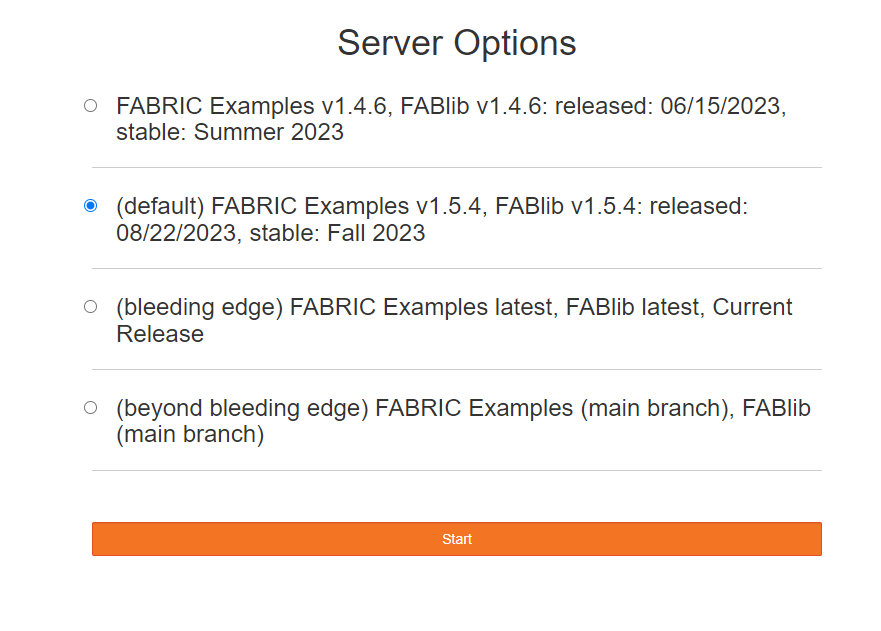
**This is going to be the second step after creating your account.**

**– Open the JupyterHub using your fabric testbed.**

**– Use the default option and sign in with CLlogon to your account.**

****

**Start with start\_here.ipynb in the latest jupyter example folder**

* + 1. **Open start\_here.ipynb. click on configure Environment.**

FIRST TIME SETUP

1. This will open **configure.ipynb** -> copy the bastion username and paste it for FABRIC\_BASTION\_USERNAME [bastion user name can be found on the user profile page in the FABRIC portal (click "My SSH Keys")].   
   We need to create a bastion key. To create Bastion key go to FABRIC Portal -> Experiments -> manage SSH keys -> click on Bastion -> scroll down to generate Bastion key pair

Write name (fabric\_bastion\_key) and description (can be anything) -> click Generate.

Download private and public keys after generating. (private key is important to download, it can’t be downloaded again)

Change FABRIC\_PROJECT\_ID -> the project ID can be found on the projects tab in the FABRIC portal

1. Go to **fabric\_config folder** - > upload the private key which was fabric\_bastion\_key.
2. Go back to **configure.ipynb**. Set the path FABRIC\_BASTION\_PRIVATE\_KEY\_LOCATION. To find the location, go to the terminal. Type following commands  
   Cd fabric\_config/ (to open folder fabric\_config)  
   realpath fabric\_bastion\_key ( to get the path of fabric\_bastion\_key)
3. Copy the path and paste after export FABRIC\_BASTION\_PRIVATE\_KEY\_LOCATION=${HOME}.
4. Run each cell in the configure python file.

Do not run “create a Downloadable Package that Deploys SSH Tunnels” - it throws error

2. IF YOU HAVE NOT USED FABRIC IN A WHILE, You need to edit your config file

Configure.ipynb // set up your configuration file called fabric\_rc

To run Configure.ipynb, you will need the <YOU\_BASTION\_USERNAME>', '<YOUR\_PROJECT\_ID>'

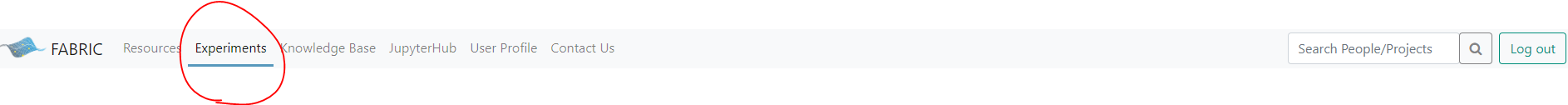
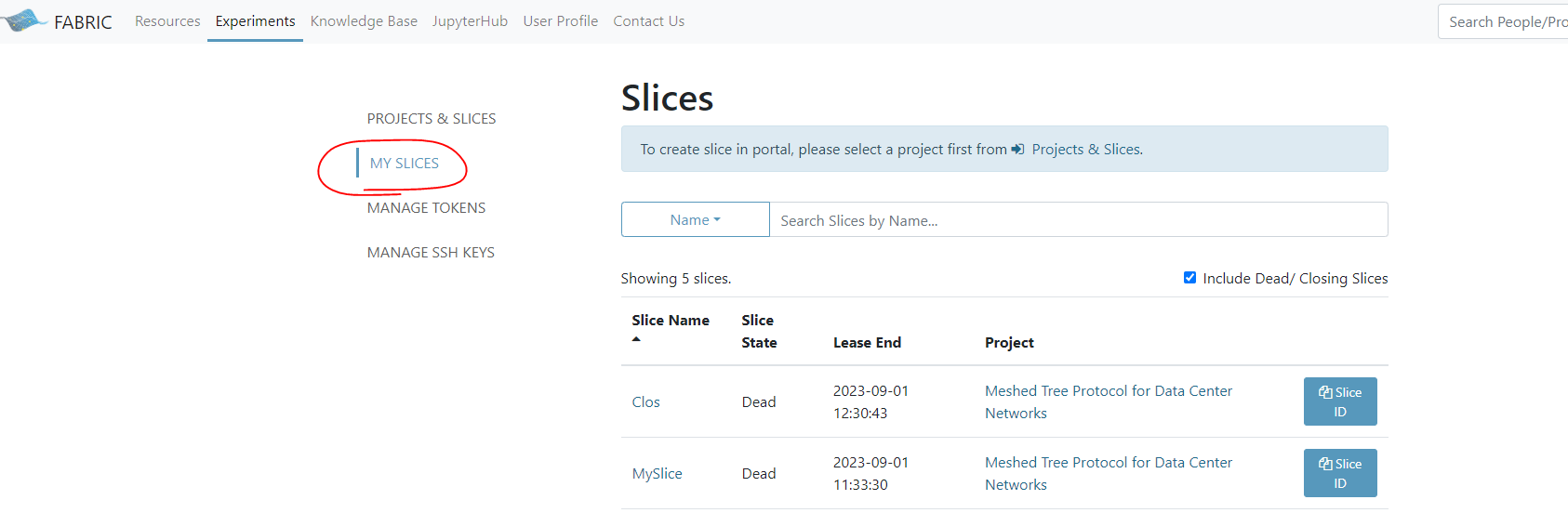
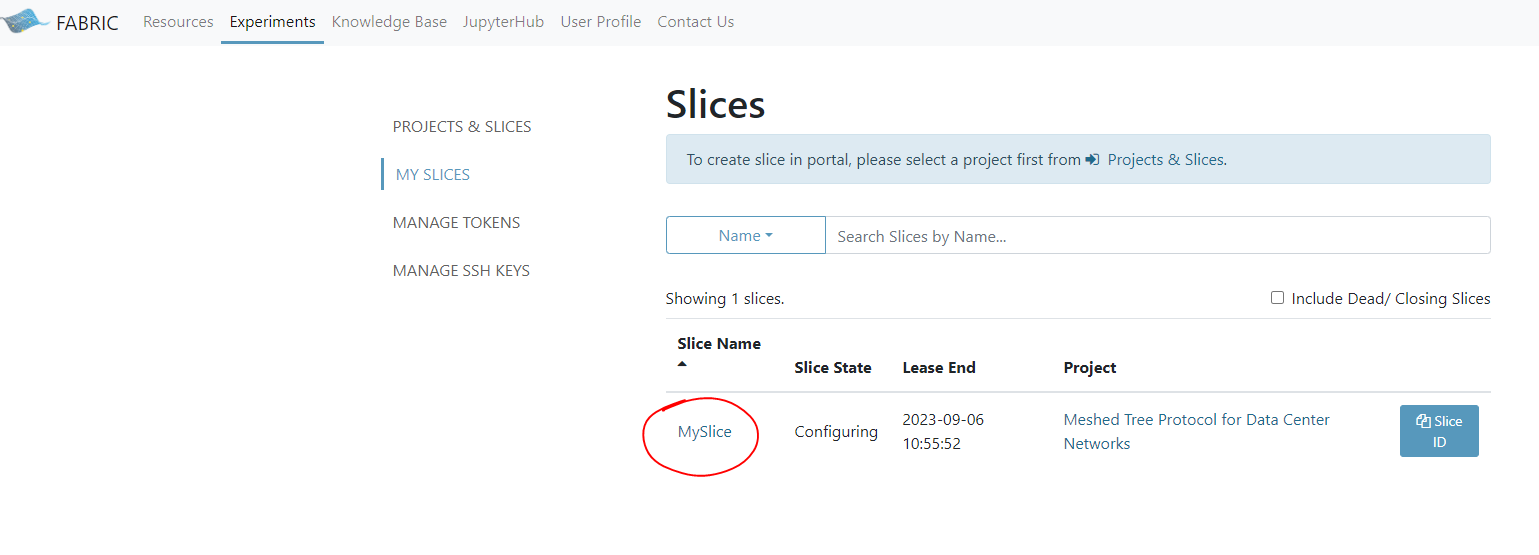
To get YOU\_BASTION\_USERNAME> go to the user profile. Click on NS(your name initials) on the top right corner.

My username is nxsvks\_0000031803

My project ID fec0d0d8-a7a8-4eac-b091-87f7914af796 for Meshed Tree Protocol

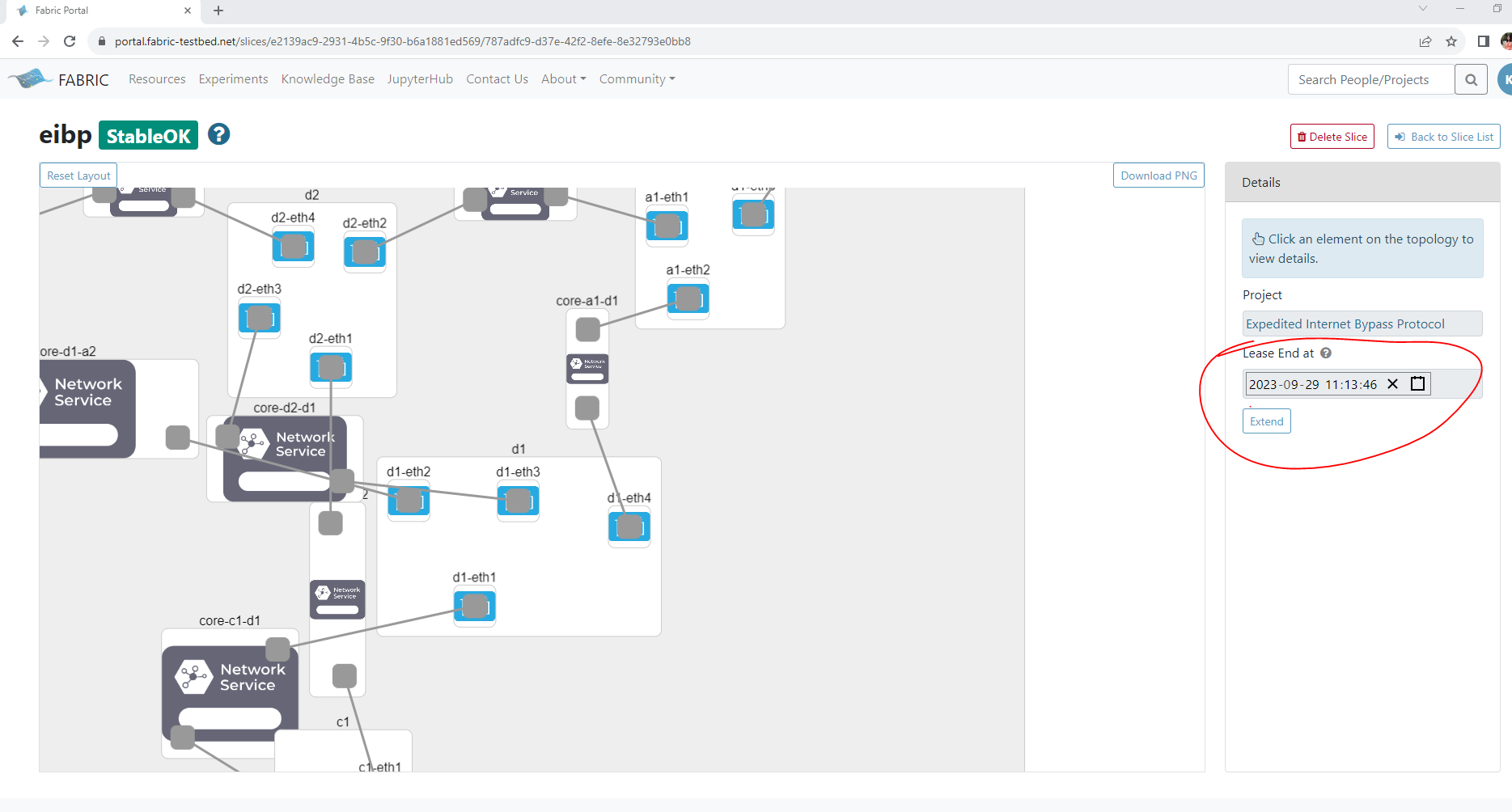
Project ID for Expedited Internet Bypass Protocol is 787adfc9-d37e-42f2-8efe-8e32793e0bb8

1. Go back to **start\_here.ipynb** file -> under heading First experiment click "Hello, FABRIC" to get to that book (hello\_fabric.ipynb). After opening hello\_fabric.ipynb. Run   
   -> Step 2: Import the FABlib Library  
   -> Step 3 (Optional): Query for Available Testbed Resources and Settings  
   -> Step 4: Create the Experiment Slice (This cell will create a slice, add a node and Submit the slice……the submit function will block until the node is ready and will display the progress of your slice being built). You may get an error if you don't specify the site. Include site =’MASS’.   
   To check if the slice is created go to FABRIC portal -> Experiments -> MY SLICES

  
then go to My Slices   
  
Click slice name  


This is the view after clicking on Slice name.

We can extend the slice lifetime – on the fabric portal. By clicking on the Slice name -> on the right side, select the date till you want to extend - click EXTEND.



OPEN GOOGLE DRIVE :

We can also use Peter’s script To increase the slice lifetime.

Go to EIBP 2023 > EIBP Intra\_AS > FABRIC-EIBP > Fabric-EIBP-Jupyter\_scripts (developed by Peter)

Go to local\_books. Open EIBPStart.ipynb. we have to be in the local\_books directory.

Copy the cell with Access the slice (change the name of the slice before copying and put it in “”)(Give the name of the slice you created in hello\_fabric.)

Go back to your terminal.

First time, click the + button on the tabs. Open a terminal. Cd to your directory.

Type python (case sensitive)– to enter interactive shell, copy paste the copied cell.

Give the name of the slice you created in hello\_fabric. It may take a couple of minutes.

Type manager.renewSlice(days) //days should not be more than 14 days

-> next step run Step 5: Observe the Slice's Attributes   
-> then run Step 6: Run the Experiment // we can upload any code on the node and test. The example from fabric allows you to print Hello Fabric.  
-> Step 7: delete the slice after you are done.

**How to load code and test on a single node**

In the hello fabric where you created a slice, locate the ssh command to connect to the node you created in fabric.

\*\*\*\*\*\*\*\*\*\*\*

If you do not want to go back to hello fabric. Open Peter’s FabUtils.py. Copy the saveSSHCommands() without the self – create manager.saveSSHCommands(). Paste in your terminal where you started a python interactive session. This will generate the ssh command txt file in the default directory.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

To upload code into the FABRIC node, we do not have a fabric utility or script. There are 2 ways-

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Upload the files from your local machine to your folder. We can use the code from MTPCode.ipynb – (ipynb – interactive python notebook) called manager.uploadDirectoryParallel(CODE\_DIR) where the CODE\_DIR- is the path to the C code). do not close after you execute the ‘access slice’ cell. This should also work.
2. We can run it in the notebook itself MTPCode.ipynb. Start from the first cell to update the slice name, directory etc. (The slice and directory info was populated when the first cell was executed).

Before compiling we should have executed MTPDeps- this will download all the tools like gcc compiler on all the nodes of the slice or you should have downloaded development tools on the node by using following commands-

**To download development tools run sudo dnf groupinstall -q -y "Development Tools". This downloads all development tools. To download only gcc related use the following command**

**sudo dnf install gcc**

To compile the code – we can fix the entries in the cell in MTPCode.ipynb. Or use the following command.

gcc –g –o Tor main.c /// to gcc compile // type command - - **COPYING FROM WORD DOES NOT HELP.**

\*\*\*\*\*\*\*\*\*\*\*\*