

## N-CAPIE Desktop Setup Guide

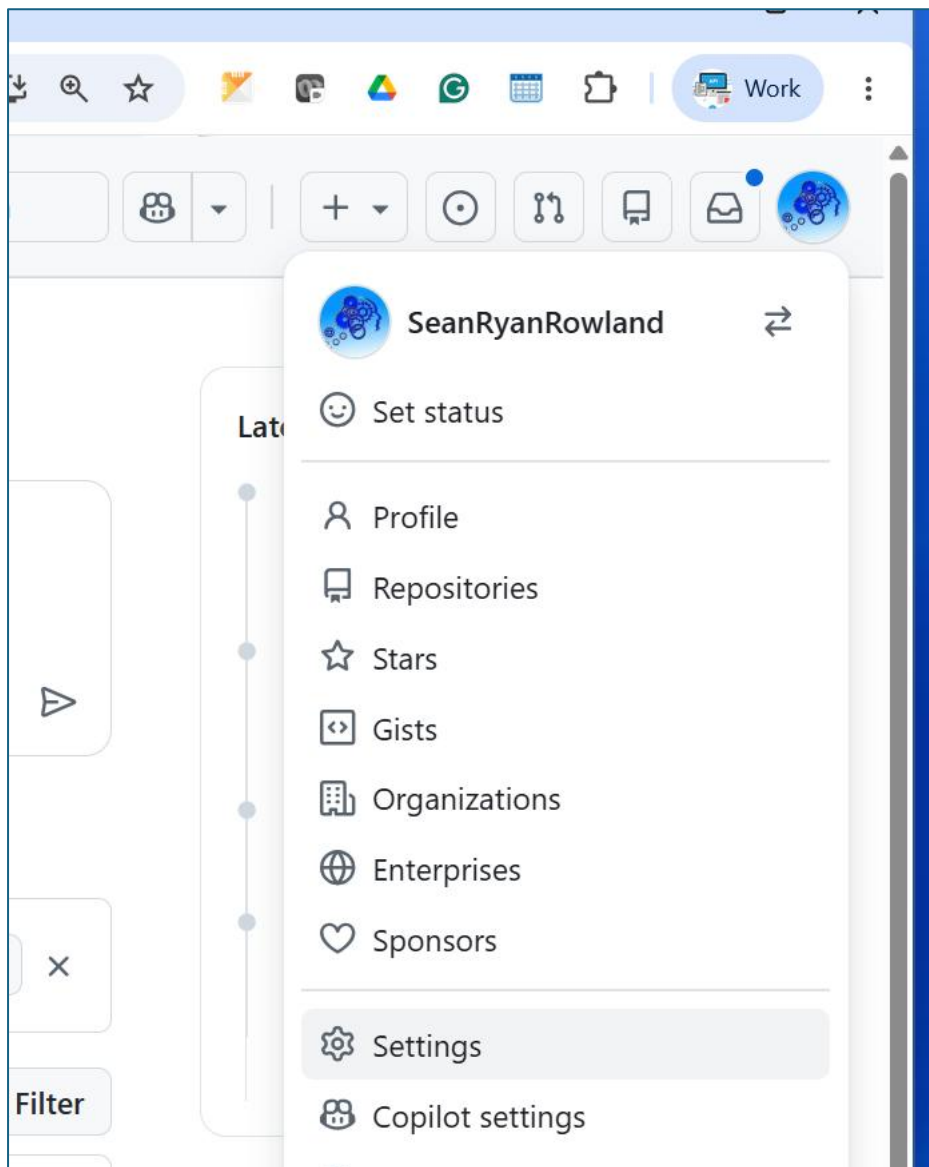
The local containerised version of N-CAPIE uses docker-compose to stand up a full environment to test locally.

To run this setup, users will need to have docker installed. In this guide, we will be using [docker desktop](#) to demonstrate setting up a new environment.

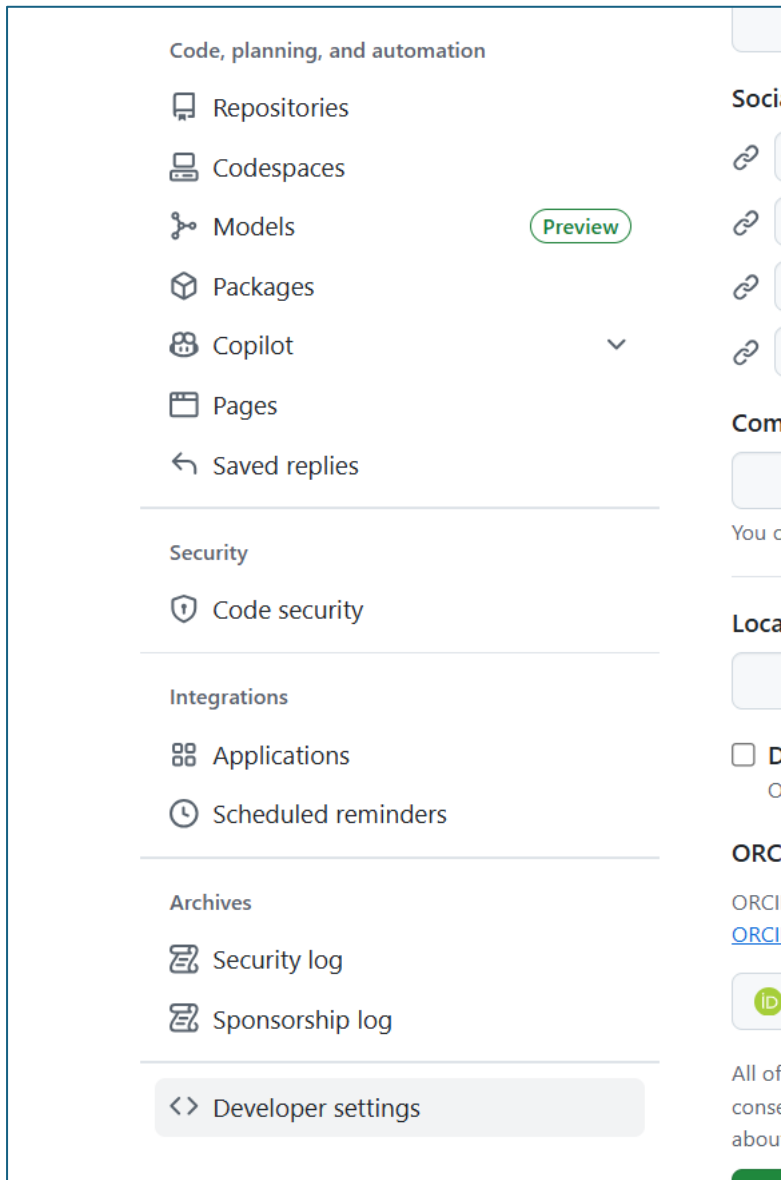
### GitHub Personal Access Token

Images are stored in Git Hub Container Repository. To pull these images during setup, users will need to create and provide a GitHub Personal Access Token.

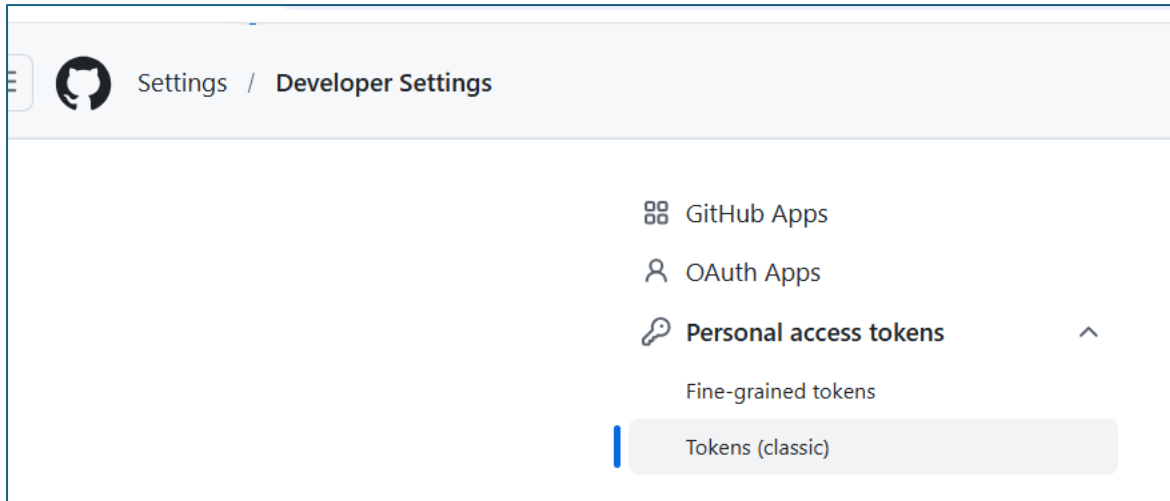
Login to GitHub, navigate to Settings in your user profile menu:



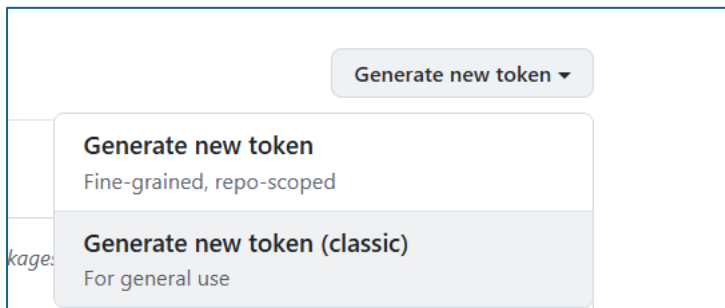
On the settings page, navigate to 'Developer Settings' at the bottom of the left-hand menu:



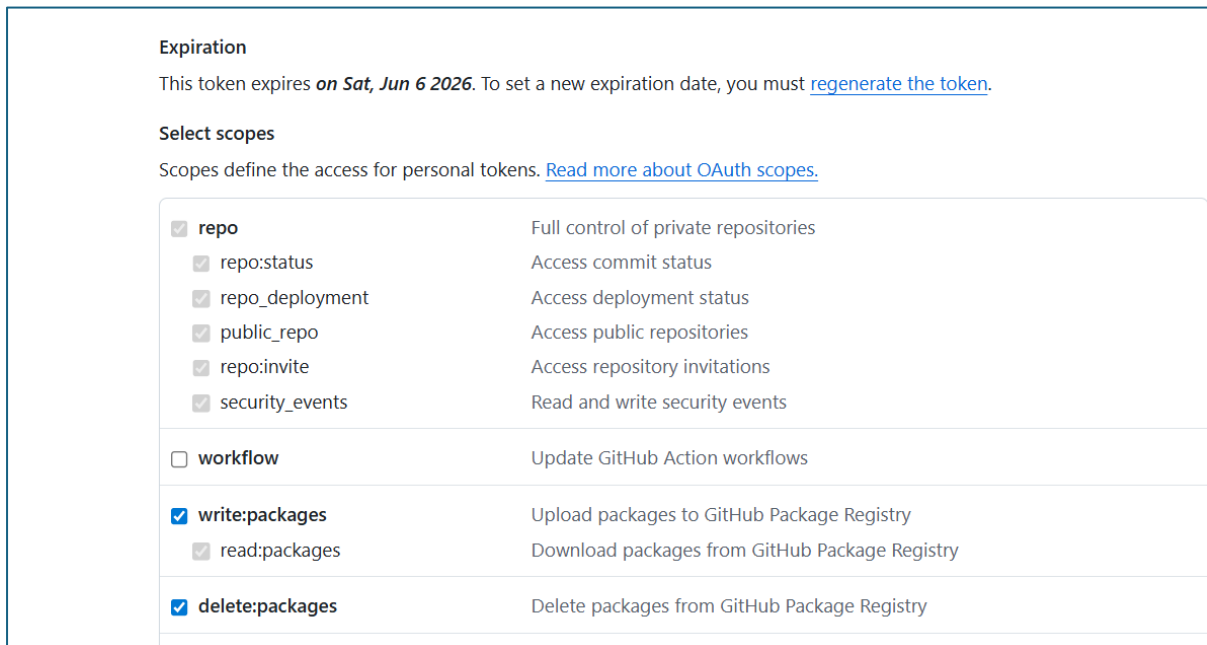
In the developer settings, select 'Tokens (Classic)' under 'Personal access tokens':



Select 'Generate new token (Classic)'



Ensure your token has read and write package permissions:

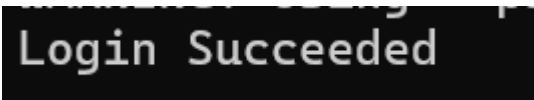


In a command window, use the following command to login to gchr.io, replacing credentials with your username and access token:

```
Administrator: Command Pro x + v
Microsoft Windows [Version 10.0.26100.32860]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator>docker login ghcr.io -u username -p accesstoken
```

You should receive a success message following the login:



In the command line interface, navigate to the directory containing the docker-compose file and issue the following command to bring up the environment:

```
'docker compose up -d'
```

This command will start the environment in detached mode so the environment will continue to run even if you close the command line terminal.

Following this command, you will see the required images being pulled. Once completed, the environment will be up and running on your system.

```
C:\Users\Administrator\Documents\docker-ncapie>docker compose up -d
[+] up 65/65
✔Image ghcr.io/ostiasoftware/resolutions/sandbox-ui:latest           Pulled      48.7s
✔Image mysql:8.0                                                     Pulled      66.4s
✔Image mongo                                                         Pulled      86.2s
✔Image ghcr.io/ostiasoftware/resolutions/sandboxondemand-app:1.0.2-snapshot Pulled      74.7s
✔Image redis:alpine                                                  Pulled      21.1s
✔Network docker-ncapie_default                                       Created     0.1s
✔Volume docker-ncapie_redis_data                                     Created     0.0s
✔Volume docker-ncapie_mongo_data                                    Created     0.0s
✔Volume docker-ncapie_mysql_data                                    Created     0.0s
✔Container mongo_db                                                 Healthy     28.2s
✔Container sandbox-ui                                               Started     3.7s
✔Container mysql_db                                                 Healthy     44.7s
✔Container redis_cache                                              Healthy     14.2s
✔Container sandboxondemand                                          Started     43.4s

What's next:
Filter, search, and stream logs from all your Compose services
in one place with Docker Desktop's Logs view. docker-desktop://dashboard/logs
```

In Docker Desktop you will now see the images and created containers:

Containers [Give feedback](#)

Container CPU usage **1.51% / 400%** (4 CPUs available)      Container memory usage **979.12MB / 1.87GB**

Search   Only show running containers

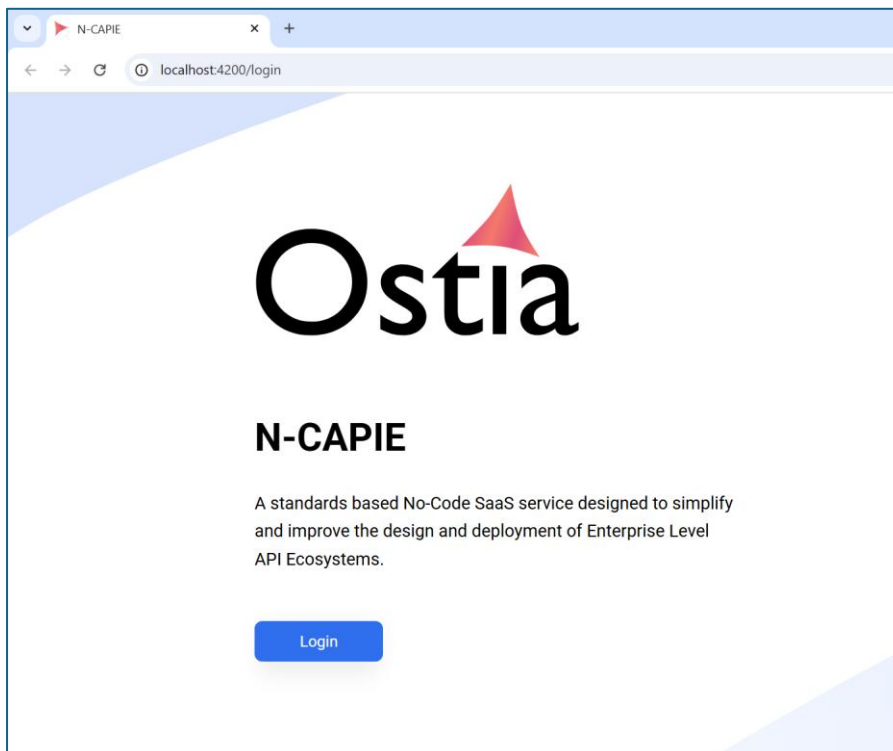
<input type="checkbox"/>	Name	Container ID	Image	Port(s)	CPU (%)	Memory usag...	Me
<input type="checkbox"/>	docker-ncapie	-	-	-	1.52%	979.12MB / 9.59i	
<input type="checkbox"/>	redis_cache	0807e4063a2f	<a href="#">redis:alpine</a>	<a href="#">16379:6379</a>	0.17%	13.61MB / 1.92G	
<input type="checkbox"/>	mongo_db	2d872b8acb14	<a href="#">mongo</a>	<a href="#">37017:27017</a>	0.53%	156.3MB / 1.92G	
<input type="checkbox"/>	mysql_db	86c7a4262b05	<a href="#">mysql:8.0</a>	<a href="#">13306:3306</a>	0.62%	152.6MB / 1.92G	
<input type="checkbox"/>	sandbox-ui	474ff455e1c9	<a href="#">ostiasoftwareolutions/sandbox-ui:latest</a>	<a href="#">5200:4200</a>	0%	45.91MB / 1.92G	
<input type="checkbox"/>	sandboxond	00bd47ffa640	<a href="#">ostiasoftwareolutions/sandboxondemand-app:1.0.2</a>	<a href="#">9090:9090</a>	0.2%	610.7MB / 1.92G	

From here, you can access the UI by selecting the link under ports for the sandbox-ui container:

<input type="checkbox"/>	●	sandbox-ui	2fb3fce06788	<a href="#">ostiasoftwareolutions/sandbox-ui:1.0.1</a>	<a href="#">4200:4200</a>
--------------------------	---	------------	--------------	--	---------------------------

or alternatively directly via a browser via the URL: <http://localhost:4200>

This will open the UI Login screen.



Once you have successfully logged in using the credentials provided, you will have full access to the N-CAPIE environment to begin testing.

We strongly recommend reading through the overviews and working through some of the tutorials provided to get a better understanding of concepts and features available.

## Managing the Environment

Following the initial setup, the environment can be stopped / Started and removed with the action buttons provided in the Docker Desktop interface.

<input type="checkbox"/>	<input type="checkbox"/>	Name	Container ID	Image	Port(s)	Actions
<input type="checkbox"/>	<input checked="" type="checkbox"/>	docker-ncapie	-	-	-	Start Stop
<input type="checkbox"/>	<input checked="" type="checkbox"/>	mongo_db	d45a73a2afd9	mongo	37017:27017	Start Stop
<input type="checkbox"/>	<input checked="" type="checkbox"/>	redis_cache	90328d2cafd1	redis:alpine	16379:6379	Start Stop
<input type="checkbox"/>	<input checked="" type="checkbox"/>	mysql_db	649197f523b6	mysql:8.0	13306:3306	Start Stop
<input type="checkbox"/>	<input checked="" type="checkbox"/>	sandbox-ui	2fb3fce06788	ostiasoftware/sandbox-ui:1.0.1	4200:4200	Start Stop
<input type="checkbox"/>	<input checked="" type="checkbox"/>	sandboxon-demand	bd27f4ca28b1	ostiasoftware/sandboxon-demand-app:1.0.2	9090:9090	Start Stop

Alternatively, you can use the command line to manage the container and images.

If you are not familiar with docker and docker-compose, we recommend the following resources:

[Getting Started with Docker Desktop](#)

[Docker Compose Overview](#)

[Docker Compose Commands](#)