



Operate First Data Science Community Meetup!

Michael Clifford



OPERATE FIRST

DATA SCIENCE COMMUNITY

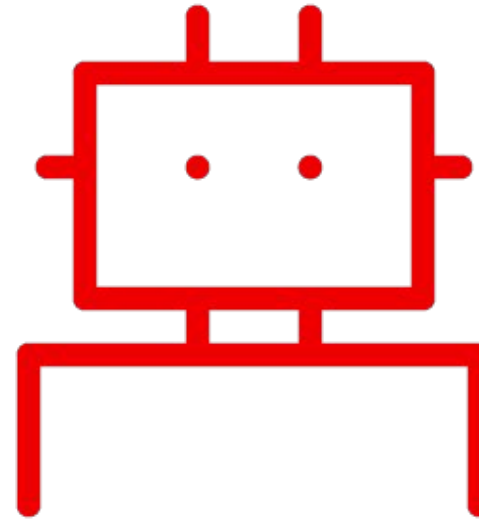
a concept to incorporate operational experience into software projects

Ready to get started?

Data Science + Operate First



Operate First



Data Science

links!

Links!

Website:

<https://www.operate-first.cloud/data-science/>

JupyterHub:

<https://Jupyterhub-opf-jupyterhub.apps.smaug.na.operate-first.cloud>

Meetup info:

<https://www.operate-first.cloud/data-science/operate-first-data-science-community/docs/meetup-landing-page.md>

Current Projects:

<https://www.operate-first.cloud/data-science/projectsoverview.md>

Slack:

https://join.slack.com/t/operatefirst/shared_invite/zt-o2gn4wn8-O39g7sthTAuPCvaCNRnLww

Youtube:

<https://youtube.com/playlist?list=PL8VBRDTEICWqC5WcZUUNJxGn9WslK6llb>

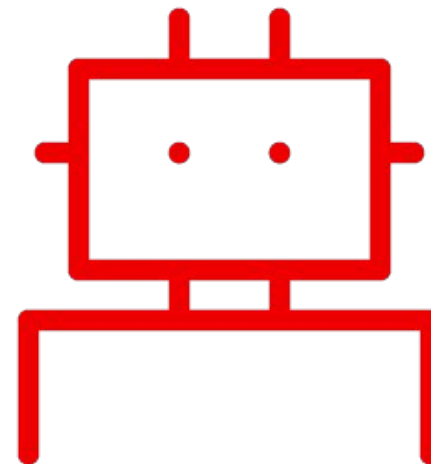
GitHub:

<https://github.com/aicoe-aiops/cloud-first-data-science-community/issues/new/choose>



Operate First

Thanks!



Reproducible Data Science with Operate First

Karan Chauhan

Data Scientist, Red Hat

Schedule

- > Reproducibility and Shareability Challenges
- > Solution with Operate-First + Thoth + AICoE-Cl
- > How-To with Examples
- > Demo
- > Q & A

How Reproducible and Shareable is Your Data Science Content?

- How are Python dependencies specified?
- Are there system-level requirements or initial setup?
- Does it need specific compute resources e.g. GPUs?

How to Share Your Data Science Content in a Reproducible Way

- How are Python dependencies specified?

Pinned dependencies in Pipfile using Thoth recs

- Are there system-level requirements or initial setup?

Project container image via AI CoE-Cl

- Does it need specific compute resources e.g. GPUs?

Cloud environment via Operate-First

Python Dependencies

- Pinned requirements in a Thoth-advised Pipfile

```
(river-demo) -----  
~/Documents/playpen/river-demo » thamos add numpy pandas onnxruntime tensorflow
```

Pipfile

```
1  [[source]]  
2  url = "https://pypi.org/simple"  
3  verify_ssl = true  
4  name = "pypi-org-simple"  
5  
6  [packages]  
7  numpy = "*"   
8  pandas = "*"   
9  onnxruntime = "*"   
10 tensorflow = "*"   
11  
12 [dev-packages]  
13  
14 [requires]  
15 python_version = "3.8"  
16  
17 [thoth]  
18 disable_index_adjustment = false  
19  
20 [thoth.allow_prereleases]
```

Python Dependencies

(river-demo)

~/Documents/playpen/river-demo » thamos advise

Application stack guidance

Link	Message	Type
https://thoth-station.ninja/j/cve_timestamp	CVE database of known vulnerabilities for Python packages was updated at '2021-12-11T00:37:34.592440'	✓INFO
https://thoth-station.ninja/j/thoth_s2i	It is recommended to use Thoth's s2i to have recommendations specific to runtime environment	✓INFO
https://thoth-station.ninja/j/rhel_ubi	Using observations for RHEL instead of UBI, RHEL is ABI compatible with UBI	⚠WARNING

Recommended stack report

Link	Message	Package name	Type
https://github.com/thoth-station/s2i-thoth/	See S2I Thoth base images and their docs for more info	-	✓INFO
https://stackoverflow.com/questions/tagged/python-3.8	See StackOverflow tags specific for Python 3.8	-	✓INFO
https://thoth-station.ninja/recommendation-types/	The recommended software stack was computed based on the 'latest' recommendation type	-	✓INFO
https://libraries.io/pypi/flatbuffers/	Information about 'flatbuffers' on libraries.io	flatbuffers	✓INFO
https://thoth-station.ninja/j/no_cve	No known CVE known for 'flatbuffers' in version '2.0'	flatbuffers	✓INFO
https://pypi.org/project/flatbuffers/2.0/	Package 'flatbuffers' in version '2.0' is released on PyPI	flatbuffers	✓INFO

Python Dependencies

```
(river-demo)
~/Documents/playpen/river-demo » ls
demo.ipynb  Pipfile  Pipfile.lock
```

Pipfile.lock

```
1      "onnxruntime": {
2          "hashes": [
3              "sha256:2d8eb89d4d62ba956f5b3392b0a02e50dfa3ba6e255561c16e3fd10c
4              "sha256:aa7346af63eade9a041b79363b2be04a60e4a5d4fe82bce4de2b1cdc
5              "sha256:2201b12c736c7c1ce683289ff012a768dc6ea0eeb81c9276fde31c86
6              "sha256:579a38f5abee3e89684ee223701d62409be20742767356d982e92cf2
7              "sha256:417d09475d3928b224225e4bd1a6aa8ce061de7458ccda2b86dfe125
8              "sha256:7395d86c1c18c5e191f6d7cfe79ecc590fe8ed30f4b01a65e1da6c6c
9              "sha256:c140299b0f94f4a3cdf872308b21941622b973868f491a37e7f3148f
10             "sha256:1c2cf7d5fe89e74f363596d90b952c367653763f16655a8b5060c7de
11             "sha256:2df2999fdc0f1f5bcc87365b93fd951adbcdb615d312b1eb69aa3ffa
12             "sha256:923721385a33d681d20878a090ae9e9e653c40eaa57665a4a87ddbcc
13             "sha256:695e9badeb538bae87e25ee17c330c50a32114ebe6eae9bf2da03d66
14             "sha256:8fd9200c327042359ff733739ca4d0e2491d52f540b31a3fa29e2805
15             "sha256:89fa6e392637c94c17e693772294ed9afe2403832f78c5efa1ccd79f
16             "sha256:a7a98e820b6a2d9a23908efad28ce4b0c7181e897273e70daa22b611
17             "sha256:affda6e92c4de3a3d7b35c8da35dedf8fc618b17638f719bb5398acc
18             "sha256:cac8e35f953ff9898f0b5fdca0b76f492ae508e99e2eeb02ea70339e
19         ],
20         "index": "pypi-org-simple",
21         "version": "==1.8.1"
22     },
23     "pandas": {
24         "hashes": [
25             "sha256:68408a39a54ebadb9014ee5a4fae27b2fe524317bc80adf56c9ac59e
```

How to Share Your Data Science Content in a Reproducible Way

- How are Python dependencies managed?

Pinned dependencies in Pipfile using Thoth recs



- Are there system-level requirements or initial setup?

Project container image via AI CoE-Cl

- Does it need specific compute resources e.g. GPUs?

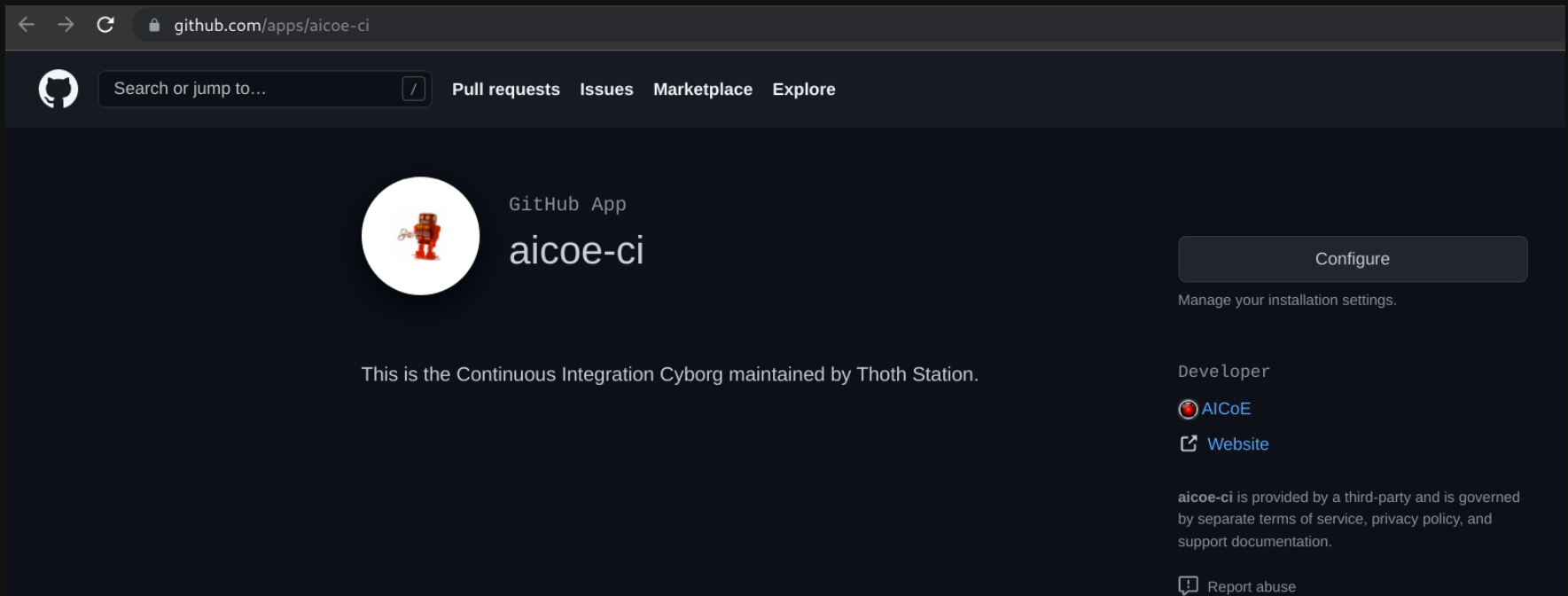
Cloud environment via Operate-First

Project Container Image via AI CoE-CI

- Package system libraries, initial setup, etc. into container image described via s2i scripts or Dockerfile
- Trigger image build automatically upon tag release or by creating issue
- If repo is within operate-first / aicoe orgs, can use already set up instance, else can host your own
- Set up in [less than] **five** easy steps

Container Image using aicoe-ci

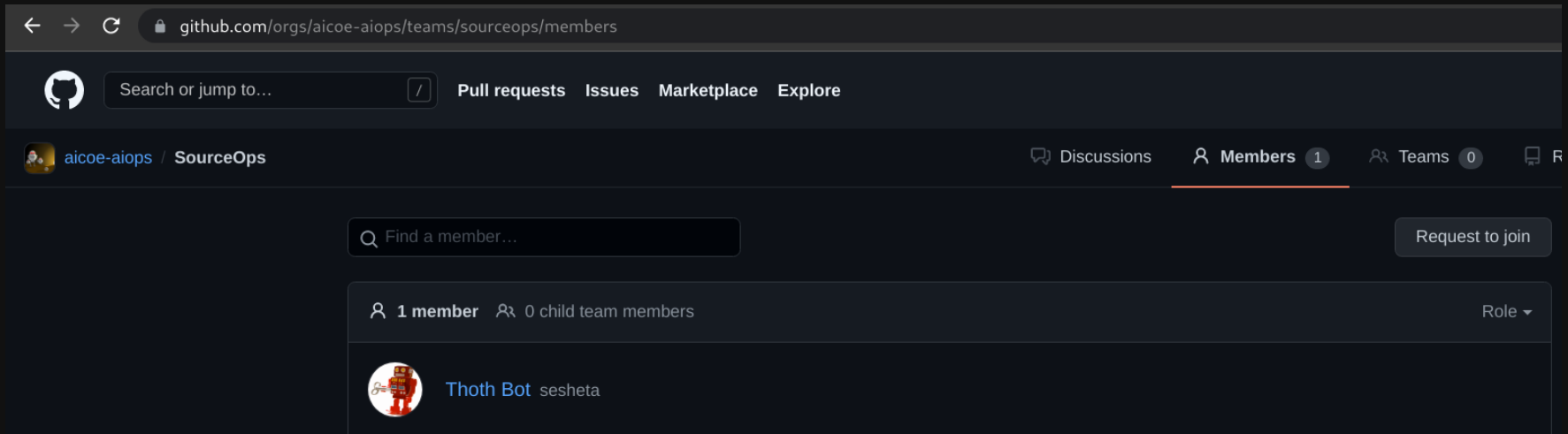
Step 1: Install aicoe-ci [github app](#) in your org/repo



Note: already done for repos within operate-first / aicoe orgs

Container Image using aicoe-ci

Step 2: Add [sesheta](#) as a collaborator to your org/repo



Note: already done for repos within operate-first / aicoe orgs

Container Image using aicoe-ci

Step 3: Add `.aicoe-ci.yaml` config file to your repo

```
1 check:
2   - thoth-build
3
4 build:
5   custom-tag: latest
6   registry: quay.io
7   registry-org: aicoe
8   registry-project: ocp-ci-analysis
9   registry-secret: aicoe-pusher-secret
10  build-strategy: Source
11  build-source-script: "image:///opt/app-root/builder"
12  base-image: quay.io/thoth-station/s2i-elyra-custom-notebook:latest
```

Example config for s2i build

Container Image using aicoe-ci

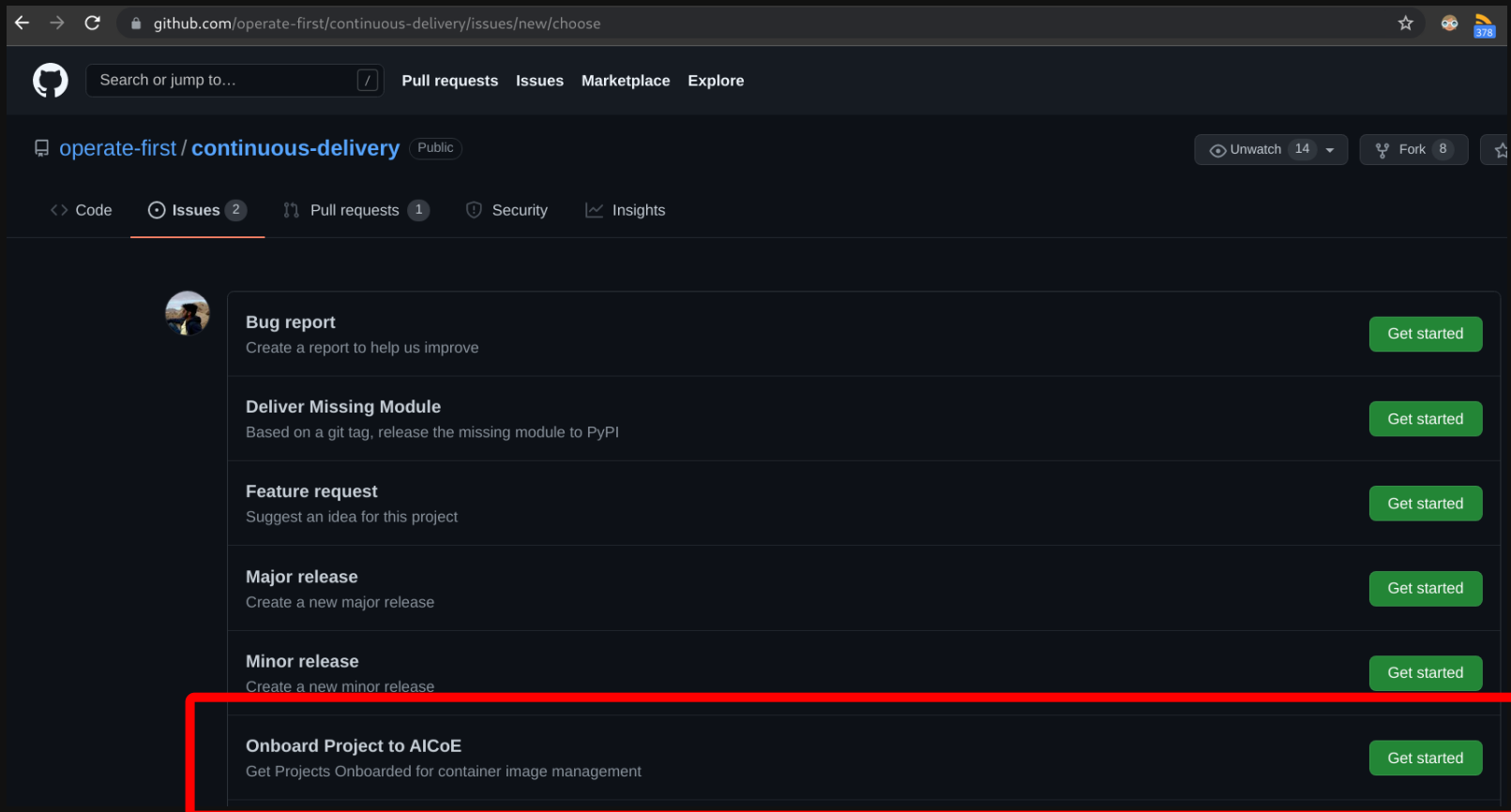
Step 3: Add `.aicoe-ci.yaml` config file to your repo

```
1 check:
2   - thoth-build
3
4 build:
5   custom-tag: latest
6   registry: quay.io
7   registry-org: os-climate
8   registry-project: aicoe-osc-demo
9   registry-secret: os-climate-pusher-secret
10  build-strategy: Dockerfile
11  dockerfile-path: Dockerfile
```

Example config for Dockerfile build

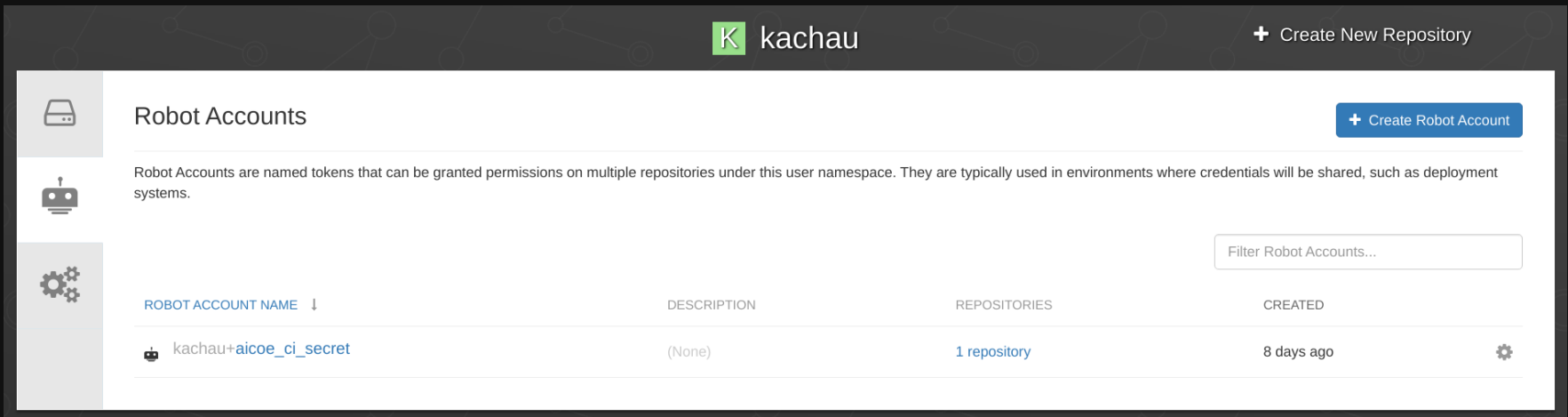
Container Image using aicoe-ci

Step 4: Open issue on [operate-first/continuous-delivery](#)
(in case of aicoe-ci instance on operate-first / aicoe orgs)





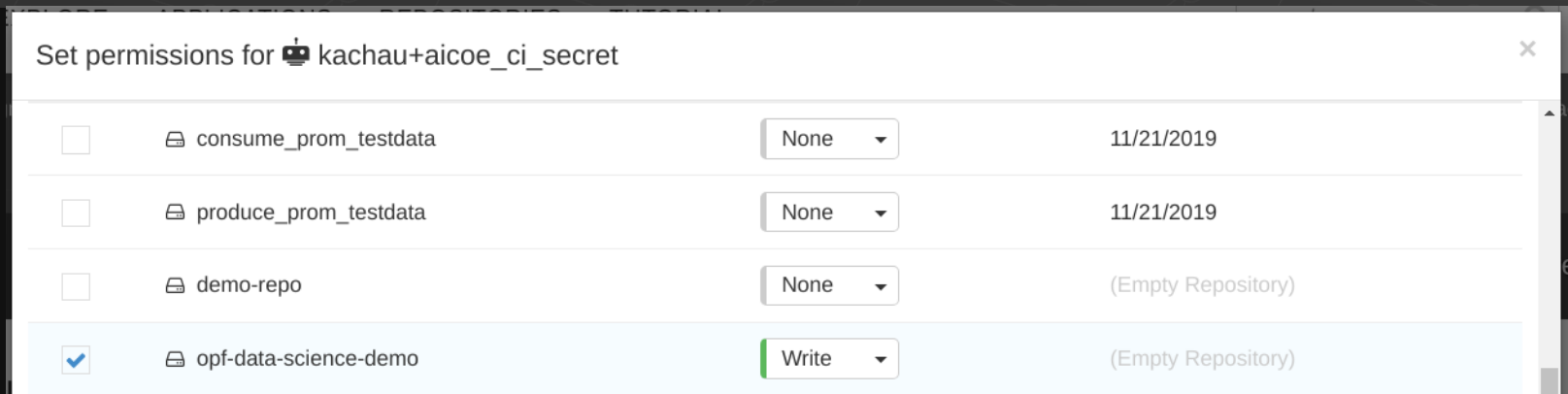
Container Image using aicoe-ci


Step 4: Create quay repo + robot account with write access
(in case of your own aicoe-ci instance)







The screenshot shows the 'Robot Accounts' page for the user 'kachau'. At the top right, there is a '+ Create New Repository' button. On the left sidebar, there are icons for a server, a robot, and settings. The main heading is 'Robot Accounts' with a '+ Create Robot Account' button on the right. Below the heading, a text block explains that Robot Accounts are named tokens for granting permissions. A search bar labeled 'Filter Robot Accounts...' is present. A table lists the robot accounts with columns for 'ROBOT ACCOUNT NAME', 'DESCRIPTION', 'REPOSITORIES', and 'CREATED'. One account is listed: 'kachau+aicoe_ci_secret' with description '(None)', '1 repository', and created '8 days ago'. A settings gear icon is next to the account name.

ROBOT ACCOUNT NAME ↓	DESCRIPTION	REPOSITORIES	CREATED
 kachau+aicoe_ci_secret	(None)	1 repository	8 days ago 

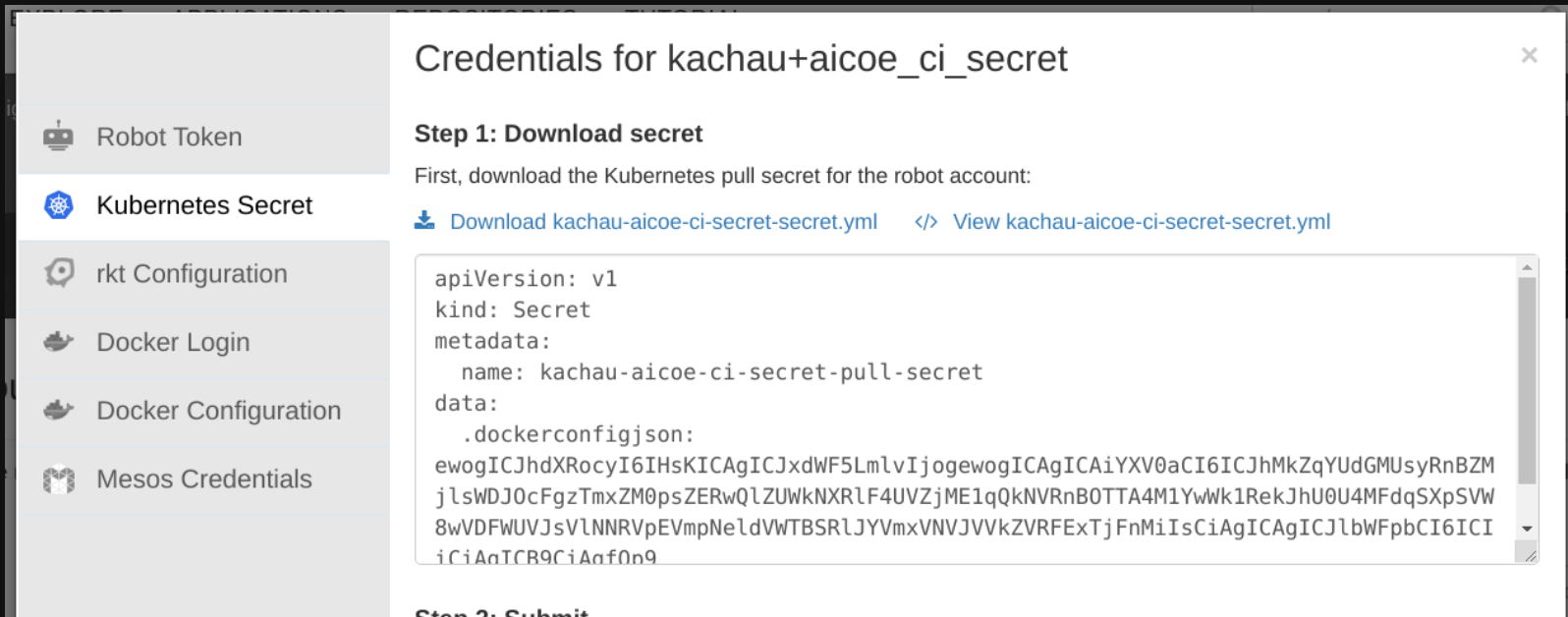


The screenshot shows a dialog titled 'Set permissions for  kachau+aicoe_ci_secret'. It contains a table with four rows, each representing a repository. The first three rows have 'consume_prom_testdata', 'produce_prom_testdata', and 'demo-repo' as repository names, all with 'None' permissions and a creation date of '11/21/2019'. The fourth row, 'opf-data-science-demo', is selected with a blue background, has 'Write' permissions, and is marked as an '(Empty Repository)'. Each row has a checkbox on the left and a dropdown menu for permissions.

	Repository	Permissions	Created
<input type="checkbox"/>	 consume_prom_testdata	None	11/21/2019
<input type="checkbox"/>	 produce_prom_testdata	None	11/21/2019
<input type="checkbox"/>	 demo-repo	None	(Empty Repository)
<input checked="" type="checkbox"/>	 opf-data-science-demo	Write	(Empty Repository)

Container Image using aicoe-ci

Step 5: Add the k8s secret to aicoe-ci instance



Credentials for kachau+aicoe_ci_secret

Step 1: Download secret

First, download the Kubernetes pull secret for the robot account:

[Download kachau-aicoe-ci-secret-secret.yml](#) [View kachau-aicoe-ci-secret-secret.yml](#)

```
apiVersion: v1
kind: Secret
metadata:
  name: kachau-aicoe-ci-secret-pull-secret
data:
  .dockerconfigjson:
ewogICJhdXRocyI6IHsKICAgICJxdWF5Lm1vIjogewogICAgICAiYXV0aCI6ICJhMkZqYUdGMUdyRnBZM
jlsWDJ0cFgzTmxZM0psZERwQlZUWkNXRlF4UVZjME1qQkNVRnB0TTA4M1YwWk1RekJhU0U4MFdqSXpSVW
8wVDFWUVJsVlNNRVpEVmpNeIdVWTBSRlJYVmxVNVJVVkZVRFEtTjFnMiIsCiAgICAgICJlbWVpYmCI6ICI
iCiAaTCB9CiAnf0n9
```

Step 2: Submit

Note: already done by issue opened in previous step, for repos within operate-first / aicoe orgs

Container Image using aicoe-ci

Result: Image created on every release

os-climate / aicoe-osc-demo Public

generated from aicoe-aiops/project-template

<> Code Issues (20) Pull requests (2) Discussions Actions Projects Wiki Security Insights Settings

Releases Tags Draft a new release Find a release

27 days ago
Shreyanand
v0.12.0
5a82b34

Compare

v0.12.0 Latest

What's Changed

- Add the final inference step in the infer_kpi notebook and config files by @Shreyanand in #101
- Add demo3 table generation for KPI Inference Model by @aakankshaduggal in #103
- Add inference pipeline by @oindrillac in #104

Full Changelog: v0.11.0...v0.12.0

Repositories os-climate / aicoe-osc-demo

Repository Tags

Compact Expanded

1 - 13 of 13 Filter Tags...

TAG	LAST MODIFIED	SECURITY SCAN	SIZE	EXPIRES	MANIFEST
<input checked="" type="checkbox"/> latest	a month ago	14 High	2.3 GB	Never	SHA256 6ffac66ba4c1
<input type="checkbox"/> v0.12.0	a month ago	14 High	2.3 GB	Never	SHA256 6ffac66ba4c1

How to Share Your Data Science Content in a Reproducible Way

- How are Python dependencies managed?

Pinned dependencies in Pipfile using Thoth recs



- What about system-level requirements?

Project container image via AI CoE-Cl



- Does it need specific compute resources e.g. GPUs?

Cloud environment via Operate-First

Share via JupyterHub

← → ↺ 🔒 jupyterhub-odh-jupyterhub.apps.odh-cl1.apps.os-climate.org/hub/spawn/chauhankaranraj?next=%2F

jupyterhub Home Token Services ▾

Start a notebook server

Select options for your notebook server.

Notebook image

<input checked="" type="radio"/> AICoE OS-Climate Demo	<input type="radio"/> Corporate Data Pipeline
<input type="radio"/> Custom Elyra Notebook Image	<input type="radio"/> Standard Data Science ⓘ Python v3.8.3
<input type="radio"/> Elyra Notebook Image	<input type="radio"/> Minimal Python ⓘ Python v3.8.3
<input type="radio"/> Minimal Python ⓘ Python v3.6.8	<input type="radio"/> SciPy Notebook Image
<input type="radio"/> Minimal Python with Apache Spark	<input type="radio"/> Minimal Python with Apache Spark and SciPy
<input type="radio"/> Tensorflow Notebook Image	

Deployment size

Container size

1 cpu, 8GB mem ▾

Number of GPUs

0 ▾

Environment variables

[+ Add more variables](#)

[Start server](#)

Add to JupyterHub

```
1 ---
2 apiVersion: image.openshift.io/v1
3 kind: ImageStream
4 metadata:
5   name: aicoe-osc-demo
6   labels:
7     opendatahub.io/notebook-image: "true"
8   annotations:
9     opendatahub.io/notebook-image-url:
10       "https://github.com/os-climate/aicoe-osc-demo"
11     opendatahub.io/notebook-image-name:
12       "AICoE OS-Climate Demo"
13     opendatahub.io/notebook-image-desc:
14       "Jupyter notebooks for the Nimbus and the Cirrus demos for OS-Climate"
15 spec:
16   lookupPolicy:
17     local: true
18   tags:
19     - name: "latest"
20     from:
21       kind: DockerImage
22       name: quay.io/os-climate/aicoe-osc-demo:latest
23     annotations:
24       openshift.io/imported-from: quay.io/os-climate/aicoe-osc-demo
25     importPolicy:
26       scheduled: true
```

Add to JupyterHub

The screenshot shows a GitHub pull request page for the repository 'operate-first / apps'. The pull request is titled 'Add imagestream for aicoe osc demo notebooks image. #1108' and is in the 'Merged' state. It was merged by HumairAK on Oct 1. The pull request includes 2 commits and 3 files changed. The files changed section shows a diff for the file 'kfdefs/overlays/osc/osc-cl1/jupyterhub/notebook-images/aicoe-osc-demo.yaml'. The diff shows the following changes:

```
@@ -0,0 +1,26 @@
1 + ---
2 + apiVersion: image.openshift.io/v1
3 + kind: ImageStream
4 + metadata:
5 +   name: aicoe-osc-demo
6 +   labels:
7 +     opendatahub.io/notebook-image: "true"
```

Add to JupyterHub

← → ↻ 🔒 jupyterhub-odh-jupyterhub.apps.odh-cl1.apps.os-climate.org/hub/spawn/chauhankaranraj

jupyterhub Home Token Services ▾

Start a notebook server

Select options for your notebook server.

Notebook image

☒ AICoE OS-Climate Demo

☐ Custom Elyra Notebook Image

☐ Elyra Notebook Image

☐ Minimal Python ⓘ
Python v3.6.8

☐ Minimal Python with Apache Spark

☐ Tensorflow Notebook Image

☐ Corporate Data Pipeline

☐ Standard Data Science ⓘ
Python v3.8.3

☐ Minimal Python ⓘ
Python v3.8.3

☐ SciPy Notebook Image

☐ Minimal Python with Apache Spark and SciPy

Deployment size

Container size

1 cpu, 8GB mem ▾

Number of GPUs

0 ▾

How to Share Your Data Science Content in a Reproducible Way

- How are Python dependencies managed?

Pinned dependencies in Pipfile using Thoth recs



- What about system-level requirements?

Project container image via AI CoE-Cl



- Does it need specific compute resources e.g. GPUs?

Cloud environment via Operate-First



Recap

- Thoth recommended, pinned dependencies in Pipfile
- Set up AICoE-CI
 - Add .aicoe-ci.yaml config file to repo
 - Open issue to onboard project
- Create release or open an issue to build image
- Add image to JupyterHub through a PR

Demo

- Example [repo](#) with aicoe-ci set up
- Create release and trigger build



Bonus: Meteor





shower.meteor.zone

The screenshot shows a web browser at the URL `shower.meteor.zone`. The page has a dark background with a starry space theme. The main heading is "Meteor" with the subtitle "Take your Jupyter Notebooks for a spin and show your impact." Below this is a form with a text input field containing the URL `https://github.com/os-climate/aicoe-osc-demo`, a green checkmark icon, and a blue button with a right arrow. Below the input field are three dropdown menus: "Branch" set to "default", "Expiration" set to "24h", and "Components" set to "2". Below the form, there is a section titled "Project Meteor" and another titled "How it works:".

Meteor

Take your Jupyter Notebooks for a spin and show your impact.

Branch  default  Expiration 24h  Components 2 

Project Meteor is an open source technology that automates the creation of interactive environments alongside publication-quality static content.

How it works:

1. Provide a link to your GitHub repo in the form above.

Q & A