Recap

Kick-start terminal instructions

```
# Create challenge folder
mkdir -p
~/code/WLaCoutur/07-ML-Ops/03-Automate-model-lifecycle/data-recap-automate-model-lifec
ycle && cd $_

# Download challenge
curl -s -H "Authorization: Token oDBkU2uqVbPdWJmtDXwiVtJW, User=WLaCoutur"
"https://kitt.lewagon.com/camps/1917/challenges/setup_script?path=07-ML-Ops%2F03-Autom
ate-model-lifecycle%2FRecap" | bash

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

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

Report an issue

Challenge not connected to Kitt yet 🙏

Download solution

Watch exercise resolution

Suggested recaps for Lifecycle Unit

We offer several possible recaps. Choose between:

- Option A) Live correction of today's Prefect challenge
- Option B) Prefect advanced: Schedule your own workflow runs
- Option C) MLflow advanced: Host your own MLFlow instance on a VM

Option A) Correction of Prefect challenge

The codebase in this repo contains a working solution.

👉 Let's read through it
👉 Play the full `make run_workflow` cycle: from Jan to Jun 2015

Option B) Register prefect workflows

🎯 The goal is to setup your prefect workflow so that it now runs once a month, ready for our new data to come in!

1) Local server

To launch the prefect server run 👇

```
prefect server start
```

Then point prefect at the server so it knows where to send flows now (as explained in the logs)!

```
prefect config set PREFECT_API_URL=http://127.0.0.1:4200/api
prefect profile use default
```

Finally checkout the your locally running prefect server at http://127.0.0.1:4200 and use `make run_workflow` and check that the flow is appearing on your local server!

☝️ So far, we've launched our flow manually from within python script `workflow.py`

What if we want to let a prefect Agent launch our flow on a regular basis?

For that, we'll need:

- A Prefect `agent` 🤖 to actually launch the run of the flows in our place
- A `workpool` 📦 into which to register our flow first
- A `deployment` 🚀 to encapsulate it all: The flow, the agent, the workpool, allowing it to be scheduled and triggered

More explanation [here]workpool

2) Create a workpool 📦

Work pools contain all the logic about what flows run and how

```
prefect work-pool create taxifare
prefect work-pool ls
```

3) Create an agent 🤖

We want an agent to pickup the work from the `taxifare` workpool so when a flow needs to be run it will be in the workpool and the agents connected to the pool will pick up the work that needs to be done. These agents can be spread across many machines!

```
prefect agent start --pool "taxifare"
```

Then go and check the health of the queue on the UI (Work Pool –> taxifare). When it is *Healthy* that means that there are agents ready to pick up the work from the pool!

4) Register your flow into a Deployment 🚀

The next part of making our flow run monthly is creating a deployment.

A deployment is a server-side concept that encapsulates a flow, allowing it to be scheduled and triggered

This contains all the data needed for the flow to be run by different agents! To create a workflow with the flow from today run 👇

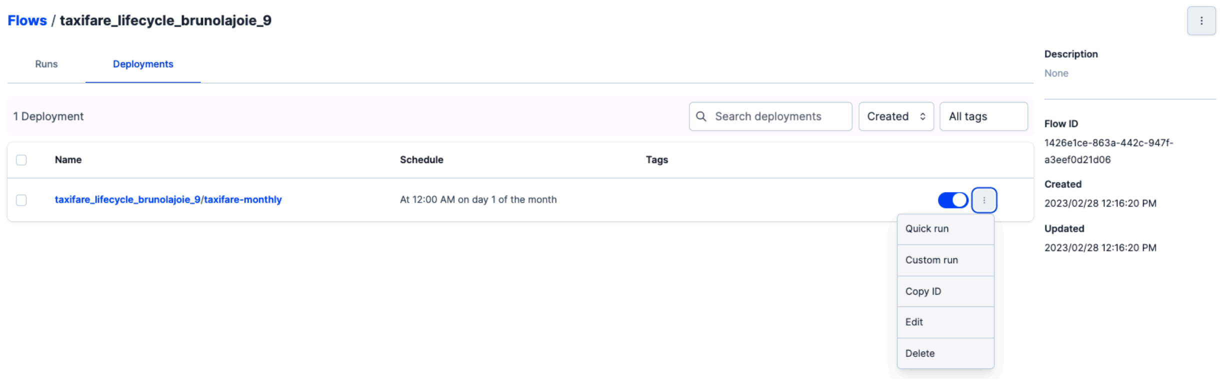
```
prefect deployment build \
        --pool taxifare \
        --name taxifare-monthly \
        --apply \
        --cron "0 0 1 * *" \
        --timezone "Europe/London" \
        taxifare/interface/workflow.py:train_flow
```

Most of the parameters should be clear to you:

- `pool` tells us where to send the flows from this deployment
- `name` for the name of the deployment
- `apply` applies the deployment as well as building it (you can build then edit the created file before applying)
- `cron` describes how often to run the deployment (this website is great for understanding cron syntax)
- `timezone` describes which timezone the cron should work from (this is the db of valid timezones)

Checkout the deployment on the ui and you will see the monthly runs planned.

Do a quick run of the deployment and checkout how the agent picks up the flow!


Flows / taxifare_lifecycle_brunolajoie_9

Runs | Deployments

1 Deployment

Name	Schedule	Tags
taxifare_lifecycle_brunolajoie_9/taxifare-monthly	At 12:00 AM on day 1 of the month	

Description
None

Flow ID
1426e1ce-863a-442c-947f-a3eef0d21d06

Created
2023/02/28 12:16:20 PM

Updated
2023/02/28 12:16:20 PM

You now have a system which can run your mlops workflow monthly and maintain the health of your production model!

💡 We could also register flow in Python

```python
# taxifare/interface/workflow.py

from prefect.deployments import Deployment
from prefect.server.schemas.schedules import CronSchedule

def register_flow():

    taxifare_deployment = Deployment.build_from_flow(
        flow=train_flow,
        name="taxifare-monthly",
        work_pool="taxifare",
        schedule=(CronSchedule(cron="0 0 1 * *", timezone="Europe/London")),
        )

    taxifare_deployment.apply()
```

Option C) Setup self hosted MLflow server!

In this unit, we've been using mlflow.lewagon.ai, a webserver instance of MLflow conveniently setup for you already!

In real life, or for projects weeks, you might want to use a clean mlflow just for your own project team!

👉 Follow the tutorial at https://kitt.lewagon.com/knowledge/cheatsheets/mlflow_hosting (the teacher should have already done instruction up to `pip3 install mlflow[extras]==2.1.1`)