

Recap

Kick-start terminal instructions

```
# Create challenge folder
mkdir -p ~/code/WLaCoutur/07-ML-Ops/02-Cloud-training/data-recap-cloud-training && cd
$_

# Download challenge
curl -s -H "Authorization: Token oDBkU2uqVbPdWJmtDXwiVtJW, User=WLaCoutur"
"https://kitt.lewagon.com/camps/1917/challenges/setup_script?path=07-ML-Ops%2F02-Cloud
-training%2FRecap" | bash

# Open challenge folder in your text editor
code .
```

 Kitt no longer picks up Github pushes two weeks after camp has finished.

Report an issue

Vertex AI Workbench

Let's explore Vertex AI Workbench as an alternative to Compute Engine for model training.

Vertex AI Workbench provides Jupyter-notebook based development environments for data science. It allows you to run ML code without having to precisely configure the environment for the code. They come prepackaged with JupyterLab, NumPy, Pandas, scikit-learn, TensorFlow and PyTorch.

They also allow you to automatically run your notebook on a recurring schedule.

Create a Workbench Instance

Create a Workbench instance:

1. Access the [Vertex AI Workbench page](#)

2. At the top, enable the Notebooks API if it's not yet enabled. This could take a few minutes. When finished, return to the Vertex AI Workbench page.
3. At the top, select the INSTANCES tab and click on the blue CREATE NEW button further below.
4. Choose a name for your instance, for example *cloud-training-recap*, and click on the CONTINUE button at the bottom.
5. In the next page *Environment*, you can select the version you want to use. For now, let's keep the settings to use the latest version. Click on CONTINUE.
6. In the next page *Machine type*, you can see the options for your machine type. You could for example select a very powerful machine with loads of memory and GPU support. At the right you will see the estimated costs for your machine. For now, let's stick to the default machine, and click on CONTINUE.
7. In the next screen you could change the size of the disks if you need more disk space. Again, stick to the default.
8. We won't change the remaining settings either. So let's scroll down and click on the blue CREATE. (If it doesn't create, try switching to a region that has sufficient resources.)

👉 The Workbench instance should be ready in a couple of minutes.

Open the Workbench instance by clicking on OPEN JUPYTERLAB.

Explore the environment

We have a complete JupyterLab environment in front of us:

- We can run a terminal
- We can run different notebooks,
 - a standard notebook for ML with numpy, pandas, and scikit-learn
 - a notebook for DL with PyTorch
 - a notebook for DL with TensorFlow

On the left, you can upload files to your instance, very much like in Google Colab. You can use it to upload your notebook, your package folders, and your data. (If your dataset is very large, it's probably better to load it directly from its source on the cloud!)

For many use cases this will be enough for you to start working.

The Workbench instance IDE also comes with an integrated git client. If you want to use it, checkout the documentation [here](#) to set up the connection. Alternatively you can do a mini-LW setup on the Workbench instance with the instructions below.

Use your own environment

If you are not happy with the default environment used by the Workbench instance, you can create your own.

You might want to do this if your project uses another TensorFlow version than the default Workbench instance.

To do so, you can create a new *conda* environment by following [these instructions](#).

Mini-LW setup on the Workbench instance

If you plan to use the instance for a long time, you can a mini setup to mimic the environment on your own machine and make it more comfortable to work with.

Go to the Workbench instance and open the Terminal.

Install zsh and oh-my-zsh

Install zsh:

```
sudo apt install zsh
```

Install oh-my-zsh:

```
sh -c "$(curl -fsSL  
https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/install.sh)"
```

Authenticate on GitHub 1/2

Install `gh`:

```
sudo apt update  
sudo apt install gh
```

Run the `gh auth login` command:

- Account: `GitHub.com`
- Protocol: `HTTPS`
- Authenticate Git with your GitHub credentials: `Yes`
- Authentication method: `Paste an authentication token`

Create a GitHub Token

Create a GitHub token to allow the workbench to access your account:

1. Access [GitHub Tokens](#)
2. Click on generate new token
3. Fill in the Note field with a meaningful name, such as *Vertex AI Workbench token*
4. Check that these scopes are enabled: 'repo', 'read:org', 'workflow'
5. Click on generate token
6. Copy the token (you will not be able to retrieve it later)

Authenticate on GitHub 2/2

Paste the token in the Vertex AI instance's Terminal

Install direnv

```
curl -sfl https://direnv.net/install.sh | bash
eval "$(direnv hook zsh)"
```

Clone your project repo

Clone your recap repo inside your Workbench instance using the `curl` command that is provided in the Kick-start terminal instructions at the top of this challenge page on Kitt.

N.B. Make sure your `GCP_PROJECT` environment variable is set in your `.env` file

```
cp .env.sample .env
nano .env
```

Then run `direnv allow .`

Install your package:

```
pip install -e .
make reset_local_files
```

Now use your package

Now that you have set up everything:

- You can run your code on your Workbench instance

- You can create new notebooks to experiment, and use the code that you already refactored into `.py` files in your package.
- Use your own machine for other things. all compute happens on the Workbench instance:.
- Synch your code using `git pull` and `git push`.

You can for example run the preprocessing and the training like before:

```
make run_preprocess run_train
```

Compute Engine vs Vertex AI Workbench

In *Compute Engine* we can see that the *Vertex AI Workbench* uses a *Compute Engine* instance behind the scenes:

The screenshot shows the Google Cloud Console interface. The left sidebar has a 'Compute Engine' tab highlighted. The main content area shows 'VM instances' under the 'INSTANCES' tab. A table lists the instances:

Status	Name	Zone	Recommendations	In use by	Internal IP	External IP	Connect
<input type="checkbox"/>	taxi-instance	europe-west1-b			10.132.0.30 (nic0)		SSH
<input type="checkbox"/>	vxai-cloud-training-recap	europe-west1-b			10.132.0.38 (nic0)		SSH

Red arrows point from the labels 'compute engine instance' and 'vertex ai workbench instance' to the 'taxi-instance' and 'vxai-cloud-training-recap' rows respectively.

The price difference between a *Compute Engine* virtual machine and *Vertex AI Workbench* instance is really small.

So, for your data science projects, you probably want to create a Workbench instance, and get both a notebook based interface for experimenting and a classic virtual machine to run your code.

