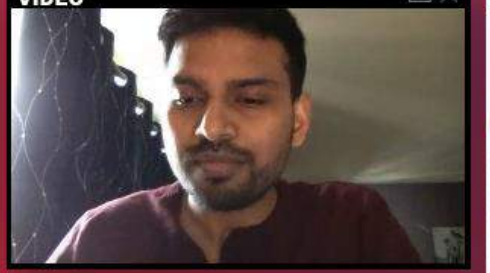
Flyte: Cloud Native Machine Learning & Data Processing Platform

onlinexperiences.com/scripts/Server.nxp

THE LINUX FOUNDATION OPEN SOURCE SUMMIT NORTH AMERICA

Embedded Linux Conference North America

VIDEO



Q&A WITH SPEAKER | SPEAKER BIO

Ketan Umare
Software engineer
Lyft

Ketan Umare is a senior staff software engineer at Lyft a
the Flyte project. Before Flyte he worked on ETA, routing
infrastructure at Lyft. He is also the founder of Flink Kub
operator and contributor to Spark on K8s. Prior to Lyft he
founding member at Oracle Gen2 Cloud and lead teams
Block Storage. Prior to that, he started and lead multiple
and Transportation optimization infrastructure at Amazo
his Masters in Computer Science from Georgia Tech spe
performance computing and his Bachelors in Engineerin
Science from VJTI Mumbai. Some of my prior talks and
Kubecon NA 2019: https://kccncna19.sched.com/event/
cloud-native-machine-learning-data-processing-platform
haytham-abuelfutuh-lyft AWS re:invent 2019:
https://www.portal.reinvent.awsevents.com/connect/se
SESSION_ID=98485&tclass=popup&csrftkn=V30O-96FA-
M7IV-HGK3-A4E0-BZII Flink Forward 2019:
https://www.ververica.com/resources/flink-forward-san-
2019managing-flink-on-kubernetes-flinkk8soperator Twi
interview: https://twimlai.com/twiml-talk-343-scalable-a
maintainable-workflows-at-lyft-with-flyte-w-haytham-abu
ketan-umare/

SLIDES

Flyte: Cloud Native Machine Learning & Data Processing Platform

onlinexperiences.com/scripts/Server.nxp

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

Q&A WITH SPEAKER SPEAKER BIO

Ketan Umare
Software engineer
Lyft

Ketan Umare is a senior staff software engineer at Lyft a
the Flyte project. Before Flyte he worked on ETA, routing
infrastructure at Lyft. He is also the founder of Flink Kub
operator and contributor to Spark on K8s. Prior to Lyft he
founding member at Oracle Gen2 Cloud and lead teams
Block Storage. Prior to that, he started and lead multiple
and Transportation optimization infrastructure at Amazc
his Masters in Computer Science from Georgia Tech spe
performance computing and his Bachelors in Engineerin
Science from VJTI Mumbai. Some of my prior talks and
Kubecon NA 2019: https://kccncna19.sched.com/event/
cloud-native-machine-learning-data-processing-platform
haytham-abuelfutuh-lyft AWS re:invent 2019:
https://www.portal.reinvent.awsevents.com/connect/se
SESSION_ID=98485&tclass=popup&csrftkn=V30O-96FA-
M7IV-HGK3-A4E0-BZII Flink Forward 2019:
https://www.ververica.com/resources/flink-forward-san-
2019managing-flink-on-kubernetes-flinkk8soperator Twi
interview: https://twimlai.com/twiml-talk-343-scalable-a
maintainable-workflows-at-lyft-with-flyte-w-haytham-abu
ketan-umare/

SLIDES

Flyte

Production Grade Orchestration for Data and ML

Ketan Umare
Software Engineer
Lyft
@KetanUmare
@KetanUmare

Matthew Smith
Software Engineer
Lyft
@matthewphsmith
@matthewphsmith

lyft

Flyte

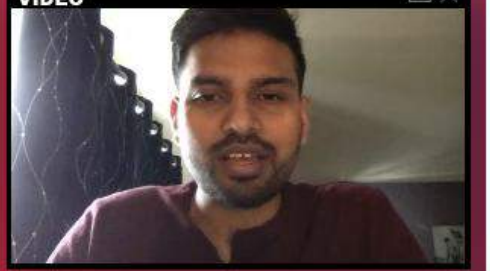
Flyte: Cloud Native Machine Learning & Data Processing Platform

onlinexperiences.com/scripts/Server.nxp

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO



ABSTRACT

Flyte is the backbone for large-scale Machine Learning and Data Processing (ETL) pipelines at Lyft. It is used across business critical applications ranging from ETA, Pricing, Mapping, Autonomous etc. At its core it is a Kubernetes native workflow engine that executes 10M+ containers per month as part of thousands of workflows. We introduced Flyte at Kubecon 2019. This talk will build on the base talk and dive deeper into the architecture and the developer experience. Thus the talk will help existing users of Flyte and new users as well.

Quick overview of Flyte concepts
Architecture of Flyte
Challenges and learnings while using Kubernetes at scale for stateful applications
A deeper look at developer ergonomics and reasons that shaped some decisions
An overview of the community and ecosystem since open sourcing Flyte
The talk will conclude with a demo.

SLIDES

Agenda

Motivation & Goal
What problem are we trying to solve?

Concepts & Features
Concepts, features and user interface

Architecture & Challenges
Architecture of Flyte and summary of challenges we faced

Case Study
How to use Flyte in the real world

Demo
Everyone loves demos!

Conclusion
Learn more, get involved, & get started

lyft

Flyte

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

ABSTRACT

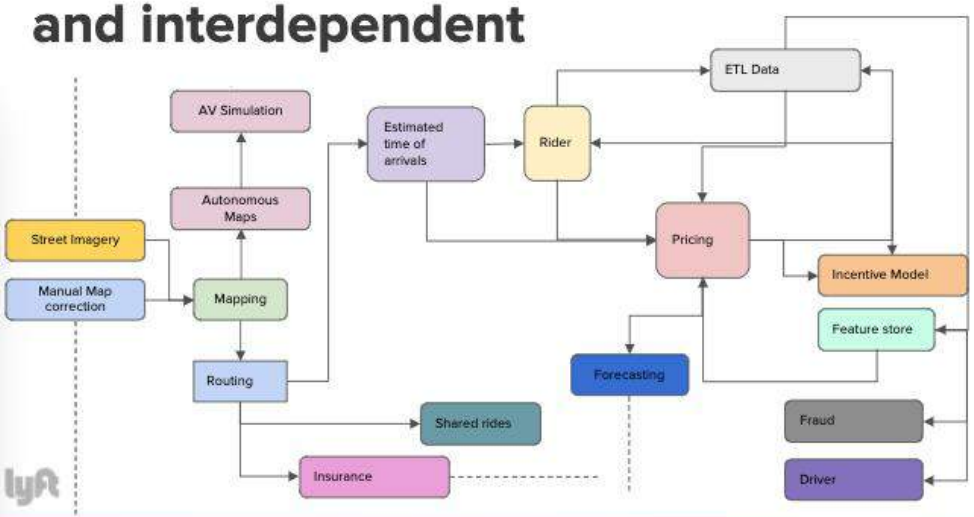
Flyte is the backbone for large-scale Machine Learning and Data Processing (ETL) pipelines at Lyft. It is used across business critical applications ranging from ETA, Pricing, Mapping, Autonomous etc. At its core it is a Kubernetes native workflow engine that executes 10M+ containers per month as part of thousands of workflows. We introduced Flyte at Kubecon 2019. This talk will build on the base talk and dive deeper into the architecture and the developer experience. Thus the talk will help existing users of Flyte and new users as well.

Quick overview of Flyte concepts
Architecture of Flyte
Challenges and learnings while using Kubernetes at scale for stateful applications
A deeper look at developer ergonomics and reasons that shaped some decisions
An overview of the community and ecosystem since open sourcing Flyte
The talk will conclude with a demo.

SLIDES

Motivation & Goal

ML & Data services are increasingly complex and interdependent



Data and ML processes often interact.

Data Flow is very complex and machine learning is more than just model code.

Flyte

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

ABSTRACT

Flyte is the backbone for large-scale Machine Learning and Data Processing (ETL) pipelines at Lyft. It is used across business critical applications ranging from ETA, Pricing, Mapping, Autonomous etc. At its core it is a Kubernetes native workflow engine that executes 10M+ containers per month as part of thousands of workflows. We introduced Flyte at Kubecon 2019. This talk will build on the base talk and dive deeper into the architecture and the developer experience. Thus the talk will help existing users of Flyte and new users as well.

Quick overview of Flyte concepts
Architecture of Flyte
Challenges and learnings while using Kubernetes at scale for stateful applications
A deeper look at developer ergonomics and reasons that shaped some decisions
An overview of the community and ecosystem since open sourcing Flyte
The talk will conclude with a demo.

SLIDES

Motivation & Goal

The parts of the stack we wanted to tackle

lyft

Source: Sculley et al: Hidden Technical Debt in Machine Learning Systems

Flyte

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

ABSTRACT

Flyte is the backbone for large-scale Machine Learning and Data Processing (ETL) pipelines at Lyft. It is used across business critical applications ranging from ETA, Pricing, Mapping, Autonomous etc. At its core it is a Kubernetes native workflow engine that executes 10M+ containers per month as part of thousands of workflows. We introduced Flyte at Kubecon 2019. This talk will build on the base talk and dive deeper into the architecture and the developer experience. Thus the talk will help existing users of Flyte and new users as well.

Quick overview of Flyte concepts
Architecture of Flyte
Challenges and learnings while using Kubernetes at scale for stateful applications
A deeper look at developer ergonomics and reasons that shaped some decisions
An overview of the community and ecosystem since open sourcing Flyte
The talk will conclude with a demo.

SLIDES

Flyte

Hosted, scalable and serverless Orchestration Platform

Fabric that connects disparate compute technologies

Extensible, Observable & shareable

Integrates best of the breed open source solutions

Auditable, Repeatable & Secure

lyft

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

ABSTRACT

Flyte is the backbone for large-scale Machine Learning and Data Processing (ETL) pipelines at Lyft. It is used across business critical applications ranging from ETA, Pricing, Mapping, Autonomous etc. At its core it is a Kubernetes native workflow engine that executes 10M+ containers per month as part of thousands of workflows. We introduced Flyte at Kubecon 2019. This talk will build on the base talk and dive deeper into the architecture and the developer experience. Thus the talk will help existing users of Flyte and new users as well.

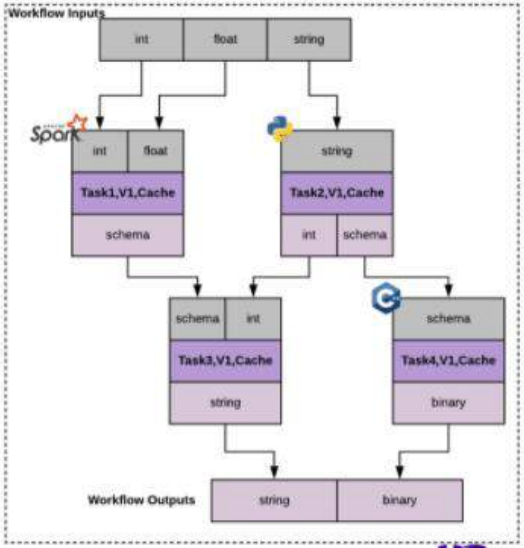
Quick overview of Flyte concepts
Architecture of Flyte
Challenges and learnings while using Kubernetes at scale for stateful applications
A deeper look at developer ergonomics and reasons that shaped some decisions
An overview of the community and ecosystem since open sourcing Flyte
The talk will conclude with a demo.

SLIDES

Concepts & Features

Tasks & Workflows

- Declarative (protobuf)
- **Versioned**
- Strongly **typed interfaces**
- Models the flow of Data
- Tasks
  - Arbitrarily complex
  - Encapsulate user code
- Workflows
  - Composable
  - Dynamic
  - DSL



lyft

Flyte

VIDEO

ABSTRACT

Flyte is the backbone for large-scale Machine Learning and Data Processing (ETL) pipelines at Lyft. It is used across business critical applications ranging from ETA, Pricing, Mapping, Autonomous etc. At its core it is a Kubernetes native workflow engine that executes 10M+ containers per month as part of thousands of workflows. We introduced Flyte at Kubecon 2019. This talk will build on the base talk and dive deeper into the architecture and the developer experience. Thus the talk will help existing users of Flyte and new users as well.

Quick overview of Flyte concepts
Architecture of Flyte
Challenges and learnings while using Kubernetes at scale for stateful applications
A deeper look at developer ergonomics and reasons that shaped some decisions
An overview of the community and ecosystem since open sourcing Flyte
The talk will conclude with a demo.

SLIDES

Concepts & Features

Tasks

Atomic unit of work & entrypoint to user code. Language agnostic.

Can be executed independently.

```scala
case class SumTaskInput(a: Long, b: Long)
case class SumTaskOutput(c: Long)
class SumTask extends … {
   override def
run(input:SumTaskInput):SumTaskOutput = {
    SumTaskOutput(input.a + input.b)
  }
}
```

https://github.com/spotify/flytekit-java

flytekit Scala

Spark Code

```python
@inputs(rides=Types.Schema[...],
k=Types.Integer)
@outputs(dest=[Types.String])
@spark_task(spark_conf={"executors":2})
def find_topk_destinations(ctx, spark_ctx,
rides, k, dest):
  '''
  Find the top k destinations for the given set
of rides ordered by frequency
```

Arbitrary containers

```python
edges = SdkRawContainerTask(
    input_data_dir="/inputs",
    output_data_dir="/outputs",
    inputs={"image": Types.Blob, "script": Types.Blob},
    outputs={"edges": Types.Blob},
    image="jjanzic/docker-python3-opencv",
    command=["python", "{{.inputs.script}}",
"/inputs/image", "/outputs/edges"],
)
```

https://github.com/lyft/flytekit

lyft

Flyte

Flyte: Cloud Native Machine Learning & Data Processing Platform

onlinexperiences.com/scripts/Server.nxp

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

ABSTRACT

Flyte is the backbone for large-scale Machine Learning and Data Processing (ETL) pipelines at Lyft. It is used across business critical applications ranging from ETA, Pricing, Mapping, Autonomous etc. At its core it is a Kubernetes native workflow engine that executes 10M+ containers per month as part of thousands of workflows. We introduced Flyte at Kubecon 2019. This talk will build on the base talk and dive deeper into the architecture and the developer experience. Thus the talk will help existing users of Flyte and new users as well.

Quick overview of Flyte concepts
Architecture of Flyte
Challenges and learnings while using Kubernetes at scale for stateful applications
A deeper look at developer ergonomics and reasons that shaped some decisions
An overview of the community and ecosystem since open sourcing Flyte
The talk will conclude with a demo.

SLIDES

Concepts & Features

Workflows

Specify the data dependency between tasks (as DAGs). **Composable** & **declarative**!

Multiple schedules for the same Workflow

Compose with other WF's

```
@workflow_class
class StaticSubWorkflowCaller(object):
    in = Input(Types.Integer, default=5,
help="Input for inner workflow")
    identity_wf_execution =
IdentityWorkflow(a=outer_a)
    out =
Output(identity_wf_execution.outputs.task_
output, sdk_type=Types.Integer)
```

ML Model Train example

```
@workflow_class
class TrainModel(object):
 # Accept inputs
 data = Input(Types.Schema[...])
 hyperparam = Input(Types.Float)
 # Split the dataset
 split = split8020(data=data)

 model = fit_xgboost(
             data=split.train,
             hyperparam=hyperparam)

 pred = eval_xgboost(data=split.val,
             m=model.outputs.v)

 metrics = compute_metrics(
             data=split.val,
             pred=pred.y_pred)
 # Create outputs
 model = Output(model.outputs.v)
 accuracy = Output(metrics.outputs.acc)
```

lyft

Flyte

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

ABSTRACT

Flyte is the backbone for large-scale Machine Learning and Data Processing (ETL) pipelines at Lyft. It is used across business critical applications ranging from ETA, Pricing, Mapping, Autonomous etc. At its core it is a Kubernetes native workflow engine that executes 10M+ containers per month as part of thousands of workflows. We introduced Flyte at Kubecon 2019. This talk will build on the base talk and dive deeper into the architecture and the developer experience. Thus the talk will help existing users of Flyte and new users as well.

Quick overview of Flyte concepts
Architecture of Flyte
Challenges and learnings while using Kubernetes at scale for stateful applications
A deeper look at developer ergonomics and reasons that shaped some decisions
An overview of the community and ecosystem since open sourcing Flyte
The talk will conclude with a demo.

SLIDES

Concepts & Features

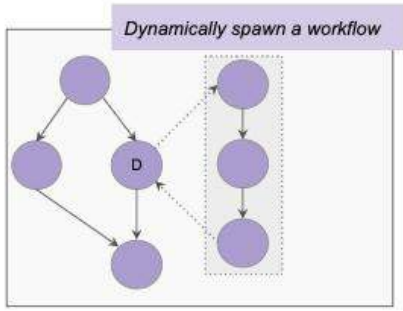
Dynamism

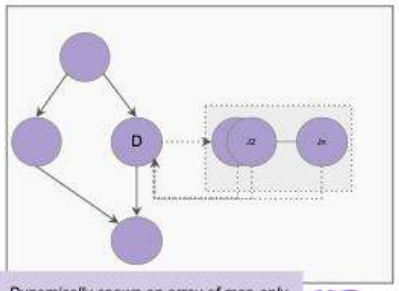
Flyte Workflows are **statically** defined and **immutable**. But, they can contain nodes - that can change the shape dynamically.

Data parallel jobs, dynamic generation of workflows

(generate logic using the available data), etc

```
@inputs(num=Types.Integer)
@outputs(out=Types.Integer)
@dynamic_task
def sub_wf_yield_task(wf_params, num, out):
    wf_params.logging.info("Running inner task... yielding a sub-workflow")
    identity_wf_execution = IdentityWorkflow(a=num)
    yield identity_wf_execution
    out.set(identity_wf_execution.outputs.task_output)
```



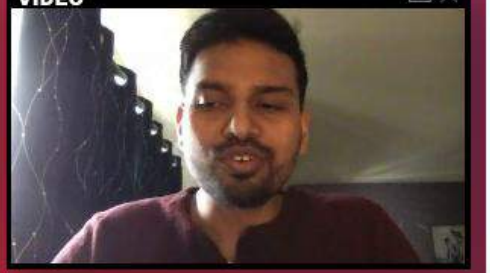


Dynamically spawn an array of map-only

lyft

Flyte

VIDEO



ABSTRACT

Flyte is the backbone for large-scale Machine Learning and Data Processing (ETL) pipelines at Lyft. It is used across business critical applications ranging from ETA, Pricing, Mapping, Autonomous etc. At its core it is a Kubernetes native workflow engine that executes 10M+ containers per month as part of thousands of workflows. We introduced Flyte at Kubecon 2019. This talk will build on the base talk and dive deeper into the architecture and the developer experience. Thus the talk will help existing users of Flyte and new users as well.

Quick overview of Flyte concepts
Architecture of Flyte
Challenges and learnings while using Kubernetes at scale for stateful applications
A deeper look at developer ergonomics and reasons that shaped some decisions
An overview of the community and ecosystem since open sourcing Flyte
The talk will conclude with a demo.

SLIDES

Concepts & Features

Grouping & Sharing

Projects, Domains & Versions

- Projects offer **logical grouping** of Workflows & Tasks and can be split across one or more repositories, one or more containers
- Domains and Versions provide **CI/CD like semantics** to Workflows & Tasks
  - Users can **push new** versions to production, **rollback** to previous version etc.
  - Users can have workflows in **integration/staging** env
- Domains are **configured globally** for the system (by administrators)

Sharing & Accounting

- Workflows can **refer to tasks and workflows** from other projects
- Executions **accounted/billed** under the **requesters project & domain (*Infraspend*)**

lyft

Flyte

VIDEO

ABSTRACT

Flyte is the backbone for large-scale Machine Learning and Data Processing (ETL) pipelines at Lyft. It is used across business critical applications ranging from ETA, Pricing, Mapping, Autonomous etc. At its core it is a Kubernetes native workflow engine that executes 10M+ containers per month as part of thousands of workflows. We introduced Flyte at Kubecon 2019. This talk will build on the base talk and dive deeper into the architecture and the developer experience. Thus the talk will help existing users of Flyte and new users as well.

Quick overview of Flyte concepts
Architecture of Flyte
Challenges and learnings while using Kubernetes at scale for stateful applications
A deeper look at developer ergonomics and reasons that shaped some decisions
An overview of the community and ecosystem since open sourcing Flyte
The talk will conclude with a demo.

SLIDES

Concepts & Features

Shareability: Flytekit Example

Project: ProjectA

```
@workflow_class
class PipelineA(object):
    in1 = Input(Types.Integer)
    in2 = Input(Types.Integer)
    …
    out1 = Output(print2.outputs.out)
```

Project: ProjectA

```
@inputs(x=Types.Integer, y=Types.Integer)
@outputs(z=Types.Integer)
@task
def my_model(x, y):
    ….
```

Project: ProjectB

```
@workflow_class
class CompositePipeline(object):

  composed_wf = lps.fetch(
          "ProjectA",
          "Production",
          "PipelineA",
          "1.0.2"
          )(in1, in2)

  t1 = local_task(composed_wf.outputs.out)

  t2 = tasks.fetch(
            "ProjectA",
            "Production",
            "my_model",
            "2.0.0"
            )(x=t1.outputs.x, y=10)
```

lyft

Flyte

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

ABSTRACT

Flyte is the backbone for large-scale Machine Learning and Data Processing (ETL) pipelines at Lyft. It is used across business critical applications ranging from ETA, Pricing, Mapping, Autonomous etc. At its core it is a Kubernetes native workflow engine that executes 10M+ containers per month as part of thousands of workflows. We introduced Flyte at Kubecon 2019. This talk will build on the base talk and dive deeper into the architecture and the developer experience. Thus the talk will help existing users of Flyte and new users as well.

Quick overview of Flyte concepts
Architecture of Flyte
Challenges and learnings while using Kubernetes at scale for stateful applications
A deeper look at developer ergonomics and reasons that shaped some decisions
An overview of the community and ecosystem since open sourcing Flyte
The talk will conclude with a demo.

SLIDES

Concepts and Features

DataCatalog: Lineage & Memoization

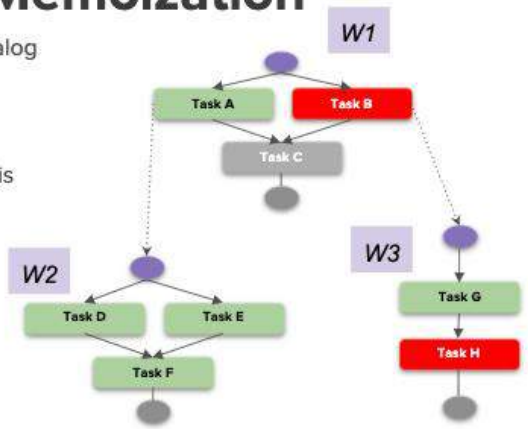
Every task execution in Flyte is **recorded** by default in Catalog Service. This enables Flyte executions to have,

Artifact Lineage

- **Causal** dependencies between data and processes is tracked

Memoization

- Each task execution has a **unique signature**, which includes the input values & version of code
- **Repeated** executions with matching signatures are cached



lyft

Flyte

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

ABSTRACT

Flyte is the backbone for large-scale Machine Learning and Data Processing (ETL) pipelines at Lyft. It is used across business critical applications ranging from ETA, Pricing, Mapping, Autonomous etc. At its core it is a Kubernetes native workflow engine that executes 10M+ containers per month as part of thousands of workflows. We introduced Flyte at Kubecon 2019. This talk will build on the base talk and dive deeper into the architecture and the developer experience. Thus the talk will help existing users of Flyte and new users as well.

Quick overview of Flyte concepts
Architecture of Flyte
Challenges and learnings while using Kubernetes at scale for stateful applications
A deeper look at developer ergonomics and reasons that shaped some decisions
An overview of the community and ecosystem since open sourcing Flyte
The talk will conclude with a demo.

SLIDES

Concepts & Features

Serverless

User should only worry about business logic

- specify **resource requirements** - CPU, GPU, Memory, #spark executors etc
- develop **multiple versions of code concurrently**
- **Multi-tenancy** unaware
- Simple **gRPC/REST** API to access all the power
- Language agnostic - **flytekit** (python) and **flytekit-java,** raw containers
- Get **notifications** and **alerts** for specific events (failures, successes etc)
- **Retrieve results** of past executions

```
@python(gpu_hint=1, cpu=4,
retries=3, timeout=30s)
def myFunction():
    ...
```

lyft

Flyte

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

ABSTRACT

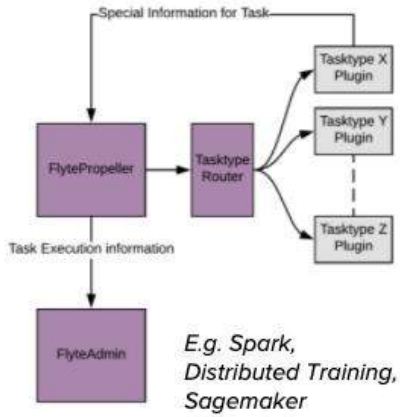
Flyte is the backbone for large-scale Machine Learning and Data Processing (ETL) pipelines at Lyft. It is used across business critical applications ranging from ETA, Pricing, Mapping, Autonomous etc. At its core it is a Kubernetes native workflow engine that executes 10M+ containers per month as part of thousands of workflows. We introduced Flyte at Kubecon 2019. This talk will build on the base talk and dive deeper into the architecture and the developer experience. Thus the talk will help existing users of Flyte and new users as well.

Quick overview of Flyte concepts
Architecture of Flyte
Challenges and learnings while using Kubernetes at scale for stateful applications
A deeper look at developer ergonomics and reasons that shaped some decisions
An overview of the community and ecosystem since open sourcing Flyte
The talk will conclude with a demo.

SLIDES

Concepts & Features

Extensible



E.g. Spark, Distributed Training, Sagemaker Flink

```python
@sensor_task
def my_test_task(ctx):
  '''
  E.g. sensor that waits for a hive partition
  to land. This is added as a contrib.
  '''
  return MyHivePartitionSensor()
```

- Container executions are purely from Flyte's POV. (you can write in python, Java etc)
- But this is limited to the implementation SDK
- Backend extensions allow extending capabilities of Flyte.

lyft

Flyte

Flyte: Cloud Native Machine Learning & Data Processing Platform

onlinexperiences.com/scripts/Server.nxp

THE LINUX FOUNDATION OPEN SOURCE SUMMIT NORTH AMERICA

Embedded Linux Conference North America

VIDEO

ABSTRACT

Flyte is the backbone for large-scale Machine Learning and Data Processing (ETL) pipelines at Lyft. It is used across business critical applications ranging from ETA, Pricing, Mapping, Autonomous etc. At its core it is a Kubernetes native workflow engine that executes 10M+ containers per month as part of thousands of workflows. We introduced Flyte at Kubecon 2019. This talk will build on the base talk and dive deeper into the architecture and the developer experience. Thus the talk will help existing users of Flyte and new users as well.

Quick overview of Flyte concepts
Architecture of Flyte
Challenges and learnings while using Kubernetes at scale for stateful applications
A deeper look at developer ergonomics and reasons that shaped some decisions
An overview of the community and ecosystem since open sourcing Flyte
The talk will conclude with a demo.

SLIDES

Architecture & Challenges

Real Production Scale

8.5k Unique Workflows defined

54k Unique Task definitions

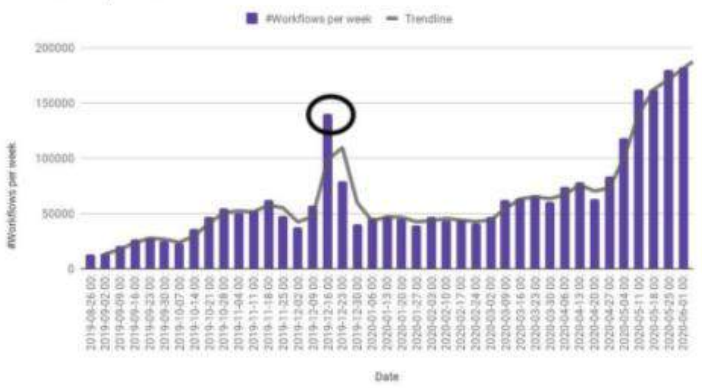
1million+ Workflow executions per month

10 million+ task executions per month

40 million+ containers executed per month

In early January this year, we had a sudden peak and ... (*we will get to the story in a bit*)

#Workflows per week



lyft

Flyte

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

ABSTRACT

Flyte is the backbone for large-scale Machine Learning and Data Processing (ETL) pipelines at Lyft. It is used across business critical applications ranging from ETA, Pricing, Mapping, Autonomous etc. At its core it is a Kubernetes native workflow engine that executes 10M+ containers per month as part of thousands of workflows. We introduced Flyte at Kubecon 2019. This talk will build on the base talk and dive deeper into the architecture and the developer experience. Thus the talk will help existing users of Flyte and new users as well.

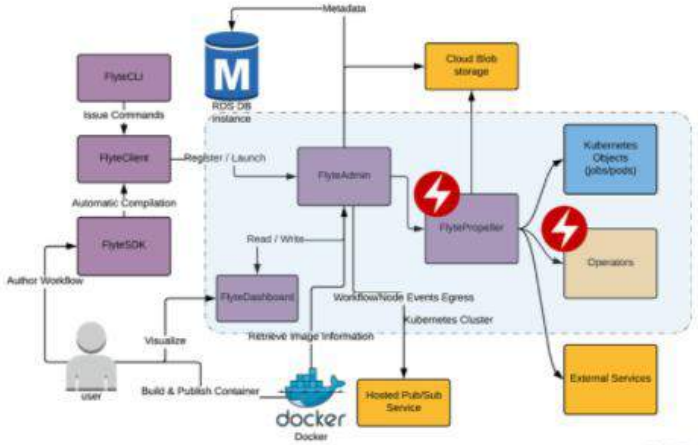
Quick overview of Flyte concepts
Architecture of Flyte
Challenges and learnings while using Kubernetes at scale for stateful applications
A deeper look at developer ergonomics and reasons that shaped some decisions
An overview of the community and ecosystem since open sourcing Flyte
The talk will conclude with a demo.

SLIDES

Architecture & Challenges

Architecture Overview

Default: Single Kubernetes cluster with scale-out options to cloud services like AWS Batch.



lyft

Flyte

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

ABSTRACT

Flyte is the backbone for large-scale Machine Learning and Data Processing (ETL) pipelines at Lyft. It is used across business critical applications ranging from ETA, Pricing, Mapping, Autonomous etc. At its core it is a Kubernetes native workflow engine that executes 10M+ containers per month as part of thousands of workflows. We introduced Flyte at Kubecon 2019. This talk will build on the base talk and dive deeper into the architecture and the developer experience. Thus the talk will help existing users of Flyte and new users as well.

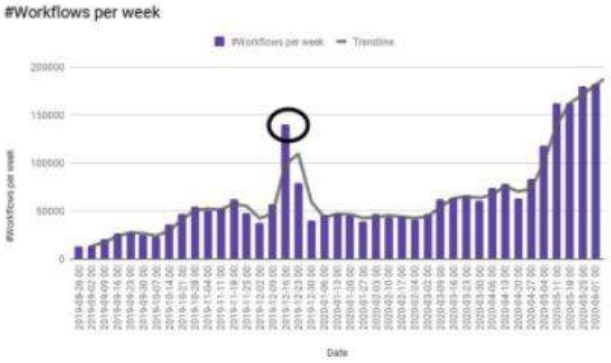
Quick overview of Flyte concepts
Architecture of Flyte
Challenges and learnings while using Kubernetes at scale for stateful applications
A deeper look at developer ergonomics and reasons that shaped some decisions
An overview of the community and ecosystem since open sourcing Flyte
The talk will conclude with a demo.

SLIDES

Architecture & Challenges

Challenges

- Super-exponential growth, spiked 6x in 2 days
- Flyte had problems
  - Users **lacked visibility**
  - System admins were overwhelmed with operations
  - The cost expenditure ballooned
  - Various scaling problems
    - K8s control plane
    - etcD and K8s Controller
    - K8s Scheduler
    - FlytePropeller is complex (data handling)
- We started diving deeper and optimizing



lyft

Flyte

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

ABSTRACT

Flyte is the backbone for large-scale Machine Learning and Data Processing (ETL) pipelines at Lyft. It is used across business critical applications ranging from ETA, Pricing, Mapping, Autonomous etc. At its core it is a Kubernetes native workflow engine that executes 10M+ containers per month as part of thousands of workflows. We introduced Flyte at Kubecon 2019. This talk will build on the base talk and dive deeper into the architecture and the developer experience. Thus the talk will help existing users of Flyte and new users as well.

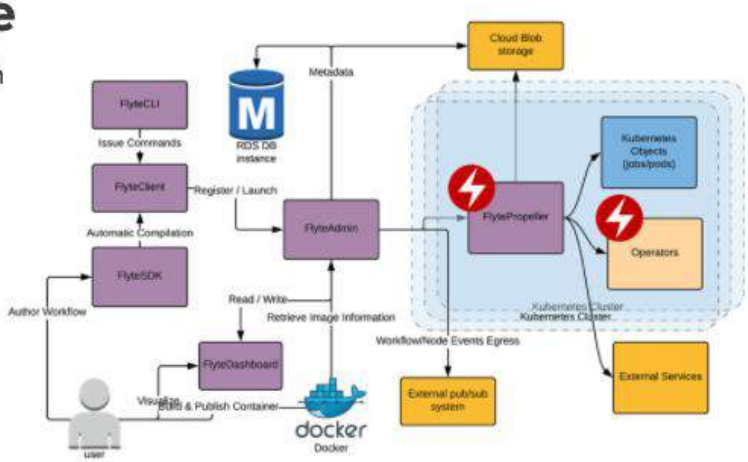
Quick overview of Flyte concepts
Architecture of Flyte
Challenges and learnings while using Kubernetes at scale for stateful applications
A deeper look at developer ergonomics and reasons that shaped some decisions
An overview of the community and ecosystem since open sourcing Flyte
The talk will conclude with a demo.

SLIDES

Architecture & Challenges

Challenge: Scale

- We observed problems with Single Kubernetes cluster -
  - API latency
  - Pod startup issues
  - etcD object size limits
- Single instance of **FlytePropeller*** can run more than **2000** workflows concurrently
  - Informer **caches**
  - Smart control loop short circuits
  - Status **compression**
  - many more!



lyft

Flyte

VIDEO

ABSTRACT

Flyte is the backbone for large-scale Machine Learning and Data Processing (ETL) pipelines at Lyft. It is used across business critical applications ranging from ETA, Pricing, Mapping, Autonomous etc. At its core it is a Kubernetes native workflow engine that executes 10M+ containers per month as part of thousands of workflows. We introduced Flyte at Kubecon 2019. This talk will build on the base talk and dive deeper into the architecture and the developer experience. Thus the talk will help existing users of Flyte and new users as well.

Quick overview of Flyte concepts
Architecture of Flyte
Challenges and learnings while using Kubernetes at scale for stateful applications
A deeper look at developer ergonomics and reasons that shaped some decisions
An overview of the community and ecosystem since open sourcing Flyte
The talk will conclude with a demo.

SLIDES

Architecture & Challenges

Challenge: Multi-tenancy

- Flyte projects are multi-tenancy primitives. Some tenants have larger use-cases as compared to others
- Flyte leverages ResourceQuotas from Kubernetes
  - Flyte Cluster resource controller allows dynamic modification of limits
  - Observability tools show current utilization
  - FlytePropeller backs off intelligently to relieve KubeAPI pressure
- Flyte provides a custom Resource manager on top of Kubernetes Quotas that ensure Gobal limits and per tenant limits
  - These limits help maintain downstream service
  - Queues to maintain fairness (OSS in progress)
- K8s CRD FairQ (inprogress)

lyft

Flyte

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

ABSTRACT

Flyte is the backbone for large-scale Machine Learning and Data Processing (ETL) pipelines at Lyft. It is used across business critical applications ranging from ETA, Pricing, Mapping, Autonomous etc. At its core it is a Kubernetes native workflow engine that executes 10M+ containers per month as part of thousands of workflows. We introduced Flyte at Kubecon 2019. This talk will build on the base talk and dive deeper into the architecture and the developer experience. Thus the talk will help existing users of Flyte and new users as well.

Quick overview of Flyte concepts
Architecture of Flyte
Challenges and learnings while using Kubernetes at scale for stateful applications
A deeper look at developer ergonomics and reasons that shaped some decisions
An overview of the community and ecosystem since open sourcing Flyte
The talk will conclude with a demo.

SLIDES

Architecture & Challenges

Challenge: Visibility

- Automatically generated user dashboard - **grafana template** (oss soon)
- Errors from execution pulled into the UI
- Logs from distributed tasks like **Spark** are pulled into the UI
- Users can **triage** amount of **CPU/memory utilized** by single execution
- **User vs System errors** are clearly separated

lyft

Flyte

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

ABSTRACT

Flyte is the backbone for large-scale Machine Learning and Data Processing (ETL) pipelines at Lyft. It is used across business critical applications ranging from ETA, Pricing, Mapping, Autonomous etc. At its core it is a Kubernetes native workflow engine that executes 10M+ containers per month as part of thousands of workflows. We introduced Flyte at Kubecon 2019. This talk will build on the base talk and dive deeper into the architecture and the developer experience. Thus the talk will help existing users of Flyte and new users as well.

Quick overview of Flyte concepts
Architecture of Flyte
Challenges and learnings while using Kubernetes at scale for stateful applications
A deeper look at developer ergonomics and reasons that shaped some decisions
An overview of the community and ecosystem since open sourcing Flyte
The talk will conclude with a demo.

SLIDES

Architecture & Challenges

Challenge: Operations

- Standardized Grafana template for Admins (OSS soon)
- Improved documentation and examples (Work in progress)
- Staged **rollouts**
- **FlyteAdmin** provides a **flexible routing** system to multiple K8s clusters
  - Allows **isolating** important usecases in different clusters
  - Deployments **bake in lower priority** clusters before proceeding

lyft

Flyte

THE LINUX FOUNDATION OPEN SOURCE SUMMIT NORTH AMERICA

Embedded Linux Conference North America

VIDEO

ABSTRACT

Flyte is the backbone for large-scale Machine Learning and Data Processing (ETL) pipelines at Lyft. It is used across business critical applications ranging from ETA, Pricing, Mapping, Autonomous etc. At its core it is a Kubernetes native workflow engine that executes 10M+ containers per month as part of thousands of workflows. We introduced Flyte at Kubecon 2019. This talk will build on the base talk and dive deeper into the architecture and the developer experience. Thus the talk will help existing users of Flyte and new users as well.

Quick overview of Flyte concepts
Architecture of Flyte
Challenges and learnings while using Kubernetes at scale for stateful applications
A deeper look at developer ergonomics and reasons that shaped some decisions
An overview of the community and ecosystem since open sourcing Flyte
The talk will conclude with a demo.

SLIDES

Architecture & Challenges

Challenge: Efficiency

- Centralized platform
  - Amortize **TCO**
  - **Efficiency** multiplier
  - Centralized tooling to visualize costs
- Utilize **Spot** instances (AWS only)
- Optimized Cluster Autoscaler (reduced spend by 25%)
- *Coming soon*: K8s scheduler optimizations

lyft

Flyte

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

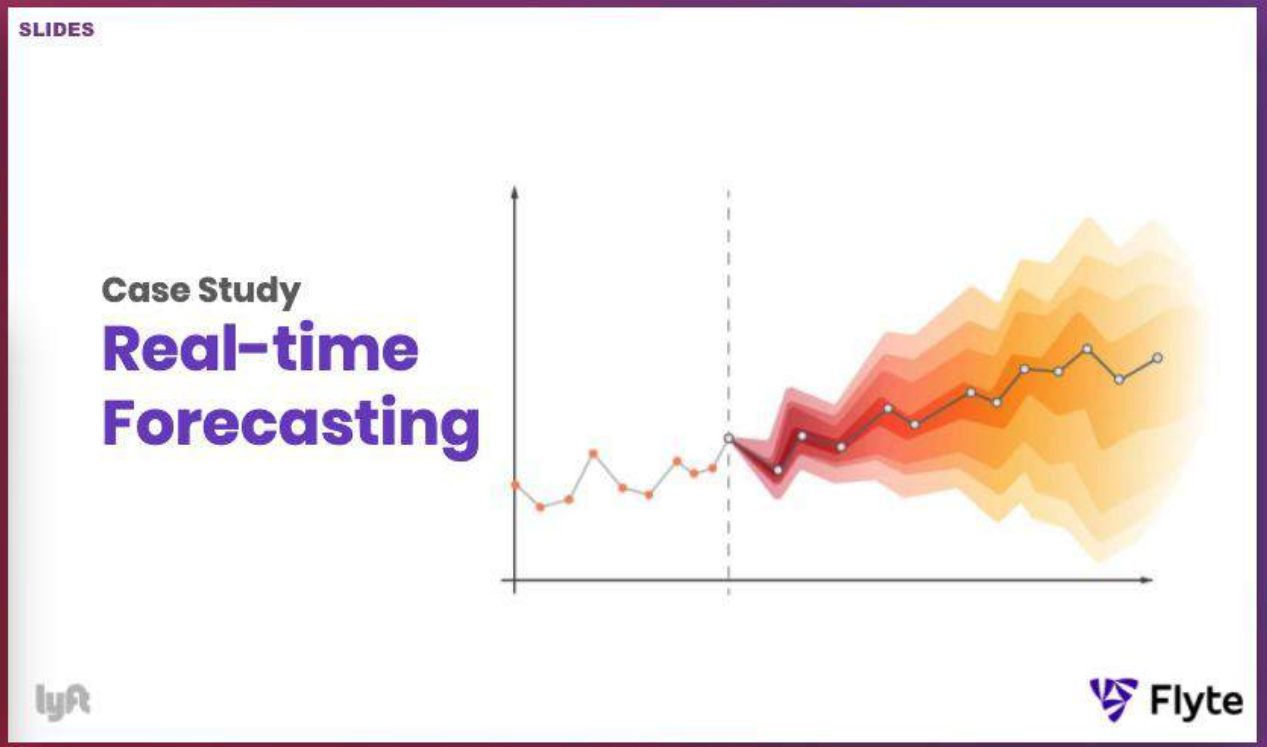
VIDEO

ABSTRACT

Flyte is the backbone for large-scale Machine Learning and Data Processing (ETL) pipelines at Lyft. It is used across business critical applications ranging from ETA, Pricing, Mapping, Autonomous etc. At its core it is a Kubernetes native workflow engine that executes 10M+ containers per month as part of thousands of workflows. We introduced Flyte at Kubecon 2019. This talk will build on the base talk and dive deeper into the architecture and the developer experience. Thus the talk will help existing users of Flyte and new users as well.

Quick overview of Flyte concepts
Architecture of Flyte
Challenges and learnings while using Kubernetes at scale for stateful applications
A deeper look at developer ergonomics and reasons that shaped some decisions
An overview of the community and ecosystem since open sourcing Flyte
The talk will conclude with a demo.

SLIDES



VIDEO

ABSTRACT

Flyte is the backbone for large-scale Machine Learning and Data Processing (ETL) pipelines at Lyft. It is used across business critical applications ranging from ETA, Pricing, Mapping, Autonomous etc. At its core it is a Kubernetes native workflow engine that executes 10M+ containers per month as part of thousands of workflows. We introduced Flyte at Kubecon 2019. This talk will build on the base talk and dive deeper into the architecture and the developer experience. Thus the talk will help existing users of Flyte and new users as well.

Quick overview of Flyte concepts
Architecture of Flyte
Challenges and learnings while using Kubernetes at scale for stateful applications
A deeper look at developer ergonomics and reasons that shaped some decisions
An overview of the community and ecosystem since open sourcing Flyte
The talk will conclude with a demo.

SLIDES

Case Study

Background: Real-Time Forecasts

- ***Business-focused*** streaming infrastructure
- **Offline training and tuning**
- **Model health**
  - Backtests
  - In-prod evaluation
- **Workflow-driven deployments**
  - Integration tests
  - Schema evolution
  - Offline data processes
  - Late-arriving trained artifacts

Flyte

lyft

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

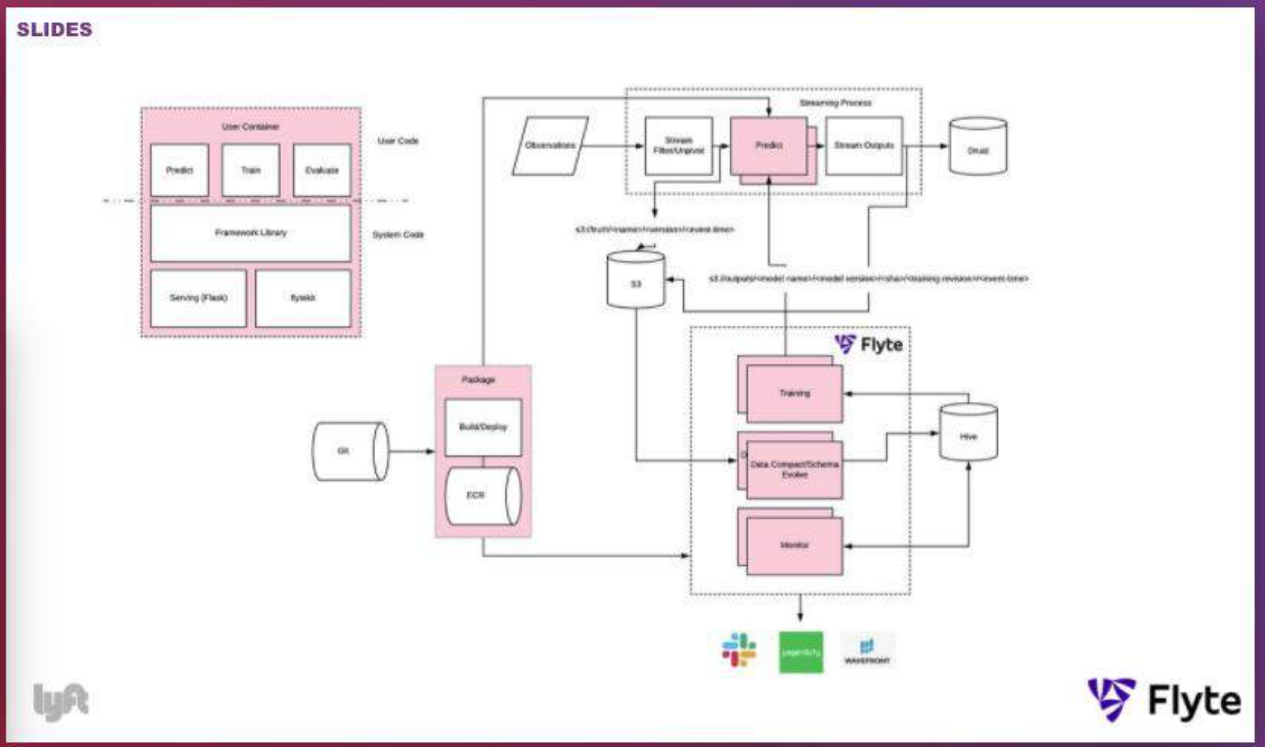
VIDEO

ABSTRACT

Flyte is the backbone for large-scale Machine Learning and Data Processing (ETL) pipelines at Lyft. It is used across business critical applications ranging from ETA, Pricing, Mapping, Autonomous etc. At its core it is a Kubernetes native workflow engine that executes 10M+ containers per month as part of thousands of workflows. We introduced Flyte at Kubecon 2019. This talk will build on the base talk and dive deeper into the architecture and the developer experience. Thus the talk will help existing users of Flyte and new users as well.

Quick overview of Flyte concepts
Architecture of Flyte
Challenges and learnings while using Kubernetes at scale for stateful applications
A deeper look at developer ergonomics and reasons that shaped some decisions
An overview of the community and ecosystem since open sourcing Flyte
The talk will conclude with a demo.

SLIDES



THE LINUX FOUNDATION OPEN SOURCE SUMMIT NORTH AMERICA

Embedded Linux Conference North America

VIDEO

ABSTRACT

Flyte is the backbone for large-scale Machine Learning and Data Processing (ETL) pipelines at Lyft. It is used across business critical applications ranging from ETA, Pricing, Mapping, Autonomous etc. At its core it is a Kubernetes native workflow engine that executes 10M+ containers per month as part of thousands of workflows. We introduced Flyte at Kubecon 2019. This talk will build on the base talk and dive deeper into the architecture and the developer experience. Thus the talk will help existing users of Flyte and new users as well.

Quick overview of Flyte concepts
Architecture of Flyte
Challenges and learnings while using Kubernetes at scale for stateful applications
A deeper look at developer ergonomics and reasons that shaped some decisions
An overview of the community and ecosystem since open sourcing Flyte
The talk will conclude with a demo.

SLIDES

Case Study

User Journey

Ideate & Iterate	Productionize	Retrieve & Replay
1. Write **business logic**	1. Promote a pipeline to **production** (CI/CD)	1. Retrieve results from **executions**
2. Test task locally and **remote**	2. Create **one or more schedules**	2. Identify production errors
3. Orchestrate multiple tasks into a **Workflow**	3. Execute **adhoc**	3. **Replay, reproduce** historical artifacts
4. **Execute** the workflow	4. **Monitor** and get notified	4. Retrieve **artifact lineage**
5. Repeat		

lyft

Flyte

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

ABSTRACT

Flyte is the backbone for large-scale Machine Learning and Data Processing (ETL) pipelines at Lyft. It is used across business critical applications ranging from ETA, Pricing, Mapping, Autonomous etc. At its core it is a Kubernetes native workflow engine that executes 10M+ containers per month as part of thousands of workflows. We introduced Flyte at Kubecon 2019. This talk will build on the base talk and dive deeper into the architecture and the developer experience. Thus the talk will help existing users of Flyte and new users as well.

Quick overview of Flyte concepts
Architecture of Flyte
Challenges and learnings while using Kubernetes at scale for stateful applications
A deeper look at developer ergonomics and reasons that shaped some decisions
An overview of the community and ecosystem since open sourcing Flyte
The talk will conclude with a demo.

SLIDES

Case Study

Ideate: Write Business Logic

Ideate & Iterate | Productionize | Retrieve & Replay

Model Author

```
In [ ]: class TrivialPredictRiders(StreamingModel):

            INPUT_SCHEMA = {
                'ts': datetime.datetime,
                'geohash6': str,
                'rider_cnt': int,
            }

            OUTPUT_SCHEMA = {
                'ts': datetime.datetime,
                'geohash6': str,
                'rider_cnt': float,
            }

            def predict(self, dataframe: pandas.DataFrame) -> pandas.DataFrame:
                # Simply choose the same value as from a day ago in the same place
                dataframe['ts'] += datetime.timedelta(days=1)
                return dataframe

            def train(self, dataframe: pandas.DataFrame) -> dict[str, File]:
                pass
```

lyft

Framework Author

```
class StreamingModel(metaclass=abc.ABCMeta):
    @abc.abstractmethod
    def predict(self, dataframe: pandas.DataFrame) -> pandas.DataFrame:
        pass

    def predict_from_flyte(
        self,
        flyte_input: FlyteTypes.Schema
    ) -> FlyteTypes.Schema:
        df = flyte_input.read_to_dataframe()
        self._assert_inputs(df)
        predict_df = self.predict(df)
        self._assert_outputs(predict_df)
        return FlyteTypes.Schema.from_dataframe(predict_df)
```

Flyte

Flyte: Cloud Native Machine Learning & Data Processing Platform

onlinexperiences.com/scripts/Server.nxp

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

ABSTRACT

Flyte is the backbone for large-scale Machine Learning and Data Processing (ETL) pipelines at Lyft. It is used across business critical applications ranging from ETA, Pricing, Mapping, Autonomous etc. At its core it is a Kubernetes native workflow engine that executes 10M+ containers per month as part of thousands of workflows. We introduced Flyte at Kubecon 2019. This talk will build on the base talk and dive deeper into the architecture and the developer experience. Thus the talk will help existing users of Flyte and new users as well.

Quick overview of Flyte concepts
Architecture of Flyte
Challenges and learnings while using Kubernetes at scale for stateful applications
A deeper look at developer ergonomics and reasons that shaped some decisions
An overview of the community and ecosystem since open sourcing Flyte
The talk will conclude with a demo.

SLIDES

Case Study

Ideate: Framework Abstracts Flyte

Ideate & Iterate | Productionize | Retrieve & Replay

```
@inputs(model_fqdn=FlyteTypes.String, training_revision=FlyteTypes.String, input_data=FlyteTypes.Schema())
@outputs(output_data=FlyteTypes.Schema())
@python_task
def predict_generic(wf_params, model_fqdn, training_revision, input_data, output_data):
    module_name, attr_name = model_fqdn.rsplit('.', -1)
    model_class = getattr(importlib.import_module(module_name), attr_name)
    model = model_class.materialize(training_revision)
    output_data.set(model.predict_from_flyte(input_data))
```

lyft

Flyte

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

ABSTRACT

Flyte is the backbone for large-scale Machine Learning and Data Processing (ETL) pipelines at Lyft. It is used across business critical applications ranging from ETA, Pricing, Mapping, Autonomous etc. At its core it is a Kubernetes native workflow engine that executes 10M+ containers per month as part of thousands of workflows. We introduced Flyte at Kubecon 2019. This talk will build on the base talk and dive deeper into the architecture and the developer experience. Thus the talk will help existing users of Flyte and new users as well.

Quick overview of Flyte concepts
Architecture of Flyte
Challenges and learnings while using Kubernetes at scale for stateful applications
A deeper look at developer ergonomics and reasons that shaped some decisions
An overview of the community and ecosystem since open sourcing Flyte
The talk will conclude with a demo.

SLIDES

Case Study

Ideate: Framework Smooths Tests

Ideate & Iterate | Productionize | Retrieve & Replay

```python
from forecastlib.flyte_tasks import predict_generic
from flytekit.common.workflow_execution import SdkWorkflowExecution

class BaseStreamingModel:
    # Continued from before
    def locally_replay_inference(
        self,
        flyte_inference_id: str,
    ) -> pandas.DataFrame
        # Grab inputs from a failed execution -- links to real production data!
        failed_workflow = SdkWorkflowExecution.fetch(flyte_inference_id)
        inputs = failed_workflow.predict_node.inputs

        # Re-run flyte task in a sandboxed context and return results for analysis
        flyte_output_dict = predict_generic.local_execute(**inputs)
        return flyte_output_dict['output_data'].to_dataframe()
```

lyft

Flyte

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

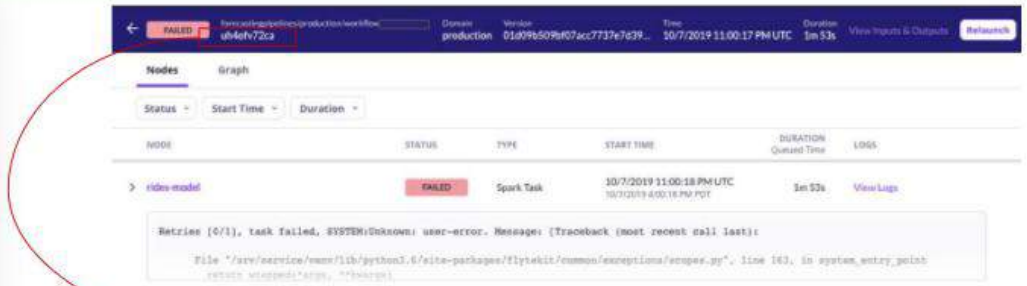
VIDEO

SLIDES

Case Study

Ideate: Rapidly Test Business Logic

Ideate & Iterate | Productionize | Retrieve & Replay



lyft

Flyte

ABSTRACT

Flyte is the backbone for large-scale Machine Learning and Data Processing (ETL) pipelines at Lyft. It is used across business critical applications ranging from ETA, Pricing, Mapping, Autonomous etc. At its core it is a Kubernetes native workflow engine that executes 10M+ containers per month as part of thousands of workflows. We introduced Flyte at Kubecon 2019. This talk will build on the base talk and dive deeper into the architecture and the developer experience. Thus the talk will help existing users of Flyte and new users as well.

Quick overview of Flyte concepts
Architecture of Flyte
Challenges and learnings while using Kubernetes at scale for stateful applications
A deeper look at developer ergonomics and reasons that shaped some decisions
An overview of the community and ecosystem since open sourcing Flyte
The talk will conclude with a demo.

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

ABSTRACT

Flyte is the backbone for large-scale Machine Learning and Data Processing (ETL) pipelines at Lyft. It is used across business critical applications ranging from ETA, Pricing, Mapping, Autonomous etc. At its core it is a Kubernetes native workflow engine that executes 10M+ containers per month as part of thousands of workflows. We introduced Flyte at Kubecon 2019. This talk will build on the base talk and dive deeper into the architecture and the developer experience. Thus the talk will help existing users of Flyte and new users as well.

Quick overview of Flyte concepts
Architecture of Flyte
Challenges and learnings while using Kubernetes at scale for stateful applications
A deeper look at developer ergonomics and reasons that shaped some decisions
An overview of the community and ecosystem since open sourcing Flyte
The talk will conclude with a demo.

SLIDES

Case Study

Ideate: Test/Evaluate at Scale

Ideate & Iterate | Productionize | Retrieve & Replay

Model Author

- Opens Pull Request
- Waits for container build
- User calls test CLI

Framework Author

- Implements generic tasks + workflows
- Tool to register Flyte entities post-build
- Tool to kickoff Flyte test jobs

lyft

Flyte

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

ABSTRACT

Flyte is the backbone for large-scale Machine Learning and Data Processing (ETL) pipelines at Lyft. It is used across business critical applications ranging from ETA, Pricing, Mapping, Autonomous etc. At its core it is a Kubernetes native workflow engine that executes 10M+ containers per month as part of thousands of workflows. We introduced Flyte at Kubecon 2019. This talk will build on the base talk and dive deeper into the architecture and the developer experience. Thus the talk will help existing users of Flyte and new users as well.

Quick overview of Flyte concepts
Architecture of Flyte
Challenges and learnings while using Kubernetes at scale for stateful applications
A deeper look at developer ergonomics and reasons that shaped some decisions
An overview of the community and ecosystem since open sourcing Flyte
The talk will conclude with a demo.

SLIDES

Case Study

Ideate: Test/Evaluate at Scale - Framework

Ideate & Iterate | Productionize | Retrieve & Replay

```
@workflow_class
class TestingWorkflow:
    model_fqdn = Input(Types.String)
    start_time = Input(Types.Datetime)
    end_time = Input(Types.Datetime)

    data_fetch_step = DataFetcherWorkflow(
        model_fqdn=model_fqdn,
        start_time=start_time,
        end_time=end_time,
    )
    training_step = TrainingWorkflow(
        model_fqdn=model_fqdn,
        data=data_fetch_step.outputs.training_data,
    )
    predict_step = predict_generic(
        model_fqdn=model_fqdn,
        training_revision=training_step.outputs.training_revision,
        data=data_fetch_step.outputs.test_data,
    )
    results_step = HealthWorkflow(
        model_fqdn=model_fqdn,
        prediction=predict_step.outputs.predictions,
        truth=data_fetch_step.outputs.test_truth,
    )
    scores = Output(results_step.outputs.scores)
```

```
@click.argument('container_sha')
@click.command()
def register_all(container_sha):
    predict_generic.register(container_sha)
    DataFetcherWorkflow.register(container_sha)
    TrainingWorkflow.register(container_sha)
    HealthWorkflow.register(container_sha)
    TestingWorkflow.register(container_sha)
```

lyft

Flyte

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

ABSTRACT

Flyte is the backbone for large-scale Machine Learning and Data Processing (ETL) pipelines at Lyft. It is used across business critical applications ranging from ETA, Pricing, Mapping, Autonomous etc. At its core it is a Kubernetes native workflow engine that executes 10M+ containers per month as part of thousands of workflows. We introduced Flyte at Kubecon 2019. This talk will build on the base talk and dive deeper into the architecture and the developer experience. Thus the talk will help existing users of Flyte and new users as well.

Quick overview of Flyte concepts
Architecture of Flyte
Challenges and learnings while using Kubernetes at scale for stateful applications
A deeper look at developer ergonomics and reasons that shaped some decisions
An overview of the community and ecosystem since open sourcing Flyte
The talk will conclude with a demo.

SLIDES

Case Study

Ideate: Test/Evaluate at Scale - Go!

Ideate & Iterate → Productionize → Retrieve & Replay

Model Author

```
forecast-cli test models.SimpleModel a23d4c --start-date=2019-01-01 --end-date=2020-01-01
```

```
Executing test...
Results:
{'avg_daily_error': 0.03235,
 'daily_error': {'FRI': 0.0321,
                 'MON': 0.0343,
                 'SAT': 0.0231,
                 'SUN': 0.0192,
                 'THR': 0.0282,
                 'TUE': 0.0423,
                 'WED': 0.0292},
 'daily_error_variance': 0.04248,
 'days_analyzed': 365}
```

lyft

Framework Author

```
from flytekit.common.sdk_launch_plan import SdkLaunchPlan

@click.command()
@click.argument('model_fqdn')
@click.argument('container_sha')
@click.option('start-time', type=click.DateTime(formats=["%Y-%m-%d"]))
@click.option('end-time', type=click.DateTime(formats=["%Y-%m-%d"]))
def test(model_fqdn, container_sha, start_time, end_time):
    wf = SdkLaunchPlan.fetch('TrainingWorkflow', container_sha)
    execution = wf.execute(
        {
            'model_fqdn': model_fqdn,
            'start_time': start_time,
            'end_time': end_time,
        }
    )
    execution.wait_for_completion()
    click.echo(execution.outputs.scores)
```

Flyte

Flyte: Cloud Native Machine Learning & Data Processing Platform

onlinexperiences.com/scripts/Server.nxp

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

ABSTRACT

Flyte is the backbone for large-scale Machine Learning and Data Processing (ETL) pipelines at Lyft. It is used across business critical applications ranging from ETA, Pricing, Mapping, Autonomous etc. At its core it is a Kubernetes native workflow engine that executes 10M+ containers per month as part of thousands of workflows. We introduced Flyte at Kubecon 2019. This talk will build on the base talk and dive deeper into the architecture and the developer experience. Thus the talk will help existing users of Flyte and new users as well.

Quick overview of Flyte concepts
Architecture of Flyte
Challenges and learnings while using Kubernetes at scale for stateful applications
A deeper look at developer ergonomics and reasons that shaped some decisions
An overview of the community and ecosystem since open sourcing Flyte
The talk will conclude with a demo.

SLIDES

Case Study

Productionize

Ideate & Iterate | Productionize | Retrieve & Replay

Model Author

- Checks-in code
- Waits for integration tests to pass
- Rolls out stage-by-stage
- Monitors dashboards/alarms

Framework Author

- Integration test harness
- Release tools
  - Automated training
  - Offline-data compaction
  - Health monitoring
- Programmatic alarms and dashboards

lyft

Flyte

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

ABSTRACT

Flyte is the backbone for large-scale Machine Learning and Data Processing (ETL) pipelines at Lyft. It is used across business critical applications ranging from ETA, Pricing, Mapping, Autonomous etc. At its core it is a Kubernetes native workflow engine that executes 10M+ containers per month as part of thousands of workflows. We introduced Flyte at Kubecon 2019. This talk will build on the base talk and dive deeper into the architecture and the developer experience. Thus the talk will help existing users of Flyte and new users as well.

Quick overview of Flyte concepts
Architecture of Flyte
Challenges and learnings while using Kubernetes at scale for stateful applications
A deeper look at developer ergonomics and reasons that shaped some decisions
An overview of the community and ecosystem since open sourcing Flyte
The talk will conclude with a demo.

SLIDES

Case Study

Productionize: Integration

Ideate & Iterate | Productionize | Retrieve & Replay

```
def integration_test(model_fqdn: str, model_config: dict[str, Any], container_sha: str) -> None:
    start_time = datetime.utcnow() - model_config['integration_window_lag']
    end_time = start_time + model_config['integretion_window_length']

    wf = TestingWorkflow.fetch(container_sha)
    execution = wf.execute(
        {
            'model_fqdn': model_fqdn,
            'start_time': start_time,
            'end_time': end_time,
        }
    )
    execution.wait_for_completion()

    assert _scores_pass_model_config(execution.outputs.scores, model_config)
```

lyft

Flyte

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

ABSTRACT

Flyte is the backbone for large-scale Machine Learning and Data Processing (ETL) pipelines at Lyft. It is used across business critical applications ranging from ETA, Pricing, Mapping, Autonomous etc. At its core it is a Kubernetes native workflow engine that executes 10M+ containers per month as part of thousands of workflows. We introduced Flyte at Kubecon 2019. This talk will build on the base talk and dive deeper into the architecture and the developer experience. Thus the talk will help existing users of Flyte and new users as well.

Quick overview of Flyte concepts
Architecture of Flyte
Challenges and learnings while using Kubernetes at scale for stateful applications
A deeper look at developer ergonomics and reasons that shaped some decisions
An overview of the community and ecosystem since open sourcing Flyte
The talk will conclude with a demo.

SLIDES

Case Study

Productionize: Release

Ideate & Iterate | Productionize | Retrieve & Replay

```python
from flytekit.common.sdk_workflow import SdkWorkflow

def release(model_fqdn: str, model_config: dict[str, Any], container_sha: str) -> None:
    training_workflow = SdkWorkflow.fetch('TrainingWorkflow', container_sha)
    for training_config in model_config['training_pipelines']:
        lp = training_workflow.create_launch_plan(
            schedule=schedules.CronSchedule(training_config['cron_expr']),
            fixed_inputs=training_config['params'],
        )
        lp.register(training_config['name'], container_sha)
        lp.update_launch_plan(LaunchPlanState.ACTIVE)

    offline_data_organizer = SdkWorkflow.fetch('OfflineDataOrganizer', container_sha)
    lp = offline_data_organizer.create_launch_plan(
        schedule=schedules.CronSchedule(model_config['offline_data_latency_cron']),
        fixed_inputs=model_config['data_schema_params'],
    )
    lp.update_launch_plan(LaunchPlanState.ACTIVE)

    # ... additional offline processes for health checking, cron-based signals, etc.
```

lyft

Flyte

Flyte: Cloud Native Machine Learning & Data Processing Platform

onlinexperiences.com/scripts/Server.nxp

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

ABSTRACT

Flyte is the backbone for large-scale Machine Learning and Data Processing (ETL) pipelines at Lyft. It is used across business critical applications ranging from ETA, Pricing, Mapping, Autonomous etc. At its core it is a Kubernetes native workflow engine that executes 10M+ containers per month as part of thousands of workflows. We introduced Flyte at Kubecon 2019. This talk will build on the base talk and dive deeper into the architecture and the developer experience. Thus the talk will help existing users of Flyte and new users as well.

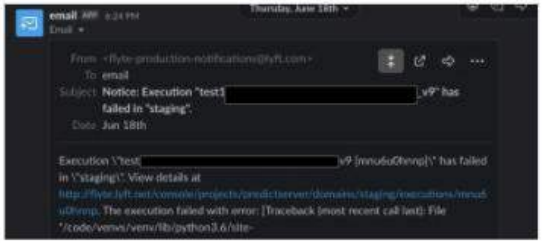
Quick overview of Flyte concepts
Architecture of Flyte
Challenges and learnings while using Kubernetes at scale for stateful applications
A deeper look at developer ergonomics and reasons that shaped some decisions
An overview of the community and ecosystem since open sourcing Flyte
The talk will conclude with a demo.

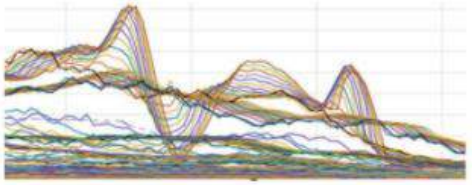
SLIDES

Case Study

Productionize: Monitor

Ideate & Iterate | Productionize | Retrieve & Replay





pagerduty Superset WAVEFRONT

lyft

Flyte

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

ABSTRACT

Flyte is the backbone for large-scale Machine Learning and Data Processing (ETL) pipelines at Lyft. It is used across business critical applications ranging from ETA, Pricing, Mapping, Autonomous etc. At its core it is a Kubernetes native workflow engine that executes 10M+ containers per month as part of thousands of workflows. We introduced Flyte at Kubecon 2019. This talk will build on the base talk and dive deeper into the architecture and the developer experience. Thus the talk will help existing users of Flyte and new users as well.

Quick overview of Flyte concepts
Architecture of Flyte
Challenges and learnings while using Kubernetes at scale for stateful applications
A deeper look at developer ergonomics and reasons that shaped some decisions
An overview of the community and ecosystem since open sourcing Flyte
The talk will conclude with a demo.

SLIDES

Case Study

Retrieve and Replay

Ideate & Iterate | Productionize | Retrieve & Replay

Model Author

- Notices anomaly in behavior
- Fetches exact input data
- If desired, fetches exact container
- Go back to 'Ideate & Iterate'

Framework Author

- One-click tools for common debug paths.
  - Retry inference
  - Compare inferences over time

lyft

Flyte

Flyte: Cloud Native Machine Learning & Data Processing Platform

onlinexperiences.com/scripts/Server.nxp

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

ABSTRACT

Flyte is the backbone for large-scale Machine Learning and Data Processing (ETL) pipelines at Lyft. It is used across business critical applications ranging from ETA, Pricing, Mapping, Autonomous etc. At its core it is a Kubernetes native workflow engine that executes 10M+ containers per month as part of thousands of workflows. We introduced Flyte at Kubecon 2019. This talk will build on the base talk and dive deeper into the architecture and the developer experience. Thus the talk will help existing users of Flyte and new users as well.

Quick overview of Flyte concepts
Architecture of Flyte
Challenges and learnings while using Kubernetes at scale for stateful applications
A deeper look at developer ergonomics and reasons that shaped some decisions
An overview of the community and ecosystem since open sourcing Flyte
The talk will conclude with a demo.

SLIDES

Case Study

Retrieve and Replay: Isolated Failures

Ideate & Iterate | Productionize | Retrieve & Replay

Model Author

```
forecast-cli debug failure --inference-id c21a8c
```

```
In [ ]: # THE CODE IN THIS CELL IS GENERATED!
from models import TrivialPredictRiders

TRAINING_REVISION = 'f12e1c'
INFERENCE_ID = 'c21a8c'

model = TrivialPredictRiders.create_from_trained_artifacts(
    TRAINING_REVISION
)
input_dataframe = model.fetch_inference_inputs(INFERENCE_ID)
output_dataframe = model.fetch_inference_outputs(INFERENCE_ID)
```

lyft

Framework Author

```
from flytekit.common.workflow_execution import SdkWorkflowExecution
from flytekit.common.launch_plan import SdkLaunchPlan
from flytekit.common.tasks.task import SdkTask
from flytekit.common.workflow import SdkWorkflow

def get_debug_params(inference_id: str) -> dict[str, Any]:
    flyte_execution_id = _map_inference_to_flyte_id(inference_id)
    execution = SdkWorkflowExecution.fetch(flyte_execution_id)
    executed_launch_plan = SdkLaunchPlan.fetch(
        execution.spec.launch_plan
    )
    executed_workflow = SdkWorkflow.fetch(
        executed_launch_plan.spec.workflow_id
    )
    executed_task = SdkTask.fetch(
        executed_workflow.nodes['predict-node'].task_node.reference_id
    )

    model_fqds = execution.inputs.model_fqds
    training_revision = ...

    return {
        'model_fqds': execution.inputs.model_fqds,
        'docker_image': executed_task.container.image,
        'training_revision': execution.inputs.training_revision,
    }
```

Flyte

Flyte: Cloud Native Machine Learning & Data Processing Platform

onlinexperiences.com/scripts/Server.nxp

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

ABSTRACT

Flyte is the backbone for large-scale Machine Learning and Data Processing (ETL) pipelines at Lyft. It is used across business critical applications ranging from ETA, Pricing, Mapping, Autonomous etc. At its core it is a Kubernetes native workflow engine that executes 10M+ containers per month as part of thousands of workflows. We introduced Flyte at Kubecon 2019. This talk will build on the base talk and dive deeper into the architecture and the developer experience. Thus the talk will help existing users of Flyte and new users as well.

Quick overview of Flyte concepts
Architecture of Flyte
Challenges and learnings while using Kubernetes at scale for stateful applications
A deeper look at developer ergonomics and reasons that shaped some decisions
An overview of the community and ecosystem since open sourcing Flyte
The talk will conclude with a demo.

SLIDES

Demo

lyft

Flyte

Flyte: Cloud Native Machine Learning & Data Processing Platform

onlinexperiences.com/scripts/Server.nxp

SCREEN SHARE

localhost:8888/notebooks/raw-container.ipynb#

jupyter raw-container (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

Step I: Run the filter_edges example within Jupyter Notebook First

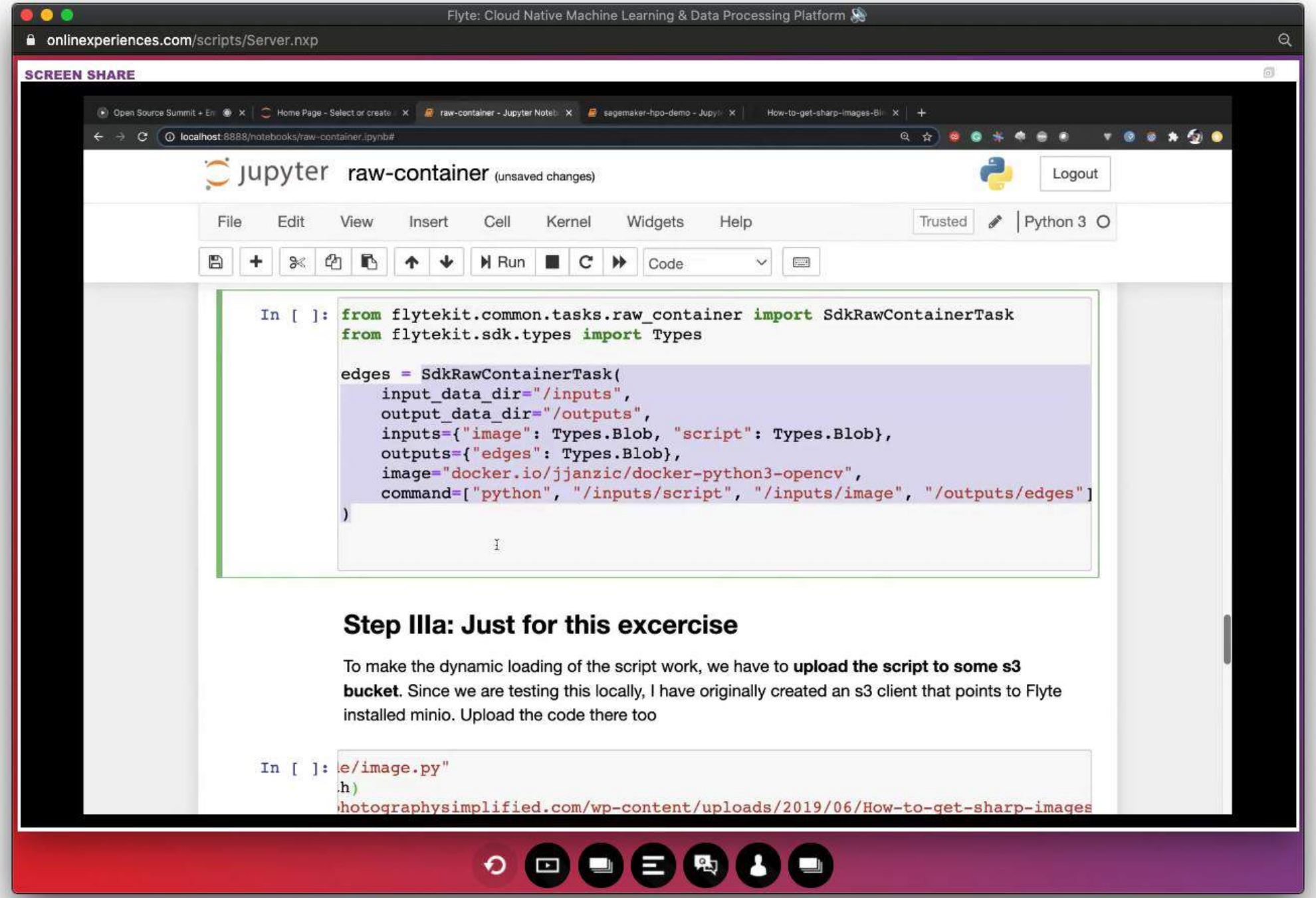
In [25]:
```
from image import filter_edges
```

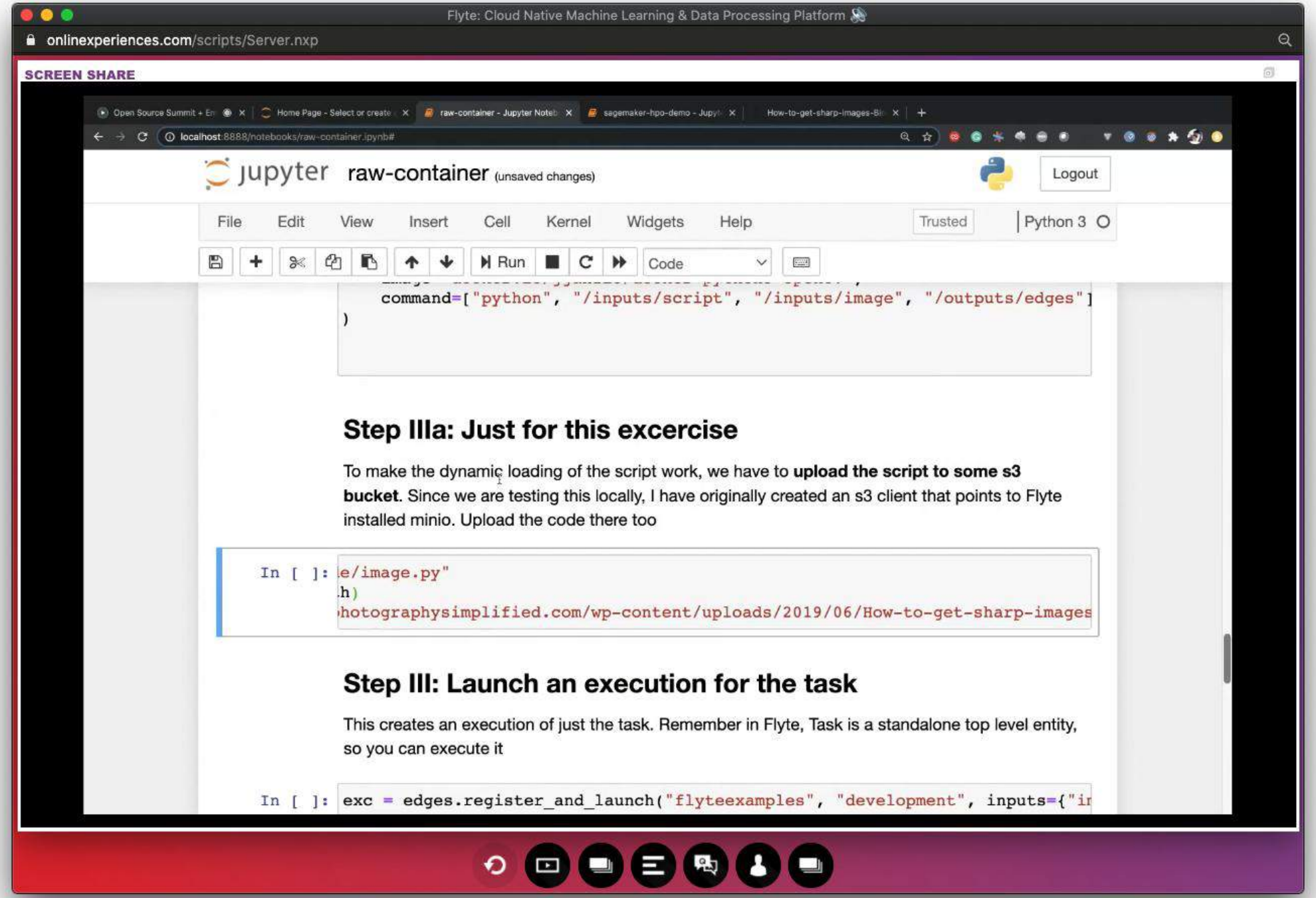
In [ ]:
```
inp = "image.png" # This should be a path to an image that is available loc
out = "edges.png" # This is the path to where we want to create the image
filter_edges(inp, out)
# Invoke the helper method that displays the image in Jupyter
display_images([inp, out])
```

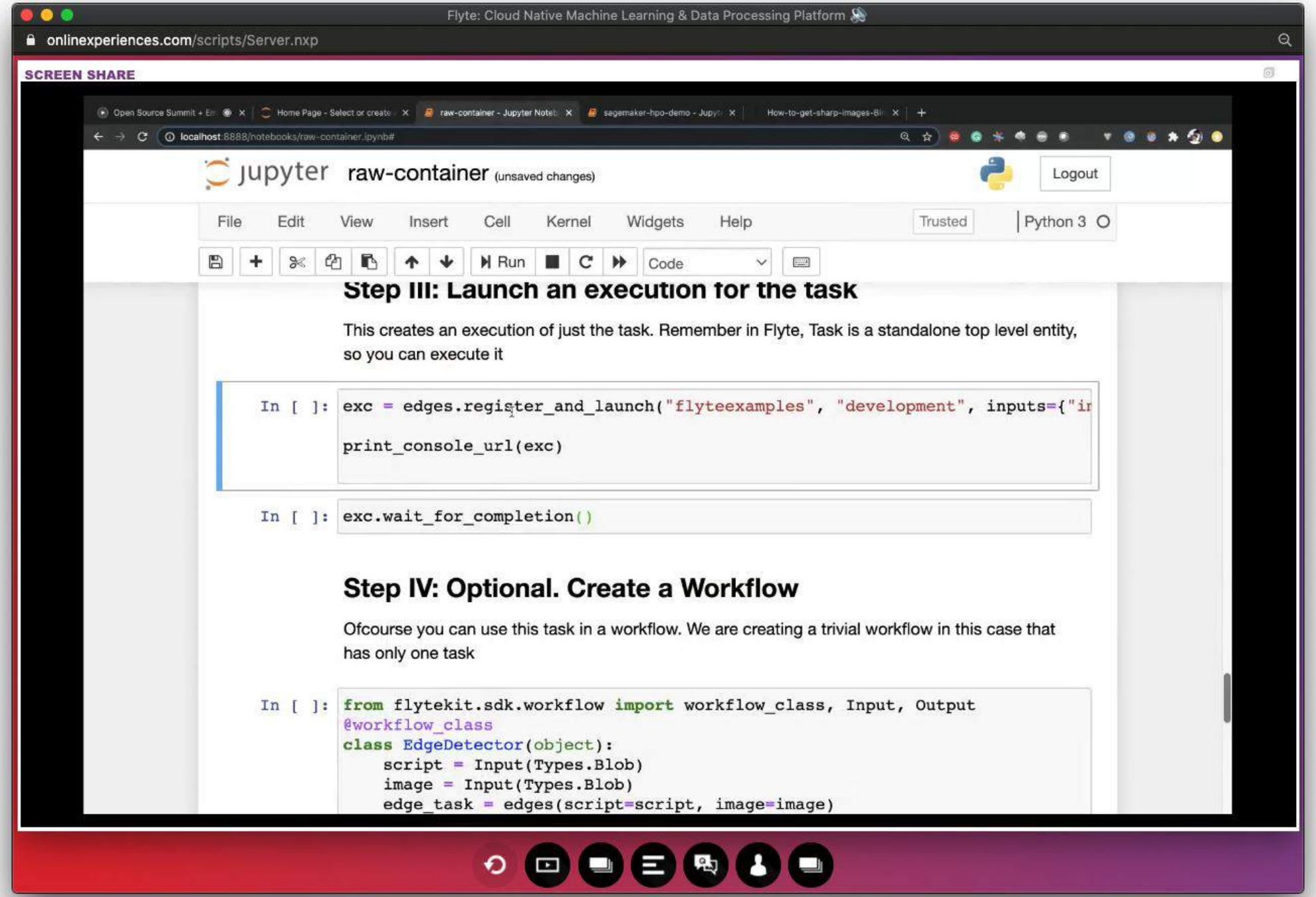
Step II: Create a Flyte Task for this function

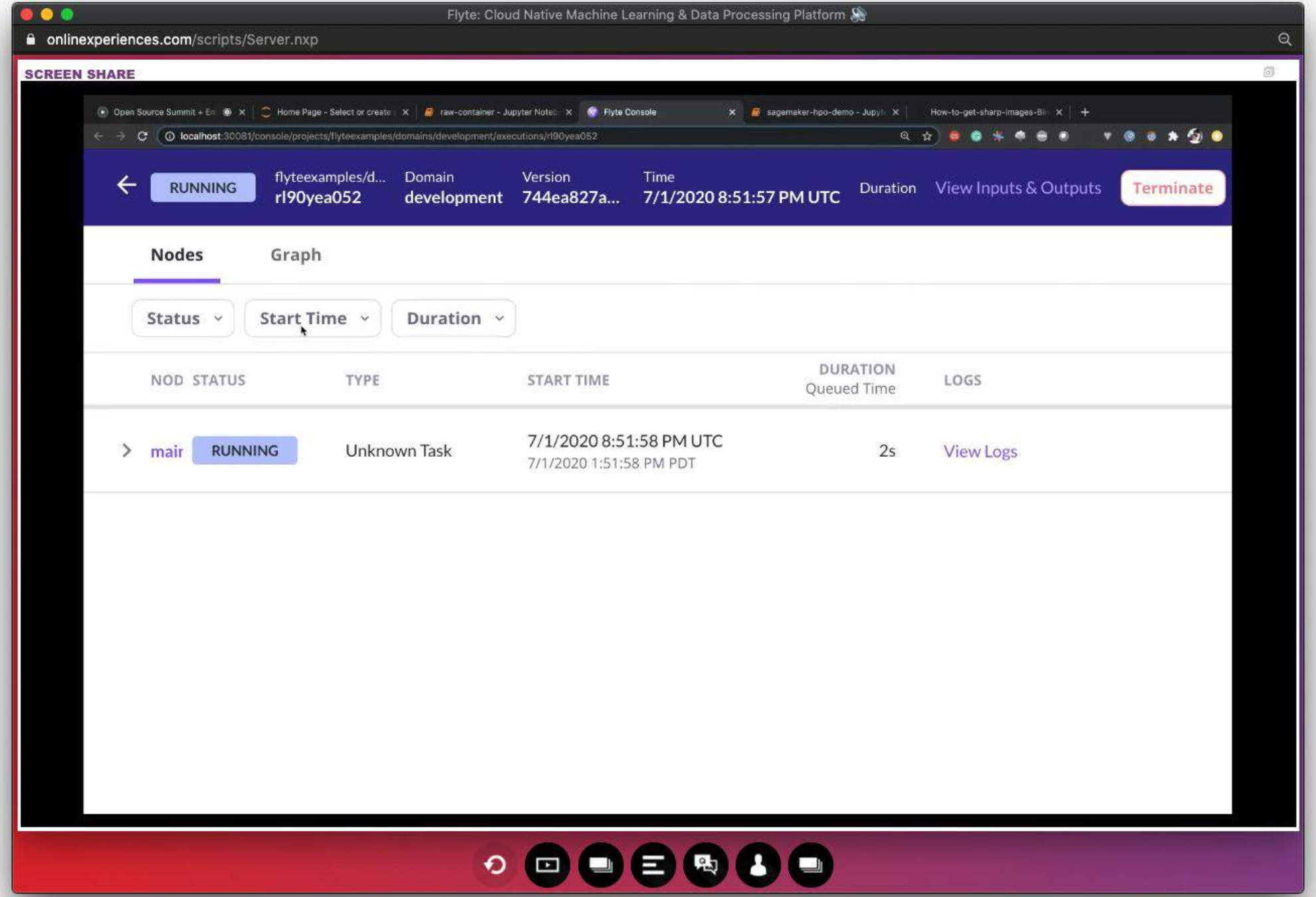
In this case we will use an SdkRawContainerTask. A raw container task is essentially a container task, where we tell Flyte, that this container does not have flytekit. So all the inputs and outputs should be auto-mounted and uploaded. The task can use an open source container `docker.io/jjanzic/docker-python3-opencv`. This container has python and OpenCV already installed.

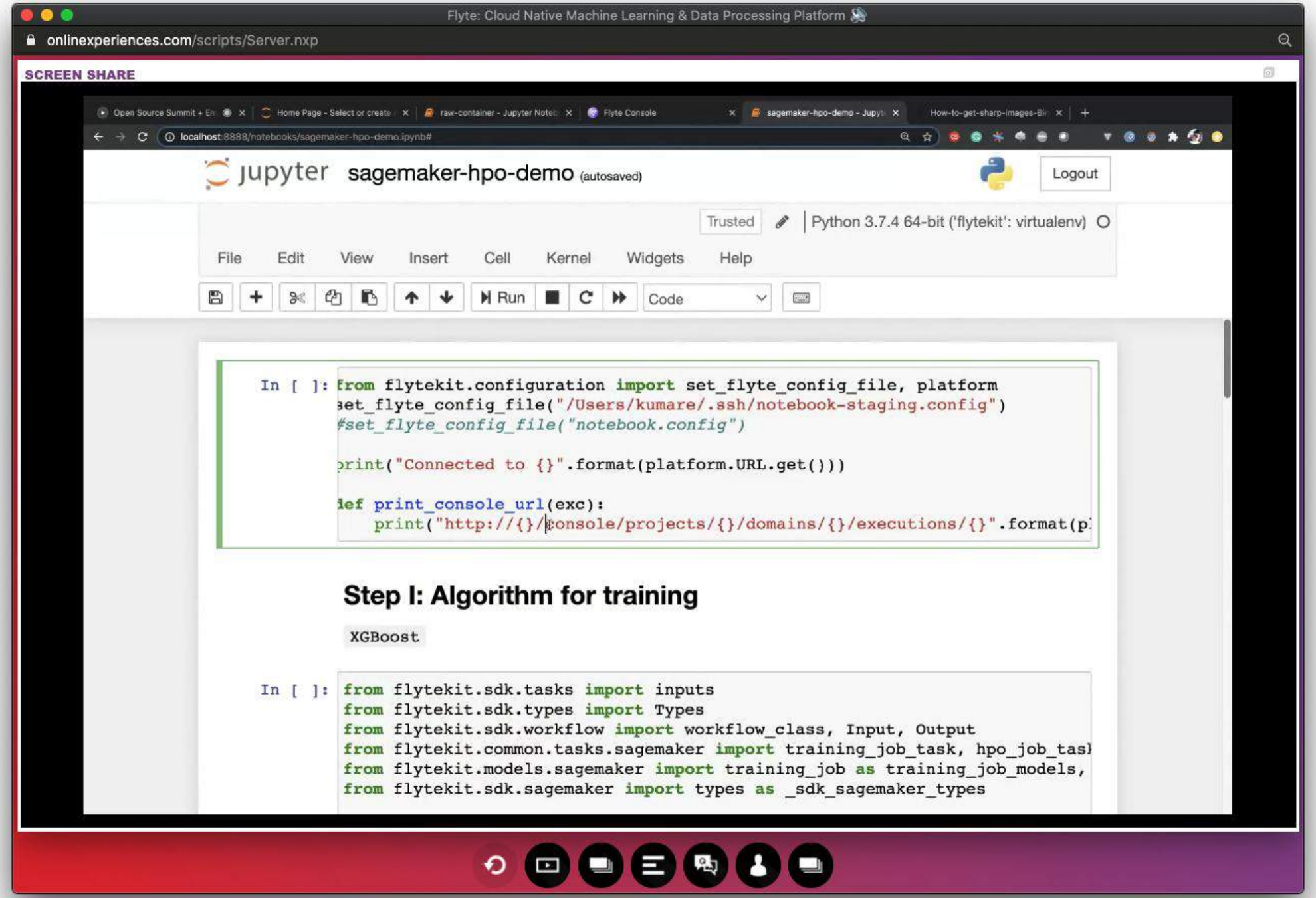
Hot load the code

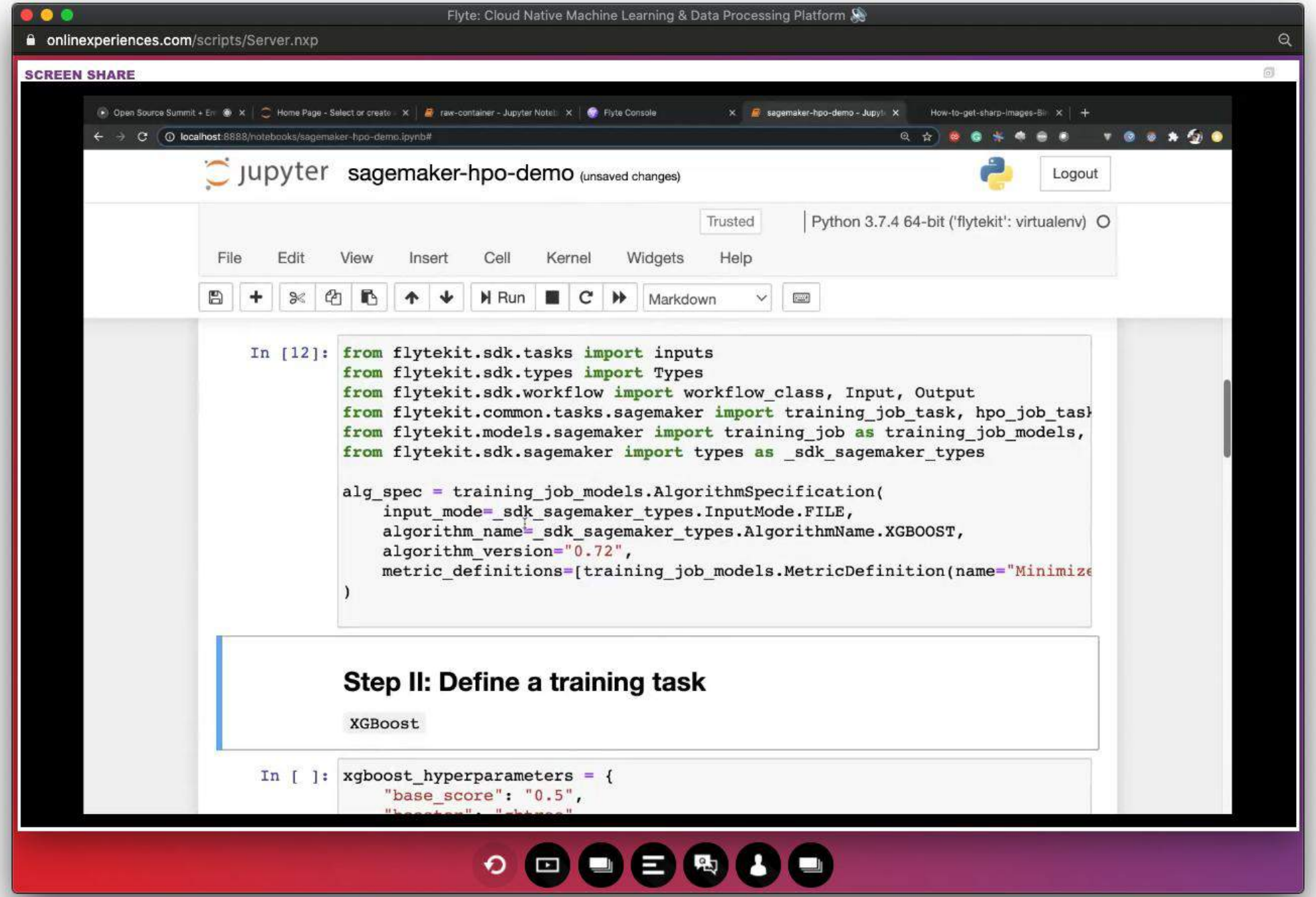

Flyte: Cloud Native Machine Learning & Data Processing Platform
onlinexperiences.com/scripts/Server.nxp
SCREEN SHARE
localhost:8888/notebooks/raw-container.ipynb#
jupyter raw-container (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
In [ ]:
from flytekit.common.tasks.raw_container import SdkRawContainerTask
from flytekit.sdk.types import Types
edges = SdkRawContainerTask(
input_data_dir="/inputs",
output_data_dir="/outputs",
inputs={"image": Types.Blob, "script": Types.Blob},
outputs={"edges": Types.Blob},
image="docker.io/jjanzic/docker-python3-opencv",
command=["python", "/inputs/script", "/inputs/image", "/outputs/edges"]
)
Step IIIa: Just for this excercise
To make the dynamic loading of the script work, we have to upload the script to some s3 bucket. Since we are testing this locally, I have originally created an s3 client that points to Flyte installed minio. Upload the code there too
In [ ]:
e/image.py"
h)
hotographysimplified.com/wp-content/uploads/2019/06/How-to-get-sharp-images

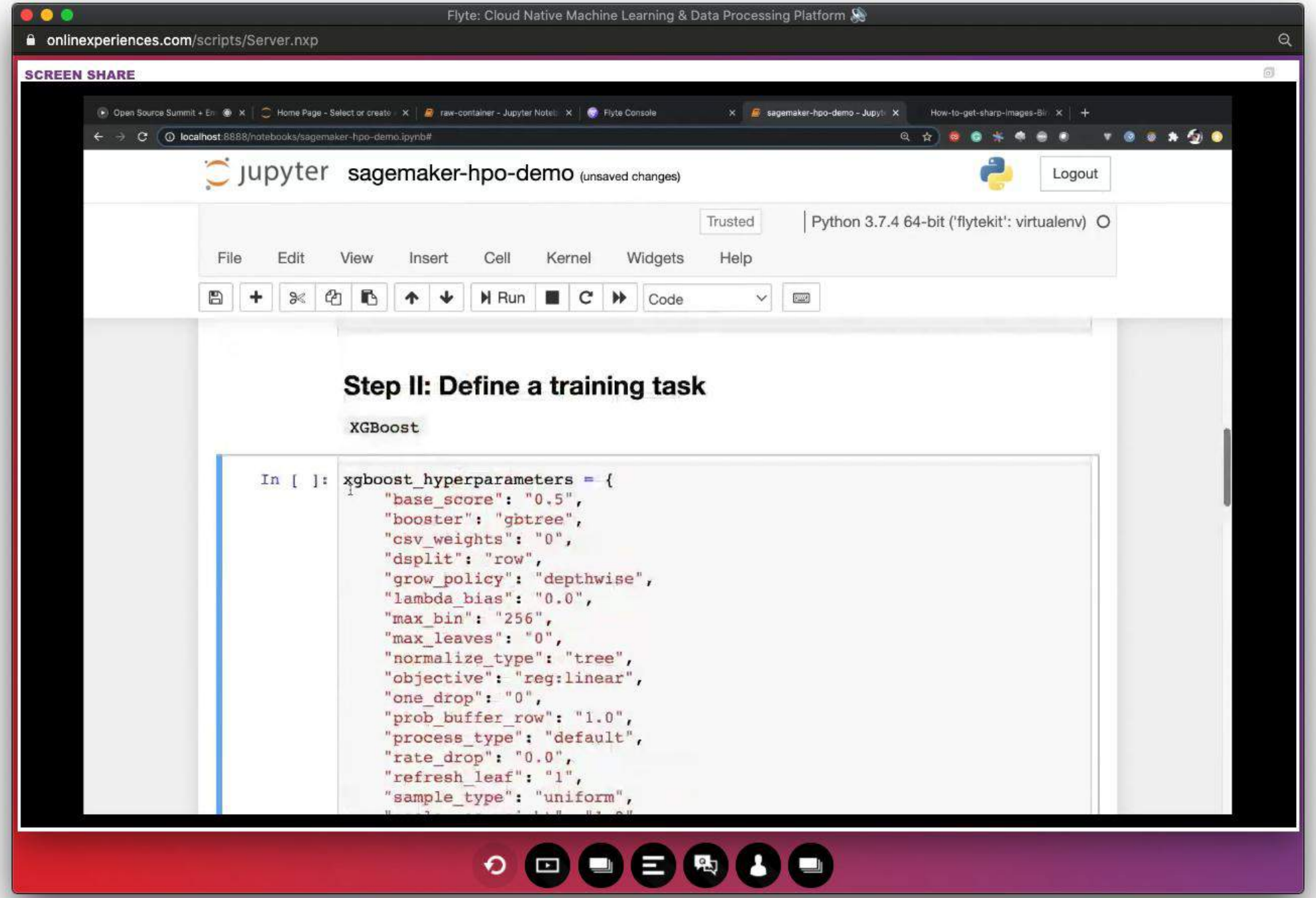

Flyte: Cloud Native Machine Learning & Data Processing Platform
onlinexperiences.com/scripts/Server.nxp
SCREEN SHARE
localhost:8888/notebooks/raw-container.ipynb#
jupyter raw-container (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
command=["python", "/inputs/script", "/inputs/image", "/outputs/edges"]
)
Step IIIa: Just for this excercise
To make the dynamic loading of the script work, we have to upload the script to some s3 bucket. Since we are testing this locally, I have originally created an s3 client that points to Flyte installed minio. Upload the code there too
In [ ]:
e/image.py"
h)
hotographysimplified.com/wp-content/uploads/2019/06/How-to-get-sharp-images
Step III: Launch an execution for the task
This creates an execution of just the task. Remember in Flyte, Task is a standalone top level entity, so you can execute it
In [ ]:
exc = edges.register_and_launch("flyteexamples", "development", inputs={"ir

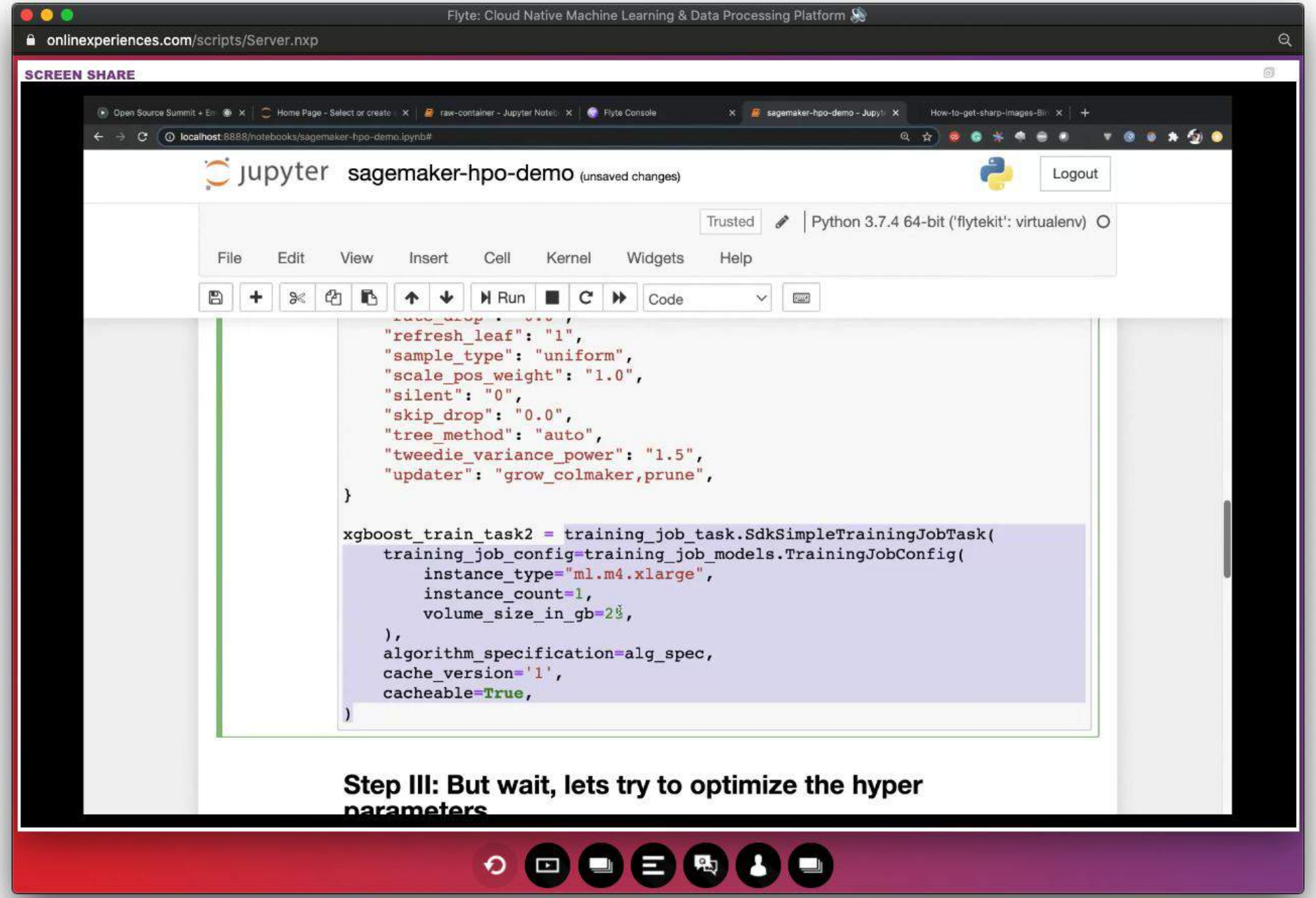

Flyte: Cloud Native Machine Learning & Data Processing Platform
onlinexperiences.com/scripts/Server.nxp
SCREEN SHARE
localhost:8888/notebooks/raw-container.ipynb#
jupyter raw-container (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Step III: Launch an execution for the task
This creates an execution of just the task. Remember in Flyte, Task is a standalone top level entity, so you can execute it
In [ ]:
exc = edges.register_and_launch("flyteexamples", "development", inputs={"ir
print_console_url(exc)
In [ ]:
exc.wait_for_completion()
Step IV: Optional. Create a Workflow
Ofcourse you can use this task in a workflow. We are creating a trivial workflow in this case that has only one task
In [ ]:
from flytekit.sdk.workflow import workflow_class, Input, Output
@workflow_class
class EdgeDetector(object):
script = Input(Types.Blob)
image = Input(Types.Blob)
edge_task = edges(script=script, image=image)

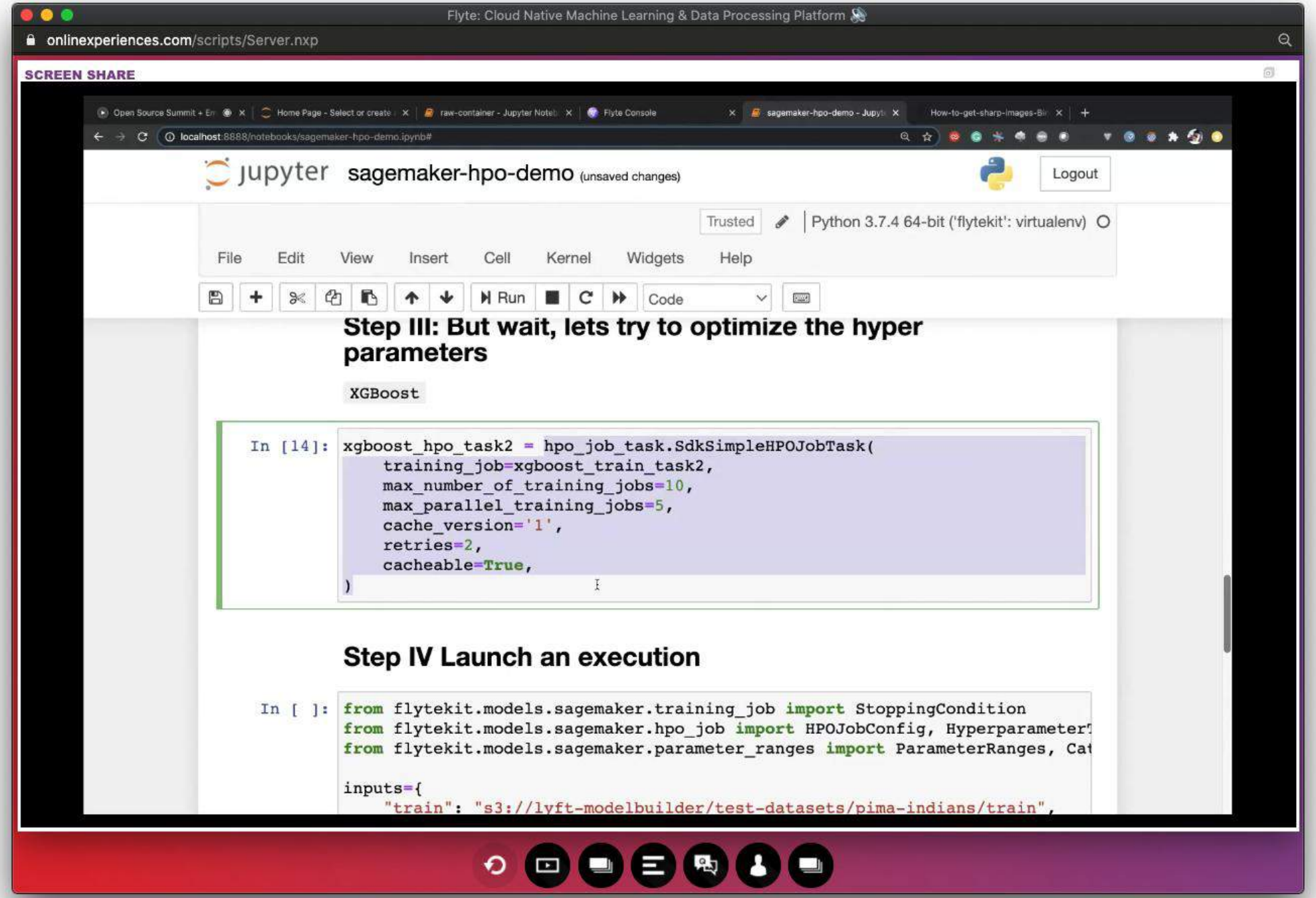

Flyte: Cloud Native Machine Learning & Data Processing Platform
onlinexperiences.com/scripts/Server.nxp
SCREEN SHARE
Open Source Summit + En
Home Page - Select or create
raw-container - Jupyter Noteb
Flyte Console
sagemaker-hpo-demo - Jupyt
How-to-get-sharp-images-Bi
localhost:30081/console/projects/flyteexamples/domains/development/executions/rl90yea052
RUNNING
flyteexamples/d...
rl90yea052
Domain
development
Version
744ea827a...
Time
7/1/2020 8:51:57 PM UTC
Duration
View Inputs & Outputs
Terminate
Nodes
Graph
Status
Start Time
Duration
NOD
STATUS
TYPE
START TIME
DURATION
Queued Time
LOGS
mair
RUNNING
Unknown Task
7/1/2020 8:51:58 PM UTC
7/1/2020 1:51:58 PM PDT
2s
View Logs

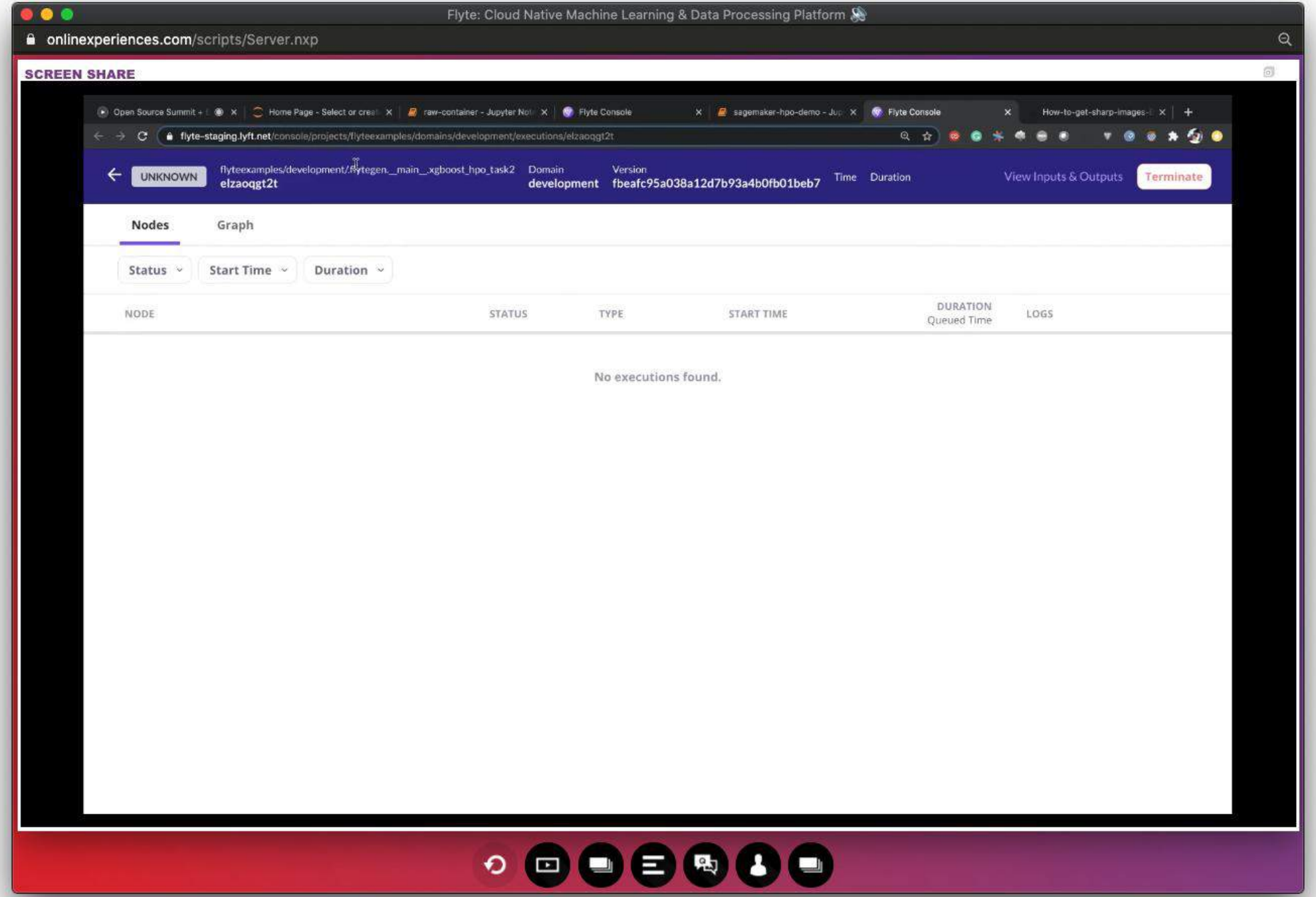

Flyte: Cloud Native Machine Learning & Data Processing Platform
onlinexperiences.com/scripts/Server.nxp
SCREEN SHARE
localhost:8888/notebooks/sagemaker-hpo-demo.ipynb#
jupyter sagemaker-hpo-demo (autosaved)
Logout
Trusted
Python 3.7.4 64-bit ('flytekit': virtualenv)
File Edit View Insert Cell Kernel Widgets Help
Run
Code
In [ ]:
from flytekit.configuration import set_flyte_config_file, platform
set_flyte_config_file("/Users/kumare/.ssh/notebook-staging.config")
#set_flyte_config_file("notebook.config")
print("Connected to {}".format(platform.URL.get()))
def print_console_url(exc):
print("http://{}/console/projects/{}/domains/{}/executions/{}".format(p
Step I: Algorithm for training
XGBoost
In [ ]:
from flytekit.sdk.tasks import inputs
from flytekit.sdk.types import Types
from flytekit.sdk.workflow import workflow_class, Input, Output
from flytekit.common.tasks.sagemaker import training_job_task, hpo_job_task
from flytekit.models.sagemaker import training_job as training_job_models,
from flytekit.sdk.sagemaker import types as _sdk_sagemaker_types

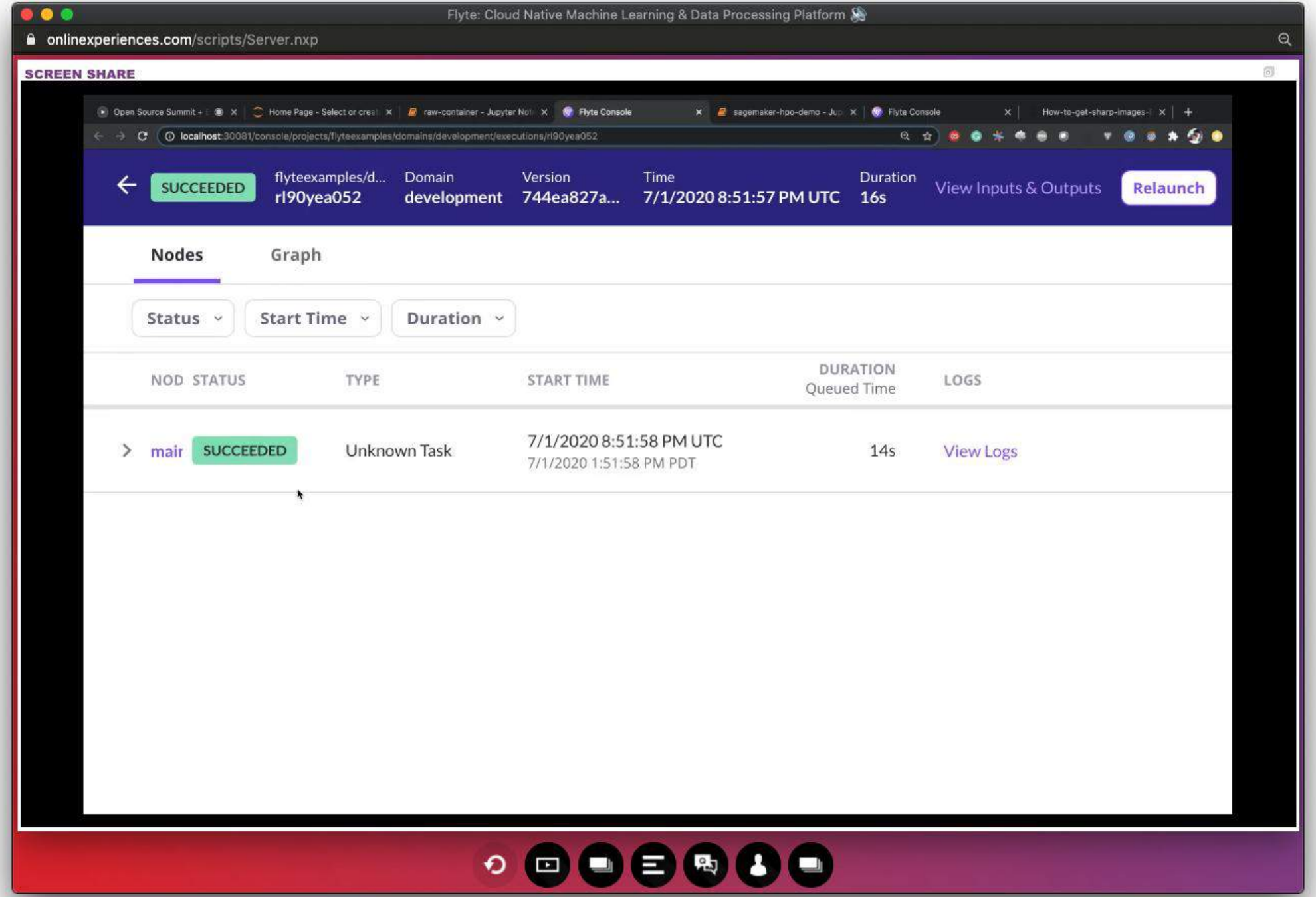

Flyte: Cloud Native Machine Learning & Data Processing Platform
onlinexperiences.com/scripts/Server.nxp
SCREEN SHARE
localhost:8888/notebooks/sagemaker-hpo-demo.ipynb#
jupyter sagemaker-hpo-demo (unsaved changes)
Logout
Trusted
Python 3.7.4 64-bit ('flytekit': virtualenv)
File Edit View Insert Cell Kernel Widgets Help
Run
Markdown
In [12]:
from flytekit.sdk.tasks import inputs
from flytekit.sdk.types import Types
from flytekit.sdk.workflow import workflow_class, Input, Output
from flytekit.common.tasks.sagemaker import training_job_task, hpo_job_task
from flytekit.models.sagemaker import training_job as training_job_models,
from flytekit.sdk.sagemaker import types as _sdk_sagemaker_types
alg_spec = training_job_models.AlgorithmSpecification(
input_mode=_sdk_sagemaker_types.InputMode.FILE,
algorithm_name=_sdk_sagemaker_types.AlgorithmName.XGBOOST,
algorithm_version="0.72",
metric_definitions=[training_job_models.MetricDefinition(name="Minimize
)
Step II: Define a training task
XGBoost
In [ ]:
xgboost_hyperparameters = {
"base_score": "0.5",

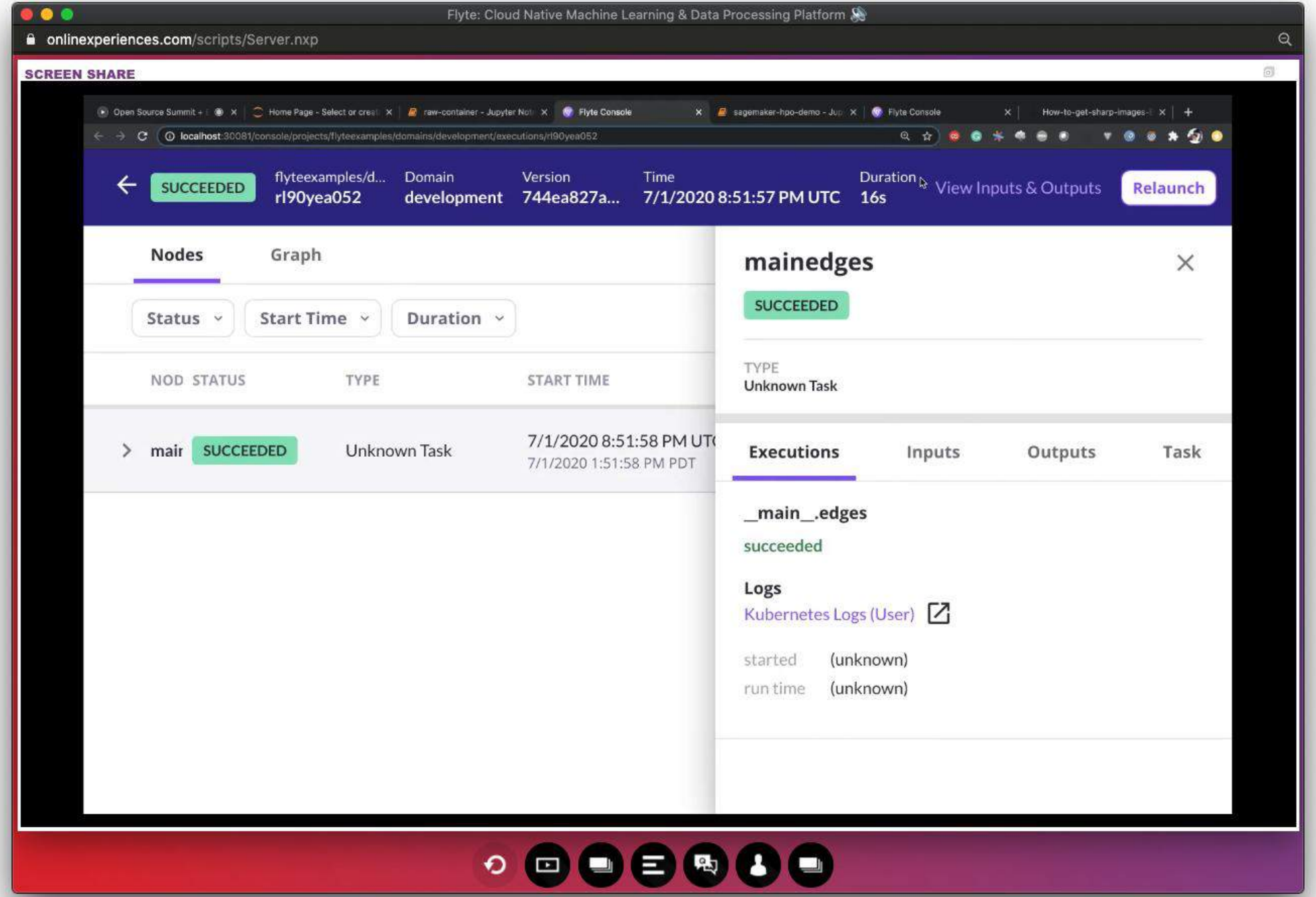

Flyte: Cloud Native Machine Learning & Data Processing Platform
onlinexperiences.com/scripts/Server.nxp
SCREEN SHARE
localhost:8888/notebooks/sagemaker-hpo-demo.ipynb#
jupyter sagemaker-hpo-demo (unsaved changes)
Logout
Trusted
Python 3.7.4 64-bit ('flytekit': virtualenv)
File Edit View Insert Cell Kernel Widgets Help
Run
Code
Step II: Define a training task
XGBoost
In [ ]:
xgboost_hyperparameters = {
"base_score": "0.5",
"booster": "gbtree",
"csv_weights": "0",
"dsplit": "row",
"grow_policy": "depthwise",
"lambda_bias": "0.0",
"max_bin": "256",
"max_leaves": "0",
"normalize_type": "tree",
"objective": "reg:linear",
"one_drop": "0",
"prob_buffer_row": "1.0",
"process_type": "default",
"rate_drop": "0.0",
"refresh_leaf": "1",
"sample_type": "uniform",

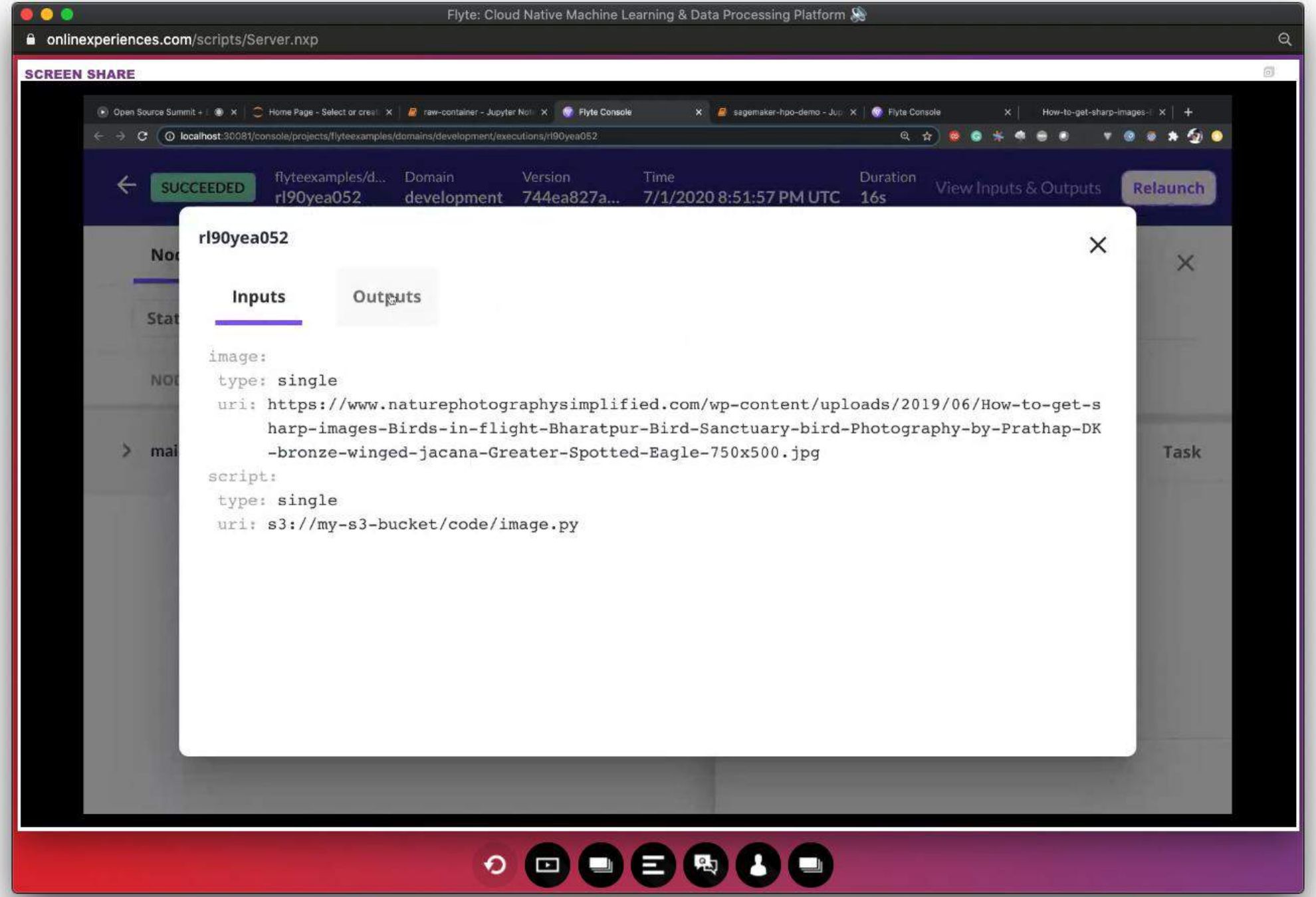

Flyte: Cloud Native Machine Learning & Data Processing Platform
onlinexperiences.com/scripts/Server.nxp
SCREEN SHARE
localhost:8888/notebooks/sagemaker-hpo-demo.ipynb#
jupyter sagemaker-hpo-demo (unsaved changes)
Logout
Trusted
Python 3.7.4 64-bit ('flytekit': virtualenv)
File Edit View Insert Cell Kernel Widgets Help
Run
Code
"refresh_leaf": "1",
"sample_type": "uniform",
"scale_pos_weight": "1.0",
"silent": "0",
"skip_drop": "0.0",
"tree_method": "auto",
"tweedie_variance_power": "1.5",
"updater": "grow_colmaker,prune",
}
xgboost_train_task2 = training_job_task.SdkSimpleTrainingJobTask(
training_job_config=training_job_models.TrainingJobConfig(
instance_type="ml.m4.xlarge",
instance_count=1,
volume_size_in_gb=25,
),
algorithm_specification=alg_spec,
cache_version='1',
cacheable=True,
)
Step III: But wait, lets try to optimize the hyper

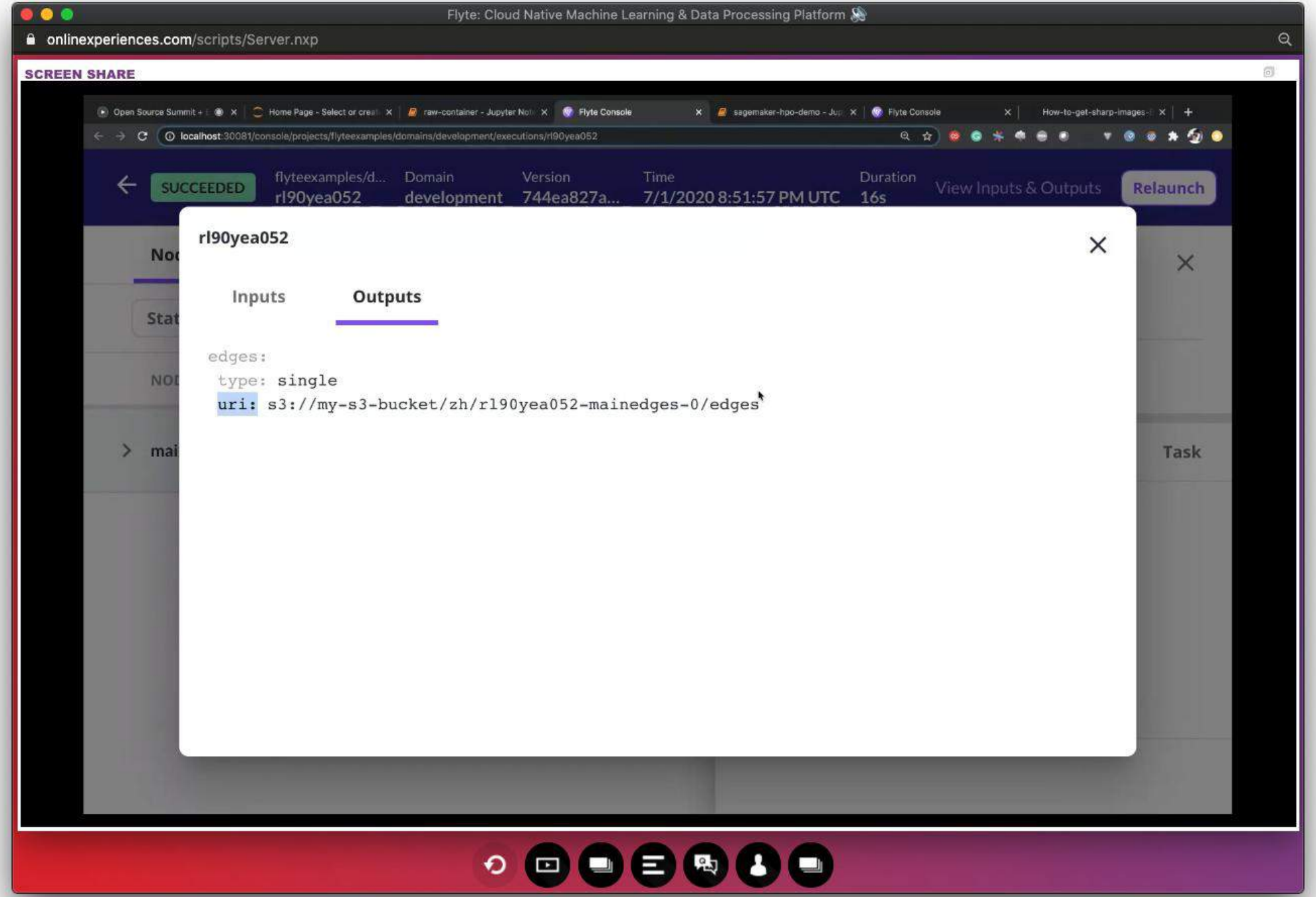

Flyte: Cloud Native Machine Learning & Data Processing Platform
onlinexperiences.com/scripts/Server.nxp
SCREEN SHARE
localhost:8888/notebooks/sagemaker-hpo-demo.ipynb#
jupyter sagemaker-hpo-demo (unsaved changes)
Logout
Trusted
Python 3.7.4 64-bit ('flytekit': virtualenv)
File Edit View Insert Cell Kernel Widgets Help
Run
Code
Step III: But wait, lets try to optimize the hyper parameters
XGBoost
In [14]:
xgboost_hpo_task2 = hpo_job_task.SdkSimpleHPOJobTask(
training_job=xgboost_train_task2,
max_number_of_training_jobs=10,
max_parallel_training_jobs=5,
cache_version='1',
retries=2,
cacheable=True,
)
Step IV Launch an execution
In [ ]:
from flytekit.models.sagemaker.training_job import StoppingCondition
from flytekit.models.sagemaker.hpo_job import HPOJobConfig, Hyperparameter
from flytekit.models.sagemaker.parameter_ranges import ParameterRanges, Cat
inputs={
"train": "s3://lyft-modelbuilder/test-datasets/pima-indians/train",


Flyte: Cloud Native Machine Learning & Data Processing Platform
onlinexperiences.com/scripts/Server.nxp
SCREEN SHARE
flyte-staging.lyft.net/console/projects/flyteexamples/domains/development/executions/elzaoqgt2t
UNKNOWN
flyteexamples/development/flytegen._main__xgboost_hpo_task2
elzaoqgt2t
Domain
development
Version
fbeafc95a038a12d7b93a4b0fb01beb7
Time
Duration
View Inputs & Outputs
Terminate
Nodes
Graph
Status
Start Time
Duration
NODE
STATUS
TYPE
START TIME
DURATION
Queued Time
LOGS
No executions found.


Flyte: Cloud Native Machine Learning & Data Processing Platform
onlinexperiences.com/scripts/Server.nxp
SCREEN SHARE
Flyte Console
localhost:30081/console/projects/flyteexamples/domains/development/executions/rl90yea052
SUCCEEDED
flyteexamples/d...
rl90yea052
Domain
development
Version
744ea827a...
Time
7/1/2020 8:51:57 PM UTC
Duration
16s
View Inputs & Outputs
Relaunch
Nodes
Graph
Status
Start Time
Duration
NOD
STATUS
TYPE
START TIME
DURATION
Queued Time
LOGS
mair
SUCCEEDED
Unknown Task
7/1/2020 8:51:58 PM UTC
7/1/2020 1:51:58 PM PDT
14s
View Logs


Flyte: Cloud Native Machine Learning & Data Processing Platform
onlinexperiences.com/scripts/Server.nxp
SCREEN SHARE
localhost:30081/console/projects/flyteexamples/domains/development/executions/rl90yea052
SUCCEEDED
flyteexamples/d...
rl90yea052
Domain
development
Version
744ea827a...
Time
7/1/2020 8:51:57 PM UTC
Duration
16s
View Inputs & Outputs
Relaunch
Nodes
Graph
Status
Start Time
Duration
NOD STATUS
TYPE
START TIME
mair
SUCCEEDED
Unknown Task
7/1/2020 8:51:58 PM UT
7/1/2020 1:51:58 PM PDT
mainedges
SUCCEEDED
TYPE
Unknown Task
Executions
Inputs
Outputs
Task
__main__.edges
succeeded
Logs
Kubernetes Logs (User)
started (unknown)
run time (unknown)


Flyte: Cloud Native Machine Learning & Data Processing Platform
onlinexperiences.com/scripts/Server.nxp
SCREEN SHARE
localhost:30081/console/projects/flyteexamples/domains/development/executions/rl90yea052
SUCCEEDED
flyteexamples/d...
rl90yea052
Domain
development
Version
744ea827a...
Time
7/1/2020 8:51:57 PM UTC
Duration
16s
View Inputs & Outputs
Relaunch
rl90yea052
Inputs
Outputs
image:
type: single
uri: https://www.naturephotographysimplified.com/wp-content/uploads/2019/06/How-to-get-sharp-images-Birds-in-flight-Bharatpur-Bird-Sanctuary-bird-Photography-by-Prathap-DK-bronze-winged-jacana-Greater-Spotted-Eagle-750x500.jpg
script:
type: single
uri: s3://my-s3-bucket/code/image.py
Task


Flyte: Cloud Native Machine Learning & Data Processing Platform
onlinexperiences.com/scripts/Server.nxp
SCREEN SHARE
Open Source Summit +
Home Page - Select or creat
raw-container - Jupyter Not
Flyte Console
sagemaker-hpo-demo - Jup
Flyte Console
How-to-get-sharp-images-
localhost:30081/console/projects/flyteexamples/domains/development/executions/rl90yea052
SUCCEEDED
flyteexamples/d...
rl90yea052
Domain
development
Version
744ea827a...
Time
7/1/2020 8:51:57 PM UTC
Duration
16s
View Inputs & Outputs
Relaunch
rl90yea052
Inputs
Outputs
edges:
type: single
uri: s3://my-s3-bucket/zh/rl90yea052-mainedges-0/edges
Task

Flyte: Cloud Native Machine Learning & Data Processing Platform

onlinexperiences.com/scripts/Server.nxp

SCREEN SHARE

jupyter raw-container (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

Step III: Launch an execution for the task

This creates an execution of just the task. Remember in Flyte, Task is a standalone top level entity, so you can execute it

In [29]:
```
exc = edges.register_and_launch("flyteexamples", "development", inputs={"ir

print_console_url(exc)
```

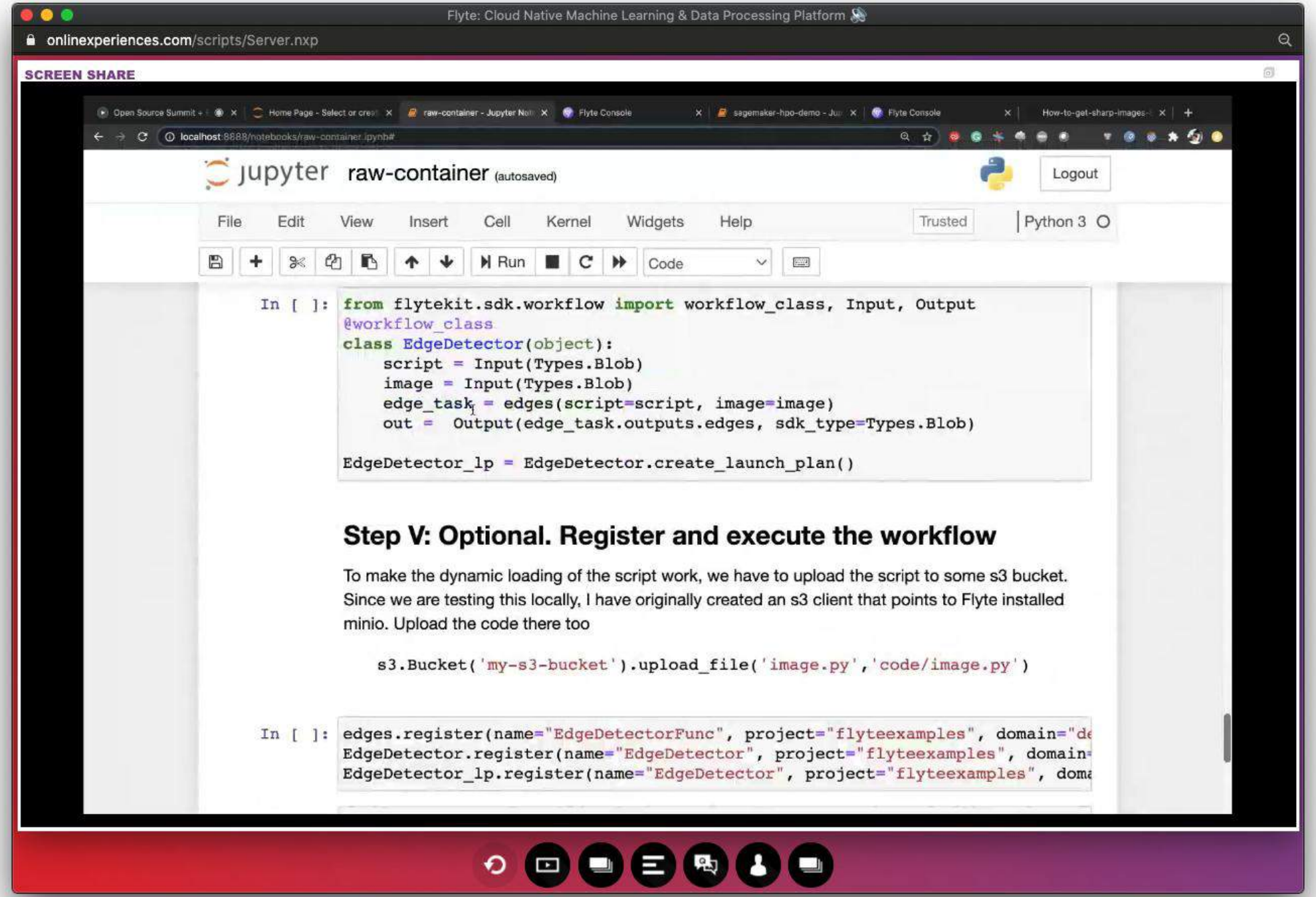
http://localhost:30081/console/projects/flyteexamples/domains/development/executions/rl90yea052

In [ ]:
```
exc.wait_for_completion()
```

Step IV: Optional. Create a Workflow

Ofcourse you can use this task in a workflow. We are creating a trivial workflow in this case that has only one task

In [ ]:
```
from flytekit.sdk.workflow import workflow_class, Input, Output
@workflow_class
class EdgeDetector(object):
```


Flyte: Cloud Native Machine Learning & Data Processing Platform
onlinexperiences.com/scripts/Server.nxp
SCREEN SHARE
localhost:8888/notebooks/raw-container.ipynb#
jupyter raw-container (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
In [ ]:
from flytekit.sdk.workflow import workflow_class, Input, Output
@workflow_class
class EdgeDetector(object):
script = Input(Types.Blob)
image = Input(Types.Blob)
edge_task = edges(script=script, image=image)
out = Output(edge_task.outputs.edges, sdk_type=Types.Blob)
EdgeDetector_lp = EdgeDetector.create_launch_plan()
Step V: Optional. Register and execute the workflow
To make the dynamic loading of the script work, we have to upload the script to some s3 bucket. Since we are testing this locally, I have originally created an s3 client that points to Flyte installed minio. Upload the code there too
s3.Bucket('my-s3-bucket').upload_file('image.py','code/image.py')
In [ ]:
edges.register(name="EdgeDetectorFunc", project="flyteexamples", domain="de
EdgeDetector.register(name="EdgeDetector", project="flyteexamples", domain=
EdgeDetector_lp.register(name="EdgeDetector", project="flyteexamples", doma

Flyte: Cloud Native Machine Learning & Data Processing Platform

onlinexperiences.com/scripts/Server.nxp

SCREEN SHARE

Open Source Summit + | Home Page - Select or creat | raw-container - Jupyter Noti | Flyte Console | sagemaker-hpo-demo - Jup | Flyte Console | How-to-get-sharp-images-

localhost:8888/notebooks/raw-container.ipynb#

jupyter raw-container (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

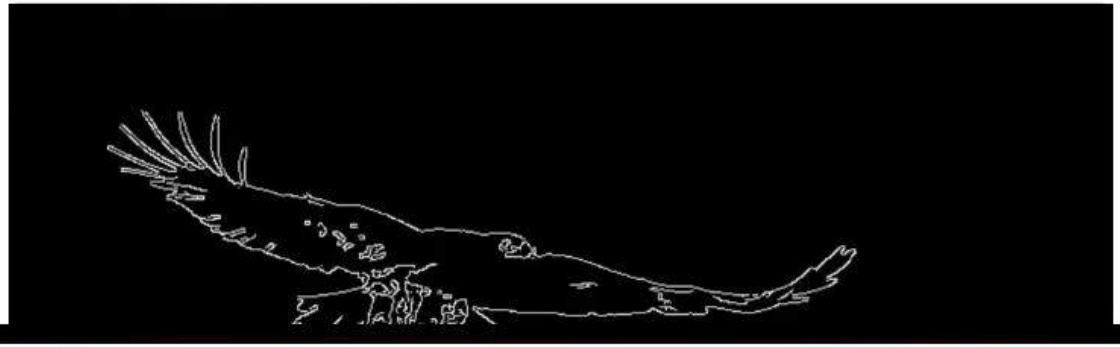
In [ ]:
```
exc = EdgeDetector_lp.execute("flyteexamples", "development", inputs={"imag
print_console_url(exc)
```

Step VI: Visualize the results

You can retrieve the results and visualize them here

In [30]:
```
key="s3://my-s3-bucket/zh/rl90yea052-mainedges-0/edges"
download_file(key, "edges.png")
```

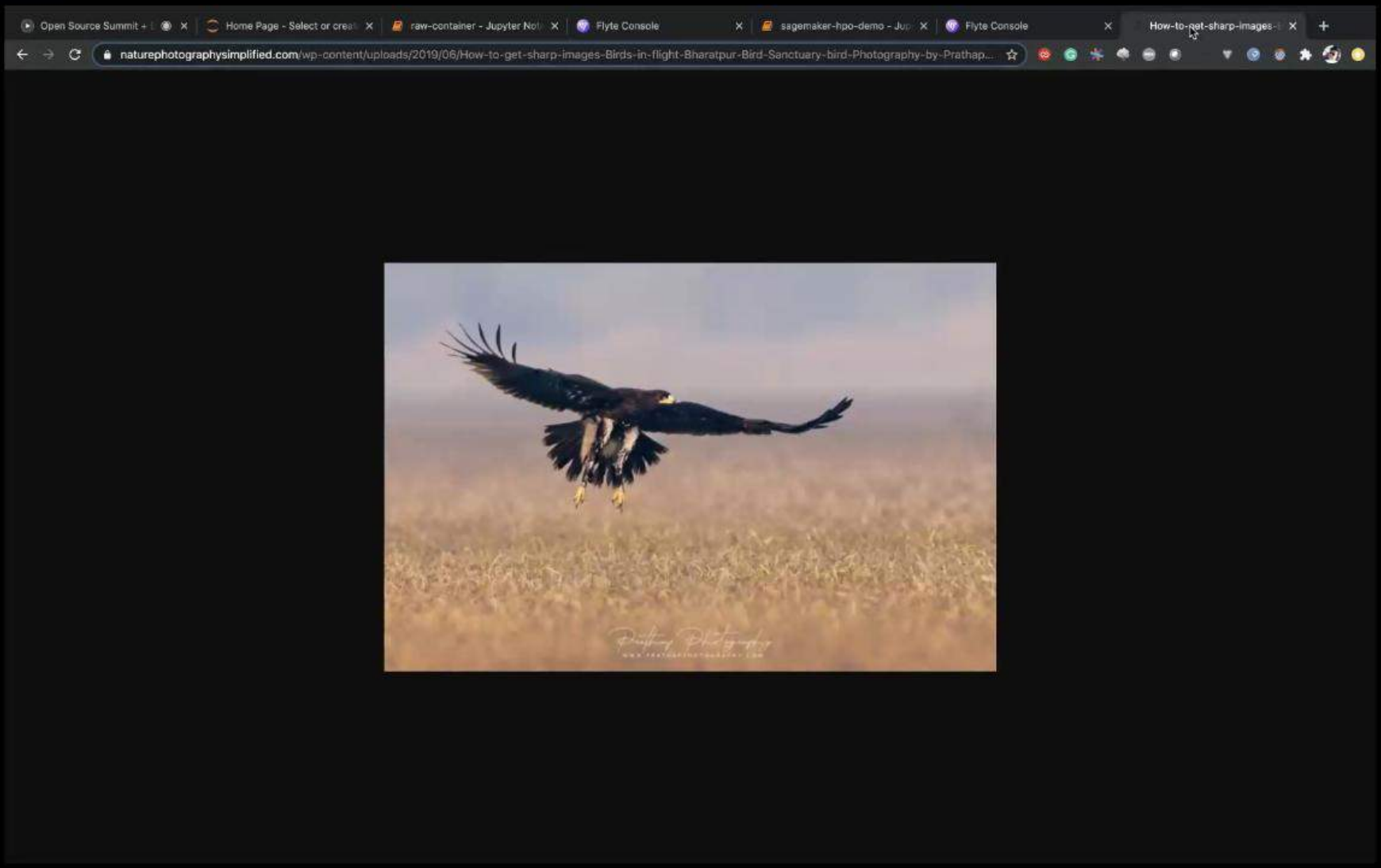
In [31]:
```
display_images(["edges.png"])
```



Flyte: Cloud Native Machine Learning & Data Processing Platform

onlinexperiences.com/scripts/Server.nxp

SCREEN SHARE



Flyte: Cloud Native Machine Learning & Data Processing Platform

onlinexperiences.com/scripts/Server.nxp

SCREEN SHARE

Open Source Summit + | Home Page - Select or crea | raw-container - Jupyter Not | Flyte Console | sagemaker-hpo-demo - Jup | Flyte Console | How-to-get-sharp-images-f

localhost:8888/notebooks/raw-container.ipynb#

jupyter raw-container (unsaved changes)

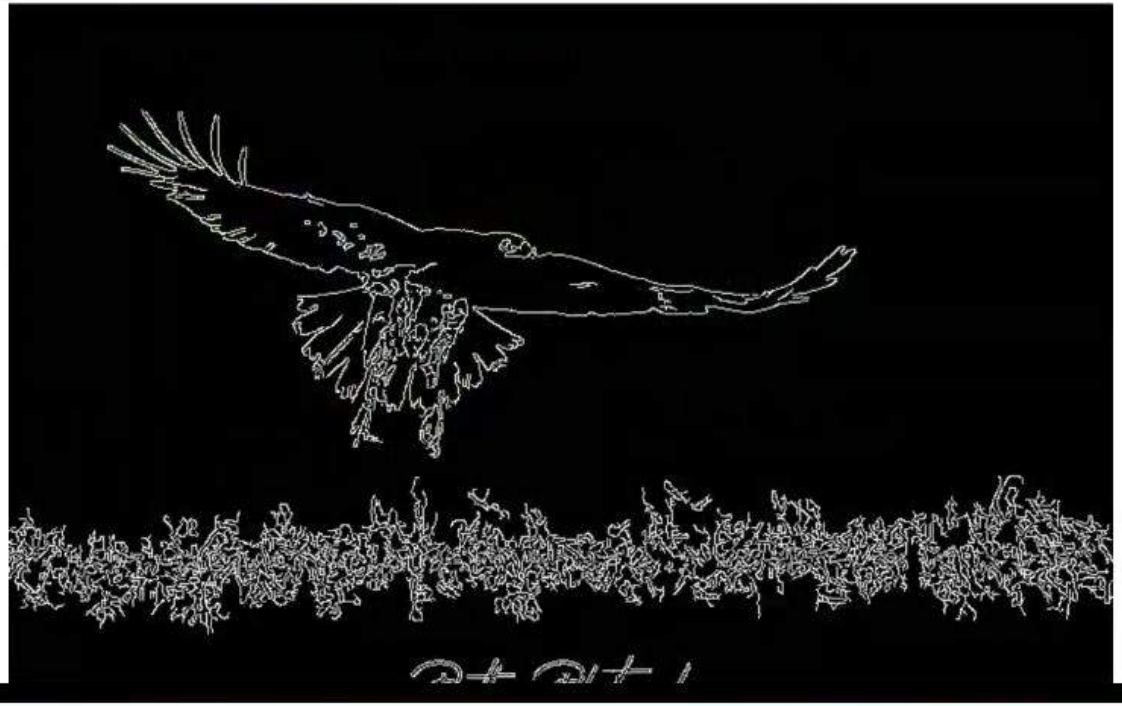
Logout

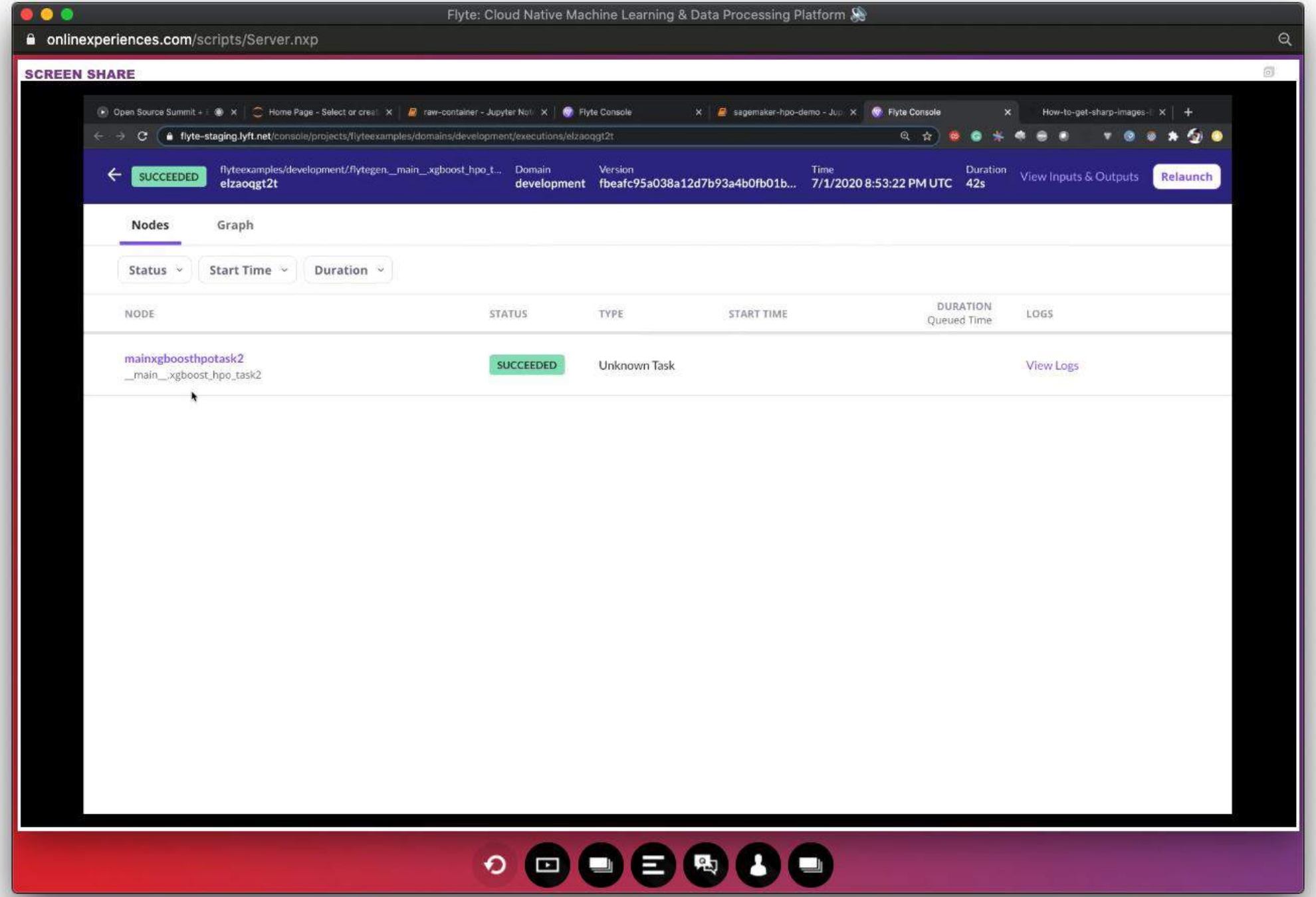
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

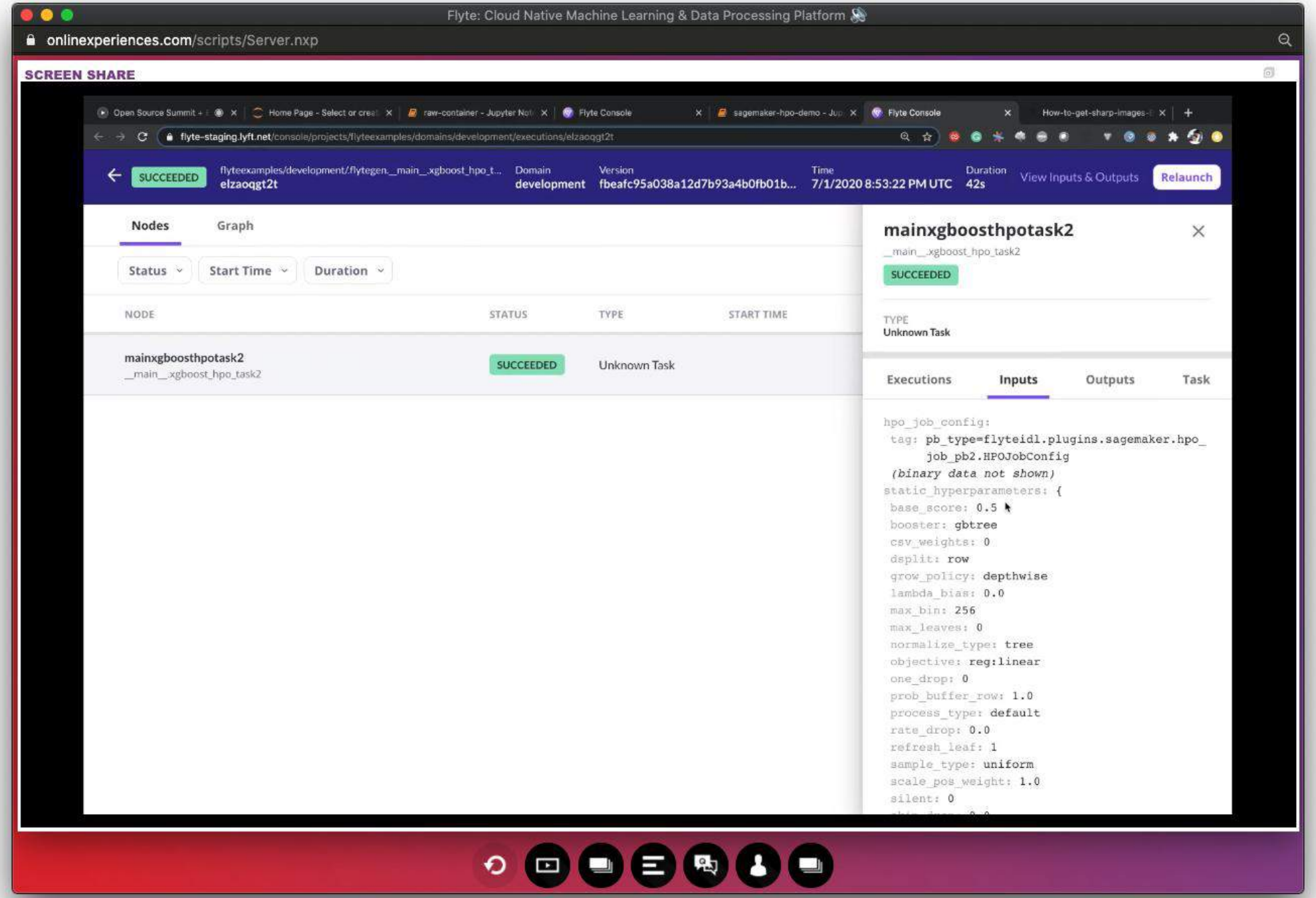
Run Code

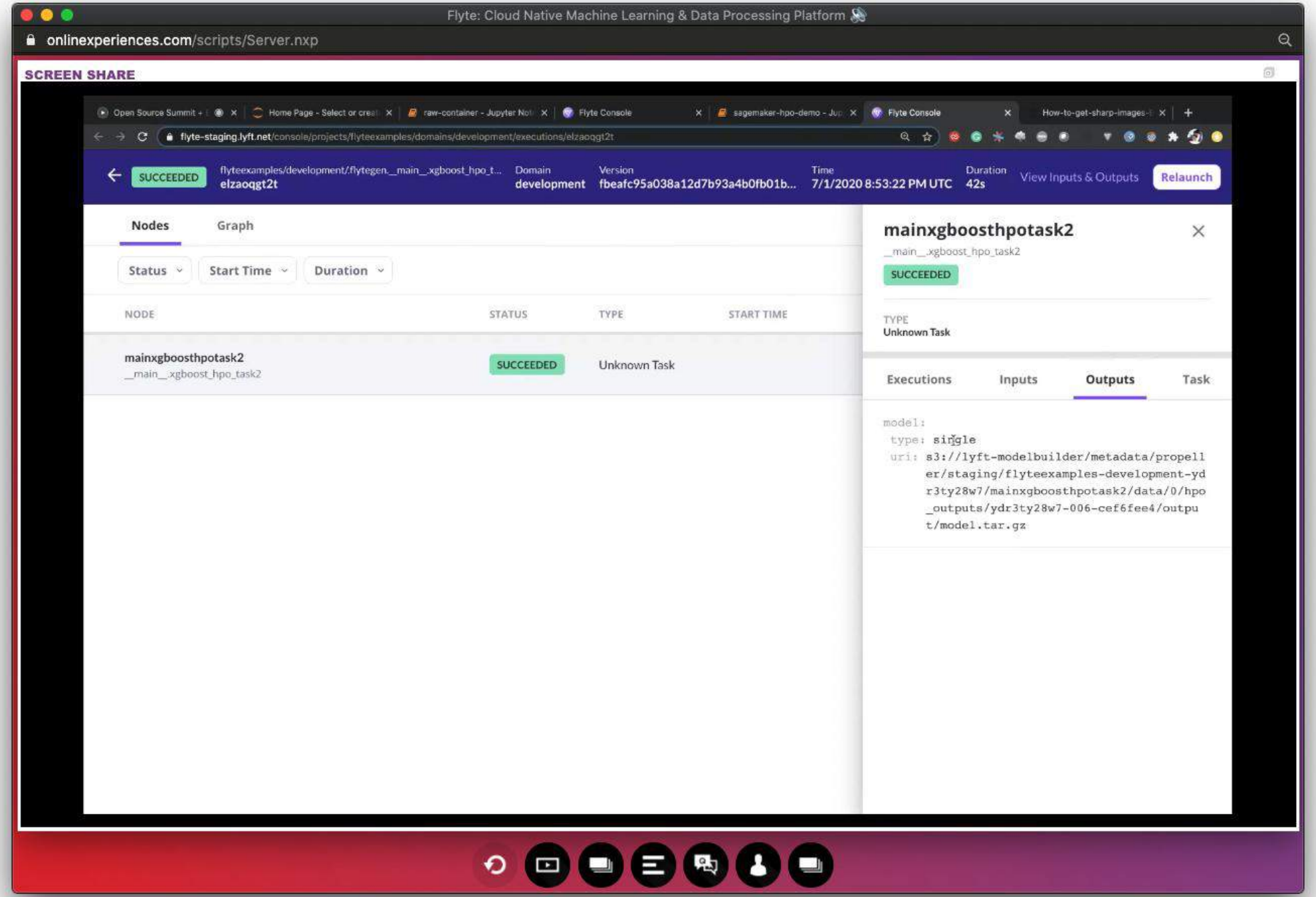
```
download_file(key, "edges.png")
```

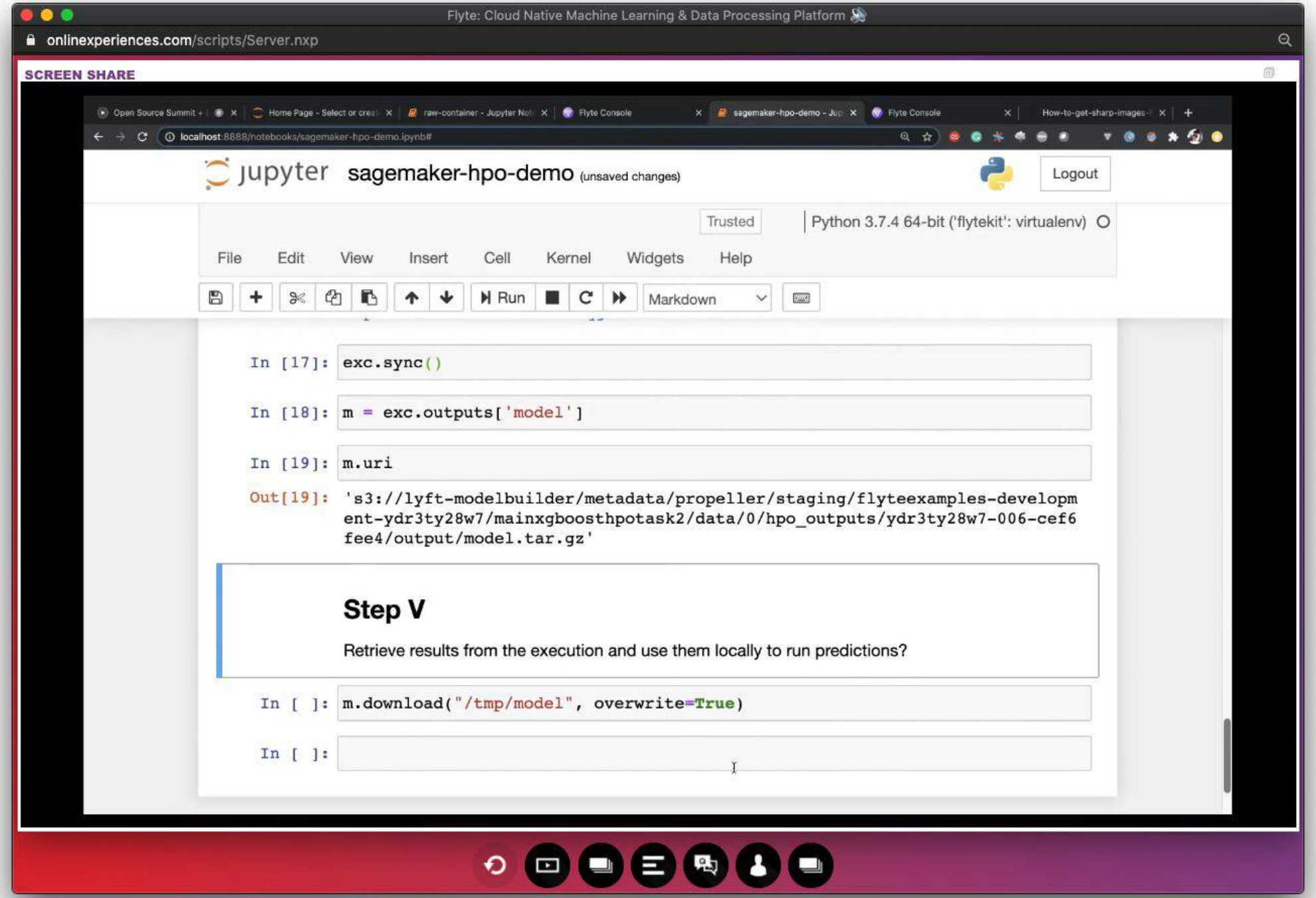
In [31]:
```
display_images(["edges.png"])
```




Flyte: Cloud Native Machine Learning & Data Processing Platform
onlinexperiences.com/scripts/Server.nxp
SCREEN SHARE
Open Source Summit +
Home Page - Select or creat
raw-container - Jupyter Not
Flyte Console
sagemaker-hpo-demo - Jup
Flyte Console
How-to-get-sharp-images-
flyte-staging.lyft.net/console/projects/flyteexamples/domains/development/executions/elzaoqgt2t
SUCCEEDED
flyteexamples/development/flytegen._main__xgboost_hpo_t...
elzaoqgt2t
Domain
development
Version
fbeafc95a038a12d7b93a4b0fb01b...
Time
7/1/2020 8:53:22 PM UTC
Duration
42s
View Inputs & Outputs
Relaunch
Nodes
Graph
Status
Start Time
Duration
NODE
STATUS
TYPE
START TIME
DURATION
Queued Time
LOGS
mainxgboosthpotask2
__main__xgboost_hpo_task2
SUCCEEDED
Unknown Task
View Logs


Flyte: Cloud Native Machine Learning & Data Processing Platform
onlinexperiences.com/scripts/Server.nxp
SCREEN SHARE
flyte-staging.lyft.net/console/projects/flyteexamples/domains/development/executions/elzaoqgt2t
SUCCEEDED
flyteexamples/development/flytegen._main__xgboost_hpo_t...
elzaoqgt2t
Domain
development
Version
fbeafc95a038a12d7b93a4b0fb01b...
Time
7/1/2020 8:53:22 PM UTC
Duration
42s
View Inputs & Outputs
Relaunch
Nodes
Graph
Status
Start Time
Duration
NODE
STATUS
TYPE
START TIME
mainxgboosthpotask2
__main__xgboost_hpo_task2
SUCCEEDED
Unknown Task
mainxgboosthpotask2
__main__xgboost_hpo_task2
SUCCEEDED
TYPE
Unknown Task
Executions
Inputs
Outputs
Task
hpo_job_config:
tag: pb_type=flyteidl.plugins.sagemaker.hpo_job_pb2.HPOJobConfig
(binary data not shown)
static_hyperparameters: {
base_score: 0.5
booster: gbtree
csv_weights: 0
dsplit: row
grow_policy: depthwise
lambda_bias: 0.0
max_bin: 256
max_leaves: 0
normalize_type: tree
objective: reg:linear
one_drop: 0
prob_buffer_row: 1.0
process_type: default
rate_drop: 0.0
refresh_leaf: 1
sample_type: uniform
scale_pos_weight: 1.0
silent: 0


Flyte: Cloud Native Machine Learning & Data Processing Platform
onlinexperiences.com/scripts/Server.nxp
SCREEN SHARE
flyte-staging.lyft.net/console/projects/flyteexamples/domains/development/executions/elzaoqgt2t
SUCCEEDED
flyteexamples/development/flytegen._main__xgboost_hpo_t...
elzaoqgt2t
Domain
development
Version
fbeafc95a038a12d7b93a4b0fb01b...
Time
7/1/2020 8:53:22 PM UTC
Duration
42s
View Inputs & Outputs
Relaunch
Nodes
Graph
Status
Start Time
Duration
NODE
STATUS
TYPE
START TIME
mainxgboosthpotask2
__main__xgboost_hpo_task2
SUCCEEDED
Unknown Task
mainxgboosthpotask2
__main__xgboost_hpo_task2
SUCCEEDED
TYPE
Unknown Task
Executions
Inputs
Outputs
Task
model:
type: single
uri: s3://lyft-modelbuilder/metadata/propeller/staging/flyteexamples-development-ydr3ty28w7/mainxgboosthpotask2/data/0/hpo_outputs/ydr3ty28w7-006-cef6fee4/output/model.tar.gz


Flyte: Cloud Native Machine Learning & Data Processing Platform
onlinexperiences.com/scripts/Server.nxp
SCREEN SHARE
localhost:8888/notebooks/sagemaker-hpo-demo.ipynb#
jupyter sagemaker-hpo-demo (unsaved changes)
Logout
Trusted
Python 3.7.4 64-bit ('flytekit': virtualenv)
File Edit View Insert Cell Kernel Widgets Help
Run
Markdown
In [17]: exc.sync()
In [18]: m = exc.outputs['model']
In [19]: m.uri
Out[19]: 's3://lyft-modelbuilder/metadata/propeller/staging/flyteexamples-development-ydr3ty28w7/mainxgboosthpotask2/data/0/hpo_outputs/ydr3ty28w7-006-cef6fee4/output/model.tar.gz'
Step V
Retrieve results from the execution and use them locally to run predictions?
In [ ]: m.download("/tmp/model", overwrite=True)
In [ ]:

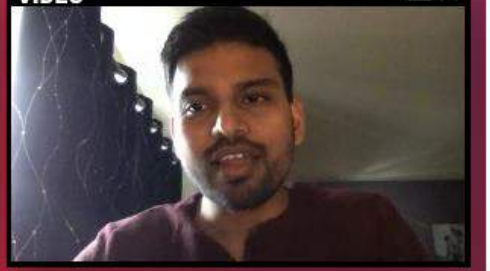
Flyte: Cloud Native Machine Learning & Data Processing Platform

onlinexperiences.com/scripts/Server.nxp

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO



ABSTRACT

Flyte is the backbone for large-scale Machine Learning and Data Processing (ETL) pipelines at Lyft. It is used across business critical applications ranging from ETA, Pricing, Mapping, Autonomous etc. At its core it is a Kubernetes native workflow engine that executes 10M+ containers per month as part of thousands of workflows. We introduced Flyte at Kubecon 2019. This talk will build on the base talk and dive deeper into the architecture and the developer experience. Thus the talk will help existing users of Flyte and new users as well.

Quick overview of Flyte concepts
Architecture of Flyte
Challenges and learnings while using Kubernetes at scale for stateful applications
A deeper look at developer ergonomics and reasons that shaped some decisions
An overview of the community and ecosystem since open sourcing Flyte
The talk will conclude with a demo.

SLIDES

Demo

lyft

Flyte

Flyte: Cloud Native Machine Learning & Data Processing Platform

onlinexperiences.com/scripts/Server.nxp

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference
North America

VIDEO

ABSTRACT

Flyte is the backbone for large-scale Machine Learning and Data Processing (ETL) pipelines at Lyft. It is used across business critical applications ranging from ETA, Pricing, Mapping, Autonomous etc. At its core it is a Kubernetes native workflow engine that executes 10M+ containers per month as part of thousands of workflows. We introduced Flyte at Kubecon 2019. This talk will build on the base talk and dive deeper into the architecture and the developer experience. Thus the talk will help existing users of Flyte and new users as well.

Quick overview of Flyte concepts
Architecture of Flyte
Challenges and learnings while using Kubernetes at scale for stateful applications
A deeper look at developer ergonomics and reasons that shaped some decisions
An overview of the community and ecosystem since open sourcing Flyte
The talk will conclude with a demo.

SLIDES

Conclusion

Ecosystem

pod
Spark
aws
Google Cloud
HIVE
jupyter
presto
Amazon SageMaker
In testing
pandas
Parquet
TensorFlow
Coming soon
Coming soon
lyft
Flyte

VIDEO

ABSTRACT

Flyte is the backbone for large-scale Machine Learning and Data Processing (ETL) pipelines at Lyft. It is used across business critical applications ranging from ETA, Pricing, Mapping, Autonomous etc. At its core it is a Kubernetes native workflow engine that executes 10M+ containers per month as part of thousands of workflows. We introduced Flyte at Kubecon 2019. This talk will build on the base talk and dive deeper into the architecture and the developer experience. Thus the talk will help existing users of Flyte and new users as well.

Quick overview of Flyte concepts
Architecture of Flyte
Challenges and learnings while using Kubernetes at scale for stateful applications
A deeper look at developer ergonomics and reasons that shaped some decisions
An overview of the community and ecosystem since open sourcing Flyte
The talk will conclude with a demo.

SLIDES

Conclusion

What's Next

Flyte is constantly evolving and new features are coming soon like,

- Flytekit python Enhancement & Flytekit JAVA primetime ready
- Richer **data catalog visualization**
- **UI** improvements
- **Reactive workflows** (respond to data publication events)
- Better documentation and more examples
- Faster getting started

To find more details **visit our docs and the Roadmap section**. Also join our fledgeling community and help us shape the future of Flyte. We appreciate contributions and suggestions.

lyft

Flyte

VIDEO

ABSTRACT

Flyte is the backbone for large-scale Machine Learning and Data Processing (ETL) pipelines at Lyft. It is used across business critical applications ranging from ETA, Pricing, Mapping, Autonomous etc. At its core it is a Kubernetes native workflow engine that executes 10M+ containers per month as part of thousands of workflows. We introduced Flyte at Kubecon 2019. This talk will build on the base talk and dive deeper into the architecture and the developer experience. Thus the talk will help existing users of Flyte and new users as well.

Quick overview of Flyte concepts
Architecture of Flyte
Challenges and learnings while using Kubernetes at scale for stateful applications
A deeper look at developer ergonomics and reasons that shaped some decisions
An overview of the community and ecosystem since open sourcing Flyte
The talk will conclude with a demo.

SLIDES

Thanks!
Learn more, get started & keep in touch at Flyte.org

@KetanUmare
@KetanUmare

@matthewphsmith
@matthewphsmith

lyft

Flyte