you will be using the resources associated with your organization, then no other action is necessary.

However, to use resources and registry spaces associated with other organizations or teams of which you are a member, you must change your configuration accordingly.

To associate the session with a different org or team, run the `ngc config set` command and then select a different org or team at the prompts.

You can also specify an org or team when you enter a command. For example, if your current context is `org_A` and `team_A`, you can specify a command to use a different org or team of which you are a member using the `--org` and `--team` options.

For example, to list the images available in `team_B`, enter the following:

```
c:/>ngc registry image list --team team_B
```

## 3.4. Getting Help Using NGC CLI

To run an NGC CLI command, enter “**ngc**” followed by the appropriate options.

To see a description of available options and command descriptions, use the option **-h** after any command or option.

**Example 1:** To view a list of all the available options for **ngc**, enter

```
C:\>ngc -h
```

**Example 2:** To view a description of the registry image command and options, enter

```
C:\>ngc registry image -h
```

**Example 2:** To view a description of the registry image info command and options, enter

```
C:\>ngc registry image info -h
```

# Chapter 4.

## RUNNING THE DIAGNOSTICS

You can view the following diagnostic information for the NGC CLI user:

- ▶ Current time
- ▶ Operating system
- ▶ Disk usage
- ▶ Current directory size
- ▶ Memory usage
- ▶ NGC CLI environment variables set
- ▶ API gateway connectivity
- ▶ API connectivity to the container registry
- ▶ API connectivity to the model registry
- ▶ External IP
- ▶ NGC CLI configuration values
- ▶ User information (ID, name, and email)
- ▶ User org roles
- ▶ User team roles

### Syntax

```
ngc diag [all,client,install,server,user]
```

where

#### **all**

Produces the maximum amount of diagnostic output

#### **client**

Produces diagnostic output only for the client machine.

#### **install**

Produces diagnostic output only for the local installation.

#### **server**

Produces diagnostic output only for the remote server.

#### **user**

Produces diagnostic output only for the user configuration.



# Chapter 5.

## USING THE CONTAINER REGISTRY

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

### 5.1. 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 | 2.37 GB   | Oct 18, | unlocked
|              |            | tu1804-ompi |           | 2019   |
|              |            | 4-mkl       |           |        |
| CANDLE        | hpc/candle | 20180326    | 1.52 GB   | Oct 18, | unlocked
|              |            |             |           | 2019   |
| ...          |            |             |           |        |
+-----+-----+-----+-----+-----+-----+
```

"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
```

## 5.2. 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.

## 5.3. 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.

## 5.4. 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>
```

# Chapter 6.

## USING THE MODEL REGISTRY

### 6.1. Models

The `ngc registry model` commands let you access ready-to-use deep learning models from the registry.

#### 6.1.1. Creating and Uploading a Model

##### 6.1.1.1. Before you begin

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

```
C:\>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.

```
C:\>ngc config set [--org <new org>][--team <new team>]
```

Or

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

##### 6.1.1.2. Creating a Model

###### Syntax

To see the syntax and options for creating a model, issue the following.

```
ngc registry model create -h
```

The output describes the required and optional arguments.

###### Example

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

```
$ ngc registry model create \
  --application OBJECT_DETECTION \
  --format "format" --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
```

### 6.1.1.3. Uploading a Model Version

#### Syntax

To see the syntax and options for uploading a model, issue the following.

```
ngc registry model upload-version -h
```

The output describes the required and optional arguments.

#### Example

The following example uploads a model using all required and optional arguments to create model version 1.

```
$ ngc registry model upload-version \
  --accuracy-reached FP32 \
  --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" \
  --source path/to/my/model/version/dir \
  nvidia/myteam/final_review_model:1
```

### 6.1.1.4. Uploading Custom Metrics Tables

You can also now 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. To add the table to the upload, use the **--metrics-file** argument.

#### 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"}
  ]
}

```

### Example Showing Model-version Upload Incorporating Custom Metrics Tables

```

$ ngc registry model upload-version \
  --accuracy-reached FP32 \
  --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" \
  --metrics-file zeppelin_table.json \
  --metrics-file rhcp_table.json \
  --source path/to/my/model/version/dir \
  nvidia/myteam/final_review_model:1

```

## 6.1.2. Sharing Models

When you upload a model, you specify the org or team registry space where you want it uploaded. The model is available to all members of the specified org or team.

### 6.1.2.1. Sharing a Model Outside of Your Org or Team

Models cannot be shared directly with another org or team. To make a model available to another org or team, you must be a member of that org or team and then upload the model to the corresponding registry space.

The following is an example showing the general process for sharing a model from team\_A with team\_B.

1. Download the model.

```

C:\>ngc registry model download-version --dest ./models nvidia/team_a/
cmr_gnmt:1

```

This example downloads the model `cmr_gnmt` version 1 to the `/models` folder on the local disk.

2. Create the model.

This step is needed only if the model does not already exist in the team\_B space.

```
C:\>ngc registry model create nvidia/team_b/cmr_gnmt
```

### 3. Upload the model version.

```
C:\>ngc registry model upload-version --source ./models nvidia/team_b/cmr_gnmt:1
```

This example uploads the model `cmr_gnmt` version 1 from the `/models` folder on your local disk.

Uploaded and shared models in one org or team are independent of the identical model in another org or team. So changes made in a model are not automatically propagated to other orgs or teams.

## 6.1.3. Viewing Model Information

There are several commands for viewing information about available models.

**To see a list of models available to you from NVIDIA, as well as any that are shared with your org and team:**

```
C:\>ngc registry model list
```

**To view detailed information about a model, you can specify the model**

```
C:\>ngc registry model info <org>/[<team>/]<model_name>
```

**or the model version.**

```
C:\>ngc registry model info <org>/[<team>/]<model_name>:<version>
```

## 6.1.4. Updating Model Information

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

### Syntax

To see the syntax and options for updating a model version, issue the following.

```
ngc registry model update -h
```

The output describes the required and optional arguments.

### Example updating a model's overview file

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

### Example updating a model-version's accuracy reached and memory footprint

```
$ ngc registry model update \
  --accuracy-reached FP16 \
  --memory-footprint 16GB \
  nvidia/myteam/final_review_model:1
```

## 6.1.5. Downloading a Model

To download a model from the registry to your local disk, specify the model name and version.

```
C:\>ngc registry model download-version <org>/[<team>/]<model-name:version>
```

**Example:** Downloading an NVIDIA-provided model to the current directory.

```
C:\>ngc registry model download-version nvidia/model/
trt_onnx_vgg16_v100_16g_int8:1
```

The following is an example showing the output confirming completion of the download:

```
Downloaded 230.92 MB in 38s, Download speed: 6.07 MB/s
-----
Transfer id: trt_onnx_vgg16_v100_16g_int8_v1 Download status: Completed.
Downloaded local path: C:\trt_onnx_vgg16_v100_16g_int8_v1
Total files downloaded: 3
Total downloaded size: 230.92 MB
Started at: 2019-03-18 14:09:31.664000
Completed at: 2019-03-18 14:10:09.712000
Duration taken: 38s seconds
-----
```

The model is downloaded to a folder that corresponds to the model name in the current directory. You can specify another path using the `-d` option.

**Example:** Downloading an NVIDIA-provided model to a specific directory (`/models`).

```
C:\>ngc registry model download-version nvidia/model/
trt_onnx_vgg16_v100_16g_int8:1 -d ./models
```

```
Downloaded 230.92 MB in 38s, Download speed: 6.07 MB/s
-----
Transfer id: trt_onnx_vgg16_v100_16g_int8_v1 Download status: Completed.
Downloaded local path: C:\models\trt_onnx_vgg16_v100_16g_int8_v1
Total files downloaded: 3
Total downloaded size: 230.92 MB
Started at: 2019-03-18 14:09:31.664000
Completed at: 2019-03-18 14:10:09.712000
Duration taken: 38s seconds
-----
```

## 6.1.6. Deleting a Model

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

Be sure the context is set appropriately for the model 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 model, including all versions of the model, enter the following.**

```
C:\>ngc registry model remove <org>/[<team>/]<model>
```

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

```
C:\>ngc registry model remove <org>/[<team>/]<model:version>
```

## 6.2. Model-scripts

The `ngc registry model-script` commands let you access ready-to-use deep learning model-scripts from the registry.

### 6.2.1. Creating a Model-script

#### Syntax

To see the syntax and options for creating a model script, issue the following.

```
ngc registry model-script create -h
```

The output describes the required and optional arguments.

#### Example

The following example creates a model-script with all required and optional arguments used:

```
$ ngc registry model-script create \
  --application CLASSIFICATION \
  --format FORMAT \
  --framework TensorFlow \
  --precision FP16 \
  --short-desc "A short description of this model-script." \
  --advanced-filename path/to/my/advanced/guide.md \
  --built-by "My Name" \
  --display-name "Model Script for Model Final Review Model" \
  --label "a" --label "b" --label "c" \
  --logo "www.example.com/mylogo.jpeg" \
  --overview-file path/to/overview/file.md \
  --performance-filename path/to/perf/chart.txt \
  --public-dataset-license LicenseType \
  --public-dataset-link "www.example.com/my/dataset" \
  --public-dataset-name "My Dataset" \
  --publisher NVIDIA \
  --quick-start-guide-filename path/to/quick/start.md \
  --setup-filename path/to/setup/instructions.md \
  nvidia/myteam/final_review_model_script
```

### 6.2.2. Uploading a Model-script Version

#### Syntax

To see the syntax and options for uploading a model script version, issue the following.

```
ngc registry model-script upload -h
```

The output describes the required and optional arguments.

#### Example



The following example uploads a model-script version with all required and optional arguments used:

```
$ ngc registry model-script upload-version \
  --accuracy-reached FP16 \
  --batch-size 2000 \
  --gpu-model "GPU MODEL" \
  --memory-footprint 16GB" \
  --num-epochs 100 \
  --desc "A full description of the target version" \
  --performance-filename path/to/perf/chart.txt \
  --quick-start-guide-filename path/to/quick/start.md \
  --release-notes-filename path/to/release/notes.md \
  --setup-filename path/to/setup/file.md \
  --source path/to/model/script/version/dir \
  nvidia/myteam/final_review_model_script:v1
```

### 6.2.3. Viewing Model-script Information

There are several commands for viewing information about available model-scripts.

**To see a list of model-scripts available to you from NVIDIA, as well as any that are shared with your org and team:**

```
C:\>ngc registry model-script list
```

**To view detailed information about a model-script, you can specify the model-script**

```
C:\>ngc registry model-script info <org>/[<team>/]<model_script_name>
```

**or the model-script version.**

```
C:\>ngc registry model-script info <org>/[<team>/]<model_script_name>:<version>
```

### 6.2.4. Updating a Model-script or Model-script Version

#### Syntax

To see the syntax and options for updating a model script version, issue the following.

```
ngc registry model-script update -h
```

The output describes the required and optional arguments.

**Example updating a model-script display name and logo**

```
$ ngc registry model-script update \
  --display-name "Awesome Model-Script Mark 1" \
  --link "www.example.com/corrected/url.png" \
  nvidia/myteam/final_review_model_script
```

**Example updating a model-script version's performance file**

```
$ ngc registry model-script update \
  --performance-filename path/to/new/perf/chart.txt \
  nvidia/myteam/final_review_model_script:v1
```

## 6.2.5. Downloading a Model-script

To download a model-script from the registry to your local disk, specify the model-script name and version.

```
C:\>ngc registry model-script download-version <org>/[<team>]/<model-script-name:version>
```

**Example:** Downloading an NVIDIA-provided model to the current directory.

```
C:\>ngc registry model-script download-version nvidia/model-script/resnet50_v1_5:1
```

The following is an example showing the output confirming completion of the download:

```
-----
Downloaded 58.64 KB in 1s, Download speed: 58.64 KB/s
-----
Transfer id: resnet50_v1_5_v1 Download status: Completed.
Downloaded local path: C:\resnet50_v1_5_v1\
Total files downloaded: 24
Total downloaded size: 58.64 KB
Started at: 2019-03-14 15:44:25.364000
Completed at: 2019-03-14 15:44:26.381000
Duration taken: 1s seconds
-----
```

The model is downloaded to a folder that corresponds to the model name in the current directory. You can specify another path using the `-d` option.

**Example:** Downloading an NVIDIA-provided model-script to a specific directory (/model-scripts).

```
C:\>ngc registry model download-version nvidia/model-script/resnet50_v1_5:1 --dest ./model-scripts
```

```
-----
Downloaded 58.64 KB in 1s, Download speed: 58.64 KB/s
-----
Transfer id: resnet50_v1_5_v1 Download status: Completed.
Downloaded local path: C:\model-scripts\resnet50_v1_5_v1
Total files downloaded: 24
Total downloaded size: 58.64 KB
Started at: 2019-03-14 15:44:25.364000
Completed at: 2019-03-14 15:44:26.381000
Duration taken: 1s seconds
-----
```

## 6.2.6. Deleting a Model-script

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

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

**To remove the model-script, including all versions of the model-script, enter the following.**

```
C:\>ngc registry model remove <org>/[<team>/]<model-script>
```

**To remove only a specific version of the model-script, enter the following.**

```
C:\>ngc registry model remove <org>/[<team>/]<model-script:version>
```

# Chapter 7.

## 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.

### 7.1. 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"
```

### 7.2. 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
-----
```

## 7.3. 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
```



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

## 7.4. 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
```



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

## Notice

THE INFORMATION IN THIS GUIDE AND ALL OTHER INFORMATION CONTAINED IN NVIDIA DOCUMENTATION REFERENCED IN THIS GUIDE IS PROVIDED “AS IS.” NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE INFORMATION FOR THE PRODUCT, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA’s aggregate and cumulative liability towards customer for the product described in this guide shall be limited in accordance with the NVIDIA terms and conditions of sale for the product.

THE NVIDIA PRODUCT DESCRIBED IN THIS GUIDE IS NOT FAULT TOLERANT AND IS NOT DESIGNED, MANUFACTURED OR INTENDED FOR USE IN CONNECTION WITH THE DESIGN, CONSTRUCTION, MAINTENANCE, AND/OR OPERATION OF ANY SYSTEM WHERE THE USE OR A FAILURE OF SUCH SYSTEM COULD RESULT IN A SITUATION THAT THREATENS THE SAFETY OF HUMAN LIFE OR SEVERE PHYSICAL HARM OR PROPERTY DAMAGE (INCLUDING, FOR EXAMPLE, USE IN CONNECTION WITH ANY NUCLEAR, AVIONICS, LIFE SUPPORT OR OTHER LIFE CRITICAL APPLICATION). NVIDIA EXPRESSLY DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR SUCH HIGH RISK USES. NVIDIA SHALL NOT BE LIABLE TO CUSTOMER OR ANY THIRD PARTY, IN WHOLE OR IN PART, FOR ANY CLAIMS OR DAMAGES ARISING FROM SUCH HIGH RISK USES.

NVIDIA makes no representation or warranty that the product described in this guide will be suitable for any specified use without further testing or modification. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer’s sole responsibility to ensure the product is suitable and fit for the application planned by customer and to do the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer’s product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this guide. NVIDIA does not accept any liability related to any default, damage, costs or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this guide, or (ii) customer product designs.

Other than the right for customer to use the information in this guide with the product, no other license, either expressed or implied, is hereby granted by NVIDIA under this guide. Reproduction of information in this guide is permissible only if reproduction is approved by NVIDIA in writing, is reproduced without alteration, and is accompanied by all associated conditions, limitations, and notices.

## Trademarks

NVIDIA, the NVIDIA logo, DGX, DGX-1, DGX-2, and DGX Station 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 NVIDIA Corporation. All rights reserved.