Run TensorFlow Distributed Training on Plexus

Contents

Run TensorFlow Distributed Training on Plexus

Background

TensorFlow Distributed Strategy is a TensorFlow API to distribute training across multiple GPUs, multiple machines, or TPUs. Using this API, you can distribute your existing models and training code with minimal code changes.

For more detailed information, see **Distributed Learning with TensorFlow.**

Example

The following implementation example describes the steps to accomplish distributed training of cifar classifier on multiple GPUs.

Implementation

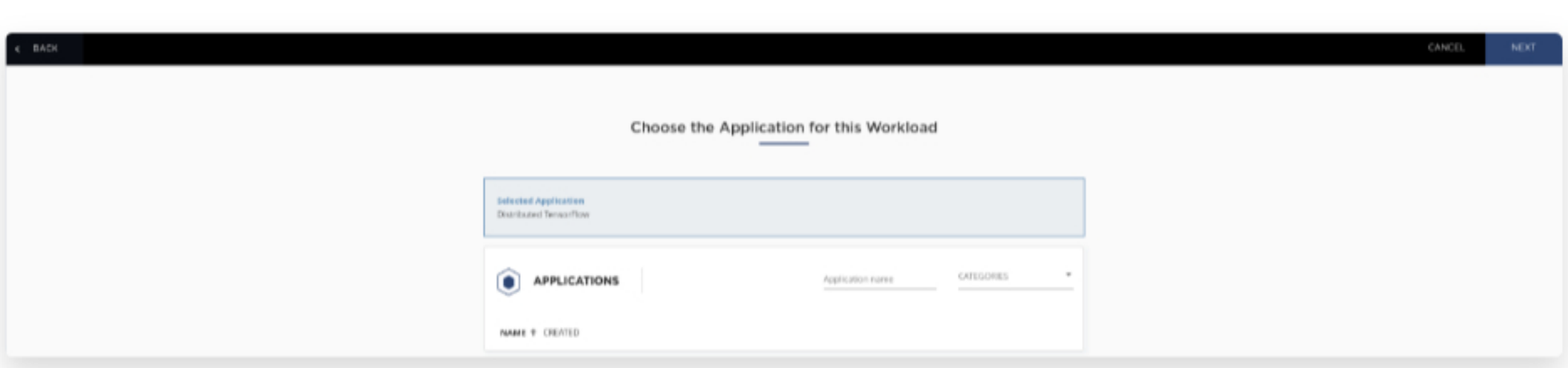
1. Sign in to Plexus, click **Applications**, and search for **Distributed TensorFlow**.

2. Select either **batch** or **interactive** mode using the switch near the top right of the screen.

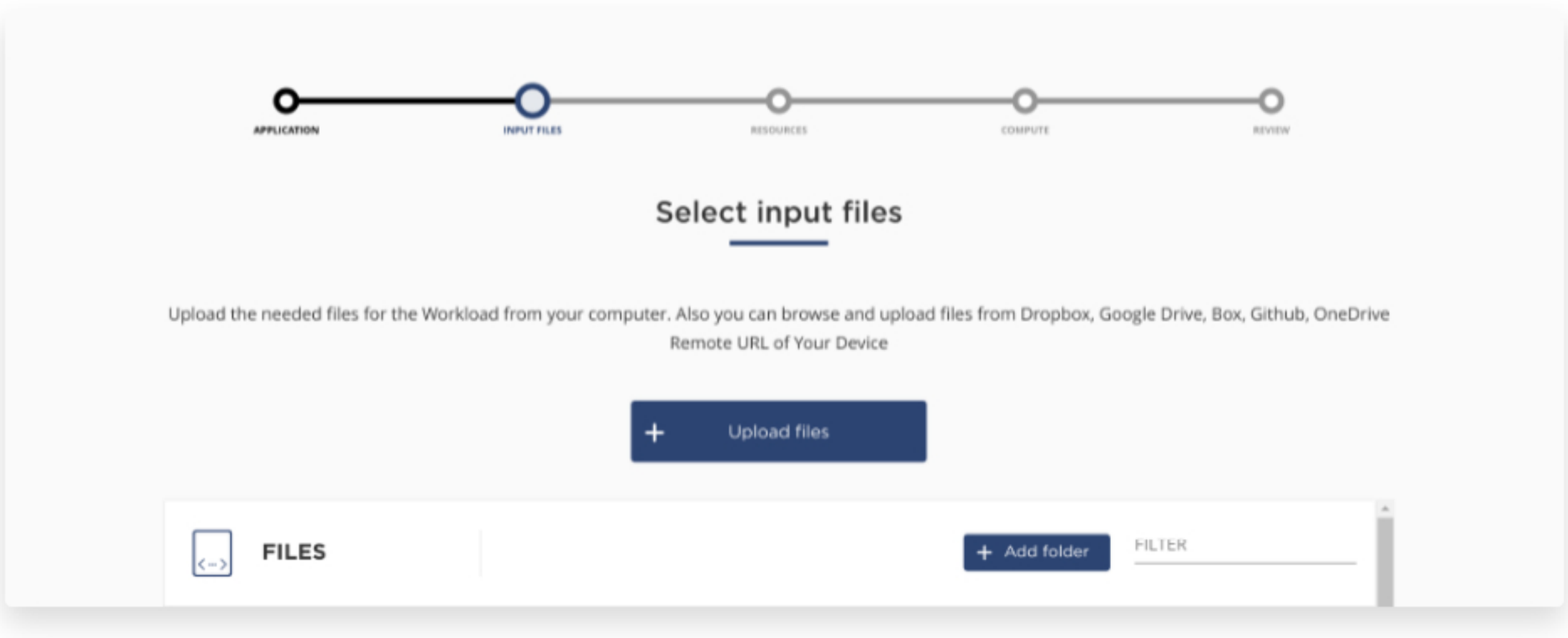
Batch mode allows you to run headless jobs by providing a python script, while interactive mode allows you to run a Jupyter notebook where you can choose to run scripts from the terminal or experiment with the notebook.

3. Click **New Workload**

4. Click **Next**



5. Upload files

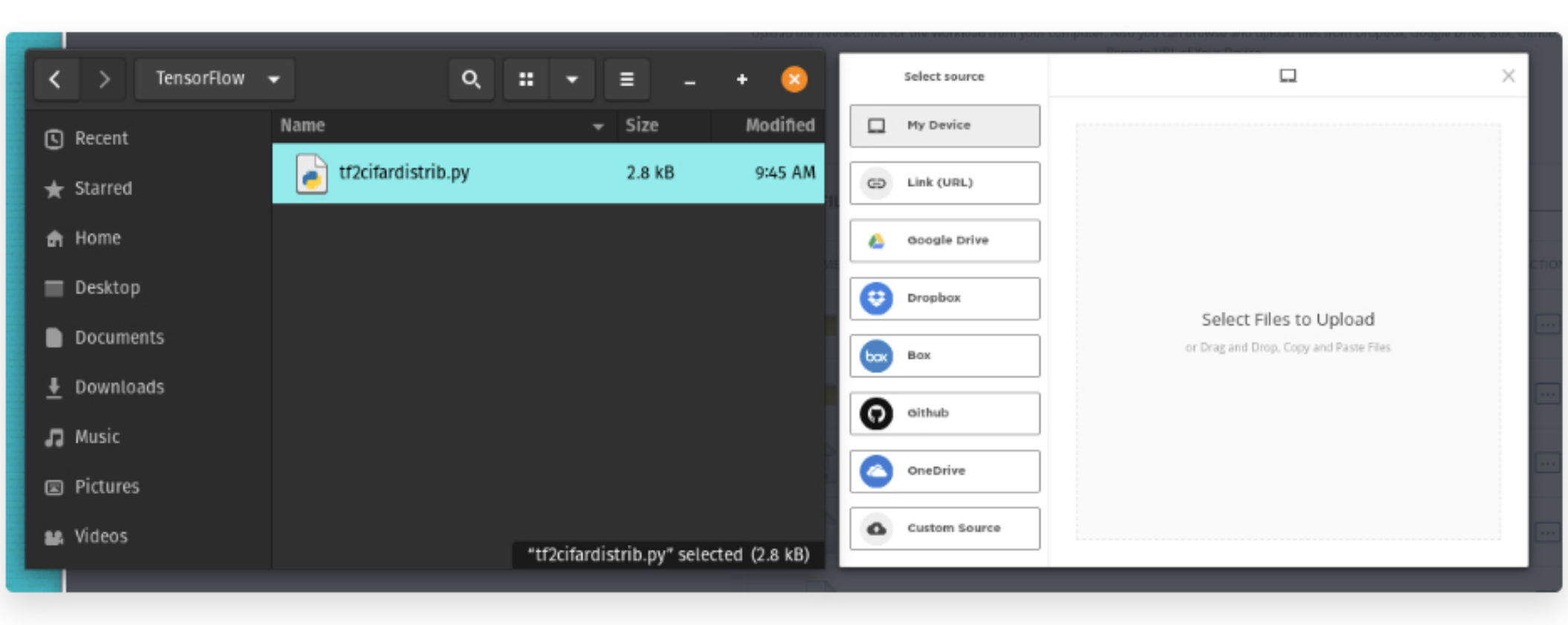


Clone the example code from the Plexus GitHub repository using the following commands from within the interactive terminal, once the workload is deployed.

```
$ git clone <https://github.com/corescientific/plexus-examples>
$ cd plexus-examples/TensorFlow
```

Upload the following file:

```
tf2cifardistrib.py
```

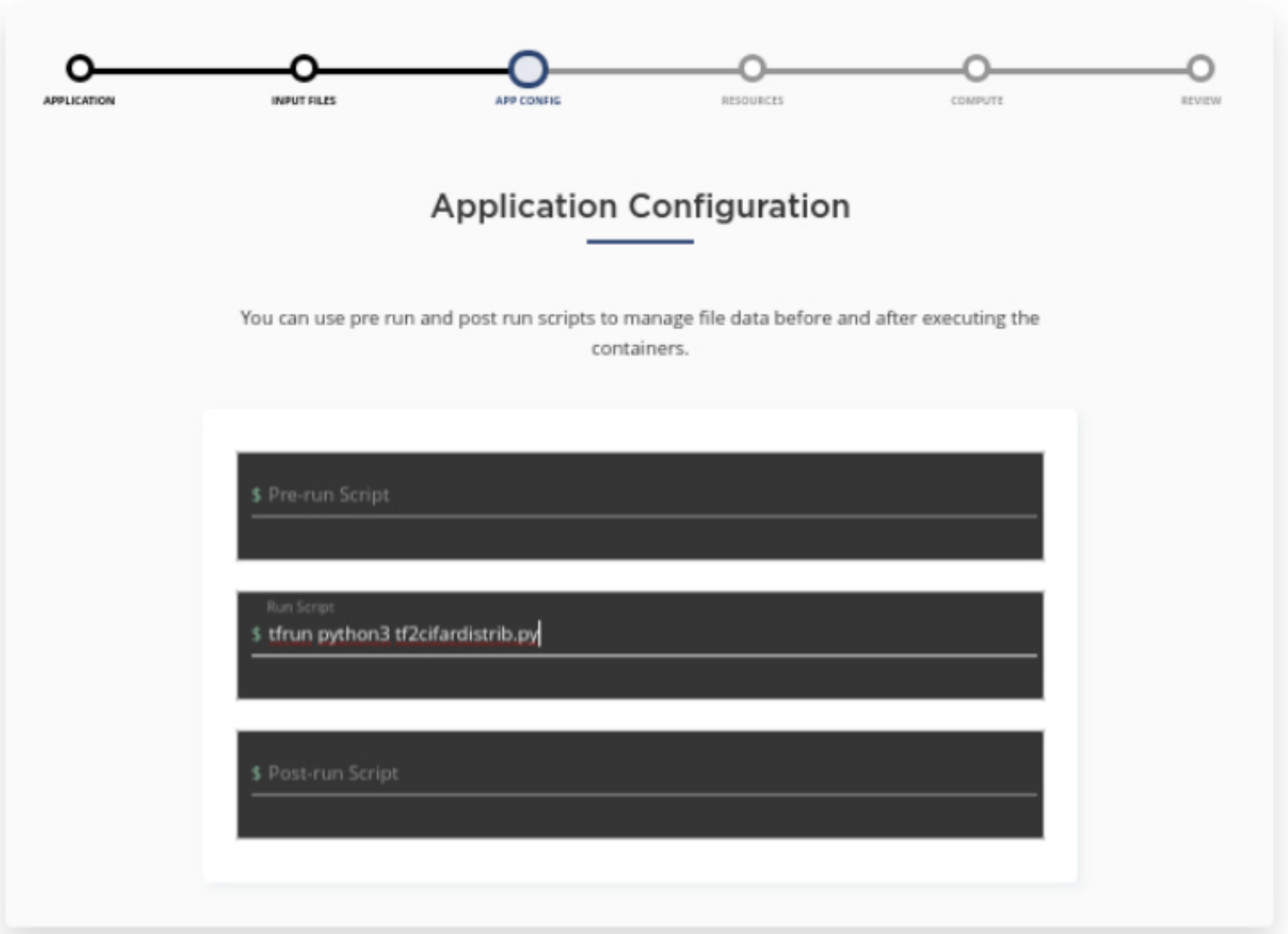


6. Once the upload is complete, click **Next**.

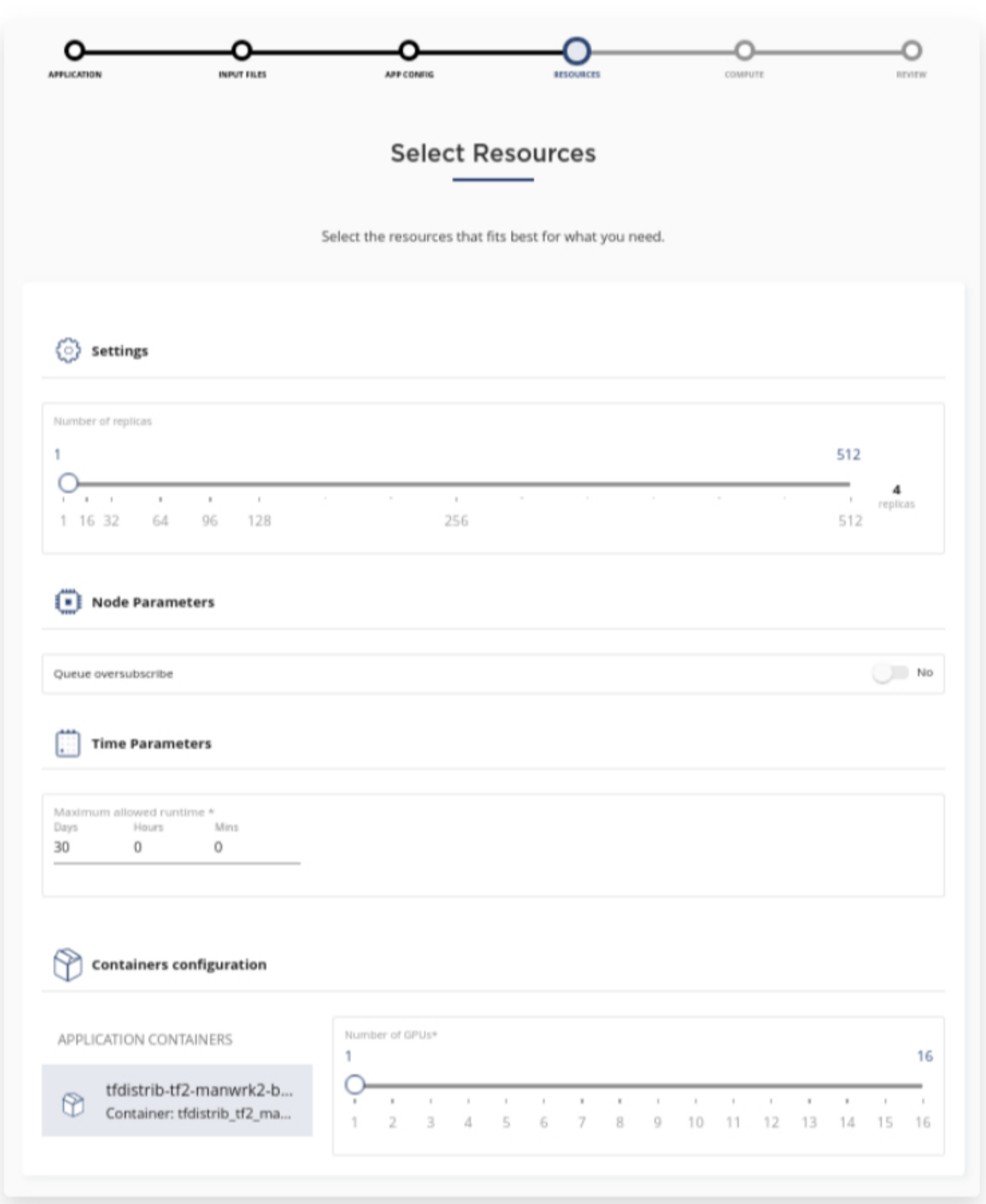
If you selected **batch mode** in step two above, and uploaded a script to run, you can tell Plexus to execute this script with TensorFlow distributed by entering:

```
tfrun python3 tf2cifardistrib.py
```

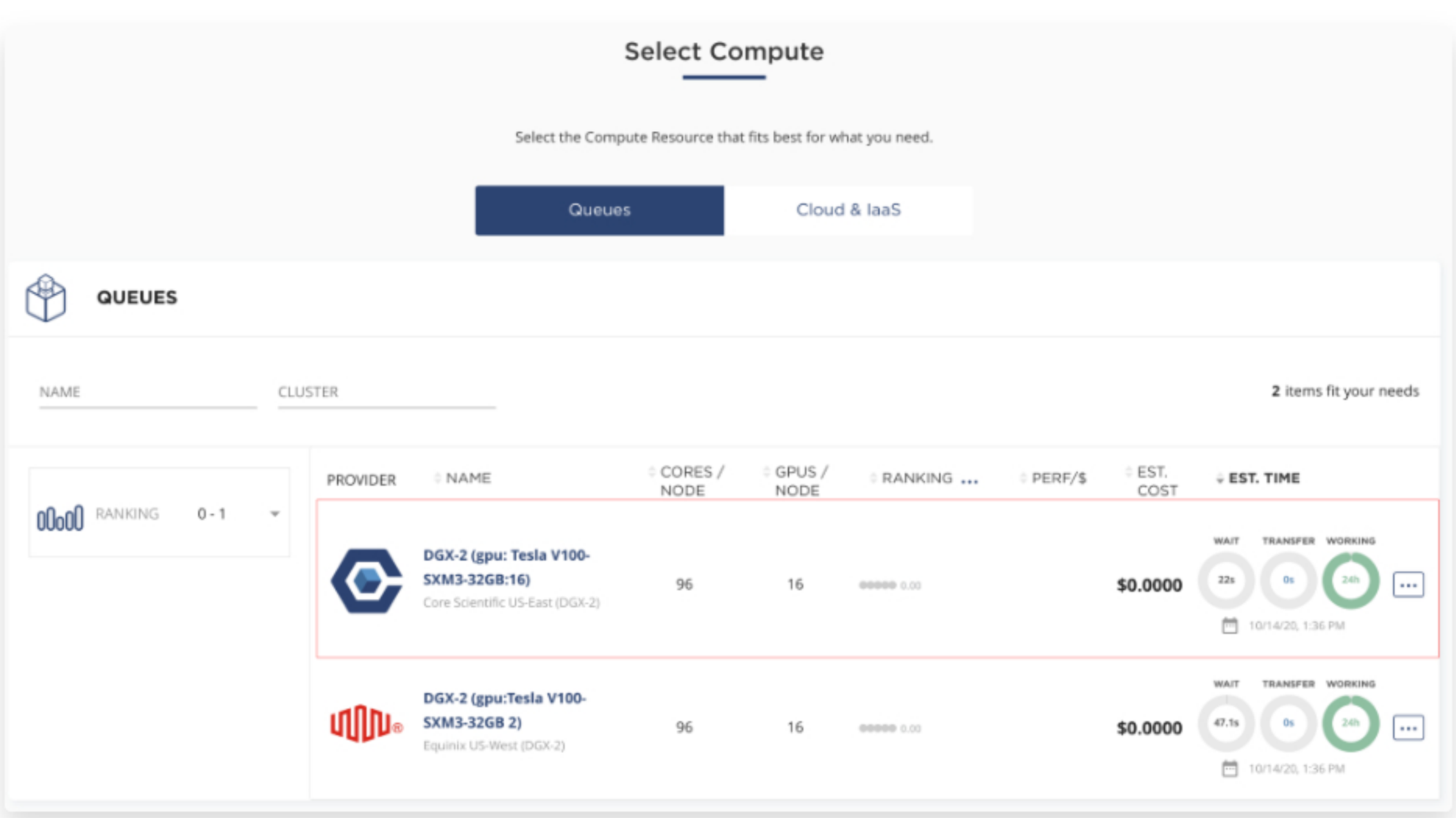
Plexus will handle how to define the cluster once the number of workers is selected in the following screens.



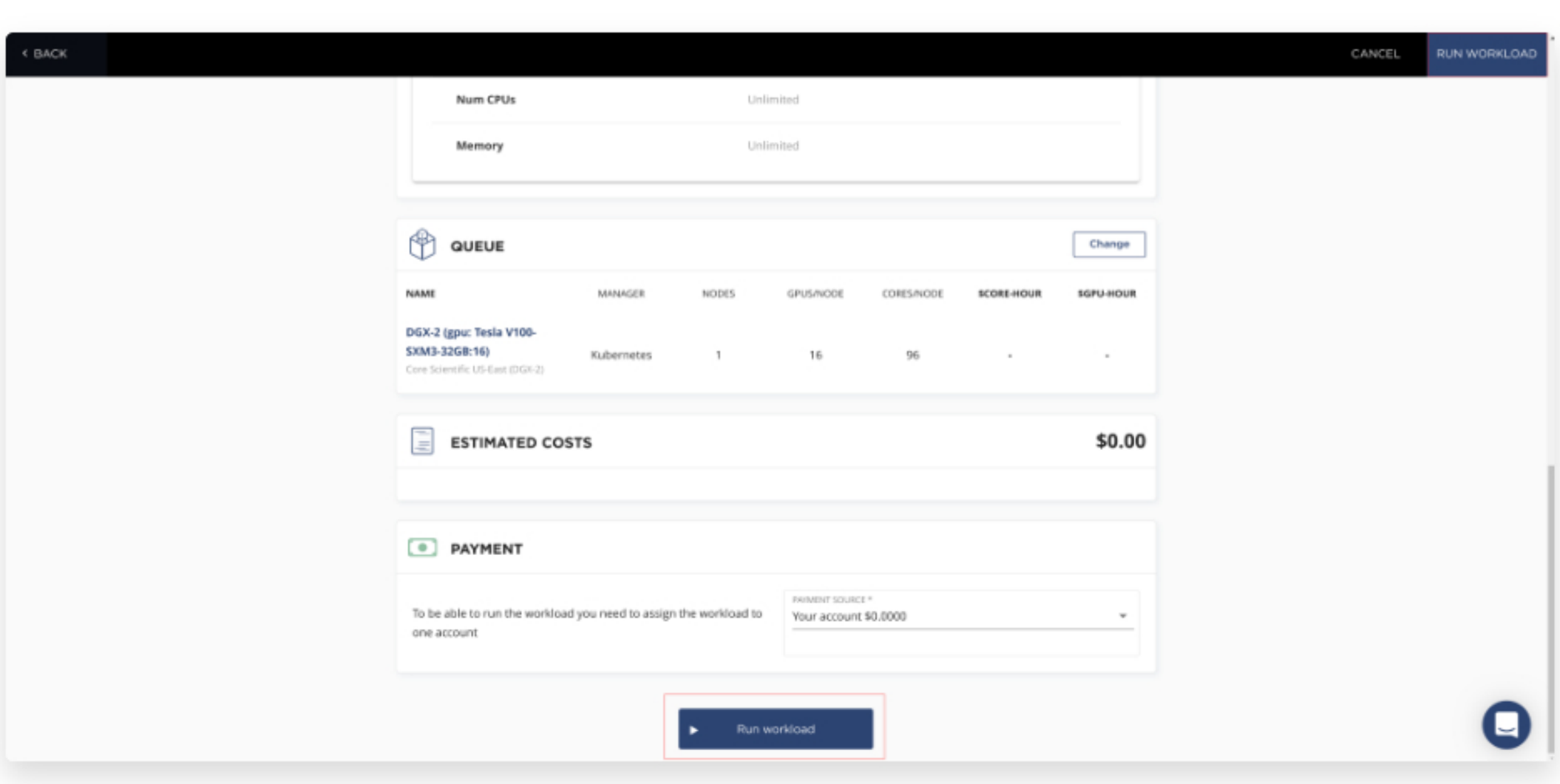
7. Select the number of workers (replicas) you would like to use with the top slider. Select the total number of GPUs per worker, and then click **Next**.



8. Select a queue, and then click **Next**

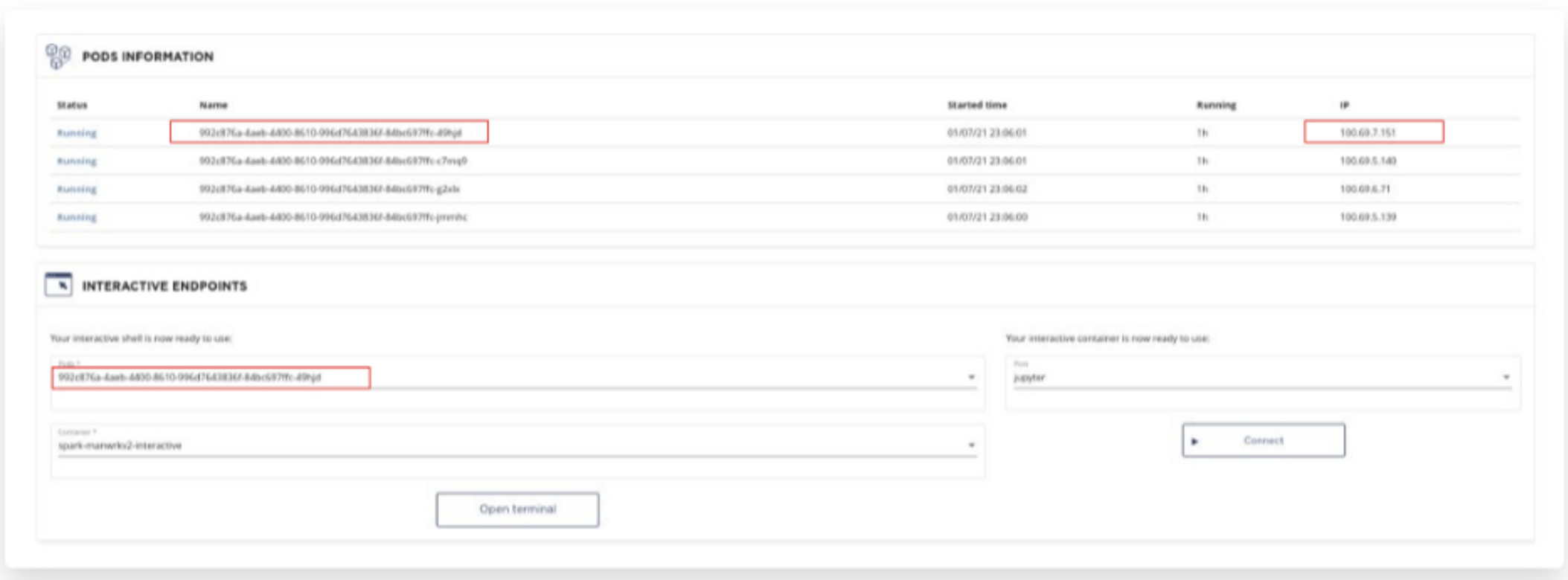


9. Review and run the workload

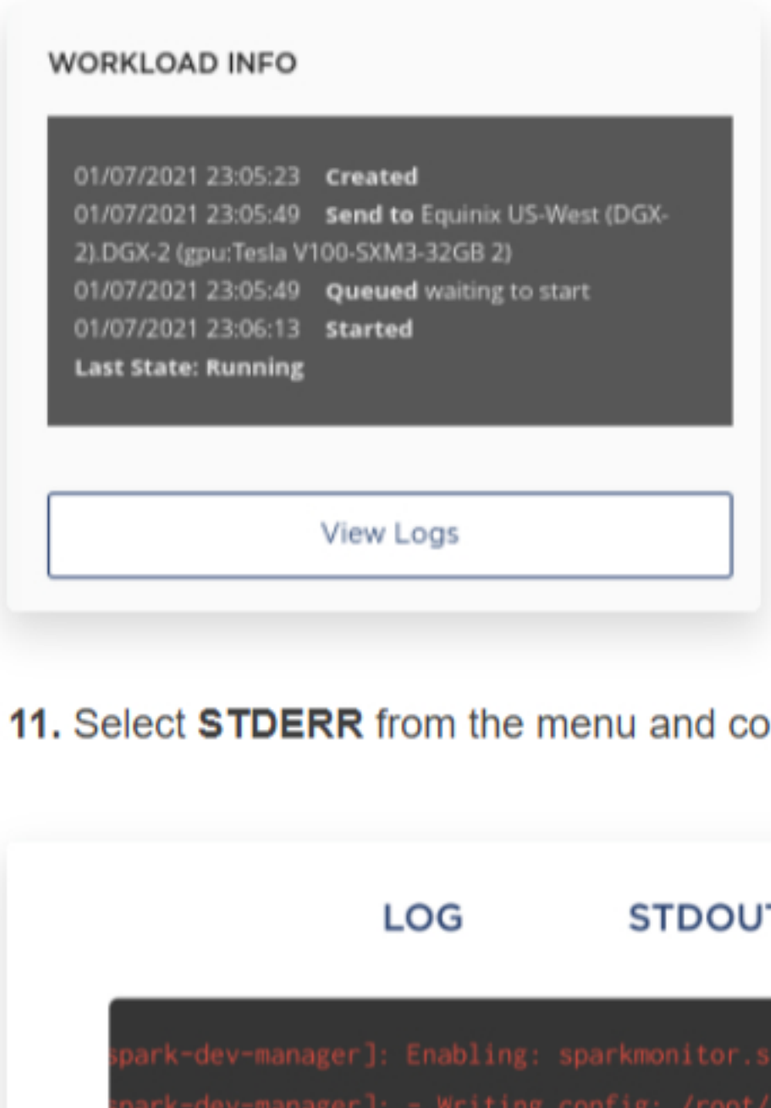


If you selected interactive mode, wait for the application to be **Running** and click on the application's name. If you selected **batch mode**, Plexus will run your script on the TensorFlow cluster.

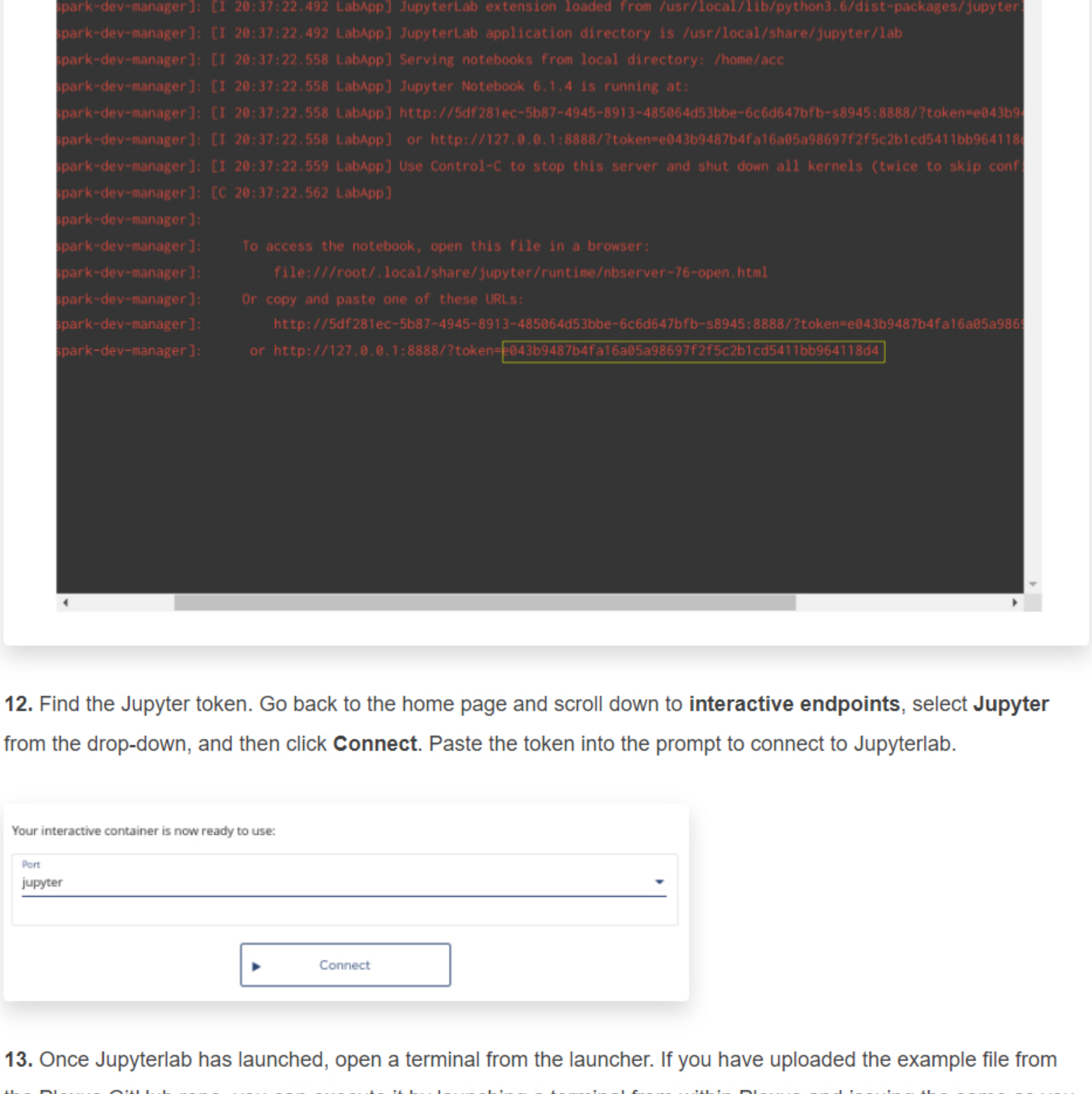
10. Select the **master** pod from the interactive endpoints



NOTE: This is currently not labeled, so you may have to cycle through endpoints until the manager is found. This will be updated soon so the manager can be found by name. Currently the manager can only be found by viewing logs. You can find tokens for use in the Jupyter notebook by clicking **View Logs**.



11. Select **STDERR** from the menu and copy the Jupyter notebook token to a .txt file



12. Find the Jupyter token. Go back to the home page and scroll down to **interactive endpoints**, select **Jupyter** from the drop-down, and then click **Connect**. Paste the token into the prompt to connect to Jupyterlab.

13. Once Jupyterlab has launched, open a terminal from the launcher. If you have uploaded the example file from the Plexus GitHub repo, you can execute it by launching a terminal from within Plexus and issuing the same as you would have used in the prerun script in step 7 above.

```
$ tfrun python3 tf2cifardistrib.py
```

About Core Scientific

Core Scientific is a leader in infrastructure and software solutions for Artificial Intelligence and Blockchain.