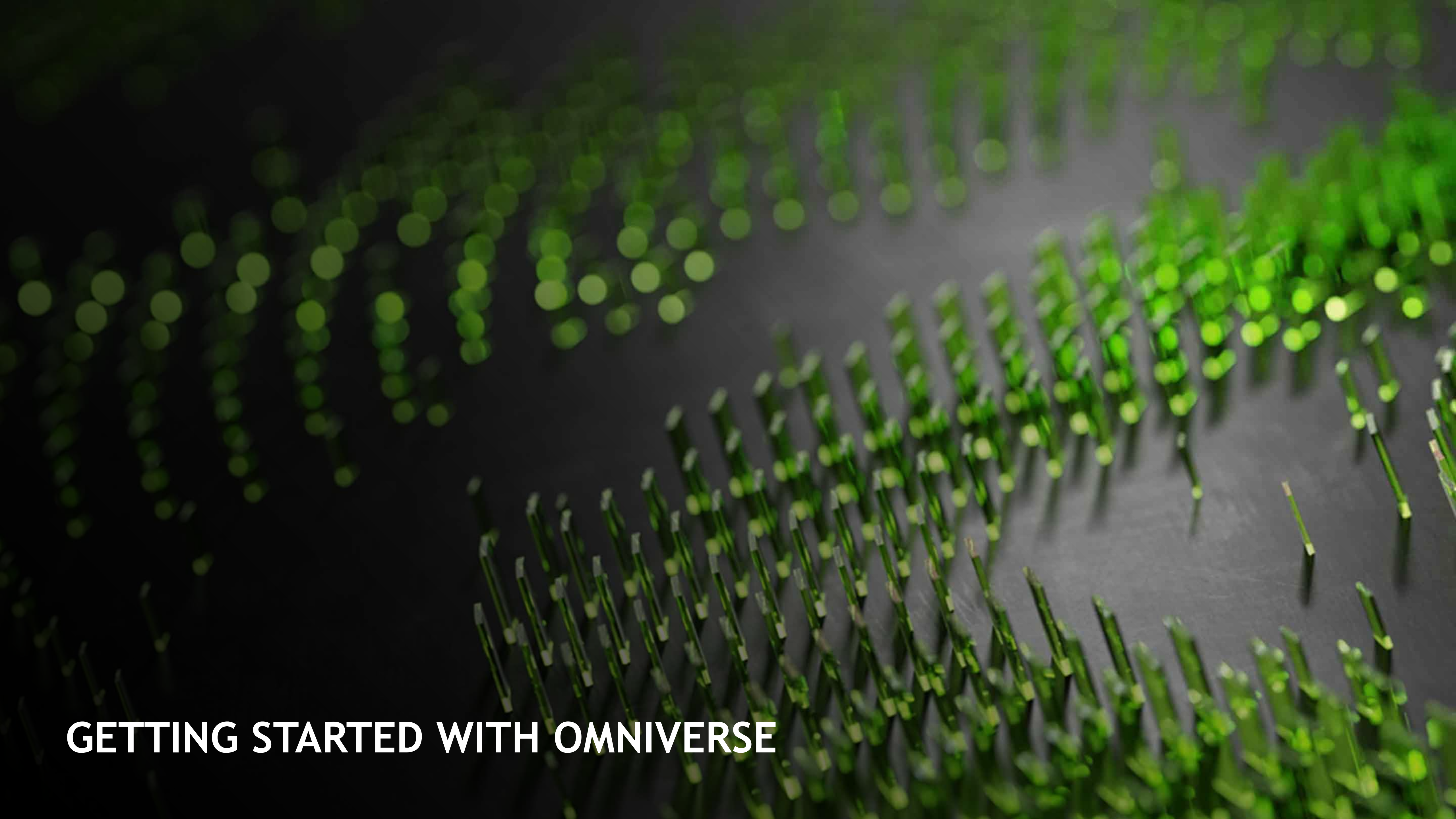
TUTORIAL

TOWARDS DIGITAL TWINS IN OMNIVERSE: VISUALIZING WITH ACTION GRAPHS

PALLAVI MOHAN

SOLUTIONS ARCHITECT,

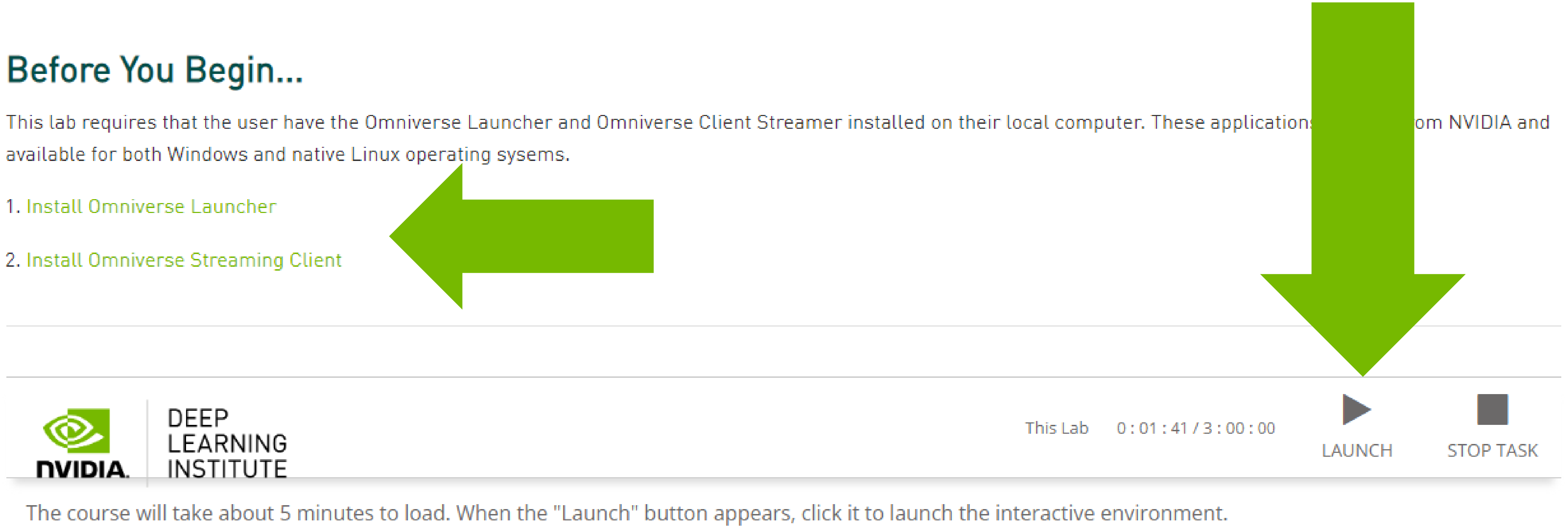
NVIDIA



GETTING STARTED WITH OMNIVERSE

A QUICK PAUSE AND CHECK

You should see this LAUNCH button



Before You Begin...

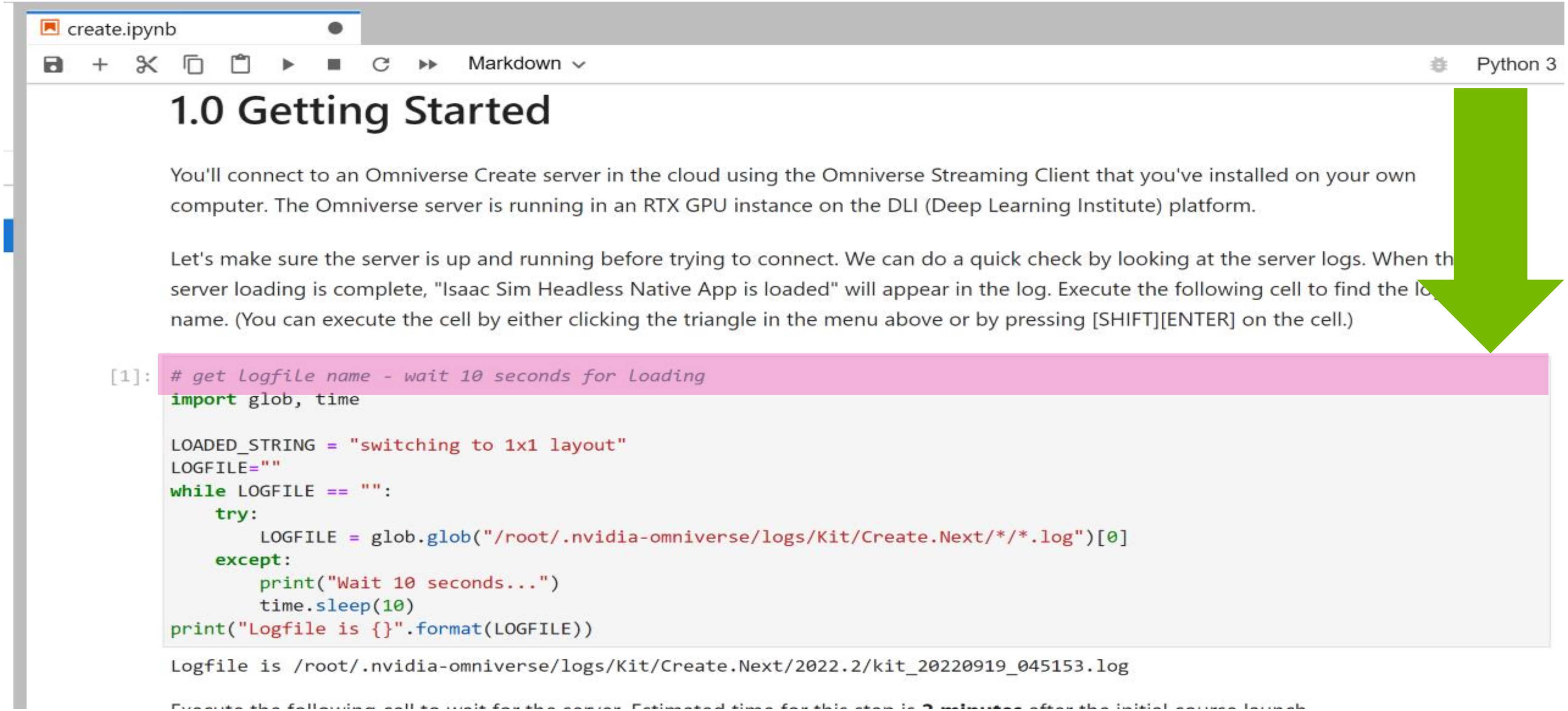
This lab requires that the user have the Omniverse Launcher and Omniverse Client Streamer installed on their local computer. These applications om NVIDIA and available for both Windows and native Linux operating sysems.

1. Install Omniverse Launcher
2. Install Omniverse Streaming Client

The course will take about 5 minutes to load. When the "Launch" button appears, click it to launch the interactive environment.

A QUICK PAUSE AND CHECK

To Launch Create: Execute the first 2 cells by clicking SHIFT+ENTER for each cell



1.0 Getting Started

You'll connect to an Omniverse Create server in the cloud using the Omniverse Streaming Client that you've installed on your own computer. The Omniverse server is running in an RTX GPU instance on the DLI (Deep Learning Institute) platform.

Let's make sure the server is up and running before trying to connect. We can do a quick check by looking at the server logs. When th server loading is complete, "Isaac Sim Headless Native App is loaded" will appear in the log. Execute the following cell to find the lo name. (You can execute the cell by either clicking the triangle in the menu above or by pressing [SHIFT][ENTER] on the cell.)

[1]:

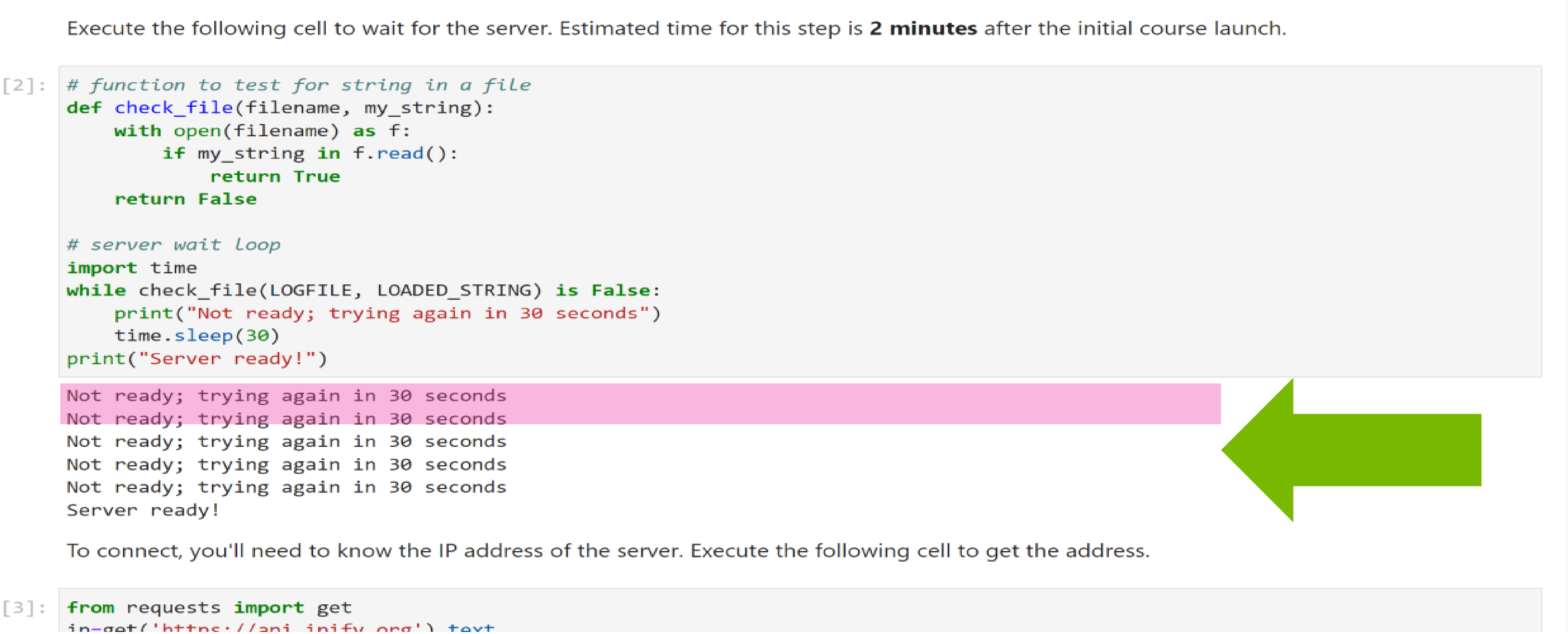
```
# get logfile name - wait 10 seconds for loading
import glob, time

LOADED_STRING = "switching to 1x1 layout"
LOGFILE=""
while LOGFILE == "":
    try:
        LOGFILE = glob.glob("/root/.nvidia-omniverse/logs/Kit/Create.Next/*/*.log")[0]
    except:
        print("Wait 10 seconds...")
        time.sleep(10)
print("Logfile is {}".format(LOGFILE))
```

```
Logfile is /root/.nvidia-omniverse/logs/Kit/Create.Next/2022.2/kit_20220919_045153.log
```

Execute the following cell to wait for the server. Estimated time for this step is 2 minutes after the initial course launch

A QUICK PAUSE AND CHECK

In the second cell, when you execute it, it will print out “Not ready…” this is normal



Execute the following cell to wait for the server. Estimated time for this step is **2 minutes** after the initial course launch.

[2]:

```
# function to test for string in a file
def check_file(filename, my_string):
    with open(filename) as f:
        if my_string in f.read():
            return True
    return False

# server wait loop
import time
while check_file(LOGFILE, LOADED_STRING) is False:
    print("Not ready; trying again in 30 seconds")
    time.sleep(30)
print("Server ready!")
```

```
Not ready; trying again in 30 seconds
Not ready; trying again in 30 seconds
Not ready; trying again in 30 seconds
Not ready; trying again in 30 seconds
Not ready; trying again in 30 seconds
Server ready!
```

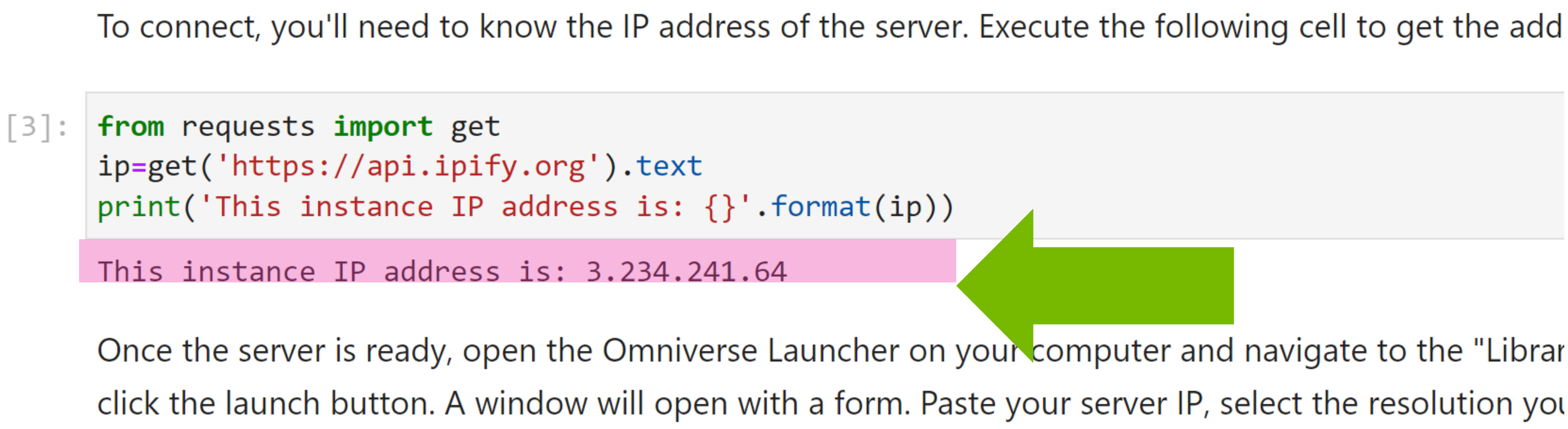
To connect, you'll need to know the IP address of the server. Execute the following cell to get the address.

[3]:

```
from requests import get
ip=get('https://api.ipify.org').text
```

QUICK CHECK AND SET UP FOR HANDS ON

Execute the next cell to get your unique IP address!



To connect, you'll need to know the IP address of the server. Execute the following cell to get the add

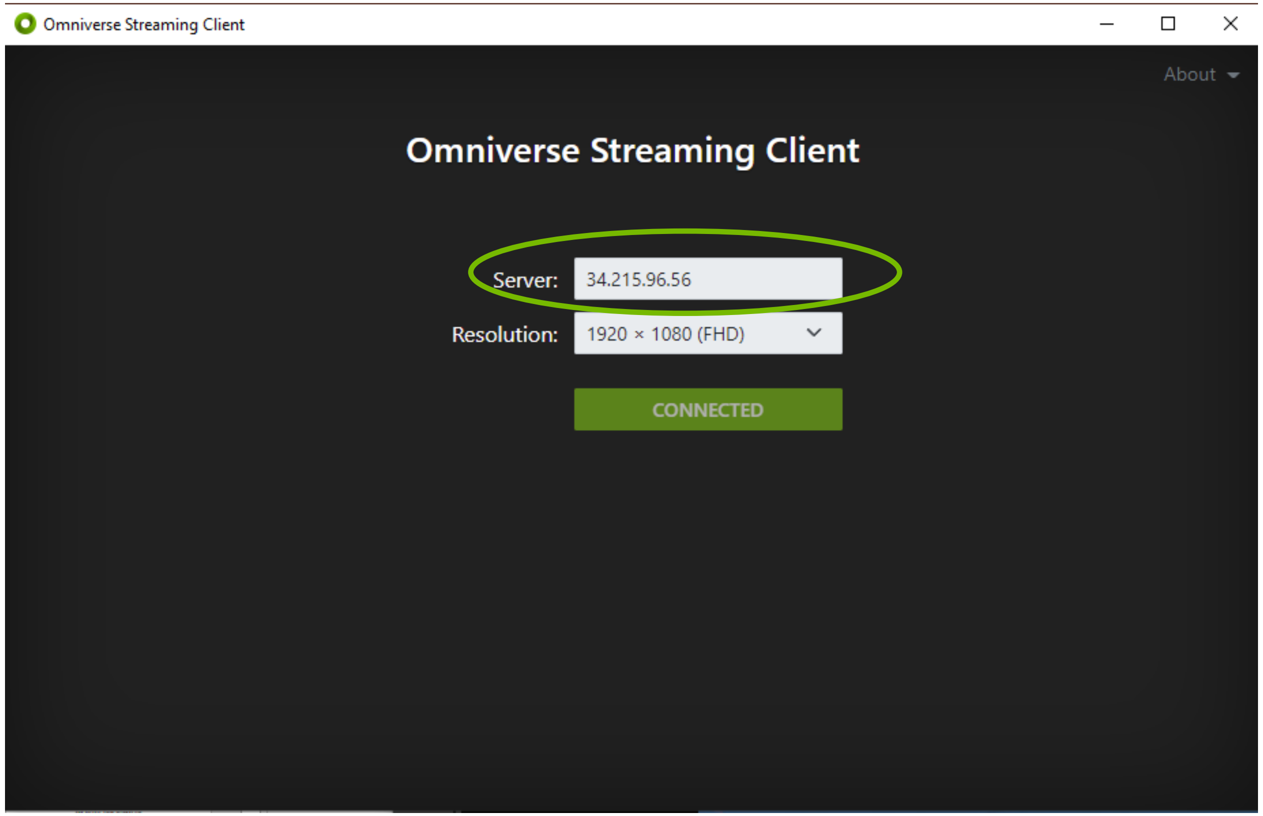
```
[3]: from requests import get
     ip=get('https://api.ipify.org').text
     print('This instance IP address is: {}'.format(ip))
```

```
This instance IP address is: 3.234.241.64
```

Once the server is ready, open the Omniverse Launcher on your computer and navigate to the "Librar

click the launch button. A window will open with a form. Paste your server IP, select the resolution yo

LAUNCHING OMNIVERSE CREATE

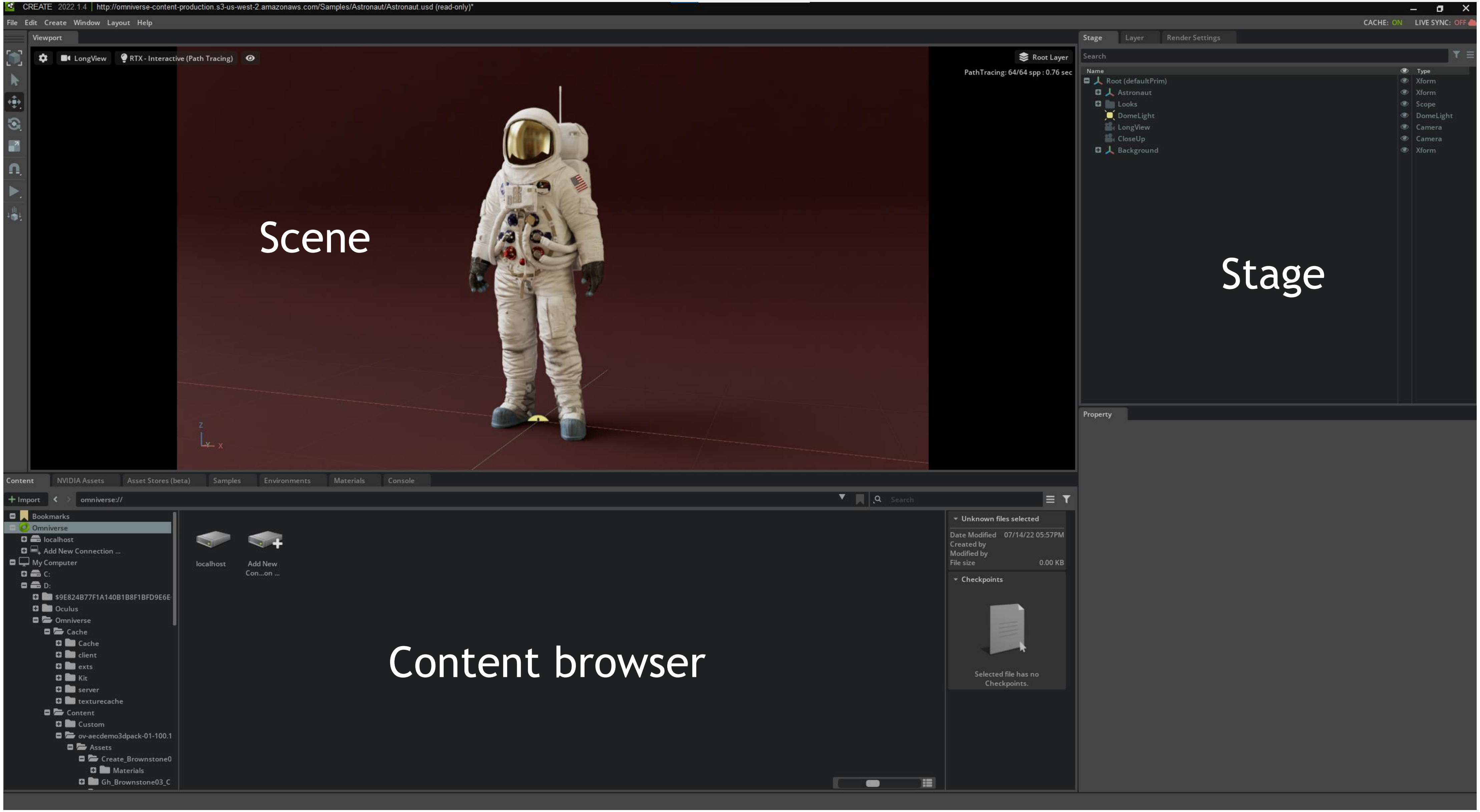


Server: 34.215.96.56

Resolution: 1920 × 1080 (FHD)

- Start Omniverse Launcher on your PC.
- Launch Omniverse Streaming Client.
- Once your course is launched, execute all the cells in the Jupyter notebook – this may take a while!
- Input the IP address of the server in your Omniverse Streaming Client and click Connect.
- Adjust the screen resolution now if you have to – it is harder to do so later.

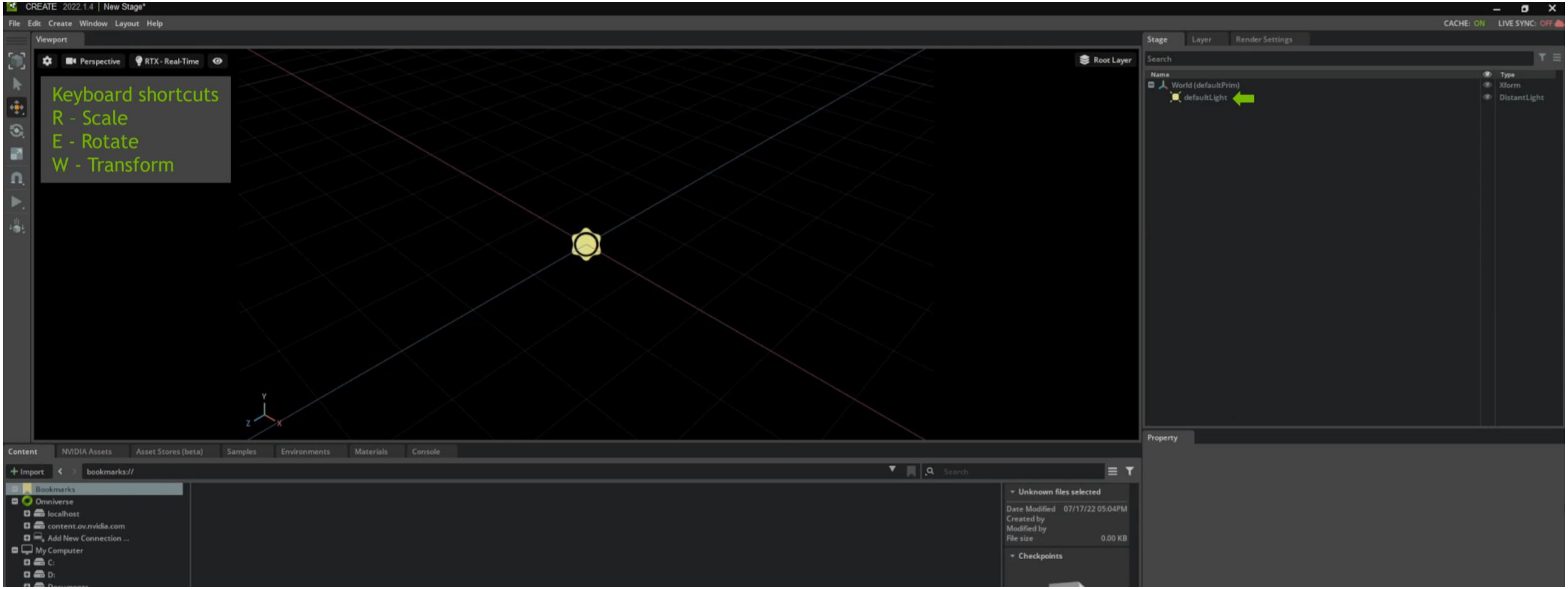
CREATE INTERFACE



Date Modified 07/14/22 05:57PM
Created by
Modified by
File size 0.00 KB

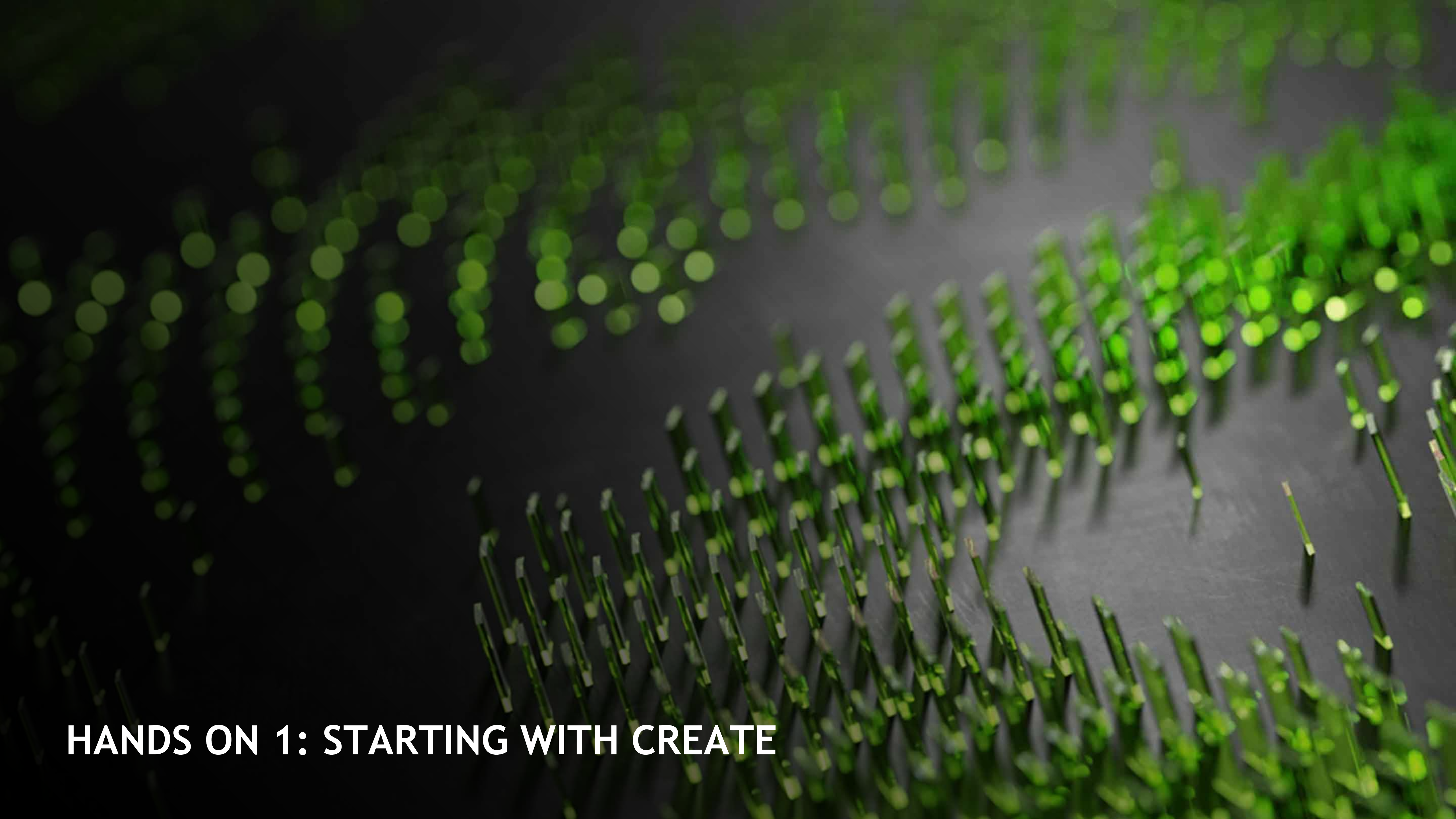
NAVIGATING IN CREATE

https://docs.omniverse.nvidia.com/app_create/prod_extensions/ext_viewport/navigation.html



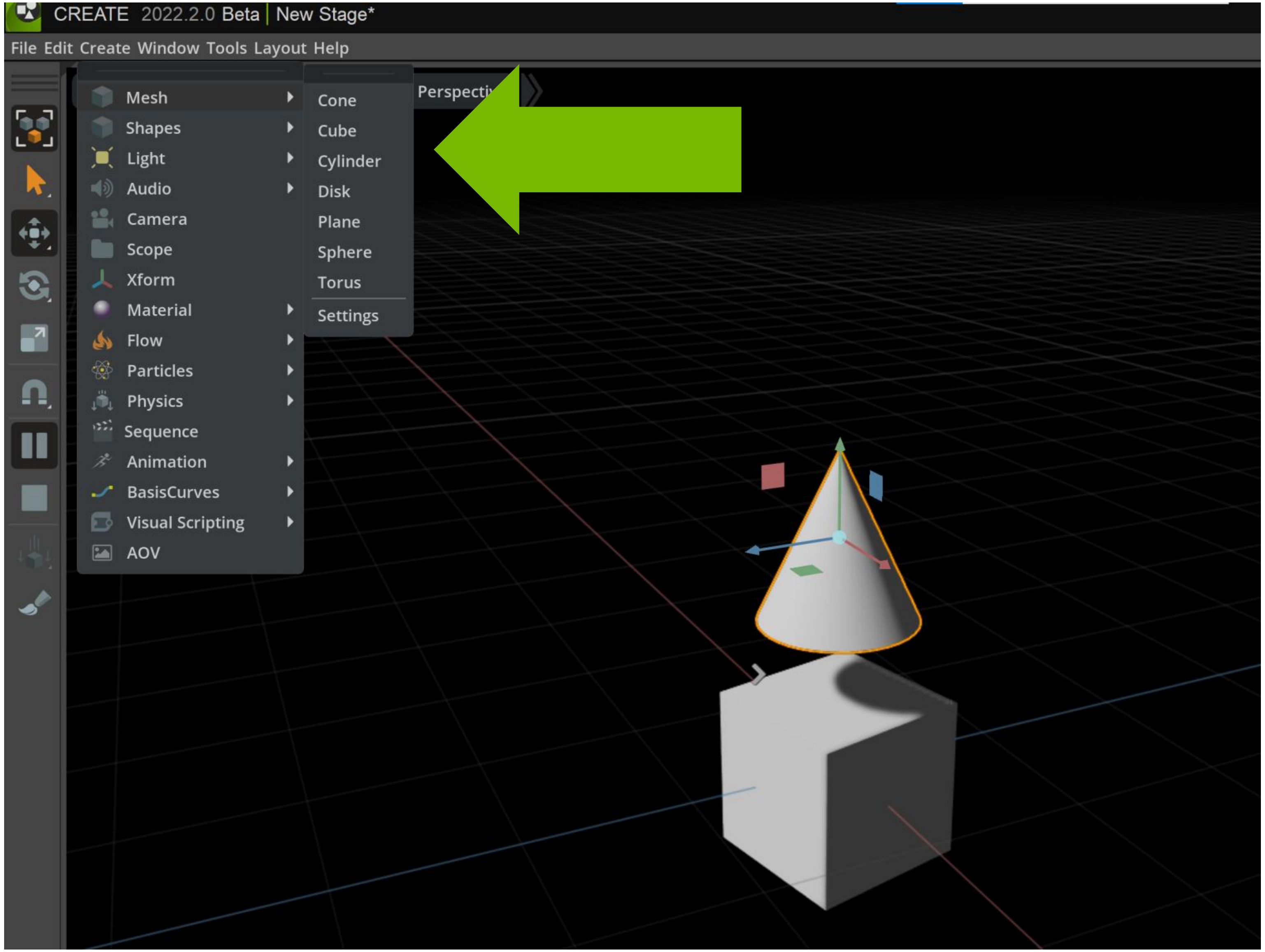
Keyboard shortcuts
R - Scale
E - Rotate
W - Transform

Date Modified 07/17/22 05:04PM
Created by
Modified by
File size 0.00 KB

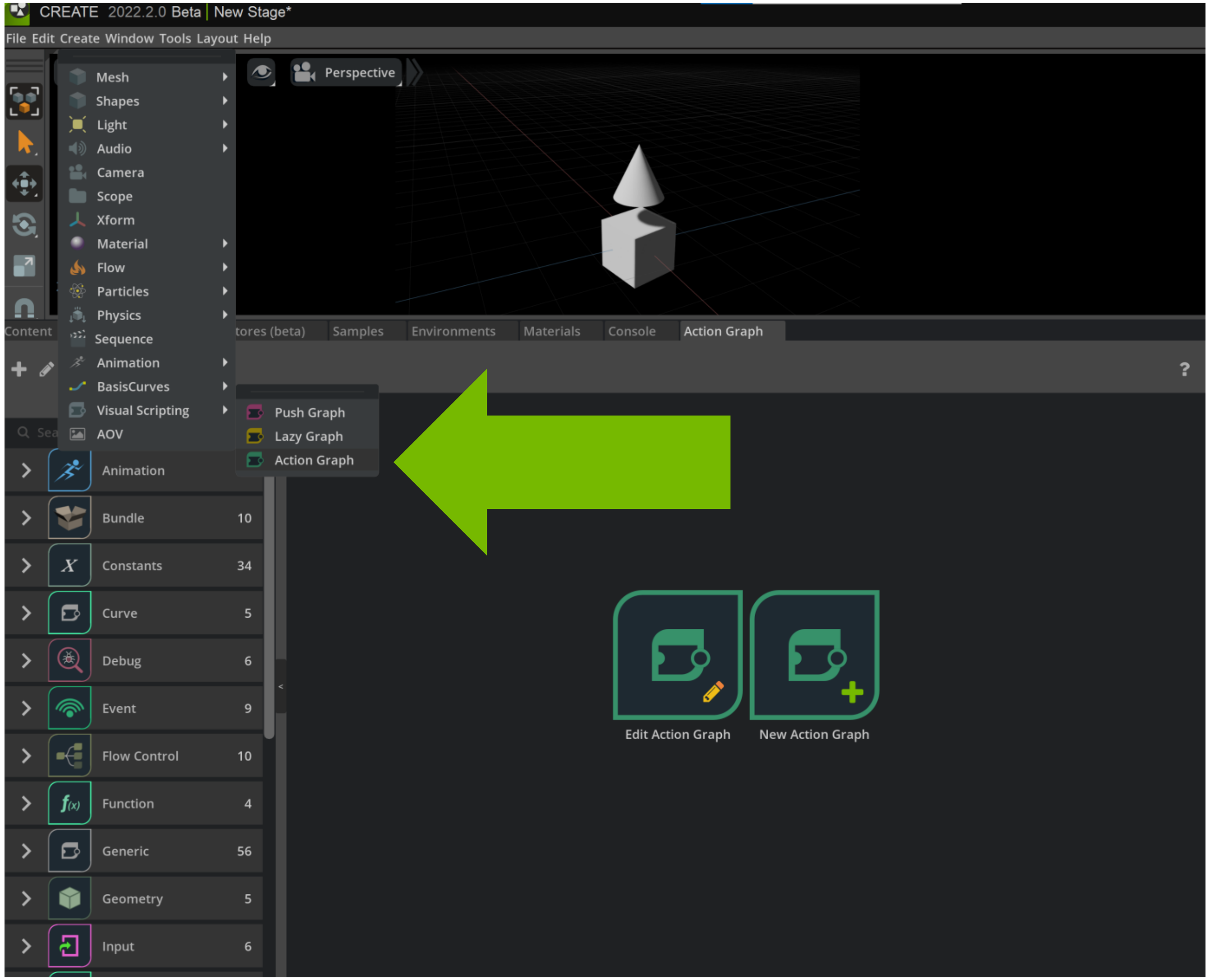


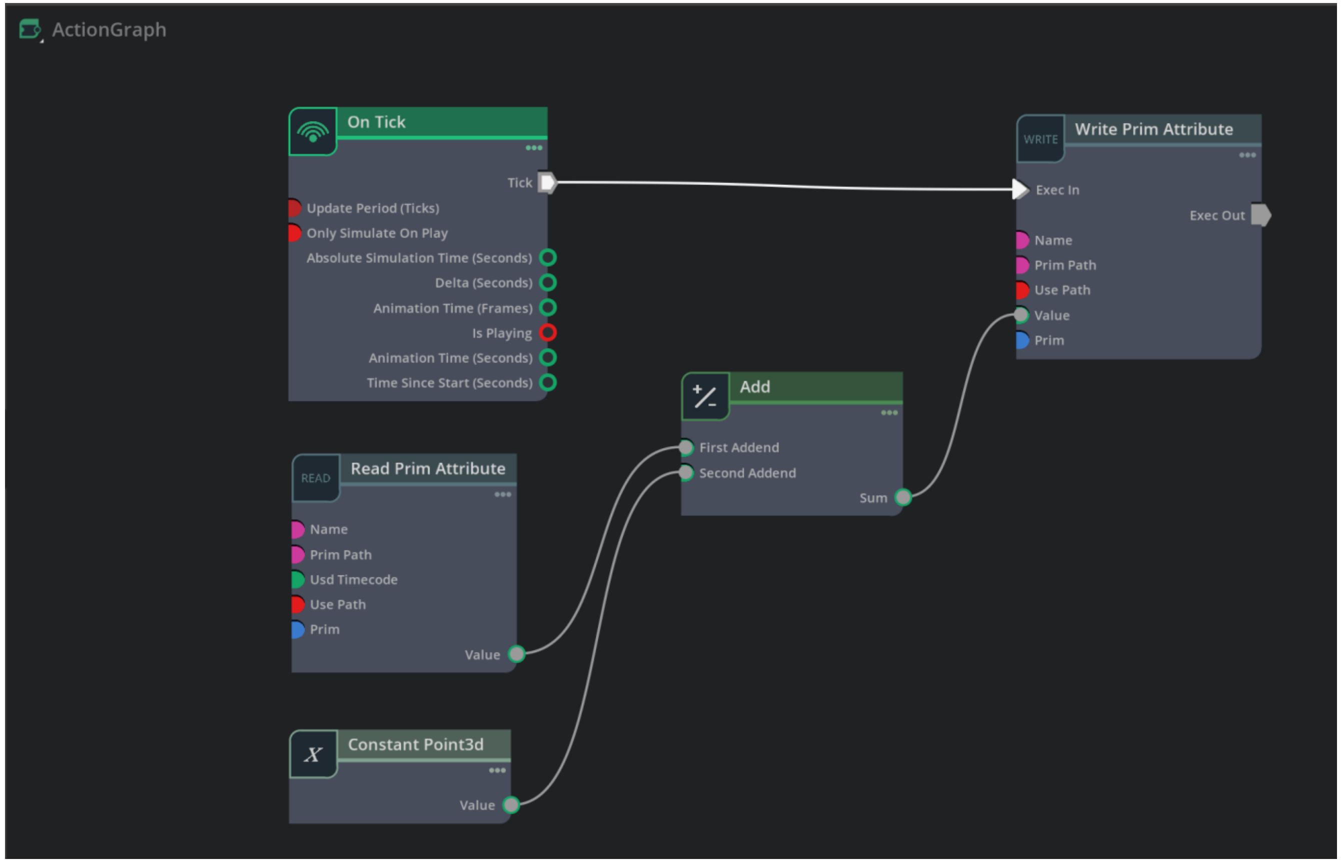
HANDS ON 1: STARTING WITH CREATE

CREATING OBJECTS WITH OMNIVERSE CREATE



CREATING ACTION GRAPHS TO TRIGGER ACTIONS



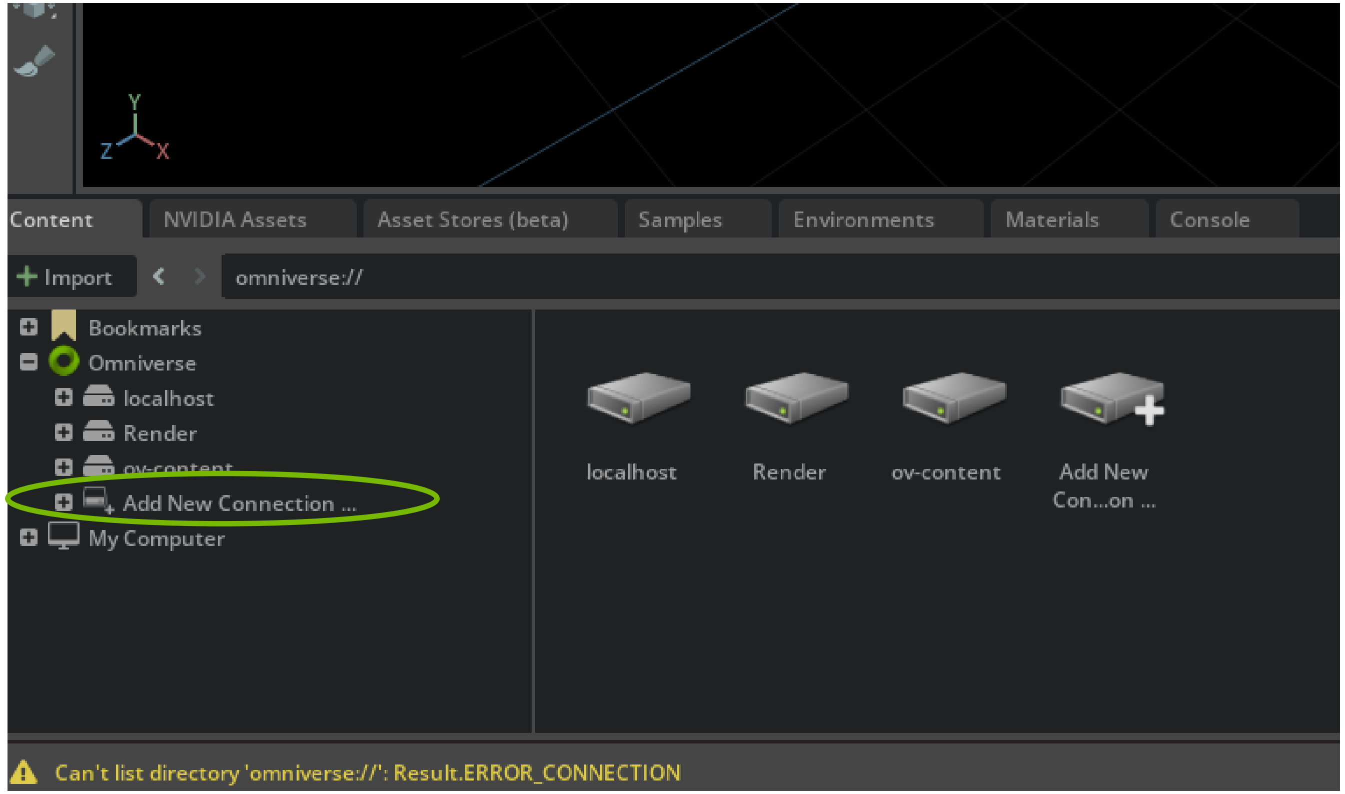


ACTION GRAPH NODES

- Try to find these nodes by searching in the search bar on the left.
- Drag the nodes onto the screen and connect outputs together as shown.
- Don’t forget to attach our object and its attribute (translate) to the Write Prim Attribute node!

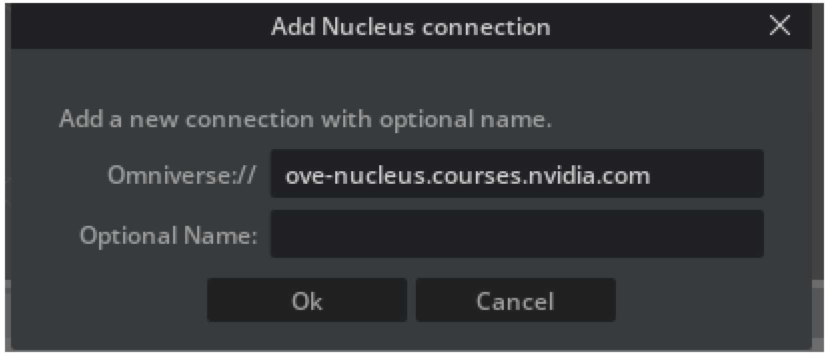
HANDS ON 2: AIRCRAFT MOVEMENT ACTION GRAPH

STARTING NUCLEUS SERVER



omniverse://

⚠ Can't list directory 'omniverse://': Result.ERROR_CONNECTION


Add Nucleus connection

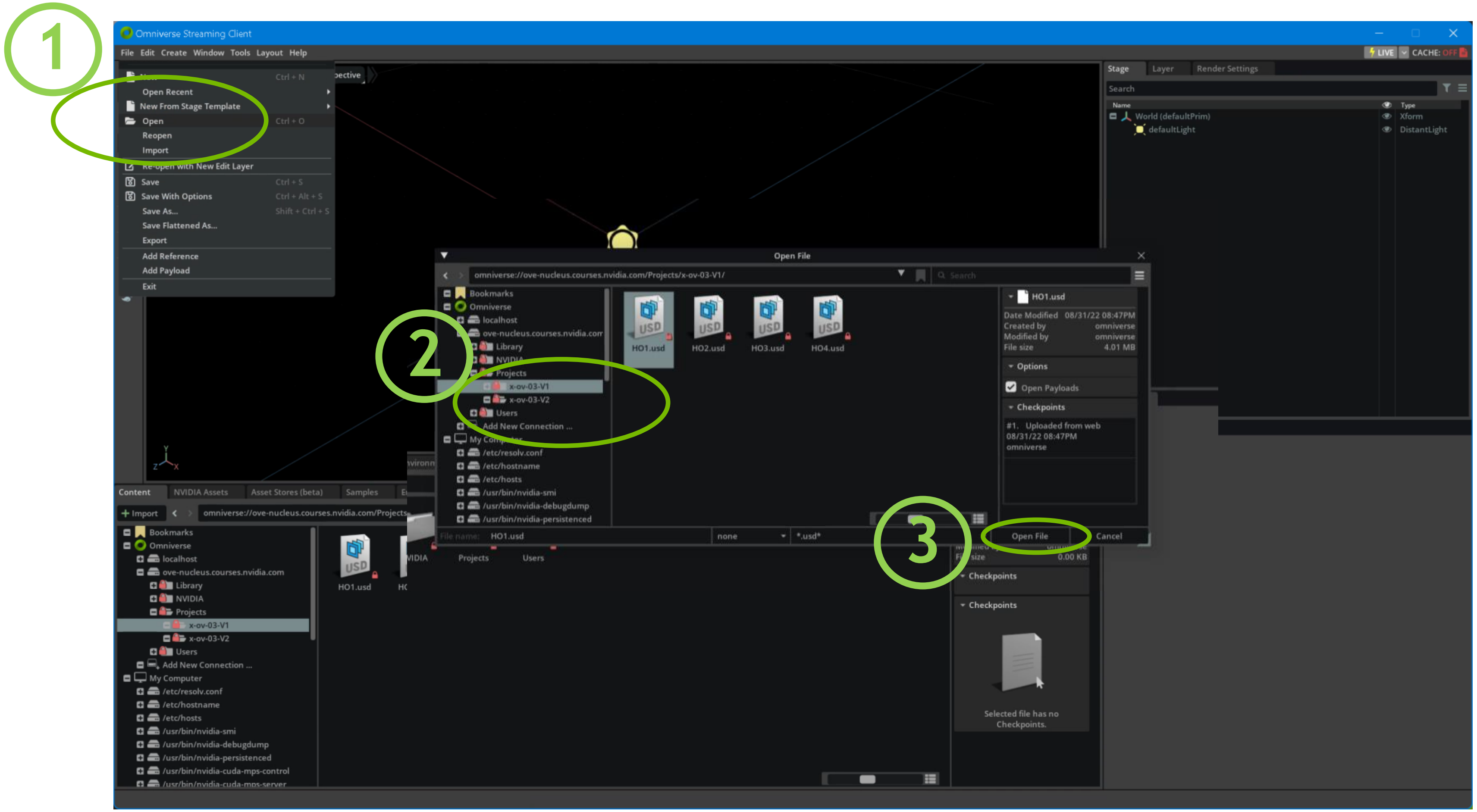
Add a new connection with optional name.

Omniverse:// ove-nucleus.courses.nvidia.com

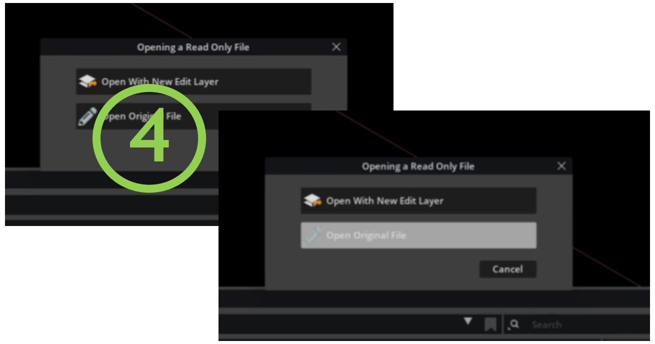
Optional Name:

- You should see an instance of Omniverse Create start up.
- Under Content-Add New Connection, enter ove-nucleus.courses.nvidia.com and connect to the server.
- Open file HO1 under **Projects/x-ov-03-V2**.

OPEN CONTENTS



Open file HO1 under **Projects/x-ov-03-V2**.



Step1.
[File] >> [Open]

Step2.
Navigate [H01] file and select

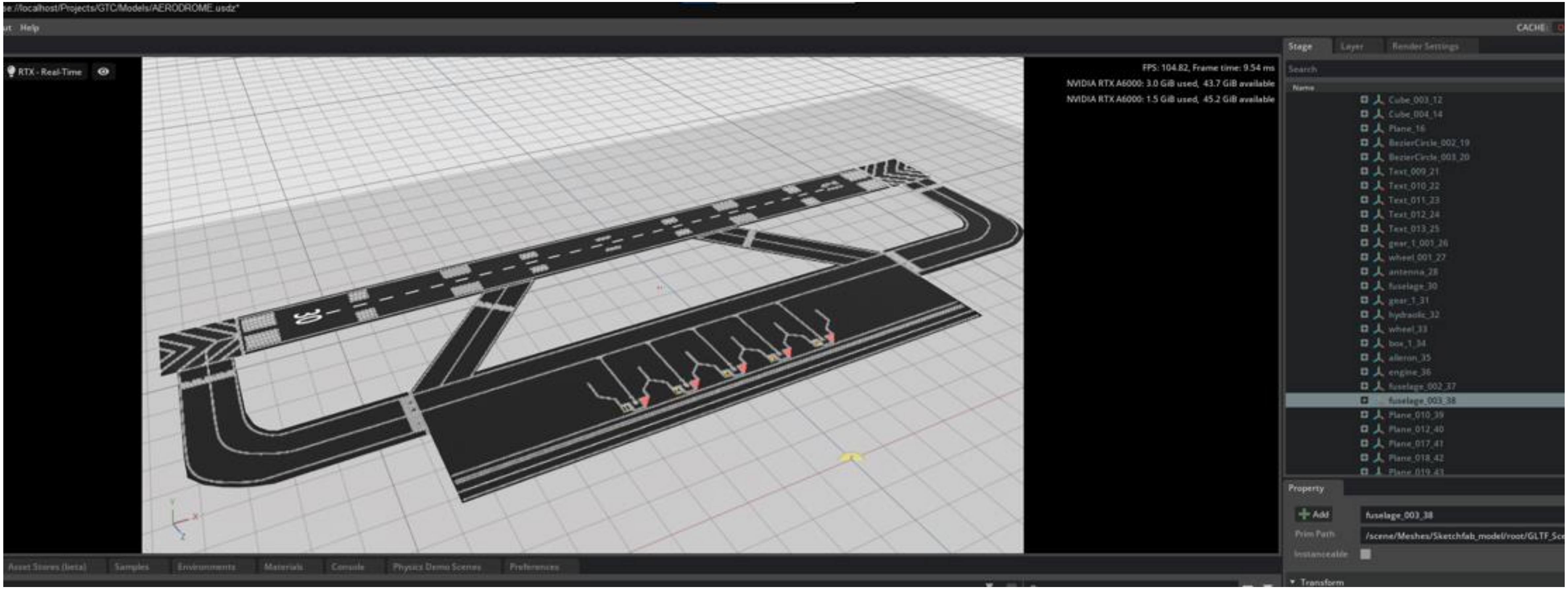
Step3.
Click [Open File]

Step4.
Select [Open Original File]!

File H01

SETTING UP THE SCENE IN CREATE

Import assets into Create



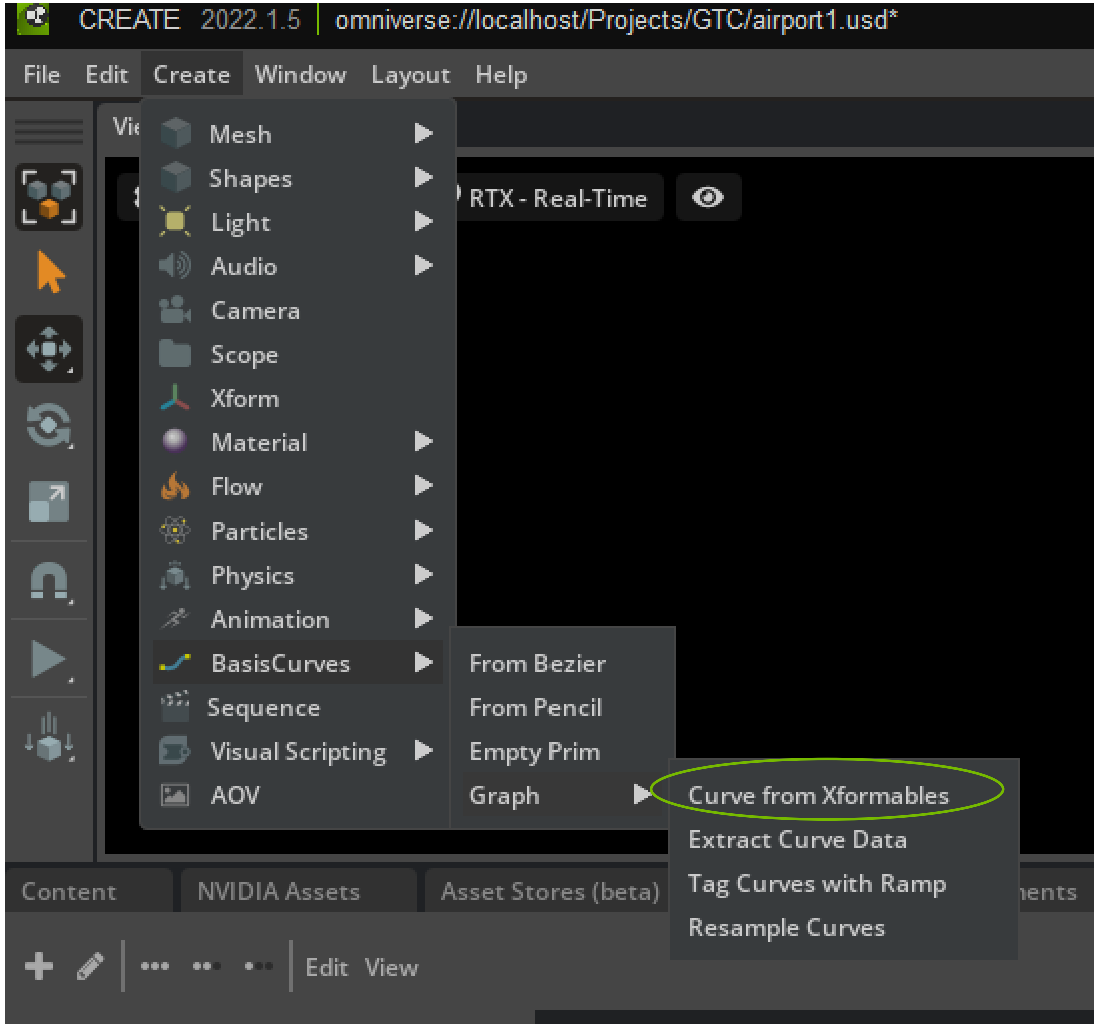
se://localhost/Projects/GTC/Models/AERODROME.usdz*

FPS: 104.82, Frame time: 9.54 ms
NVIDIA RTX A6000: 3.0 GiB used, 43.7 GiB available
NVIDIA RTX A6000: 1.5 GiB used, 45.2 GiB available

- Cube_003_12
- Cube_004_14
- Plane_16
- BezierCircle_002_19
- BezierCircle_003_20
- Text_009_21
- Text_010_22
- Text_011_23
- Text_012_24
- Text_013_25
- gear_1_001_26
- wheel_001_27
- antenna_28
- fuselage_30
- gear_1_31
- hydraulic_32
- wheel_33
- box_1_34
- aileron_35
- engine_36
- fuselage_002_37
- fuselage_003_38
- Plane_010_39
- Plane_012_40
- Plane_017_41
- Plane_018_42
- Plane_019_43

fuselage_003_38
Prim Path /scene/Meshes/Sketchfab_model/root/GLTF_Sce
Instanceable □

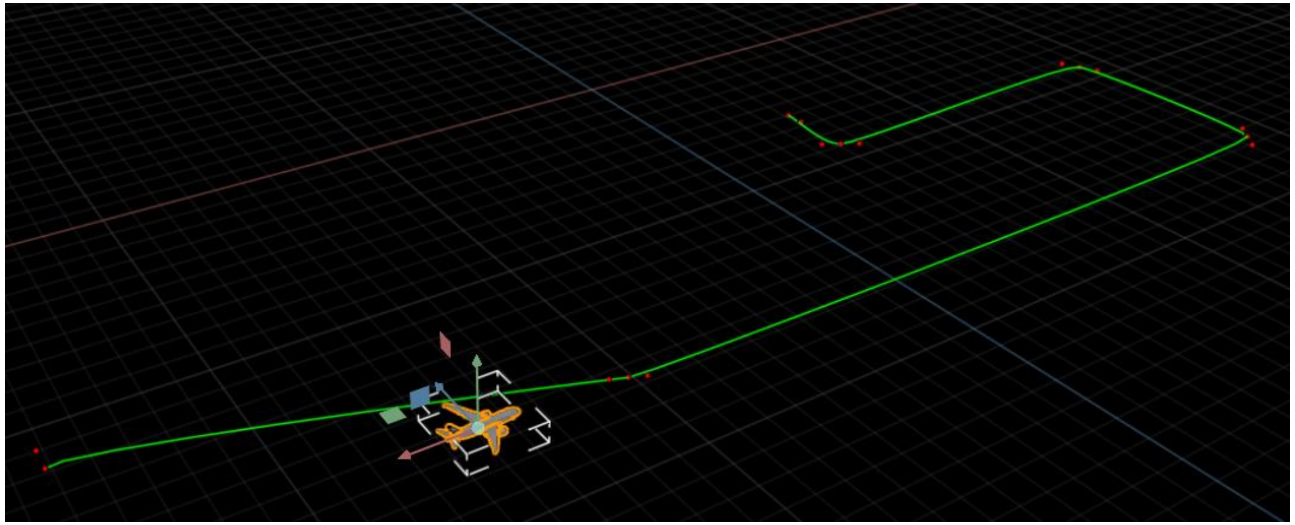
File H02



DRAW A CURVE FOR AIRCRAFT PATH

- Add points on the runway as new Xforms
- To transform the points into a single curve, select all the newly created Xforms in the Stage.
- Go to Create – BasisCurves – Graph – Curve from Xformables.

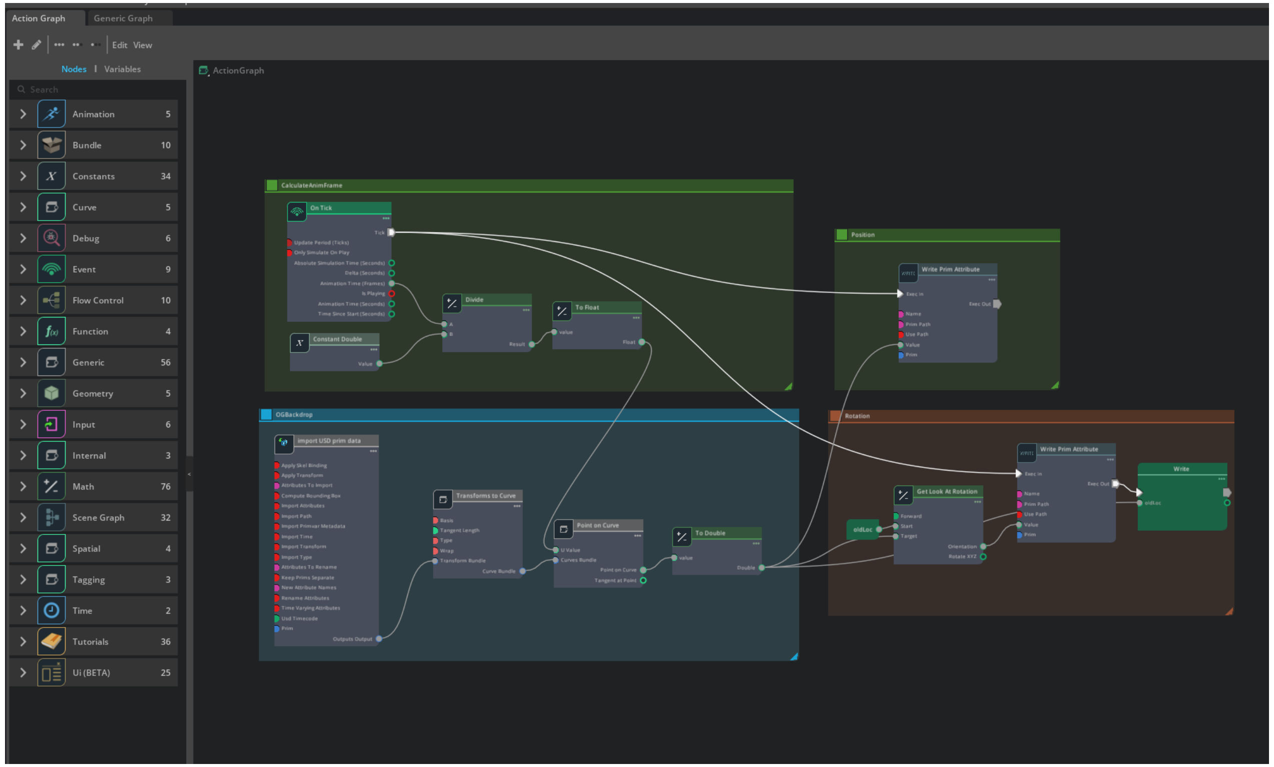
File H02



VISUALIZE AND EDIT THE CURVE

- Open the newly created Push Graph from the Stage.
- To visualize the curve, enable Curve visualizer from the Push Graph.
- Adjust the tangent length and rotate each tangent by transforming the respective Xform points on the curve, until your curve looks the way you want it to.

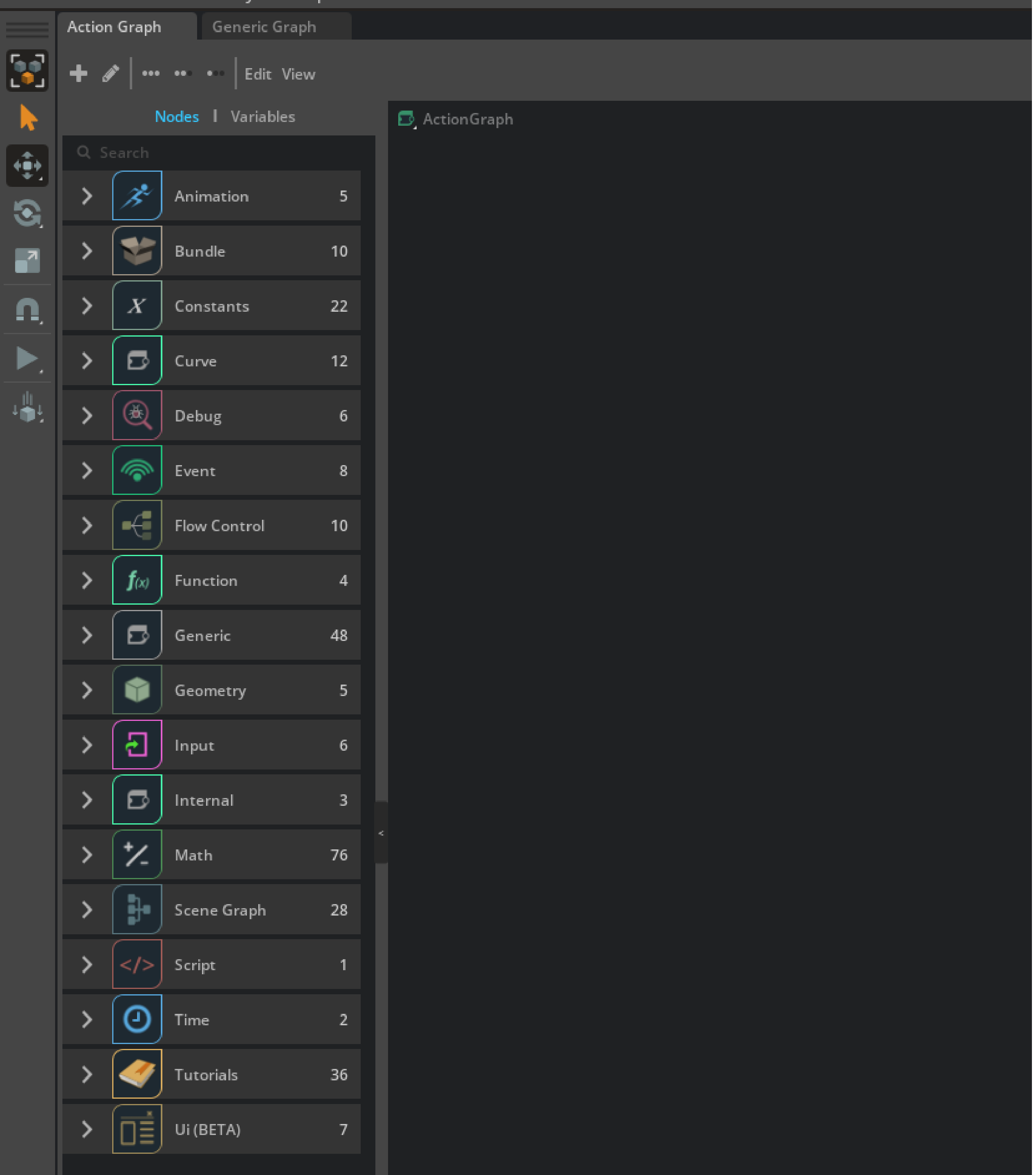
File HO3



MOVING THE AIRCRAFT ALONG THE CURVE

- To link our aircraft to follow the curve, we want to change its Xform:translate values to go through consecutive points on the curve at every frame.
- We need an Action Graph to simulate this movement.

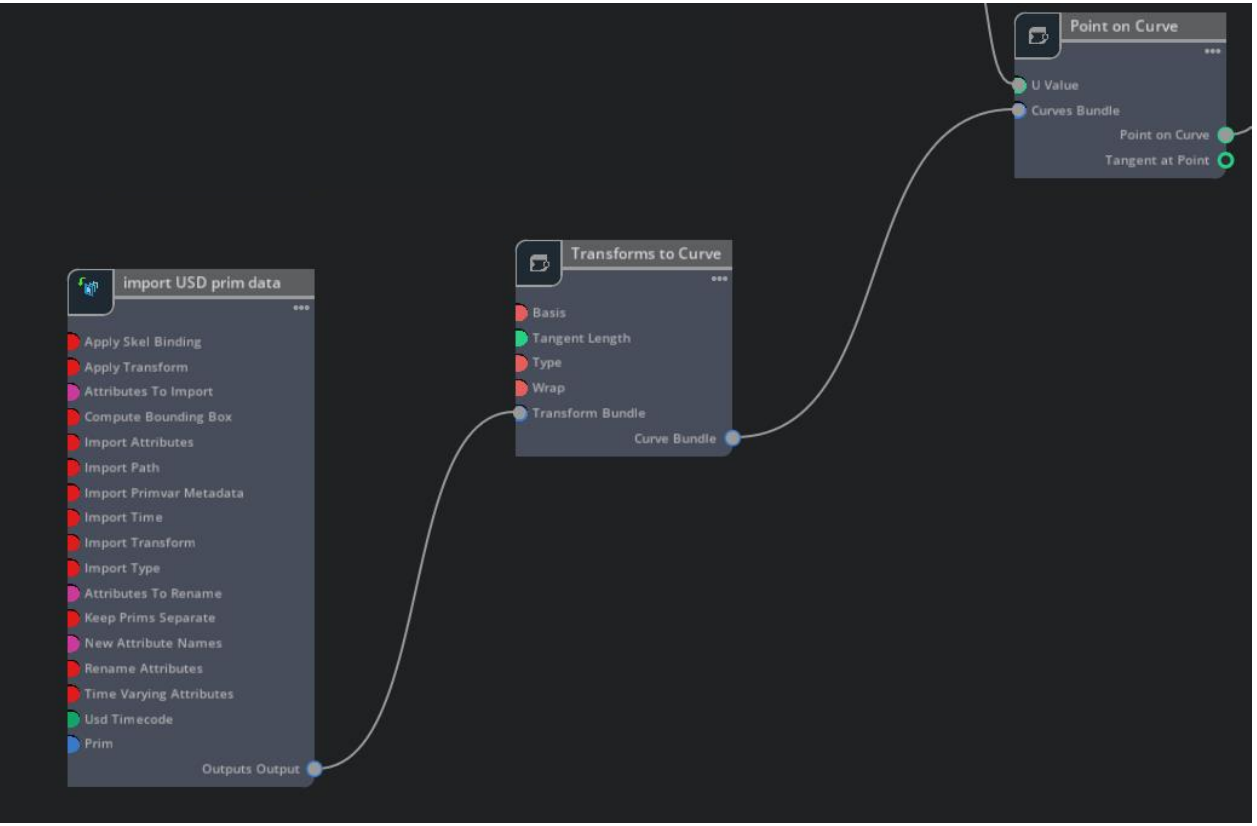
Action Graphs are a part of OmniGraph, the compute engine of Omniverse. Action graphs allow event driven behaviour for Omniverse entities/prims.



File H03

USING ACTION GRAPHS TO SIMULATE MOVEMENT

- Ensure that the following extension bundle is loaded: *omni.graph.bundle.action*
- Open up the graph editor UI by clicking on Window -> Visual Scripting -> Action Graph
- In the new window, click on New Action Graph to create an empty action graph at the default path
- Let’s name it AircraftController.



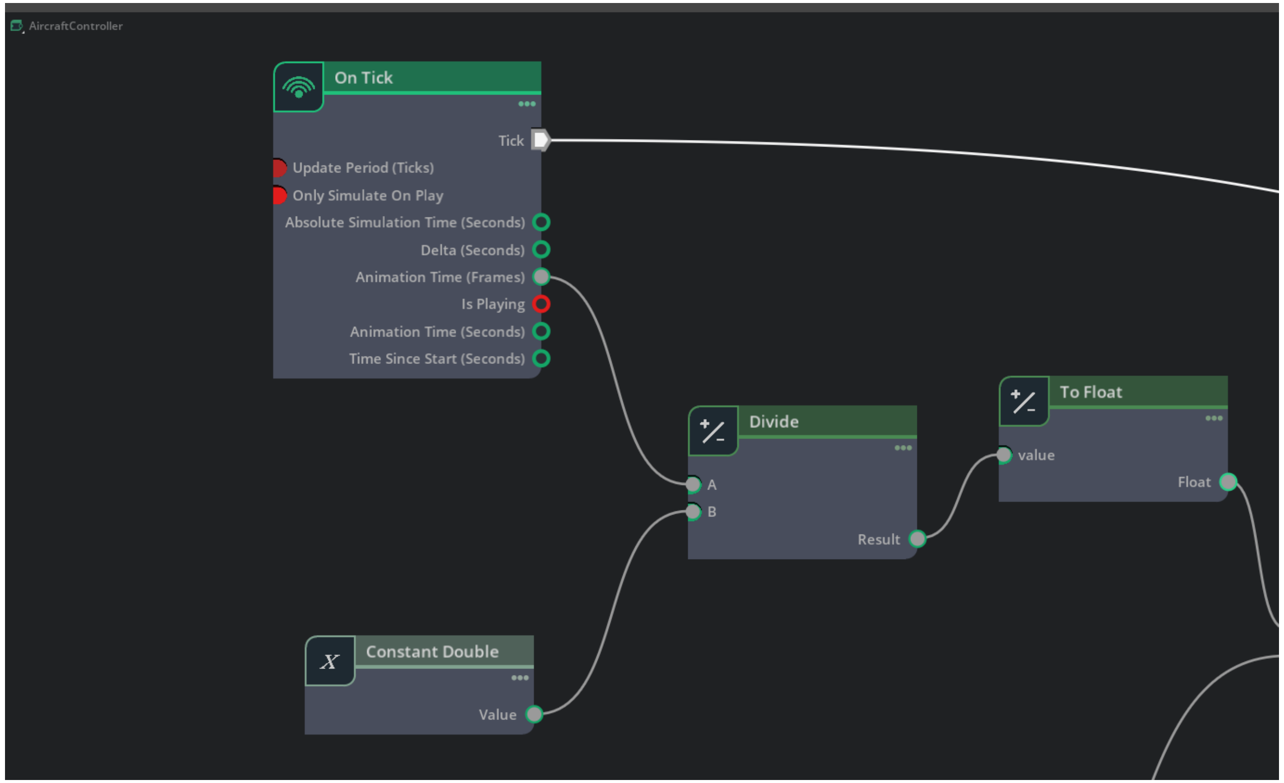
File H03

MOVING THE AIRCRAFT WITH ACTION GRAPHS

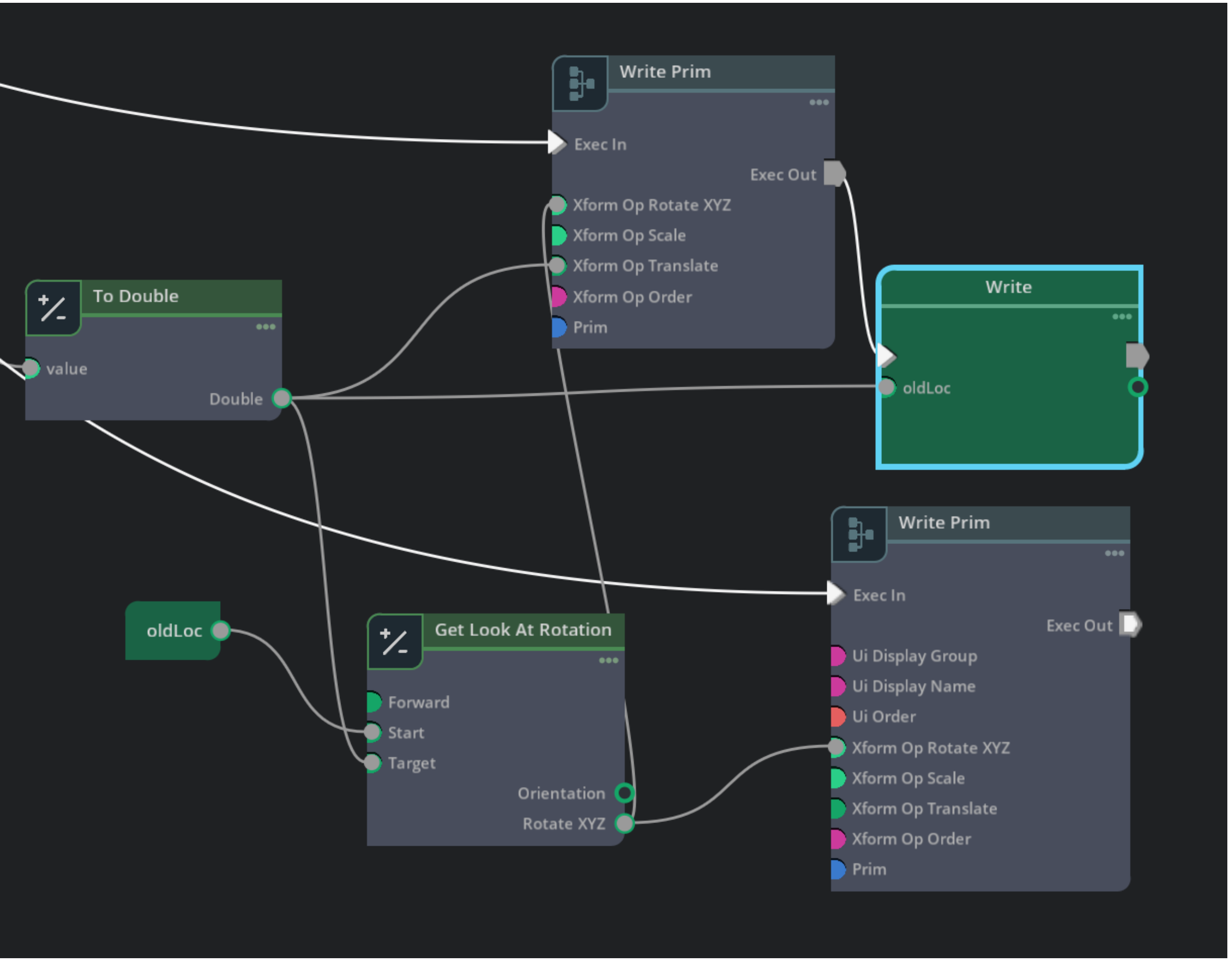
- Add curves to the graph with **Import USD prim data** node
- Select our curve Xform prims.
- Link its Outputs node to Transform Bundle on a new **Transforms to Curve** node.
- Now we select a point on the curve using the **Point on Curve** node.
- We feed it to the transform of the aircraft. For this, create a new **Write Prim** node. Add our aircraft prim as the target in the property window and connect Point on Curve to its location transform.

File H03

MOVING THE AIRCRAFT WITH ACTION GRAPHS



- We need a U value for the curve, to tell the curve how much to move when on it.
- Drop down an **On Tick** node. This can be done using the search box on the left.
- The On Tick node triggers actions at every clock tick –i.e. at every time!
- **Divide** the time by the number of frames (in my case it’s 3000), and convert that to float using **To Float**.



File HO3

MOVING THE AIRCRAFT WITH ACTION GRAPHS

- Now if we play we see the aircraft follow the path, but it does not rotate correctly.
- Add Get Look at Rotation and a new variable OldLoc to store the previous rotation, which connects to the Start attribute of Get Look at Rotation.
- Connect the previous output from To Double to the Target attribute.
- The Orientation output should connect to Write Prim’s Orient.
- Now we can see the aircraft should follow the curve we chose, defined by a few Xform points.

SEE YOU IN OMNIVERSE!