Google

Colaboratory

Local runtimes

Cla boat o l es ou connrect oa l oal r utnim eusing Jupt e. Thisall ow s out o exeout e cod e on our l oal lardwar ea ch la vea ccesst o our l oal lfi es st em.

Security considerations

Mak esur e outr ust hea ut hos of a n rob ebook befo e exeout ing ti. Wh l oal connrect io,rt he cod e ou exeout e a n ead, wr ti,ea chd le te lfi es on our com put e.

Connrect ingt oa Jupt e rob ebook se ver u nning on our l oal ma chine a np ovd ema n be refi s. Wh t hese be refi s com ese ious pb etrail r isk s. B connrect ingt oa l oal r utnim e ouar eall ow ingt he Cla boat o rf otnecht o exeout e cod e ith he rob ebook usingt hel oal r esour ces on our ma chine. Thism a nst latt he rob ebook could:

- Invok ear bitrar comma chs(i e"`rm -rf /`")
- Accesst hel oal lfi es st em
- Ru nmal icious cotnetn on our ma chine

Befo eatt emp ingt o connrect oa l oal r utnim e,ma k esur e outr ust hea ut hos of he rob ebooka ch ensur e ou u dhe sta chw lat cod e is being exeout ed. Fo m o e inf omat io no th eJupt e rob ebook se ve 's secur ti m d le, consult Jupt e 'sd ocum etat io n

Setup instructions

Inod et oall ow Cla boat o t o connrect o our l oall r u nning Jupt e se ve, oull' need t o pe fom t he ftb ow ing st eps.

Step 1: Install Jupyter

Inst all Jupt e on our l oal ma chine

Step 2: Install and enable the `jupyter_http_over_ws` jupyter extension (one-time)

The `jupyter_http_over_ws` extension is authored by the Chrome team and is available on GitHub.

```
pip install jupyter_http_over_ws
jupyter serverextension enable --py jupyter_http_over_ws
```

Step 3: Start server and authenticate

New notebook servers are started normally, though you will need to take advantage of expanded trusts WebSocket connections from the Chrome origin.

```
jupyter notebook \
  --NotebookApp.allow_origin='https://colab.research.google.com' \
  --port=8888 \
  --NotebookApp.port_retries=0
```

Once the server has started, it will print a message with the initial token used for authentication. Make a copy of this URL as you'll need to provide this in the next step.

Step 4: Connect to the local runtime

In Chrome, click the "Connect" button and select "Connect to local runtime...". Enter the URL from the previous step in the dialog that appears and click the "Connect" button. After this, you should now be connected to your local runtime.

Browser-specific settings

Note: If you're using Mozilla Firefox, you'll need to set the `network.websocket.allowInsecureFromHTTPS` preference within the Firefox configuration. Chrome makes connections to your local kernel using a WebSocket. By default, Firefox disallows connections from HTTPS domains using standard WebSockets.

Sharing

If you share your notebook with others, their runtime on your local machine will be shared. When others open the shared notebook, they will be connected to a standard Colab runtime by default.

By default, all code cell outputs are stored in Google Drive. If your local connection will access sensitive data and you would like to omit code cell outputs, select *Edit > Notebook settings > Omit code cell output when saving this notebook*.

Uninstallation

You can disable/enable/remove the `jupyter_http_over_ws` jupyter extension by running the following:

```
jupyter serverextension disable --py jupyter_http_over_ws
pip uninstall jupyter_http_over_ws
```

Connecting to a runtime on a Google Compute Engine instance

If the Jupyter notebook server you'd like to connect to is running on a remote machine (e.g. Google Compute Engine instance), you can setup SSH local port forwarding to allow Colaboratory to connect to it.

Note Google Cloud Platform provides Deep Learning VM images with Colaboratory local backend support pre-configured. Follow the how-to guide to setup your Google Compute Engine instance with local SSH port forwarding. If you use these images, skip directly to Step 4: Connect to the local runtime (using port 8888).

First, setup your Jupyter notebook server using the instructions above.

Second, establish an SSH connection from your local machine to the remote instance (e.g. Google Compute Engine instance) and specify the '-L' flag. For example, to forward port 8888 on your local machine to port 8888 on your Google Compute Engine instance, run the following:

```
gcloud compute ssh --zone YOUR_ZONE YOUR_INSTANCE_NAME -- -L 8888:localhost:8888
```

Finally, make the connection within Colaboratory to connect to the forwarded port (follow the same instructions as those in Step 4: Connect to the local runtime).