Lab - Getting started with machine teaching using Cartpole and Bonsai using the CLI

In this lab, you'll be introduced to machine teaching using Bonsai using the CLI in the new Windows Terminal. You'll create a brain to train the OpenAI Gym environment for Cartpole, a simple balance control problem.

Prerequisites

- Experience with Anaconda Distribution, Python 3.7 version
- Git for Windows

Clone the Cartpole GitHub repo

1. Open Edge from your taskbar. The cartpole lab repo for this lab is the starting page.

2. On the GitHub repo page, click on the **Clone or download** button, as in the following illustration:
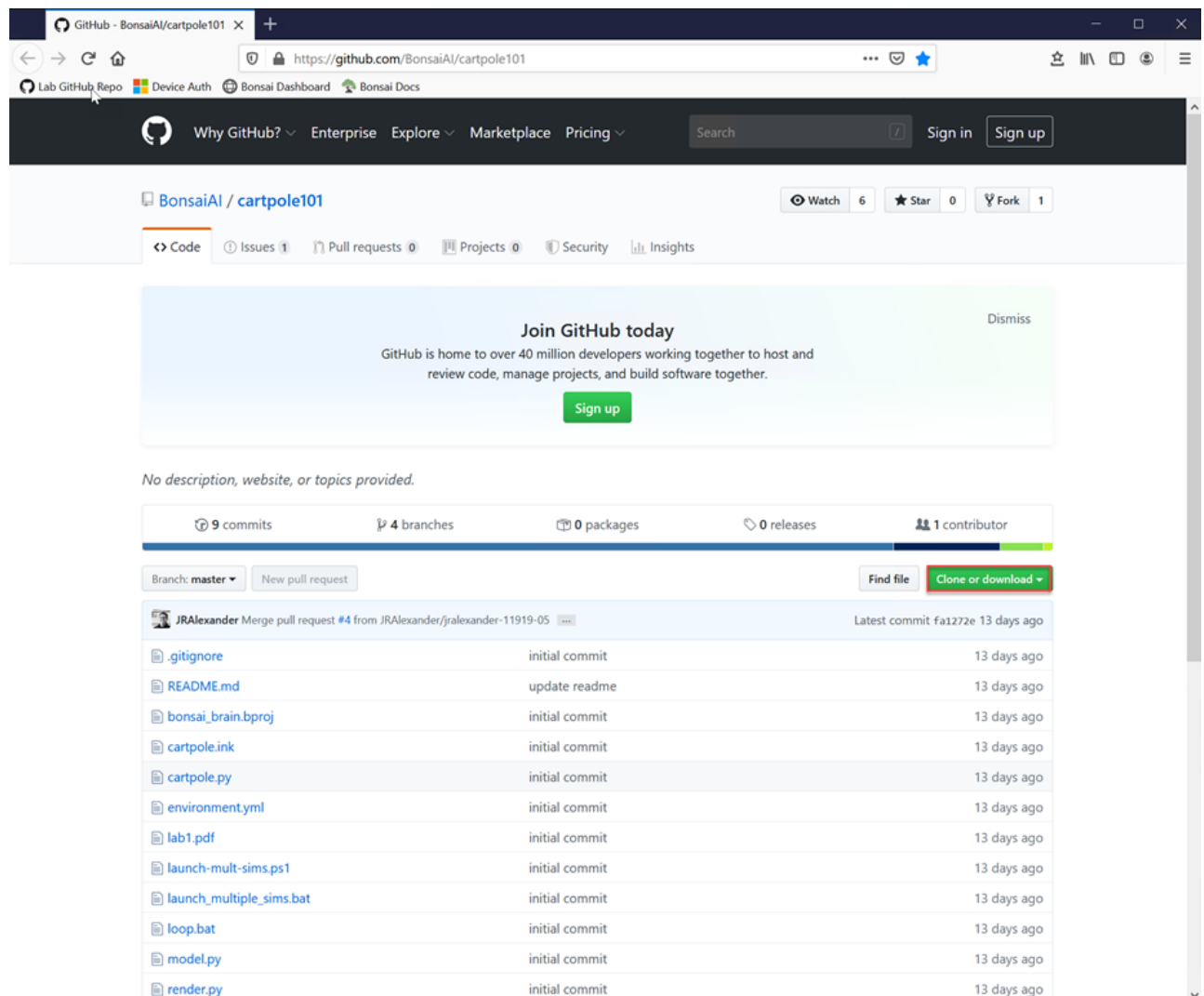
3. On the GitHub repo page, click on the **Copy to clipboard** button, as in the following illustration:
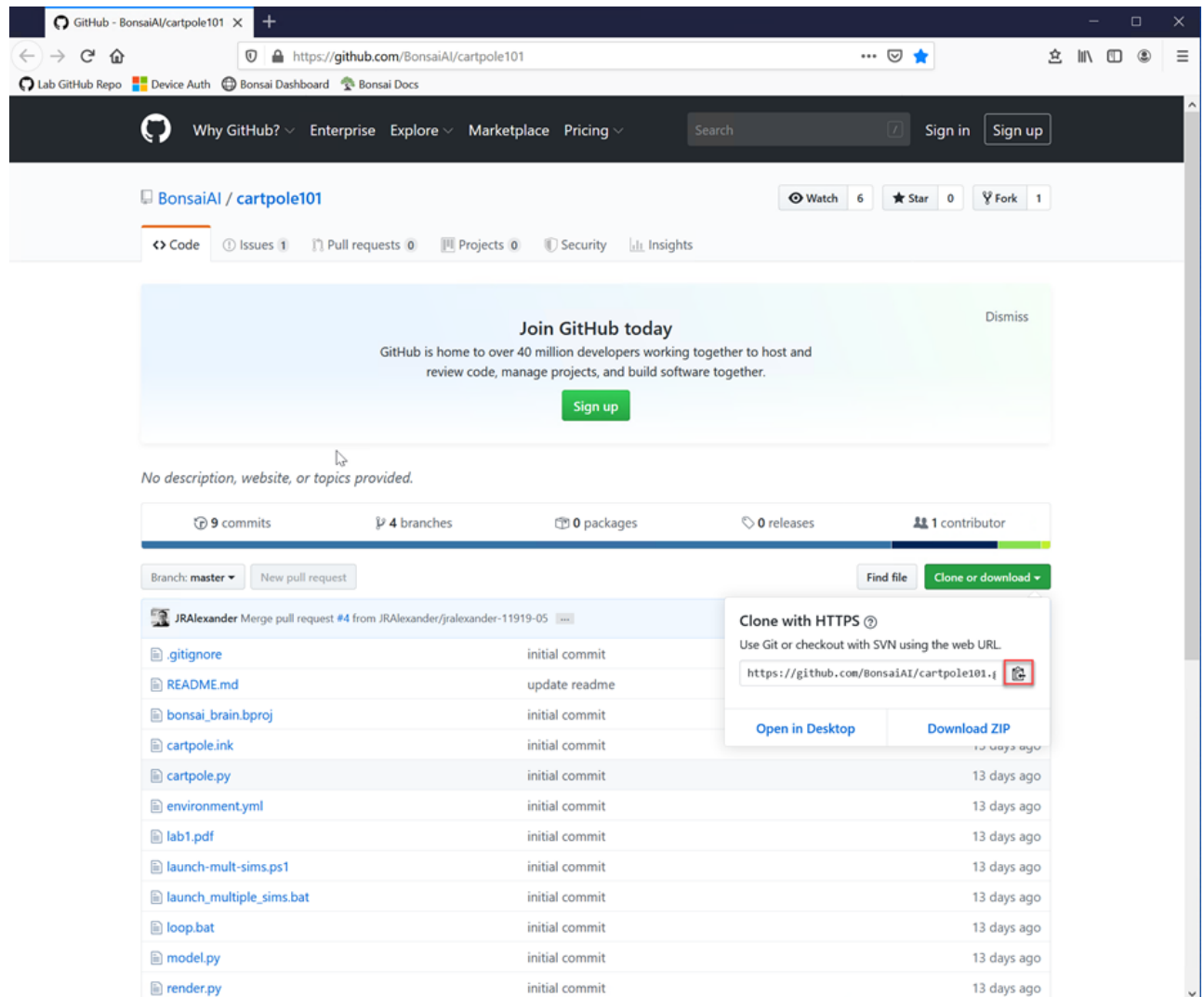
Create and activate the environment

5. In the taskbar, click on **Windows Terminal** and clone the **Cartpole101** GitHub repo with the following command that was copied to the clipboard:

```
git clone https://github.com/BonsaiAI/cartpole101.git
```

6. In **Windows Terminal**, create an environment with the following commands:

  o Change to the `cartpole101` directory you just created with `git clone` with the following command:

```
cd cartpole101
```

7. Next, activate the conda environment with the following command:

```
conda activate bonsai
```

Configure Bonsai

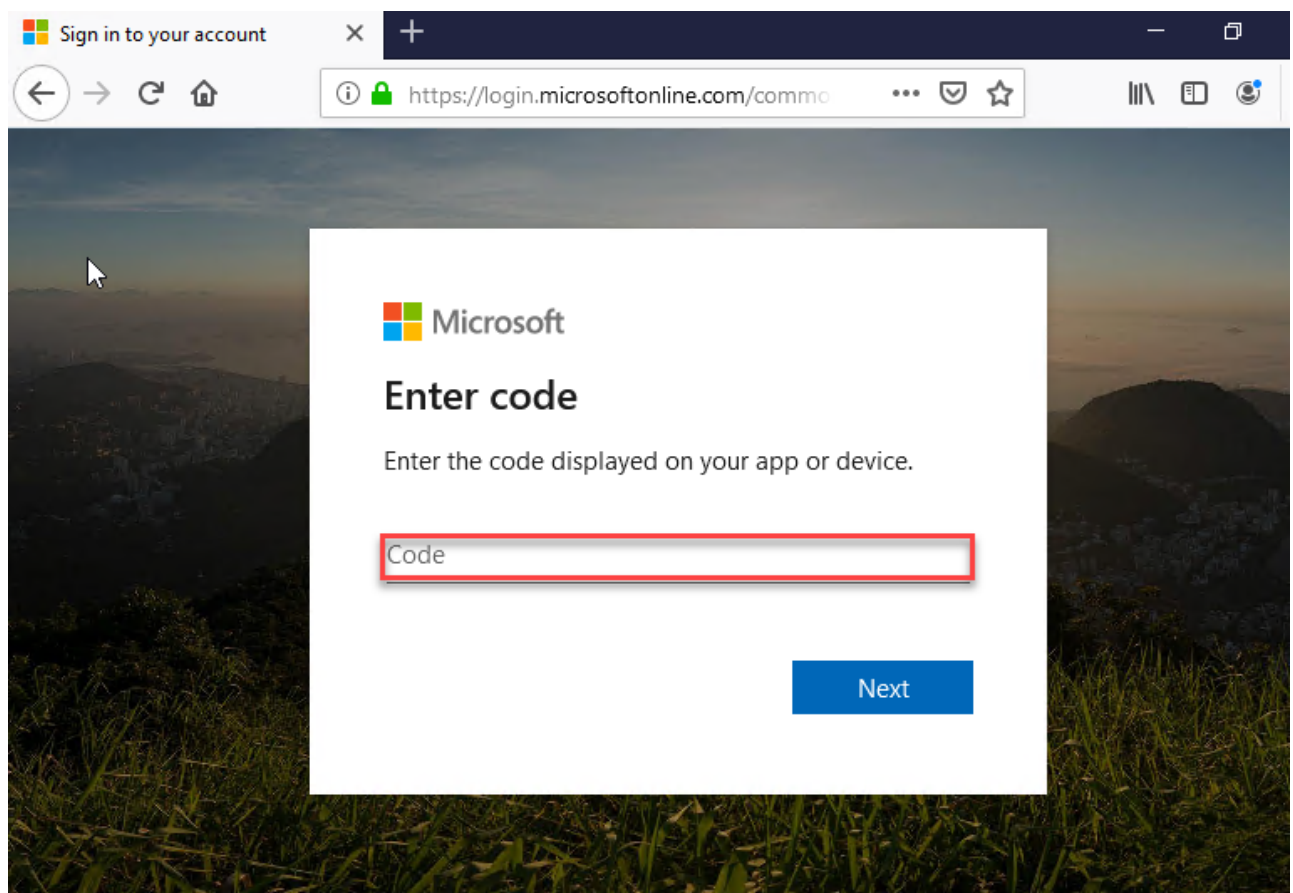8. Authenticate and configure Bonsai with the following command:

```
bonsai configure
```

- `bonsai configure` will request a device login with a code at devicelogin (there is a bookmark in your browser), as in the following illustration:
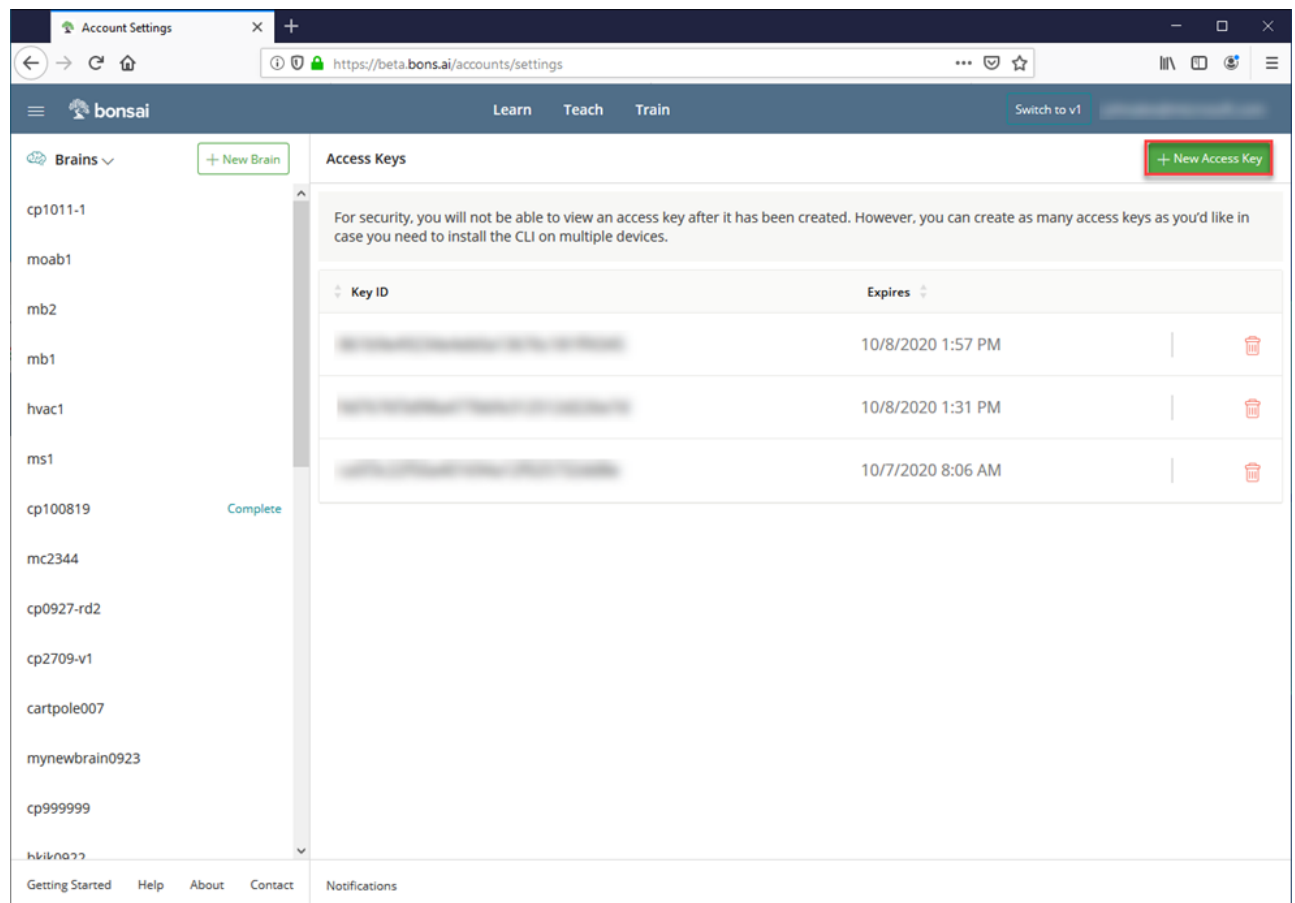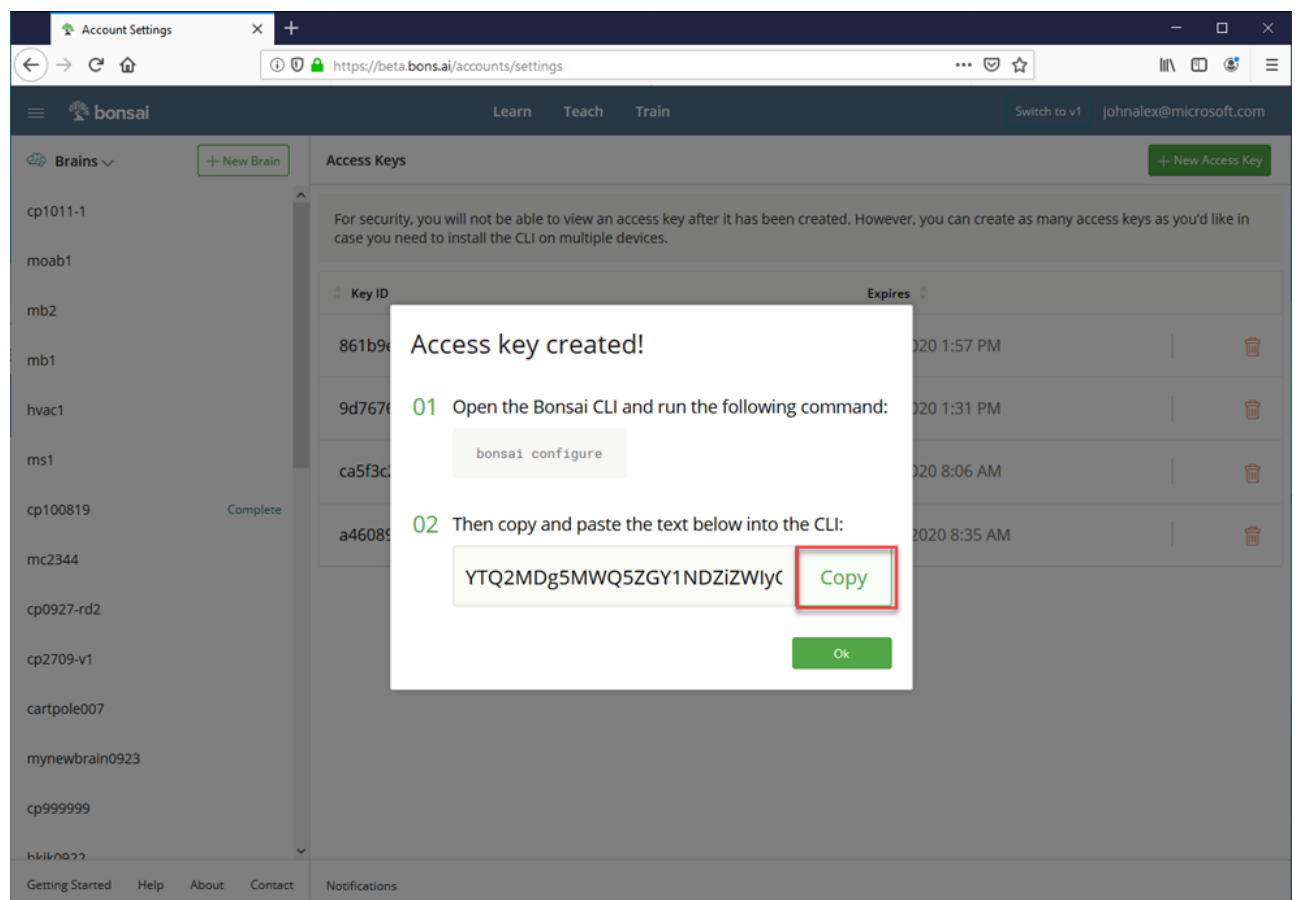

- Authenticate with your Azure account on the AAD sign in, as in the following screenshot:

- Sign into beta.bons.ai and create an access key, and paste it into the Windows Terminal prompt, as in the following illustration:

- Copy the access key and paste it into the Anaconda prompt, as in the following illustration:

Verify bonsai configuration with bonsai diagnose

bonsai diagnose runs several tests to ensure your local configuration is set up correctly. These tests are:

- Version of the CLI, to check if it is up-to-date.

- Status of http://beta.bons.ai

- Local configuration

- Run `bonsai diagnose` with the following command:

```
bonsai diagnose
```

Create the brain and upload related files

9. Create your brain (and its related project file) with bonsai create and give it a name (in the format cp-
   <today's date>) using the following command:

```
bonsai create cp-mmddyy
```

Use bonsai push to upload the `cp-mmddyy` brain and its associated files to the Bonsai AI Engine for
training:

```
bonsai push
```

Train the brain

10. Use the bonsai train start command to enable the Bonsai AI Engine training mode to train the BRAIN:

```
bonsai train start
```

Connecting a simulator

Now that your brain is ready to train, connect it to the `sim.py` file, and start the brain training.

> [!NOTE] Training doesn't start until the brain is connected to a simulator.

11. Connect your BRAIN to the python simulator for OpenAI Gym (`sim.py`) with the following command:

```
python sim.py --render
```

The `--render` parameter displays the simulation's graphical rendering, as in the following illustration:
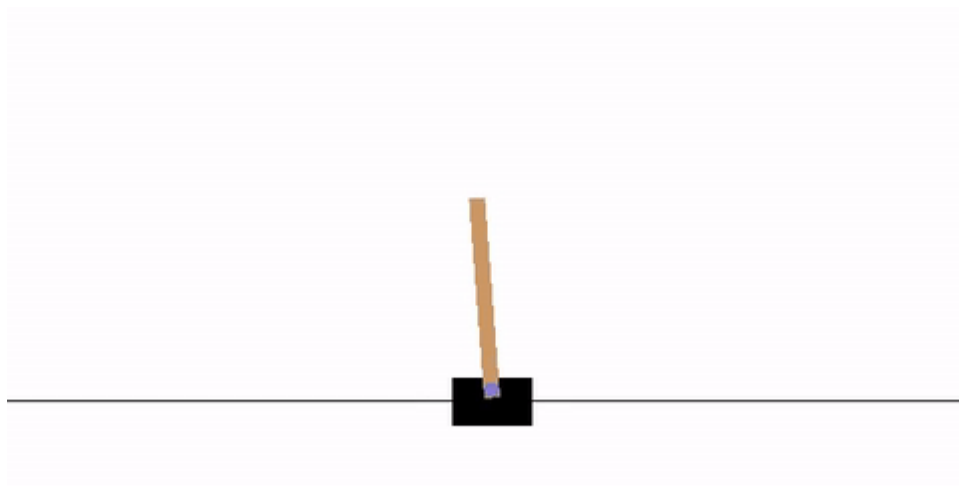
Connect additional simulators to maximize training time

12. Connect additional simulators to train your brain at a faster rate with a batch file.

    ○ Open a new instance of **Windows Terminal** by clicking on the plus sign (+) next to your current tab (Right-Click, Open).

    ○ Change to the `cartpole101` directory with the following command:
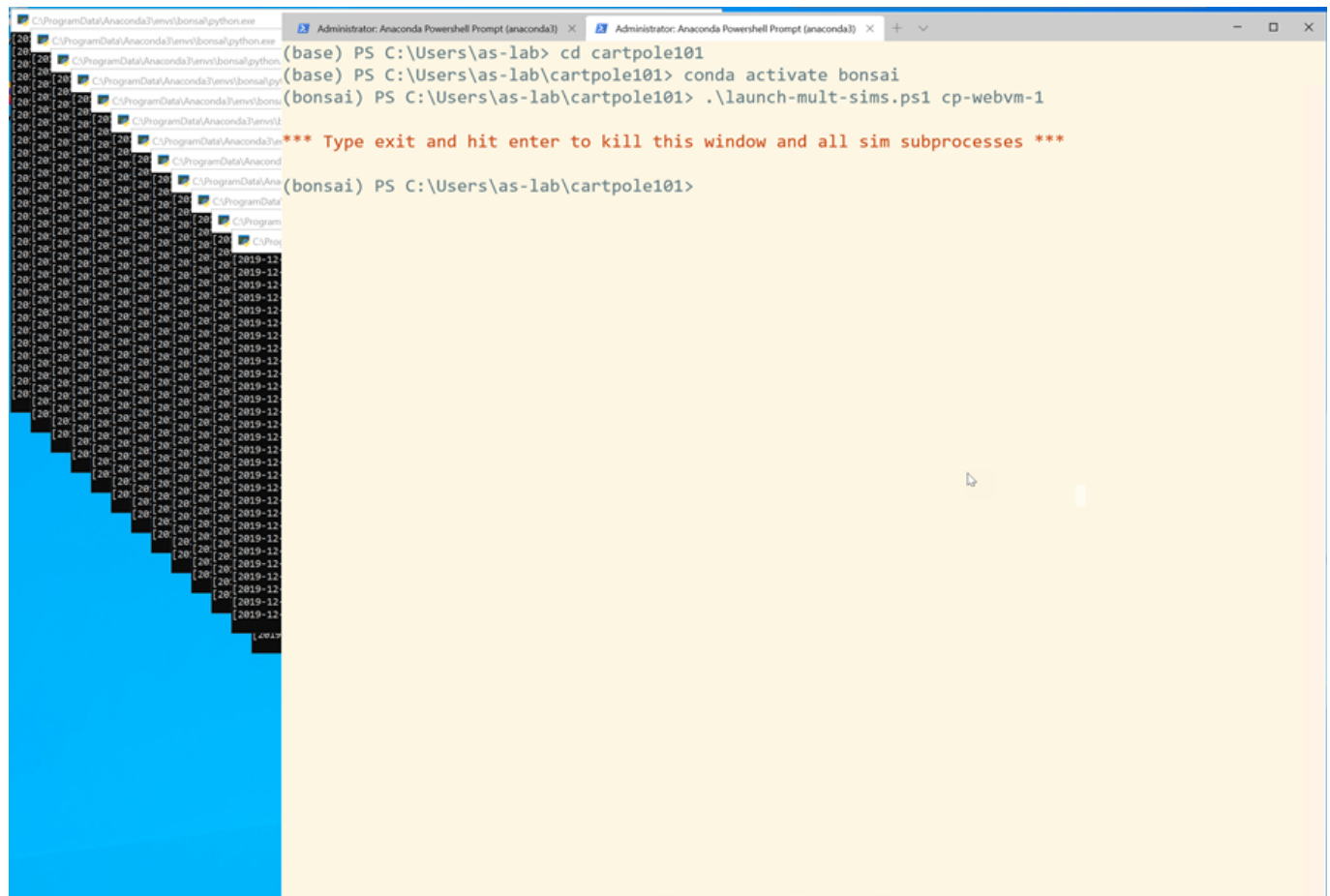
    ```
    cd cartpole101
    ```

    ○ Activate the the bonsai environment with the following command:

    ```
    conda activate bonsai
    ```

- Run the `launch-mult-sims.ps1` along with your brain name to launch 16 more instances of sim.py with the following command:

    ```
    launch-mult-sims.ps1 <your brain name>
    ```

The additional `sim.py` instances aren't graphically rendered to maximize speed. To kill all the spawned simulator processes, type `exit` in your Windows terminal instance, as in the following illustration:

Monitor training

While the brain is training, you can monitor the progress both locally and on the Bonsai website.
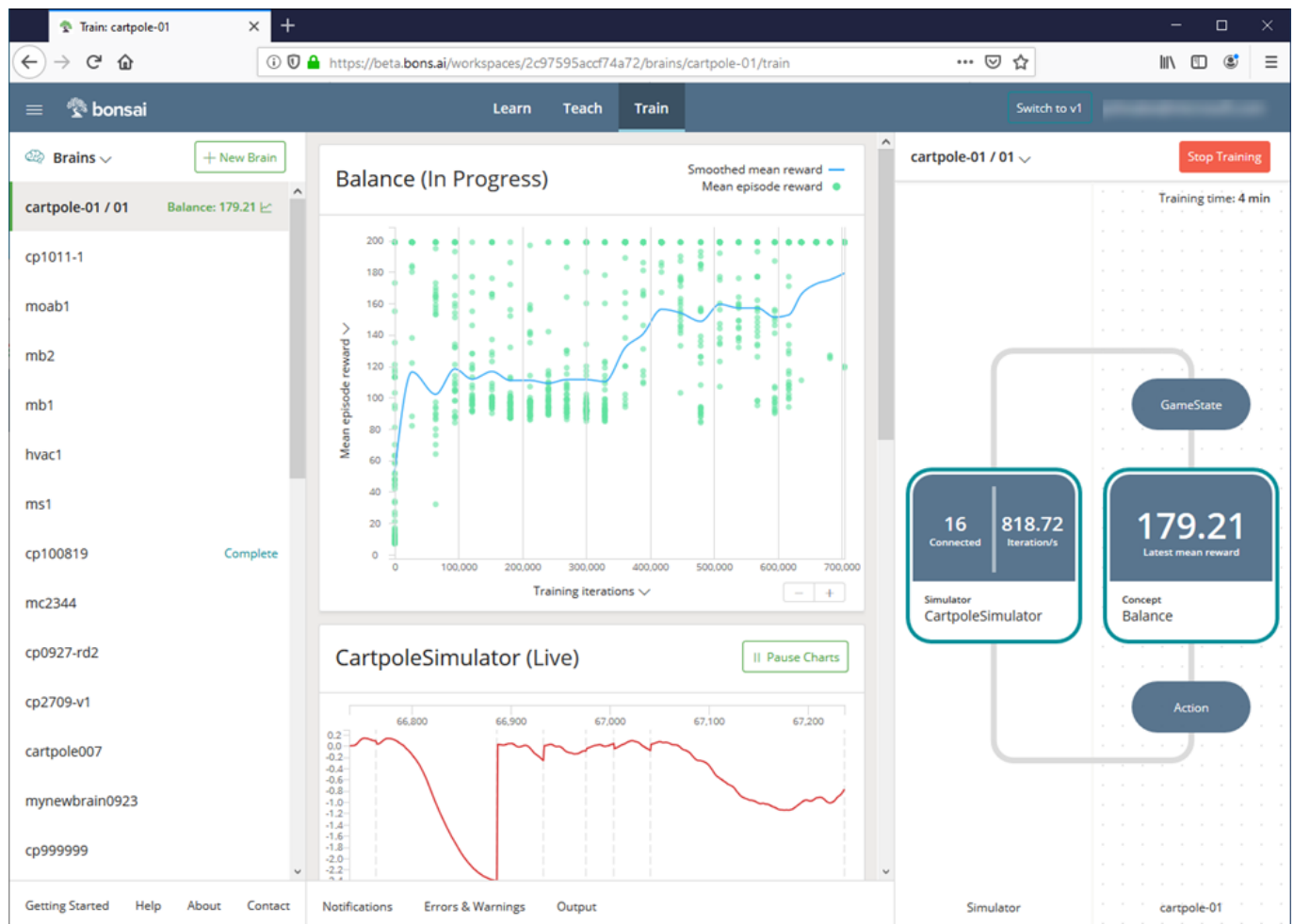
This graph will display each of your concepts to be trained (if you look in the Inkling code you'll see that Cartpole only has one).

Viewing simulator graphs

To monitor state variables from the simulator, click on the simulator box on the right hand side, as in the following illustration:
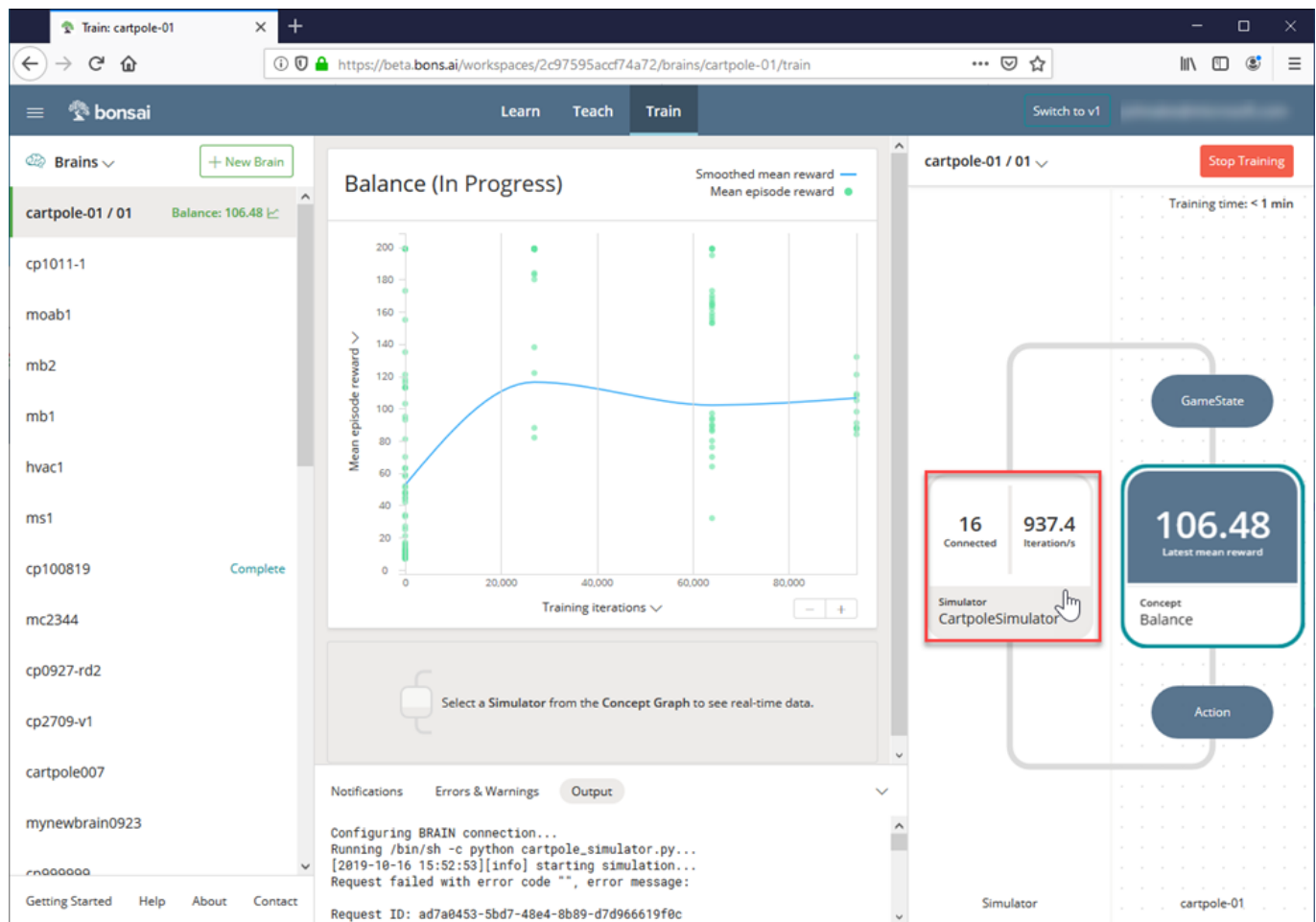
You can either monitor all the simulator output in a single graph, or add graphs for each variable. Click the **reward** button to deselect it, and the **position** button and then click **Add Chart**, as in the following illustration:
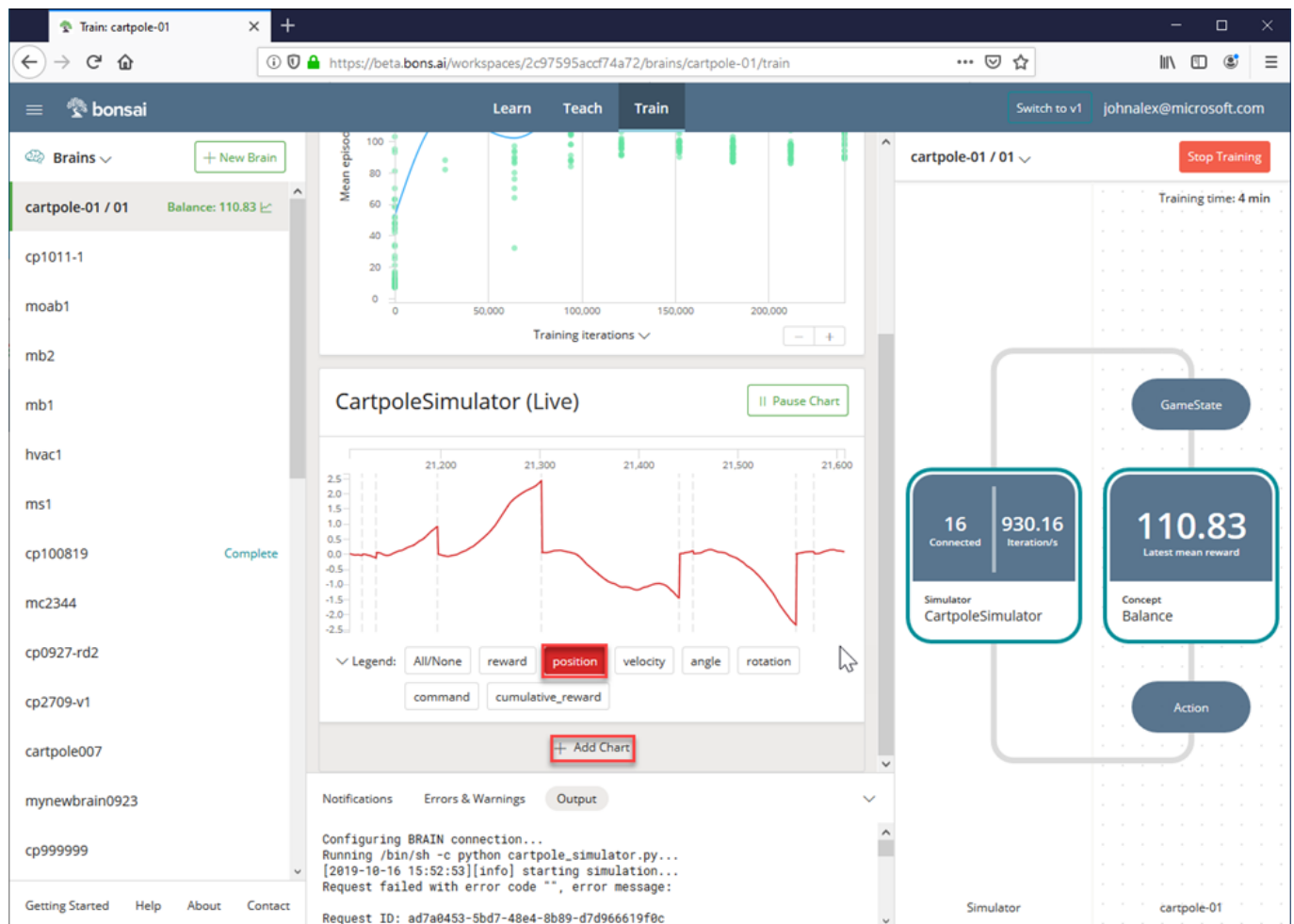
Next, click the **reward** button to deselect it, and the **cumulative_reward** button and then click **Add Chart**, as in the following illustration:
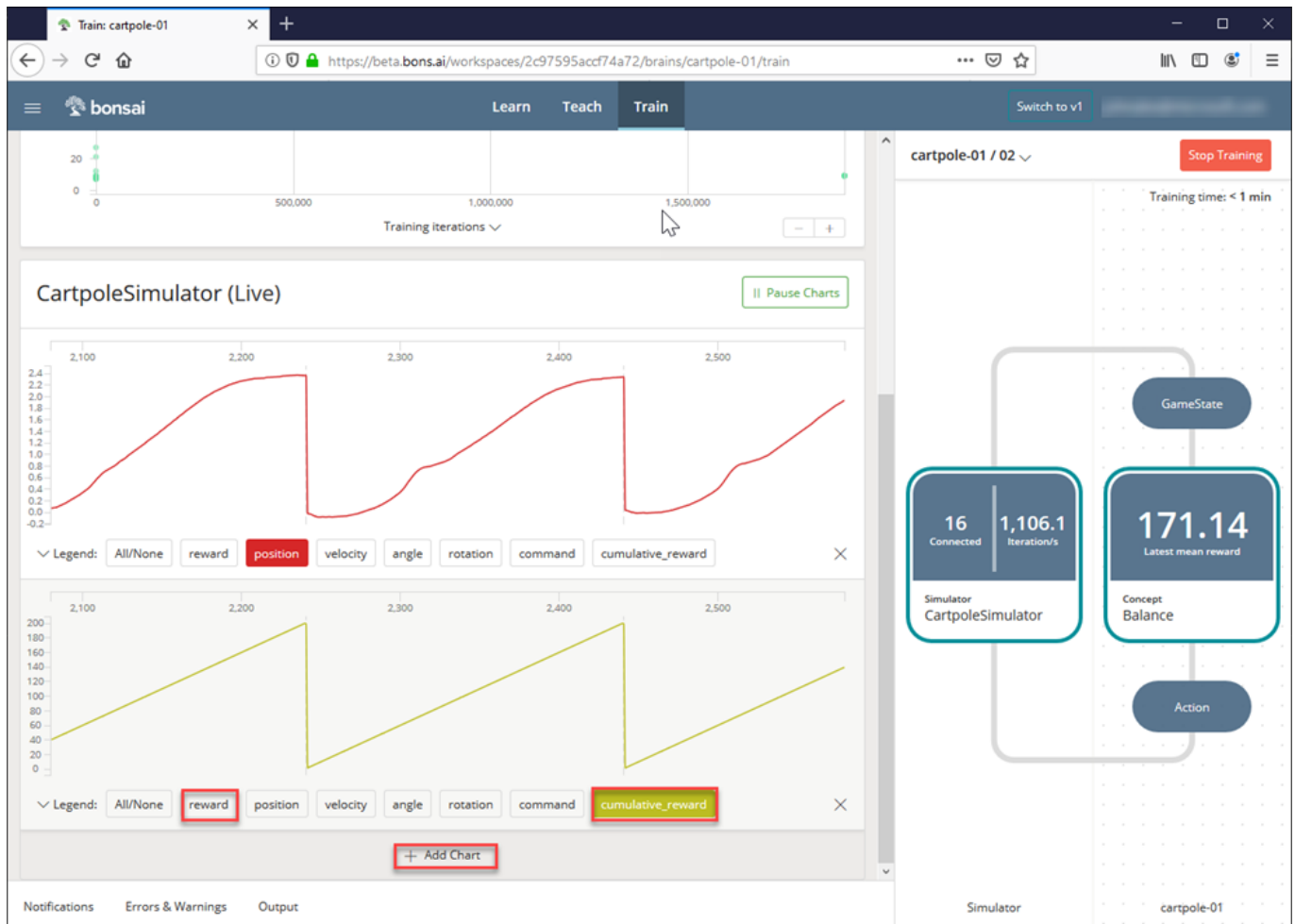
Continue the same process for **velocity**, **angle**, and **rotation**.

Training Cartpole for about 15 minutes will enable reasonable performance for this particular task. The longer the brain is trained, the longer the cart will be able to balance the pole before it falls over. Your graphs will vary because the training takes random actions, so don't worry if yours don't look like these.

Finish training

12. Once the brain is trained, use `bonsai train stop` , with the following command:

```
bonsai train stop
```

Predict the cartpole

13. Now that you've trained your brain, connect it again to the python simulator for OpenAI Gym (`sim.py`) with the following command:

```
python sim.py --render --predict
```

Congratulations! You've successfully trained your brain and used it for predictions.