



## Colaboratory

# Local runtimes

Colaboratory lets you connect to a local runtime using Jupyter. This allows you to execute code on your local hardware and have access to your local file system.

## Security considerations

Make sure you trust the authors of any notebook before executing it. With a local connection, the code you execute can read, write, and delete files on your computer.

Connecting to a Jupyter notebook server running on your local machine can provide many benefits. With these benefits come serious potential risks. By connecting to a local runtime, you are allowing the Colaboratory frontend to execute code in the notebook using the local resources on your machine. This means that the notebook could:

- Invoke arbitrary commands (i.e. `rm -rf /`)
- Access the local file system
- Run malicious content on your machine

Before attempting to connect to a local runtime, make sure you trust the authors of the notebook and ensure you understand what code is being executed. For more information on the Jupyter notebook server's security model, consult [Jupyter's documentation](#).

## Setup instructions

In order to allow Colaboratory to connect to your locally running Jupyter server, you'll need to perform the following steps.

### Step 1: Install Jupyter

Install [Jupyter](#) on your local machine.

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

The `jupyter_http_over_ws` extension is authored by the Colaboratory team and 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 set a flag to explicitly trust WebSocket connections from the Colaboratory frontend.

```
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 backend URL 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 Colaboratory, 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 config editor](#). Colaboratory makes a connection 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, the runtime on your local machine will not be shared. When others open the shared notebook, they will be connected to a standard Cloud 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 and 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 another machine (e.g. Google Compute Engine instance), you can set up 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 preconfigured. Follow the [how-to guides](#) to set up 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, set up 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 by connecting to the forwarded port (follow the same instructions under Step 4: Connect to the local runtime).