M3s: Apache Mesos and K3s

Resource-conscious Platform for ML and Data-Processing

Apachecon 2022 - Marcel Neuhausler

ISSGOVERNANCE.COM

Agenda

- Motivation
- PaaS - M3s Developer Cloud
  - Apache Mesos
  - Frameworks - M3s
  - M3s Management
- SaaS - M3s Data-Science
  - Jupyterlab
  - Apache Spark
  - Data Pipeline - Apache NiFi. Apache Airflow
  - ML Platform, MLOps
- M3trics
- M3s Future
- Summary

Motivation

ISS Tech Innovation Lab

Taking ML to Production - a Journey

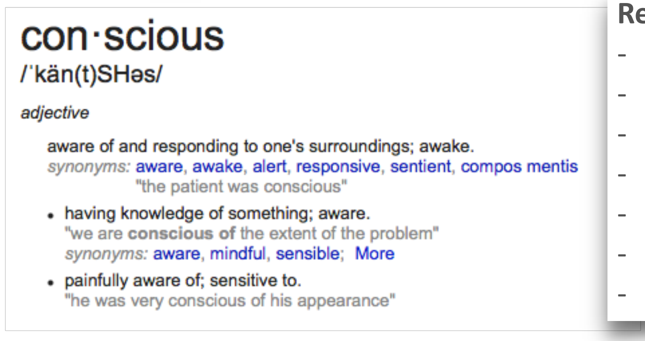
Innovation: Data Inspired Continuous Exploration

Guiding Principles

- Strive for Simplicity
- Partition where and whenever you can .. and you have tools to support and automate it.
- Be Resource Conscious .. throughout each layer of the stack.
- Prioritize Independence .. “Don’t be driven by technology, drive it" - BMW
- It is all about a healthy balance .. which to find is the hardest part.

You Build It You Run It

“Simplicity is a great virtue but it requires hard work to achieve it and education to appreciate it. And to make matters worse: complexity sells better.”
- Edsger Dijkstra



con·scious
/ˈkän(t)SHəs/

adjective

aware of and responding to one's surroundings; awake.
synonyms: aware, awake, alert, responsive, sentient, compos mentis
"the patient was conscious"

- having knowledge of something; aware.
  "we are **conscious of** the extent of the problem"
  synonyms: aware, mindful, sensible; More
- painfully aware of; sensitive to.
  "he was very conscious of his appearance"

Resource Conscious Infrastructure

- Hardware Constraints
- Energy Consumption
- Head Count
- Operational Overhead
- Configuration, Deployment Complexity
- Effort to Debug, to Detect Anomalies
- Time to Develop versus Time to Operate

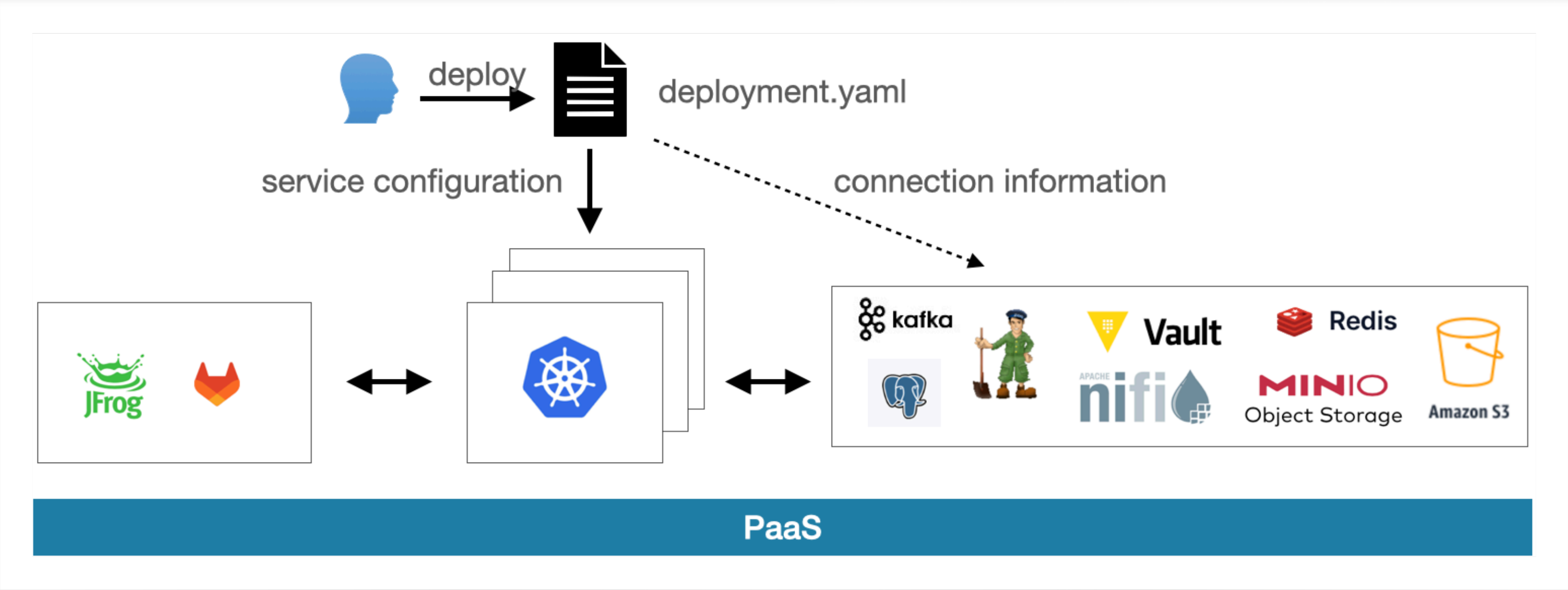
Be careful ...

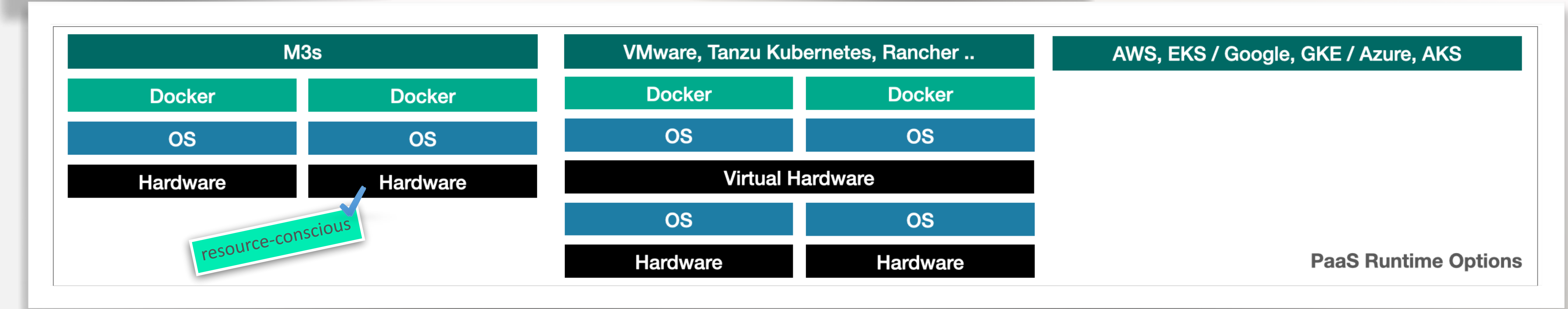
The anomaly of cheap complexity

It is often more cost-effective to take a very complicated device, and make it simulate simplicity, than to make a simpler device.

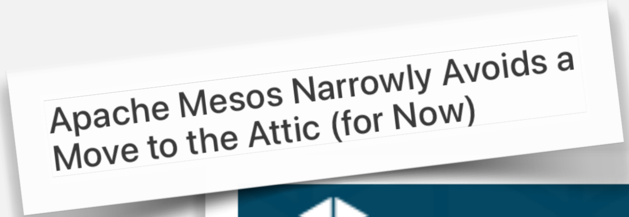
https://freedom-to-tinker.com/2022/08/03/the-anomaly-of-cheap-complexity/

PaaS - Developer Cloud

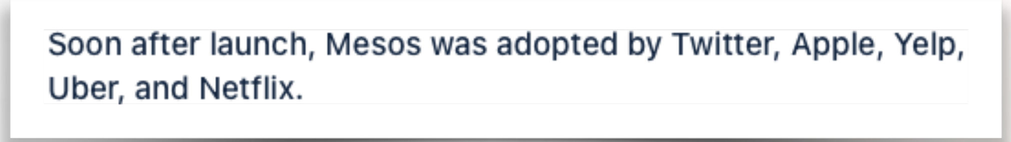




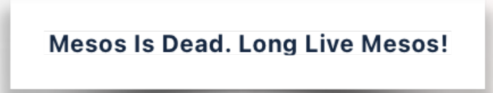
Apache Mesos - Past & Present



Apache Mesos Narrowly Avoids a Move to the Attic (for Now)

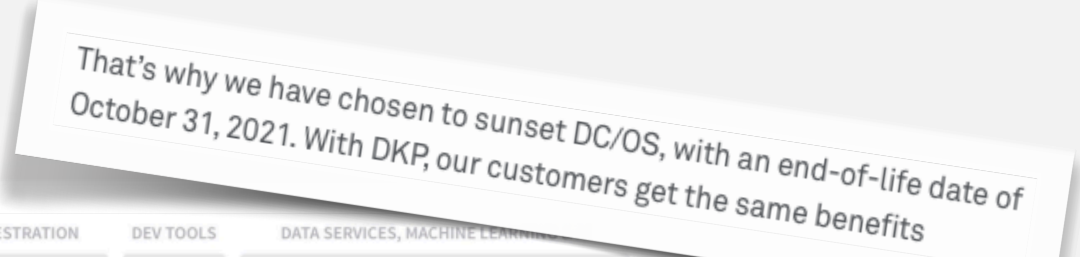


Soon after launch, Mesos was adopted by Twitter, Apple, Yelp, Uber, and Netflix.

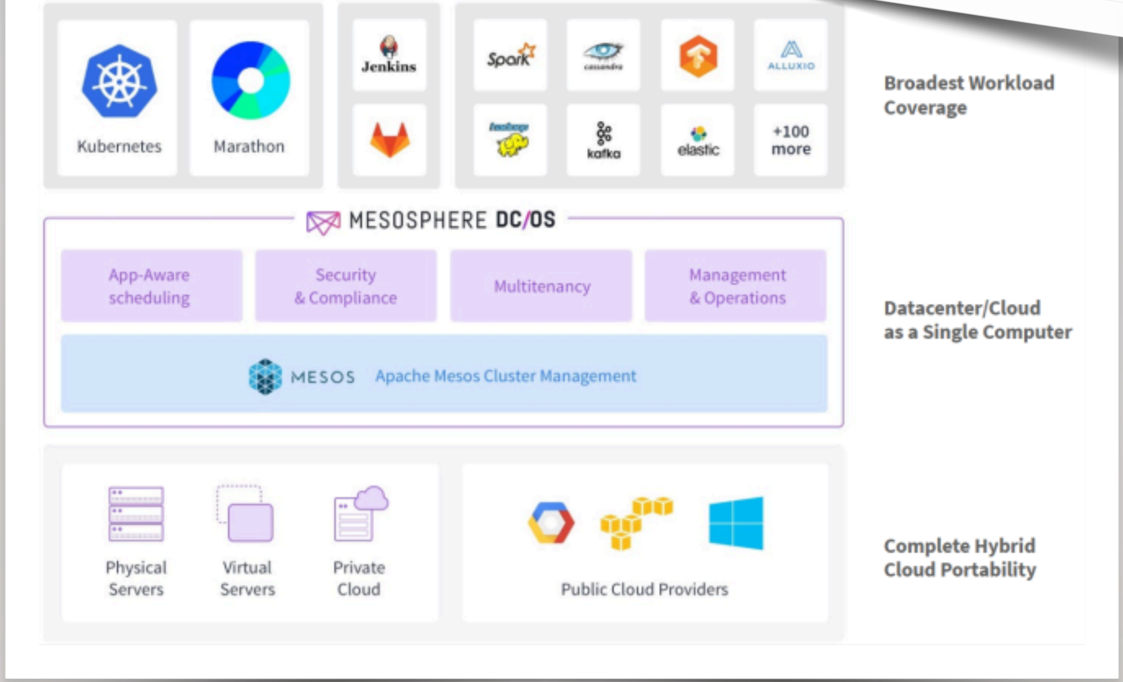


Mesos Is Dead. Long Live Mesos!

Thanks to the current Apache Mesos main committers!
- Charles-Françoise Natali
- Qian Zhang
- Andreas Peters
- …

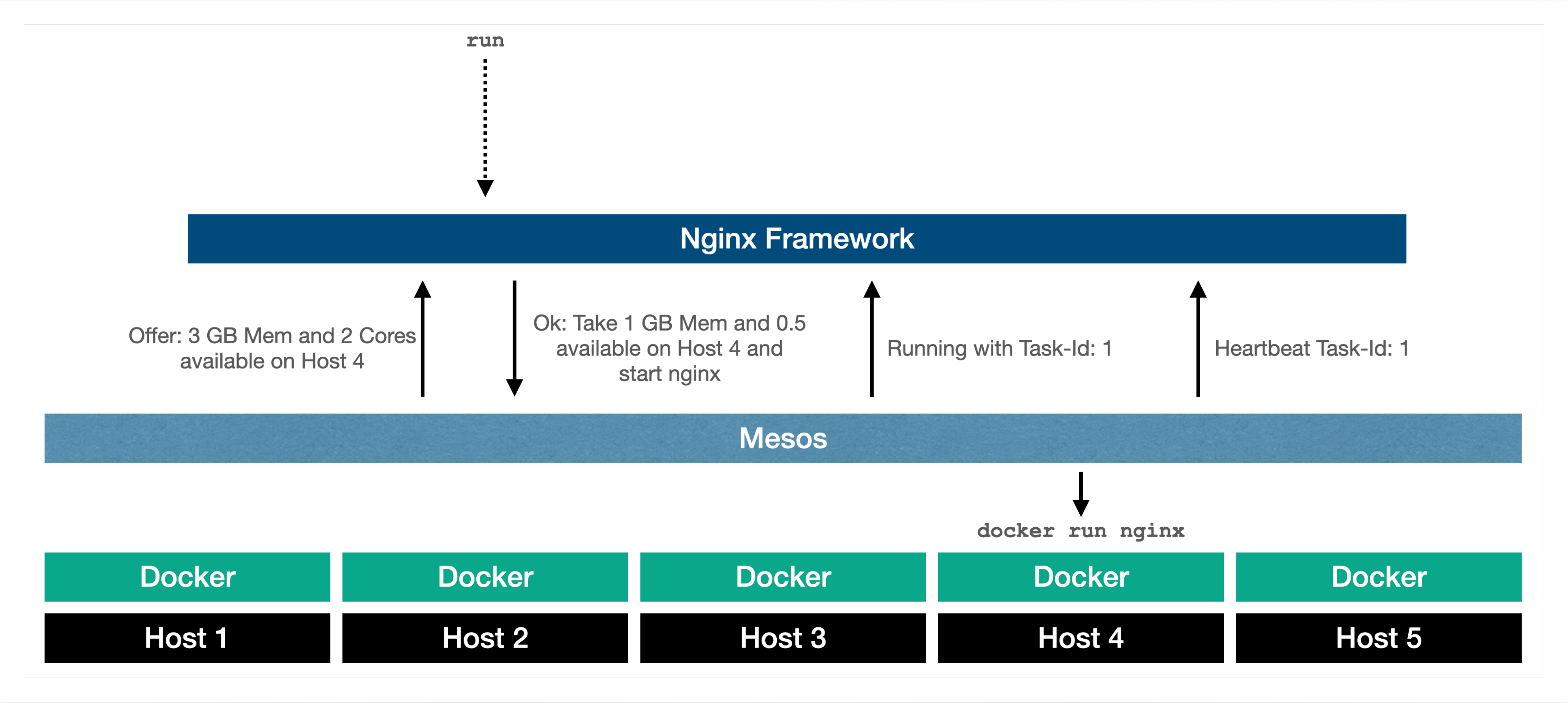


That's why we have chosen to sunset DC/OS, with an end-of-life date of October 31, 2021. With DKP, our customers get the same benefits



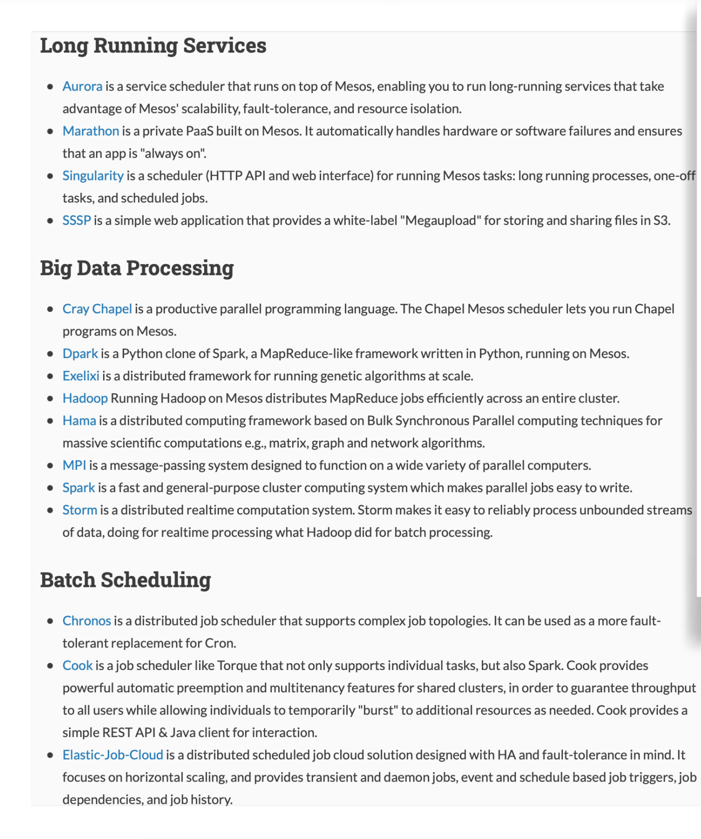
https://mesos.apache.org/
https://thenewstack.io/apache-mesos-narrowly-avoids-a-move-to-the-attic-for-now/
https://d2iq.com/products/dcos

Apache Mesos



Apache Mesos - Frameworks

The Past



Long Running Services

- Aurora is a service scheduler that runs on top of Mesos, enabling you to run long-running services that take advantage of Mesos' scalability, fault-tolerance, and resource isolation.
- Marathon is a private PaaS built on Mesos. It automatically handles hardware or software failures and ensures that an app is "always on".
- Singularity is a scheduler (HTTP API and web interface) for running Mesos tasks: long running processes, one-off tasks, and scheduled jobs.
- SSSP is a simple web application that provides a white-label "Megaupload" for storing and sharing files in S3.

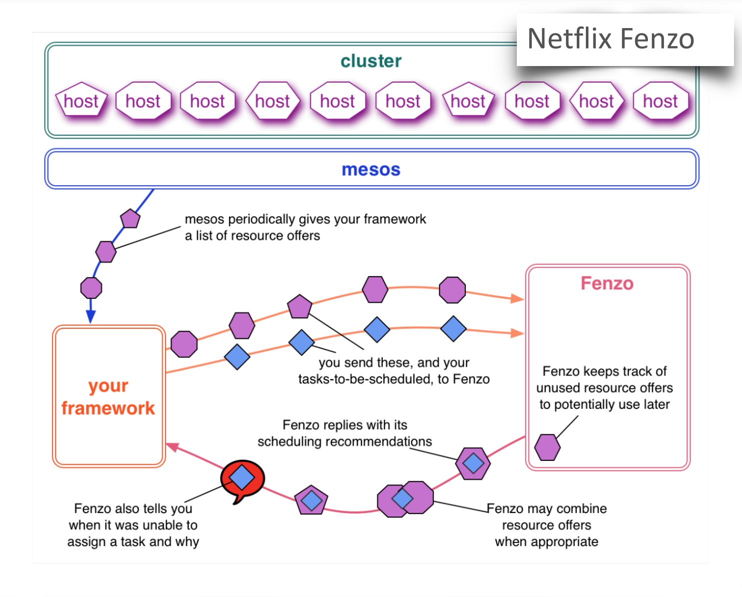
Big Data Processing

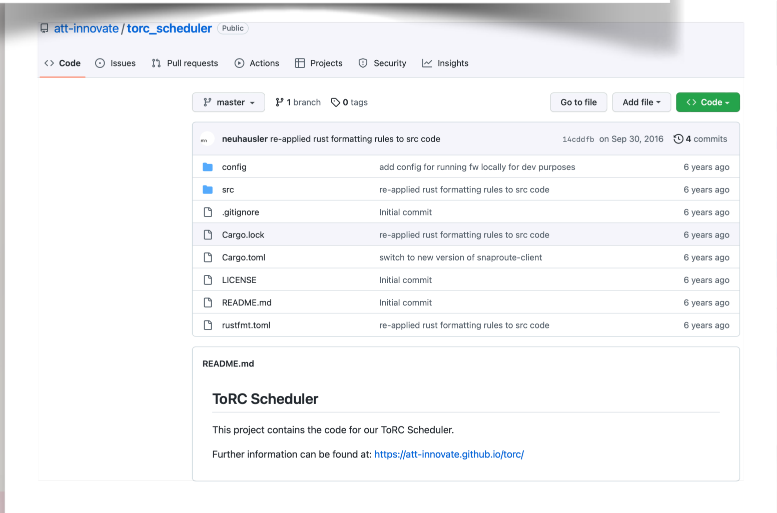
- Cray Chapel is a productive parallel programming language. The Chapel Mesos scheduler lets you run Chapel programs on Mesos.
- Dpark is a Python clone of Spark, a MapReduce-like framework written in Python, running on Mesos.
- Exelixi is a distributed framework for running genetic algorithms at scale.
- Hadoop Running Hadoop on Mesos distributes MapReduce jobs efficiently across an entire cluster.
- Hama is a distributed computing framework based on Bulk Synchronous Parallel computing techniques for massive scientific computations e.g., matrix, graph and network algorithms.
- MPI is a message-passing system designed to function on a wide variety of parallel computers.
- Spark is a fast and general-purpose cluster computing system which makes parallel jobs easy to write.
- Storm is a distributed realtime computation system. Storm makes it easy to reliably process unbounded streams of data, doing for realtime processing what Hadoop did for batch processing.

Batch Scheduling

- Chronos is a distributed job scheduler that supports complex job topologies. It can be used as a more fault-tolerant replacement for Cron.
- Cook is a job scheduler like Torque that not only supports individual tasks, but also Spark. Cook provides powerful automatic preemption and multitenancy features for shared clusters, in order to guarantee throughput to all users while allowing individuals to temporarily "burst" to additional resources as needed. Cook provides a simple REST API & Java client for interaction.
- Elastic-Job-Cloud is a distributed scheduled job cloud solution designed with HA and fault-tolerance in mind. It focuses on horizontal scaling, and provides transient and daemon jobs, event and schedule based job triggers, job dependencies, and job history.

Netflix Fenzo


cluster
host
host
host
host
host
host
host
host
host
host
mesos
mesos periodically gives your framework a list of resource offers
your framework
Fenzo
you send these, and your tasks-to-be-scheduled, to Fenzo
Fenzo keeps track of unused resource offers to potentially use later
Fenzo replies with its scheduling recommendations
Fenzo also tells you when it was unable to assign a task and why
Fenzo may combine resource offers when appropriate



neuhausler re-applied rust formatting rules to src code 14cddfb on Sep 30, 2016 4 commits

config	add config for running fw locally for dev purposes	6 years ago
src	re-applied rust formatting rules to src code	6 years ago
.gitignore	Initial commit	6 years ago
Cargo.lock	re-applied rust formatting rules to src code	6 years ago
Cargo.toml	switch to new version of snaproute-client	6 years ago
LICENSE	Initial commit	6 years ago
README.md	Initial commit	6 years ago
rustfmt.toml	re-applied rust formatting rules to src code	6 years ago

README.md

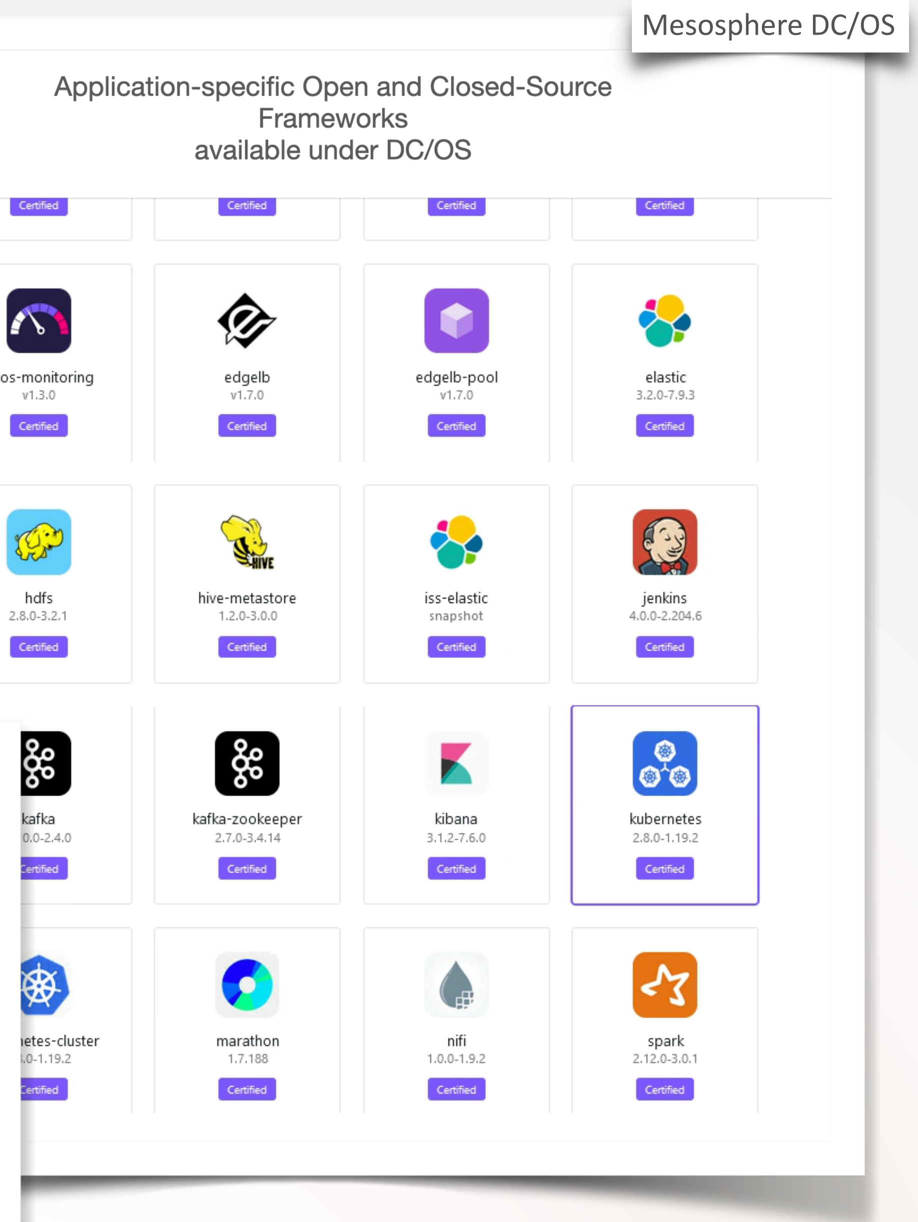
ToRC Scheduler

This project contains the code for our ToRC Scheduler.

Further information can be found at: https://att-innovate.github.io/torc/

Mesosphere DC/OS

Application-specific Open and Closed-Source Frameworks available under DC/OS

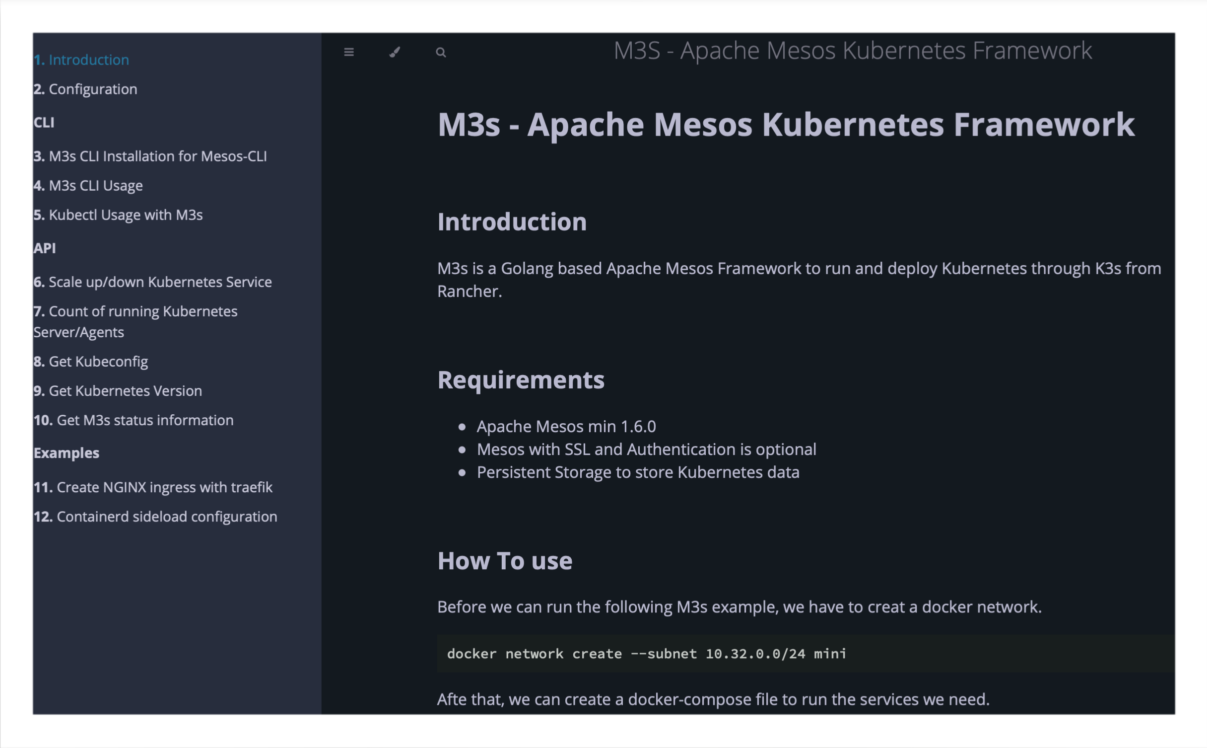


https://mesos.readthedocs.io/en/latest/frameworks/
https://netflixtechblog.com/fenzo-oss-scheduler-for-apache-mesos-frameworks-5c340e77e543
https://github.com/att-innovate/torc_scheduler

Apache Mesos - M3s Frameworks

Current

Aventer UG - Andreas Peters



1. Introduction
2. Configuration

CLI

3. M3s CLI Installation for Mesos-CLI
4. M3s CLI Usage
5. Kubectl Usage with M3s

API

6. Scale up/down Kubernetes Service
7. Count of running Kubernetes Server/Agents
8. Get Kubeconfig
9. Get Kubernetes Version
10. Get M3s status information

Examples

11. Create NGINX ingress with traefik
12. Containerd sideload configuration

M3S - Apache Mesos Kubernetes Framework

M3s - Apache Mesos Kubernetes Framework

Introduction

M3s is a Golang based Apache Mesos Framework to run and deploy Kubernetes through K3s from Rancher.

Requirements

- Apache Mesos min 1.6.0
- Mesos with SSL and Authentication is optional
- Persistent Storage to store Kubernetes data

How To use

Before we can run the following M3s example, we have to creat a docker network.

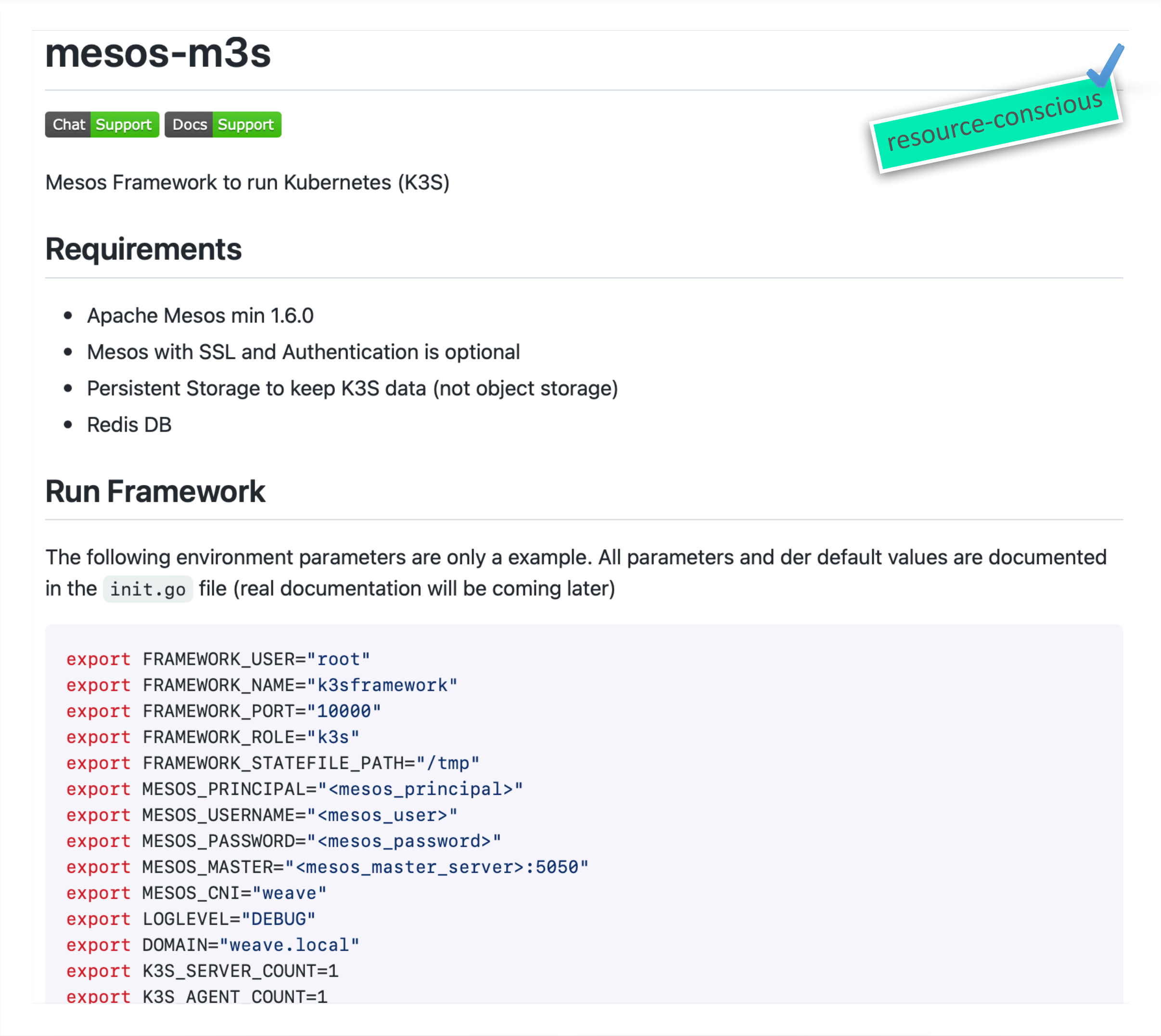
```
docker network create --subnet 10.32.0.0/24 mini
```

Afte that, we can create a docker-compose file to run the services we need.

https://aventer-ug.github.io/mesos-m3s/index.html
https://github.com/AVENTER-UG/mesos-m3s
https://k3s.io/

K3S

resource-conscious ✓



mesos-m3s

Chat Support Docs Support

resource-conscious ✓

Mesos Framework to run Kubernetes (K3S)

Requirements

- Apache Mesos min 1.6.0
- Mesos with SSL and Authentication is optional
- Persistent Storage to keep K3S data (not object storage)
- Redis DB

Run Framework

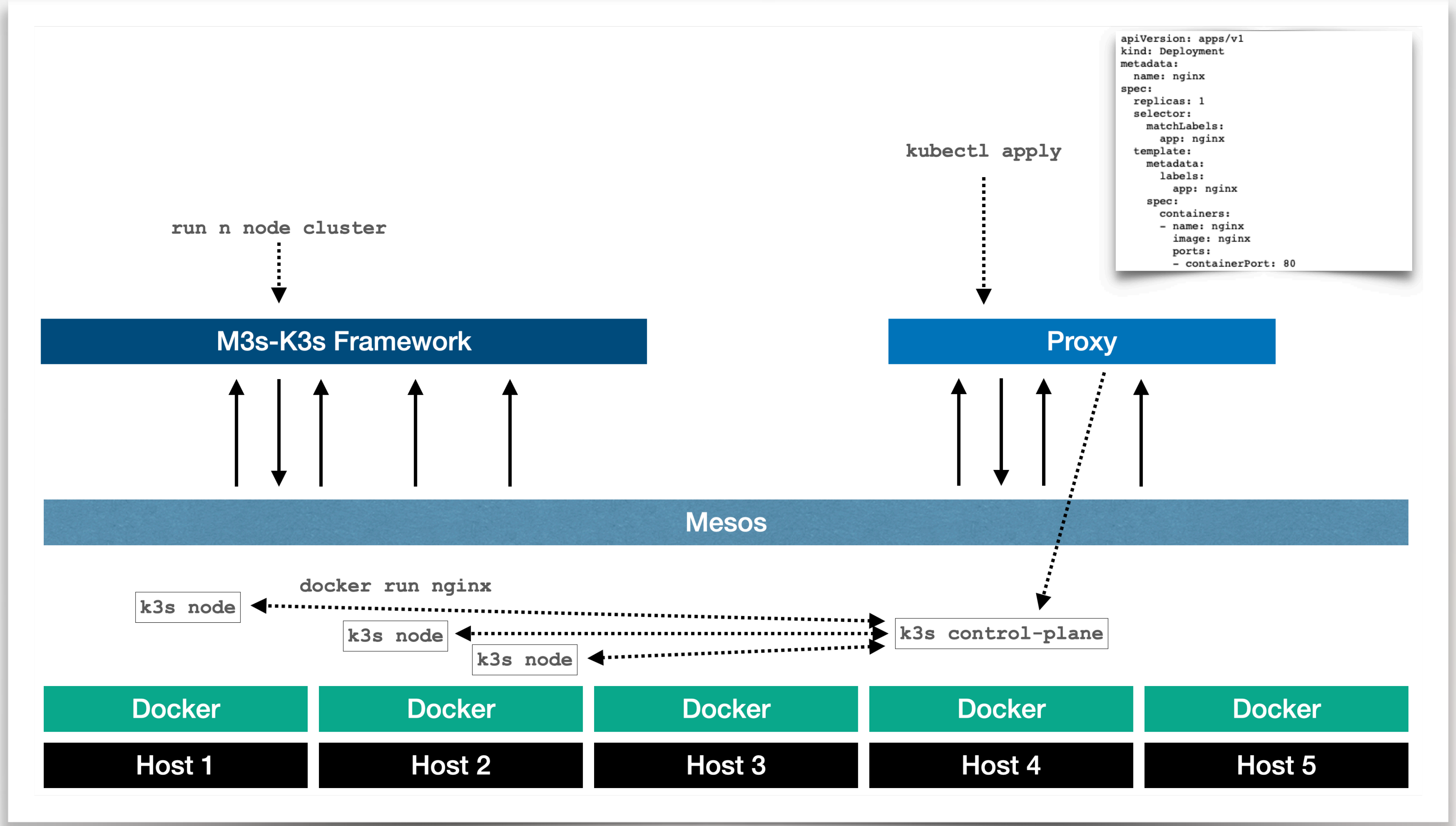
The following environment parameters are only a example. All parameters and der default values are documented in the `init.go` file (real documentation will be coming later)

```
export FRAMEWORK_USER="root"
export FRAMEWORK_NAME="k3sframework"
export FRAMEWORK_PORT="10000"
export FRAMEWORK_ROLE="k3s"
export FRAMEWORK_STATEFILE_PATH="/tmp"
export MESOS_PRINCIPAL="<mesos_principal>"
export MESOS_USERNAME="<mesos_user>"
export MESOS_PASSWORD="<mesos_password>"
export MESOS_MASTER="<mesos_master_server>:5050"
export MESOS_CNI="weave"
export LOGLEVEL="DEBUG"
export DOMAIN="weave.local"
export K3S_SERVER_COUNT=1
export K3S_AGENT_COUNT=1
```

Apache Mesos - M3s

Baremetal-installs:
- Mesos-Master/Agent
- Mesos-DNS
- Apache Zookeeper
- Hashicorp Vault
- Docker

resource-conscious

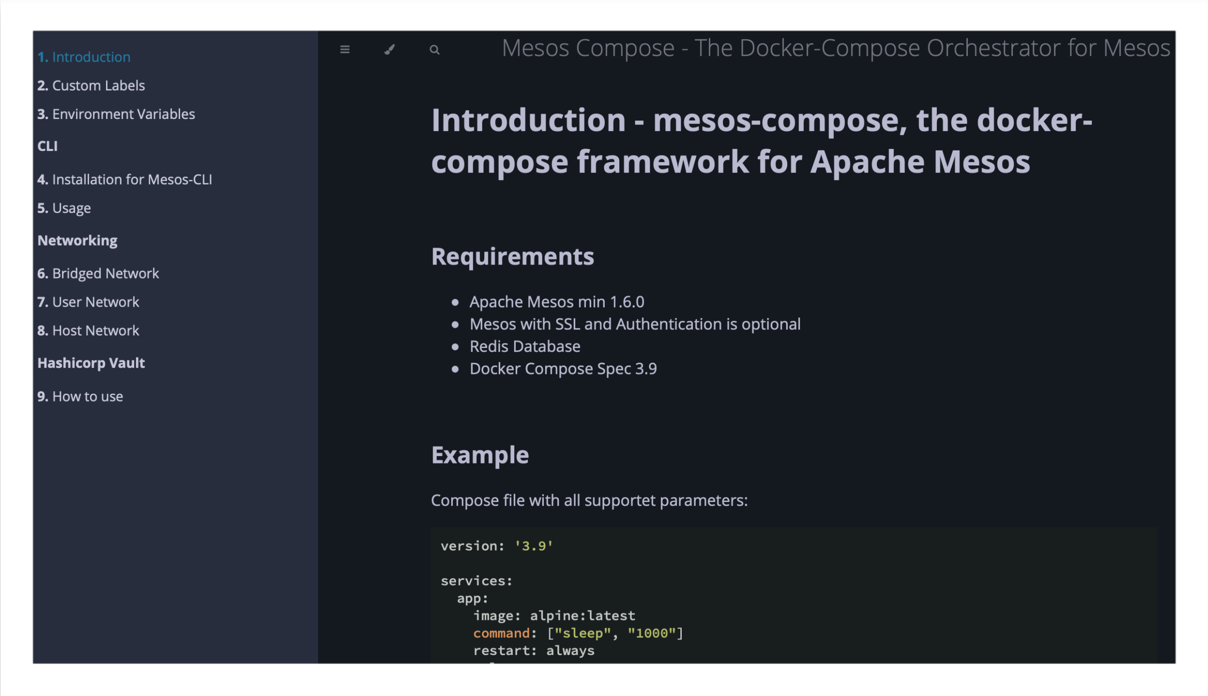


```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx
        ports:
        - containerPort: 80
```

Apache Mesos - M3s Frameworks

Current

Aventer UG - Andreas Peters



Mesos Compose - The Docker-Compose Orchestrator for Mesos

Introduction - mesos-compose, the docker-compose framework for Apache Mesos

Requirements

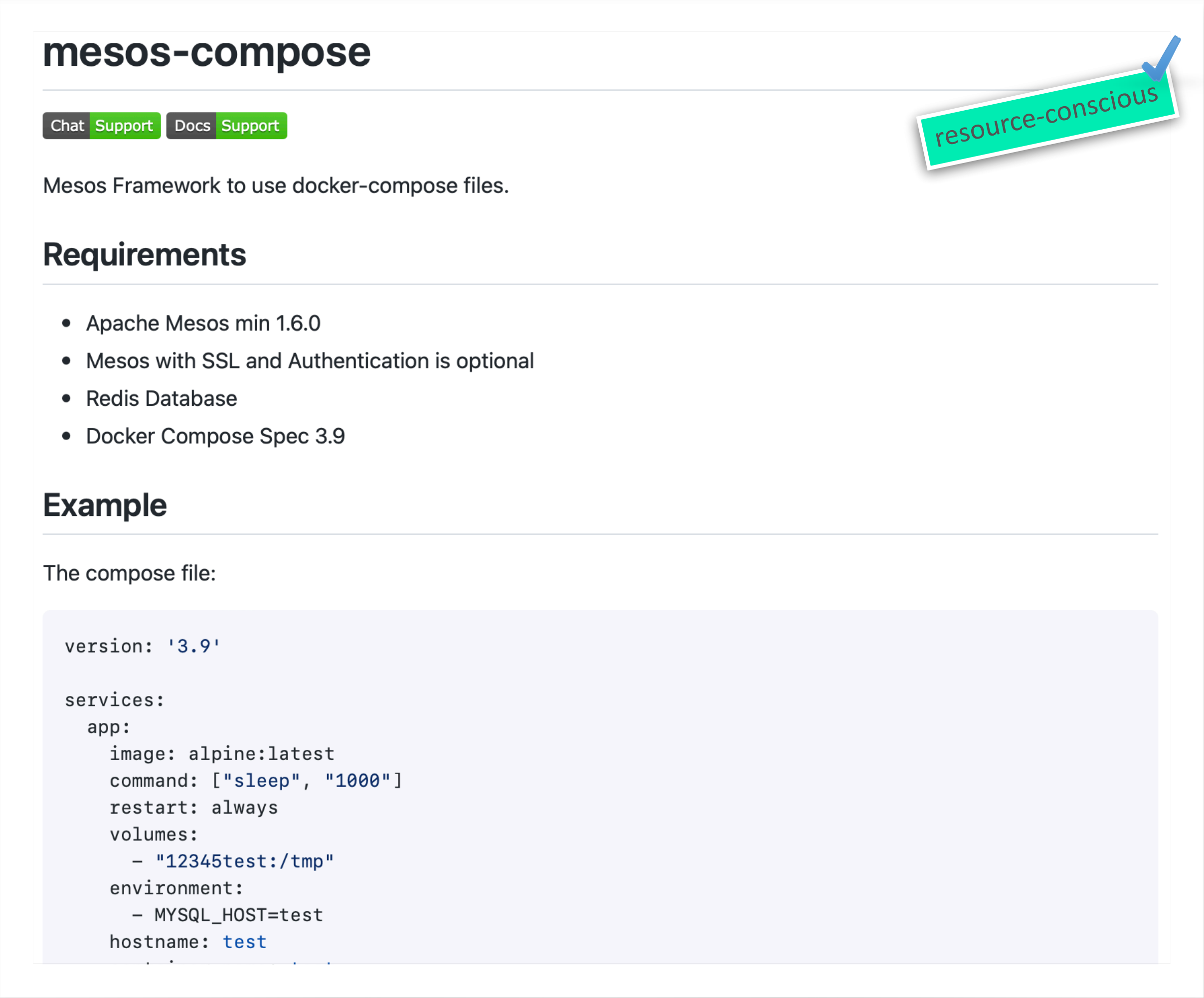
- Apache Mesos min 1.6.0
- Mesos with SSL and Authentication is optional
- Redis Database
- Docker Compose Spec 3.9

Example

Compose file with all supportet parameters:

```
version: '3.9'

services:
  app:
    image: alpine:latest
    command: ["sleep", "1000"]
    restart: always
```

https://github.com/AVENTER-UG/mesos-compose
https://aventer-ug.github.io/mesos-compose/



mesos-compose

Chat Support Docs Support

resource-conscious ✓

Mesos Framework to use docker-compose files.

Requirements

- Apache Mesos min 1.6.0
- Mesos with SSL and Authentication is optional
- Redis Database
- Docker Compose Spec 3.9

Example

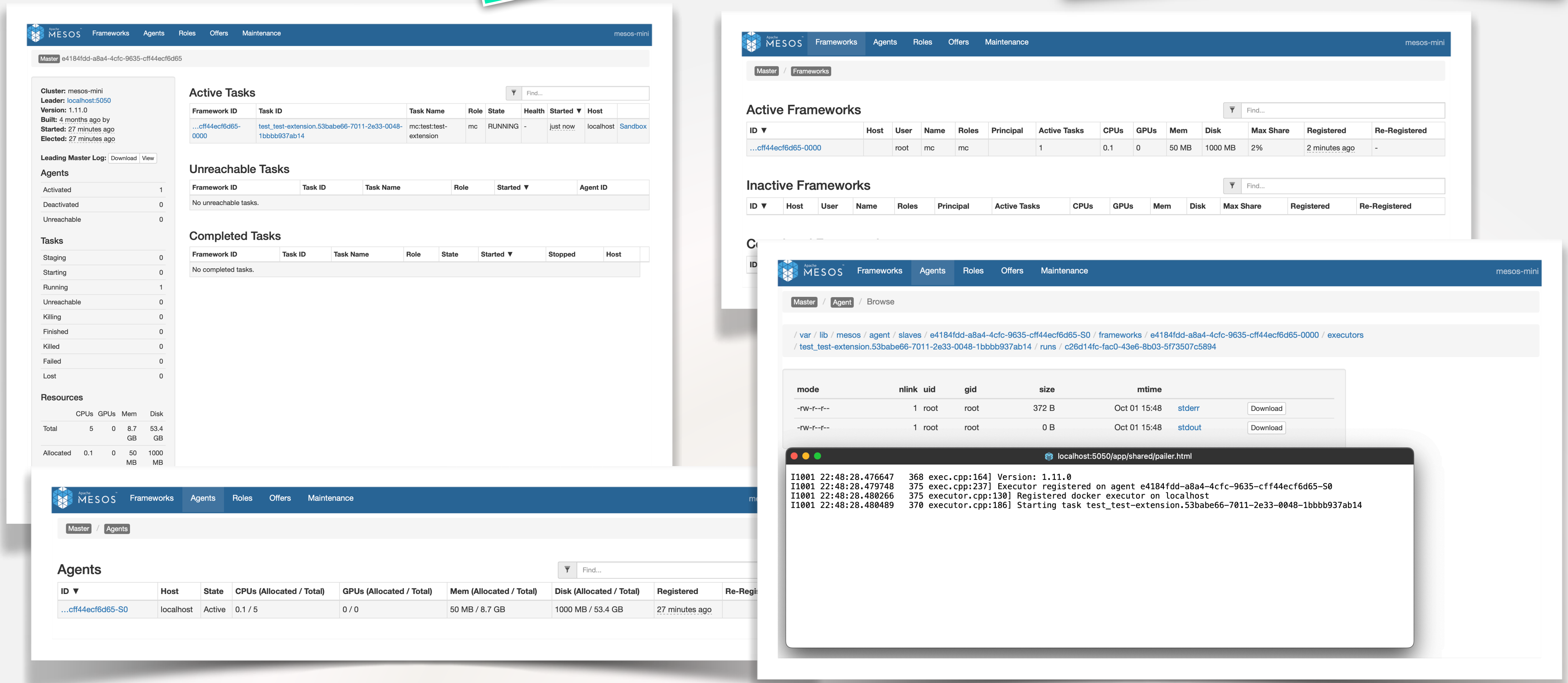
The compose file:

```
version: '3.9'

services:
  app:
    image: alpine:latest
    command: ["sleep", "1000"]
    restart: always
    volumes:
      - "12345test:/tmp"
    environment:
      - MYSQL_HOST=test
    hostname: test
```

Apache Mesos UI

resource-conscious ✓

In case you want to try it out:
https://hub.docker.com/extensions/avhost/docker-mesos-extension



Master e4184fdd-a8a4-4cfc-9635-cff44ecf6d65

Cluster: mesos-mini
Leader: localhost:5050
Version: 1.11.0
Built: 4 months ago by
Started: 27 minutes ago
Elected: 27 minutes ago

Leading Master Log: Download View

Agents

Activated	1
Deactivated	0
Unreachable	0

Tasks

Staging	0
Starting	0
Running	1
Unreachable	0
Killing	0
Finished	0
Killed	0
Failed	0
Lost	0

Resources

	CPUs	GPUs	Mem	Disk
Total	5	0	8.7 GB	53.4 GB
Allocated	0.1	0	50 MB	1000 MB

Active Tasks

Find...

Framework ID	Task ID	Task Name	Role	State	Health	Started ▼	Host	
...cff44ecf6d65-0000	test_test-extension.53babe66-7011-2e33-0048-1bbbb937ab14	mc:test:test-extension	mc	RUNNING	-	just now	localhost	Sandbox

Unreachable Tasks

Framework ID	Task ID	Task Name	Role	Started ▼	Agent ID
No unreachable tasks.					

Completed Tasks

Framework ID	Task ID	Task Name	Role	State	Started ▼	Stopped	Host
No completed tasks.							

Master / Frameworks

Active Frameworks

Find...

ID ▼	Host	User	Name	Roles	Principal	Active Tasks	CPUs	GPUs	Mem	Disk	Max Share	Registered	Re-Registered
...cff44ecf6d65-0000		root	mc	mc		1	0.1	0	50 MB	1000 MB	2%	2 minutes ago	-

Inactive Frameworks

Find...

ID ▼	Host	User	Name	Roles	Principal	Active Tasks	CPUs	GPUs	Mem	Disk	Max Share	Registered	Re-Registered

Master / Agents

Agents

Find...

ID ▼	Host	State	CPUs (Allocated / Total)	GPUs (Allocated / Total)	Mem (Allocated / Total)	Disk (Allocated / Total)	Registered	Re-Regi
...cff44ecf6d65-S0	localhost	Active	0.1 / 5	0 / 0	50 MB / 8.7 GB	1000 MB / 53.4 GB	27 minutes ago	

Master / Agent / Browse

/ var / lib / mesos / agent / slaves / e4184fdd-a8a4-4cfc-9635-cff44ecf6d65-S0 / frameworks / e4184fdd-a8a4-4cfc-9635-cff44ecf6d65-0000 / executors / test_test-extension.53babe66-7011-2e33-0048-1bbbb937ab14 / runs / c26d14fc-fac0-43e6-8b03-5f73507c5894

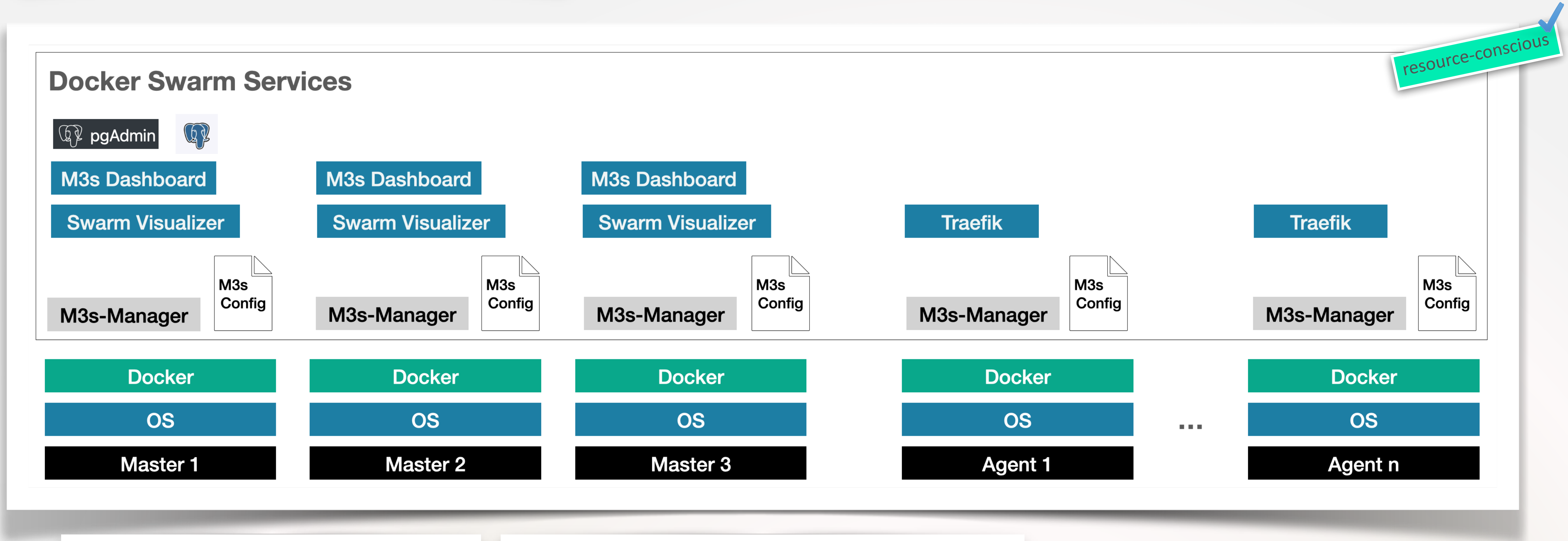
mode	nlink	uid	gid	size	mtime		
-rw-r--r--	1	root	root	372 B	Oct 01 15:48	stderr	Download
-rw-r--r--	1	root	root	0 B	Oct 01 15:48	stdout	Download

localhost:5050/app/shared/pailer.html

```
I1001 22:48:28.476647   368 exec.cpp:164] Version: 1.11.0
I1001 22:48:28.479748   375 exec.cpp:237] Executor registered on agent e4184fdd-a8a4-4cfc-9635-cff44ecf6d65-S0
I1001 22:48:28.480266   375 executor.cpp:130] Registered docker executor on localhost
I1001 22:48:28.480489   370 executor.cpp:186] Starting task test_test-extension.53babe66-7011-2e33-0048-1bbbb937ab14
```

M3s Management

Docker Swarm as replacement for systemd/puppet/ansible



https://docs.docker.com/engine/swarm/
https://www.postgresql.org/
https://www.pgadmin.org/

https://traefik.io/
https://github.com/dockersamples/docker-swarm-visualizer
https://www.hashicorp.com/products/vault

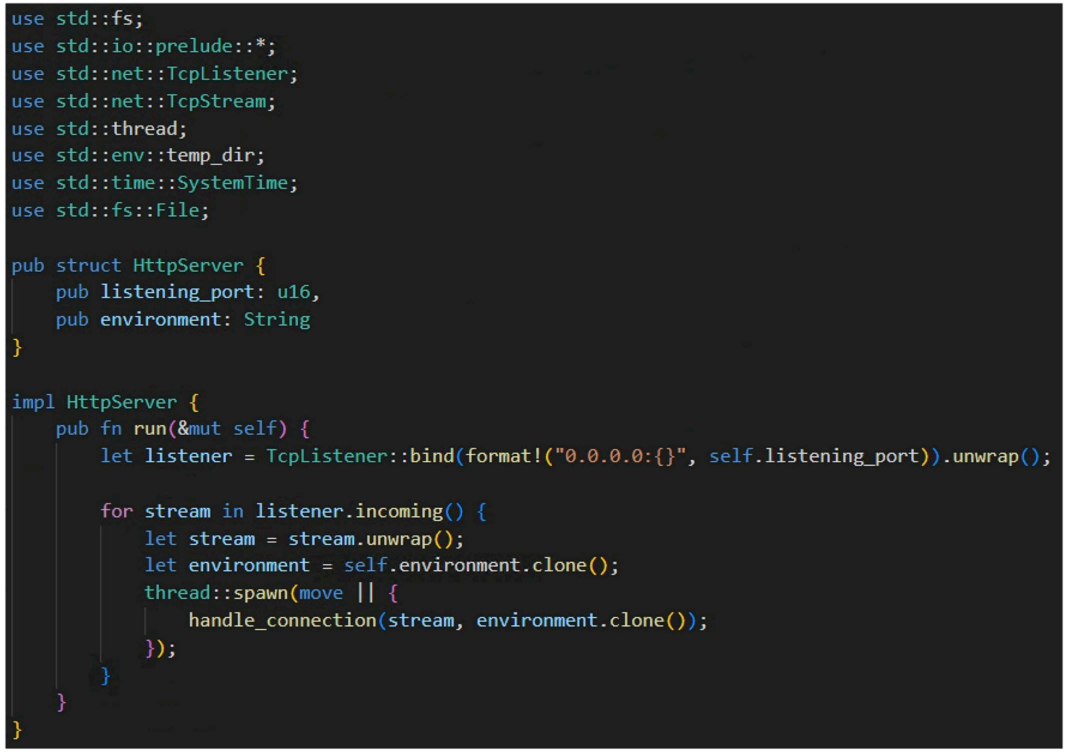
M3s Management Service

resource-conscious ✓

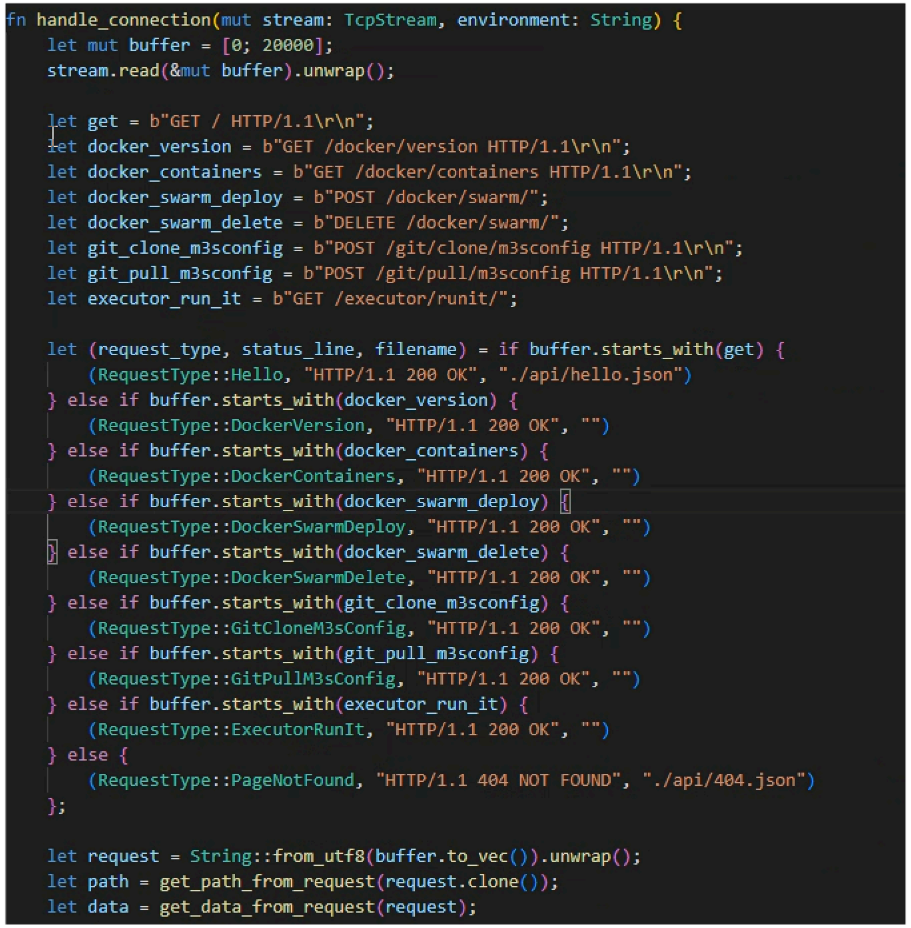
Infrastructure as Code and APIs

Provides access to Docker Engine, M3s Frameworks, Shell commands defined in M3s config

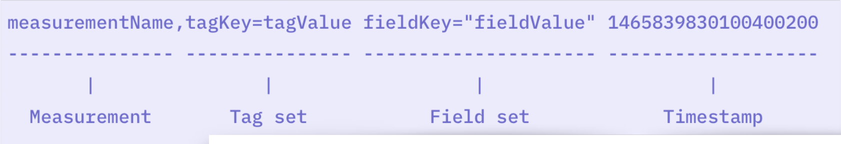
Will be open-sourced later this year.



```rust
use std::fs;
use std::io::prelude::*;
use std::net::TcpListener;
use std::net::TcpStream;
use std::thread;
use std::env::temp_dir;
use std::time::SystemTime;
use std::fs::File;

pub struct HttpServer {
    pub listening_port: u16,
    pub environment: String
}

impl HttpServer {
    pub fn run(&mut self) {
        let listener = TcpListener::bind(format!("0.0.0.0:{}", self.listening_port)).unwrap();

        for stream in listener.incoming() {
            let stream = stream.unwrap();
            let environment = self.environment.clone();
            thread::spawn(move || {
                handle_connection(stream, environment.clone());
            });
        }
    }
}
```



```rust
fn handle_connection(mut stream: TcpStream, environment: String) {
    let mut buffer = [0; 20000];
    stream.read(&mut buffer).unwrap();

    let get = b"GET / HTTP/1.1\r\n";
    let docker_version = b"GET /docker/version HTTP/1.1\r\n";
    let docker_containers = b"GET /docker/containers HTTP/1.1\r\n";
    let docker_swarm_deploy = b"POST /docker/swarm/";
    let docker_swarm_delete = b"DELETE /docker/swarm/";
    let git_clone_m3sconfig = b"POST /git/clone/m3sconfig HTTP/1.1\r\n";
    let git_pull_m3sconfig = b"POST /git/pull/m3sconfig HTTP/1.1\r\n";
    let executor_run_it = b"GET /executor/runit/";

    let (request_type, status_line, filename) = if buffer.starts_with(get) {
        (RequestType::Hello, "HTTP/1.1 200 OK", "./api/hello.json")
    } else if buffer.starts_with(docker_version) {
        (RequestType::DockerVersion, "HTTP/1.1 200 OK", "")
    } else if buffer.starts_with(docker_containers) {
        (RequestType::DockerContainers, "HTTP/1.1 200 OK", "")
    } else if buffer.starts_with(docker_swarm_deploy) {
        (RequestType::DockerSwarmDeploy, "HTTP/1.1 200 OK", "")
    } else if buffer.starts_with(docker_swarm_delete) {
        (RequestType::DockerSwarmDelete, "HTTP/1.1 200 OK", "")
    } else if buffer.starts_with(git_clone_m3sconfig) {
        (RequestType::GitCloneM3sConfig, "HTTP/1.1 200 OK", "")
    } else if buffer.starts_with(git_pull_m3sconfig) {
        (RequestType::GitPullM3sConfig, "HTTP/1.1 200 OK", "")
    } else if buffer.starts_with(executor_run_it) {
        (RequestType::ExecutorRunIt, "HTTP/1.1 200 OK", "")
    } else {
        (RequestType::PageNotFound, "HTTP/1.1 404 NOT FOUND", "./api/404.json")
    };

    let request = String::from_utf8(buffer.to_vec()).unwrap();
    let path = get_path_from_request(request.clone());
    let data = get_data_from_request(request);
```



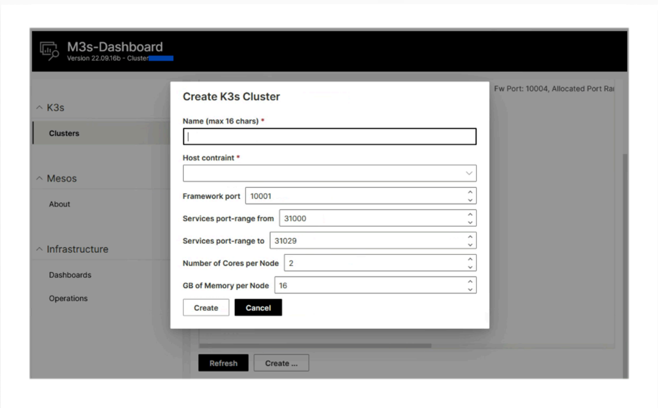
```
measurementName,tagKey=tagValue fieldKey="fieldValue" 1465839830100400200
--------------- --------------- --------------------- -------------------
       |               |                  |                    |
  Measurement       Tag set           Field set            Timestamp
```

Example: InflxuxDB Line Protocol
https://docs.influxdata.com/influxdb/cloud/reference/syntax/line-protocol/

M3s - Managing K3s Cluster

resource-conscious ✓

M3s-Dashboard
Will be open-sourced later this year.


M3s-Dashboard
Version 22.09.16b - Cluster [redacted]

Fw Port: 10004, Allocated Port Ra

Create K3s Cluster

Name (max 16 chars) *

Host contraint *

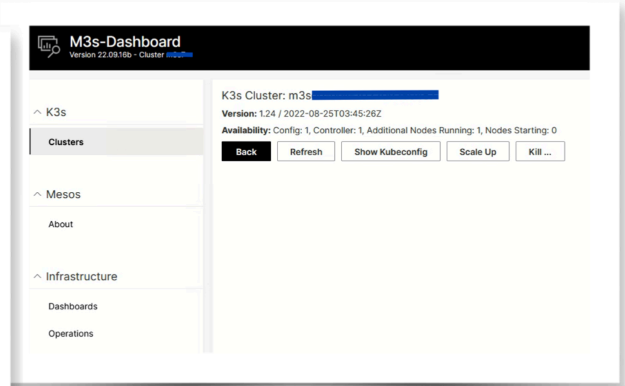
Framework port: 10001

Services port-range from: 31000

Services port-range to: 31029

Number of Cores per Node: 2

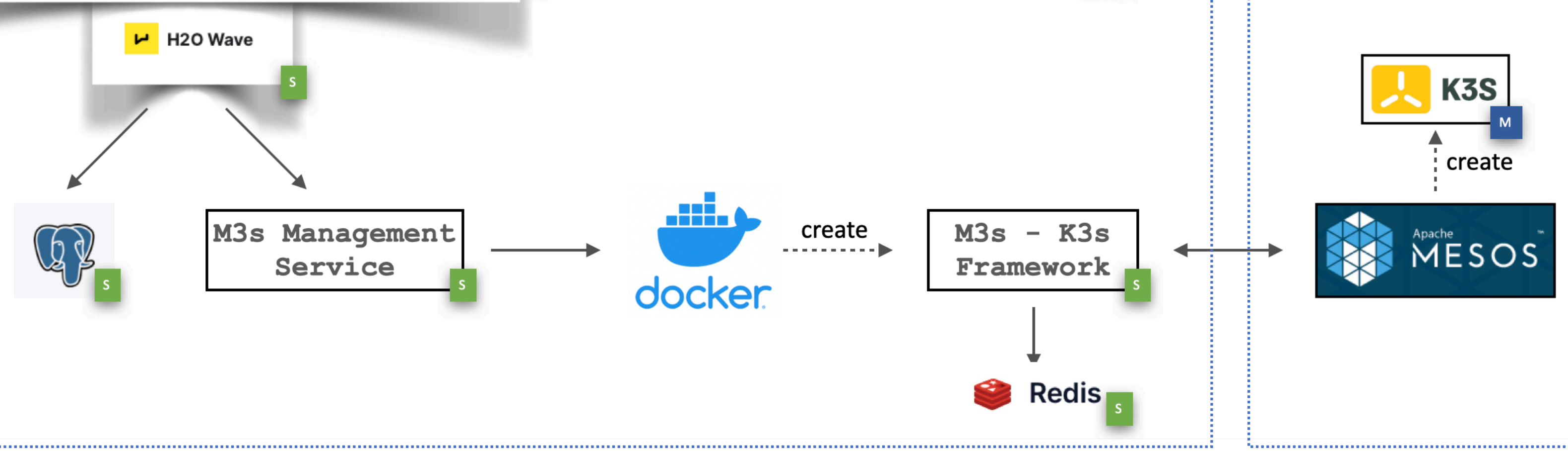
GB of Memory per Node: 16


M3s-Dashboard
Version 22.09.16b - Cluster [redacted]

K3s Cluster: m3s[redacted]

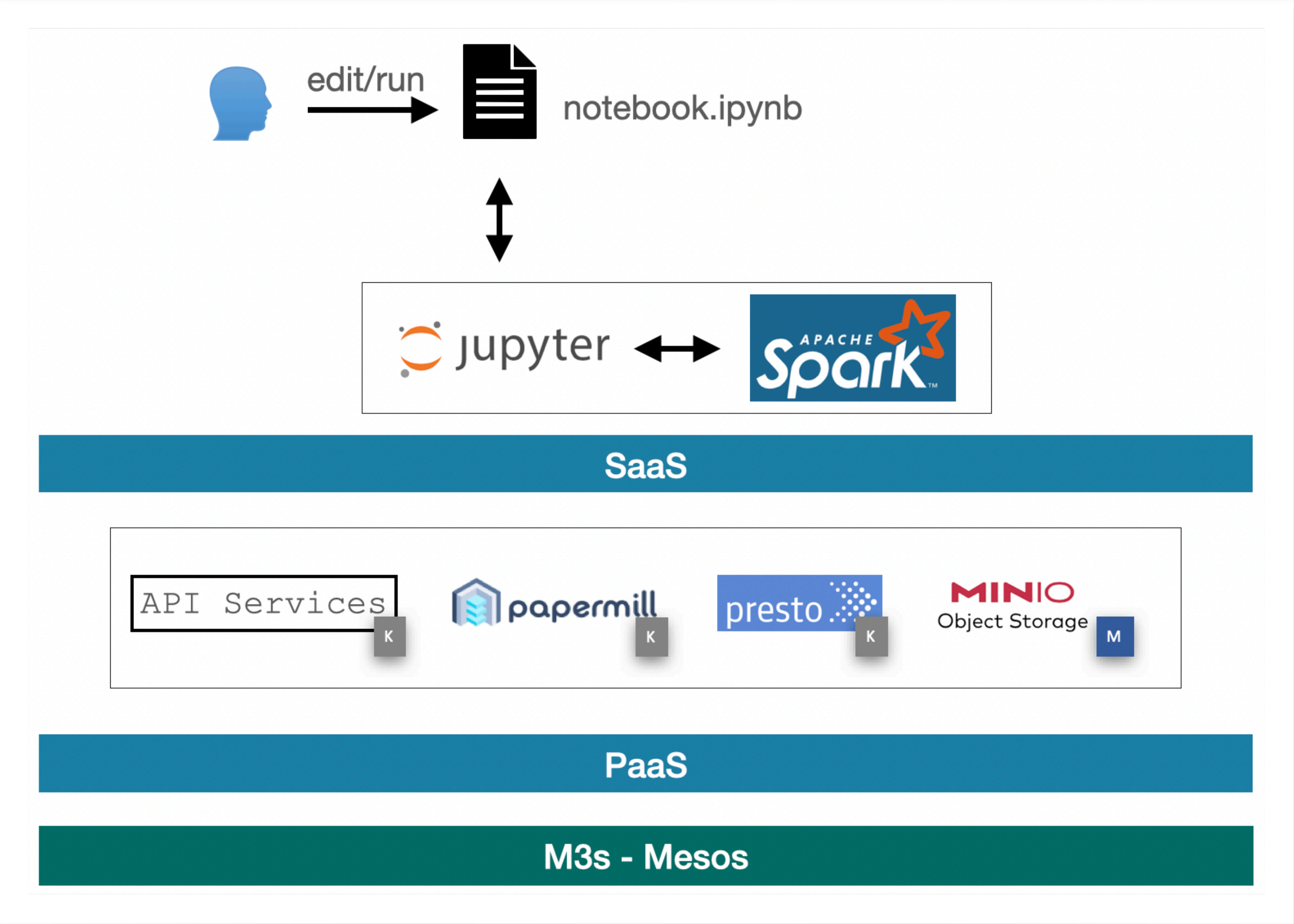
Version: 1.24 / 2022-08-25T03:45:26Z

Availability: Config: 1, Controller: 1, Additional Nodes Running: 1, Nodes Starting: 0



https://wave.h2o.ai/
https://docs.docker.com/engine/swarm/
https://www.postgresql.org/
https://redis.io/
https://k3s.io/

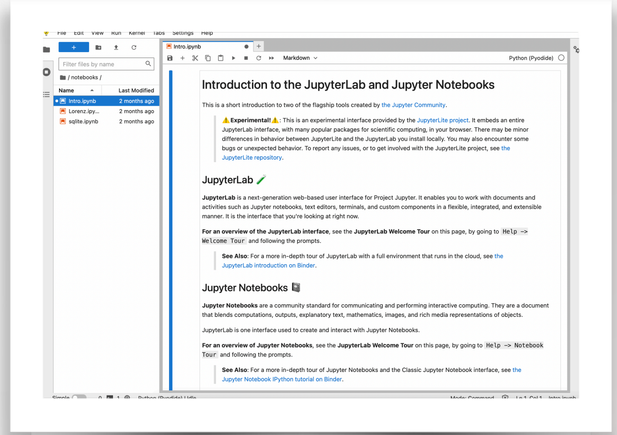
SaaS - M3s Data-Science



https://jupyter.org/
https://spark.apache.org/
https://papermill.readthedocs.io/en/latest/#
https://prestodb.io/
https://min.io/

Jupyterlab

Jupyterlab as a Pod vs Jupyterlab as a Mesos Taks



Introduction to the JupyterLab and Jupyter Notebooks

This is a short introduction to two of the flagship tools created by the Jupyter Community.

> ⚠️ **Experimental!** ⚠️: This is an experimental interface provided by the JupyterLite project. It embeds an entire JupyterLab interface, with many popular packages for scientific computing, in your browser. There may be minor differences in behavior between JupyterLite and the JupyterLab you install locally. You may also encounter some bugs or unexpected behavior. To report any issues, or to get involved with the JupyterLite project, see the JupyterLite repository.

JupyterLab 🧪

JupyterLab is a next-generation web-based user interface for Project Jupyter. It enables you to work with documents and activities such as Jupyter notebooks, text editors, terminals, and custom components in a flexible, integrated, and extensible manner. It is the interface that you're looking at right now.

For an overview of the JupyterLab interface, see the **JupyterLab Welcome Tour** on this page, by going to `Help -> Welcome Tour` and following the prompts.

> **See Also**: For a more in-depth tour of JupyterLab with a full environment that runs in the cloud, see the JupyterLab introduction on Binder.

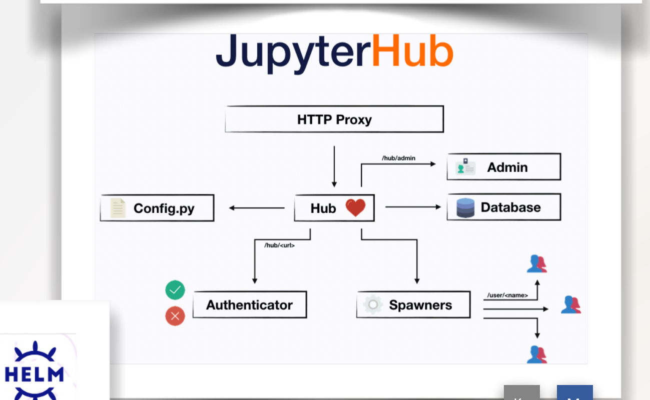
Jupyter Notebooks 📓

Jupyter Notebooks are a community standard for communicating and performing interactive computing. They are a document that blends computations, outputs, explanatory text, mathematics, images, and rich media representations of objects.

JupyterLab is one interface used to create and interact with Jupyter Notebooks.

For an overview of Jupyter Notebooks, see the **JupyterLab Welcome Tour** on this page, by going to `Help -> Notebook Tour` and following the prompts.

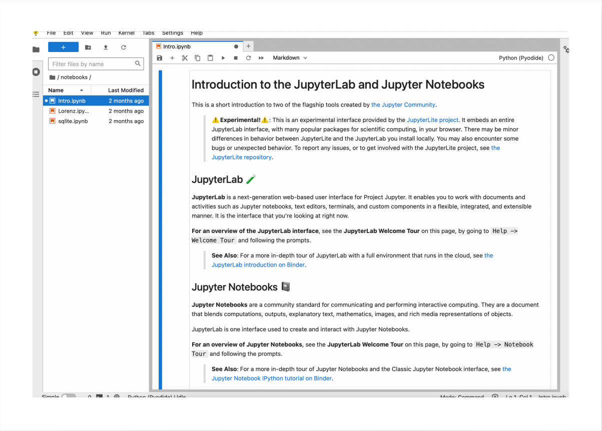
> **See Also**: For a more in-depth tour of Jupyter Notebooks and the Classic Jupyter Notebook interface, see the Jupyter Notebook IPython tutorial on Binder.



K M

HELM

Zero to JupyterHub with Kubernetes
https://zero-to-jupyterhub.readthedocs.io/en/stable


Introduction to the JupyterLab and Jupyter Notebooks

This is a short introduction to two of the flagship tools created by the Jupyter Community.

> ⚠️ **Experimental!** ⚠️: This is an experimental interface provided by the JupyterLite project. It embeds an entire JupyterLab interface, with many popular packages for scientific computing, in your browser. There may be minor differences in behavior between JupyterLite and the JupyterLab you install locally. You may also encounter some bugs or unexpected behavior. To report any issues, or to get involved with the JupyterLite project, see the JupyterLite repository.

JupyterLab 🧪

JupyterLab is a next-generation web-based user interface for Project Jupyter. It enables you to work with documents and activities such as Jupyter notebooks, text editors, terminals, and custom components in a flexible, integrated, and extensible manner. It is the interface that you're looking at right now.

For an overview of the JupyterLab interface, see the **JupyterLab Welcome Tour** on this page, by going to `Help -> Welcome Tour` and following the prompts.

> **See Also**: For a more in-depth tour of JupyterLab with a full environment that runs in the cloud, see the JupyterLab introduction on Binder.

Jupyter Notebooks 📓

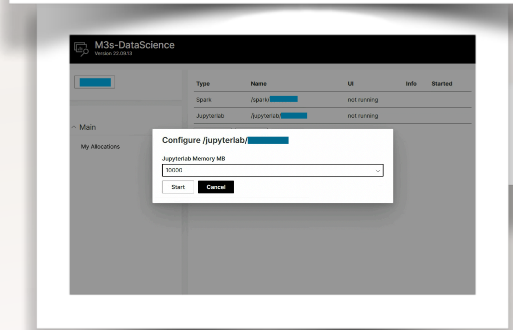
Jupyter Notebooks are a community standard for communicating and performing interactive computing. They are a document that blends computations, outputs, explanatory text, mathematics, images, and rich media representations of objects.

JupyterLab is one interface used to create and interact with Jupyter Notebooks.

For an overview of Jupyter Notebooks, see the **JupyterLab Welcome Tour** on this page, by going to `Help -> Notebook Tour` and following the prompts.

> **See Also**: For a more in-depth tour of Jupyter Notebooks and the Classic Jupyter Notebook interface, see the Jupyter Notebook IPython tutorial on Binder.

resource-conscious ✓

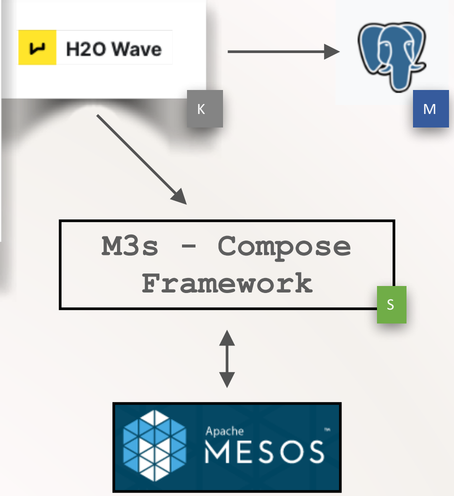


[redacted]

Type	Name	UI	Info	Started
Spark	/spark/[redacted]	not running		
Jupyterlab	/jupyterlab/[redacted]	not running		

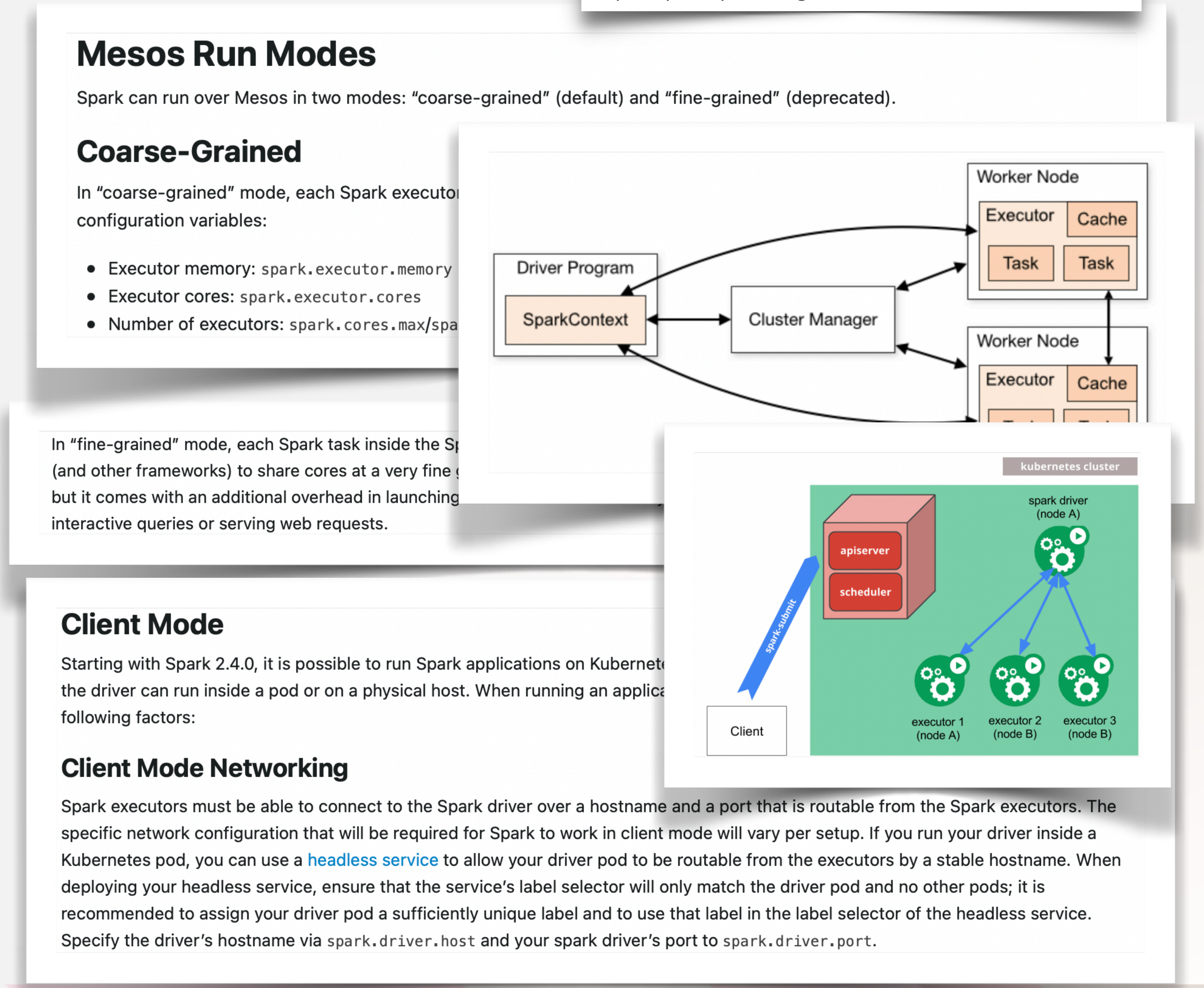
Configure /jupyterlab/[redacted]

Jupyterlab Memory MB: 10000



Apache Spark

https://spark.apache.org/docs/latest/index.html



Mesos Run Modes

Spark can run over Mesos in two modes: “coarse-grained” (default) and “fine-grained” (deprecated).

Coarse-Grained

In “coarse-grained” mode, each Spark executo
configuration variables:

- Executor memory: `spark.executor.memory`
- Executor cores: `spark.executor.cores`
- Number of executors: `spark.cores.max/spa`

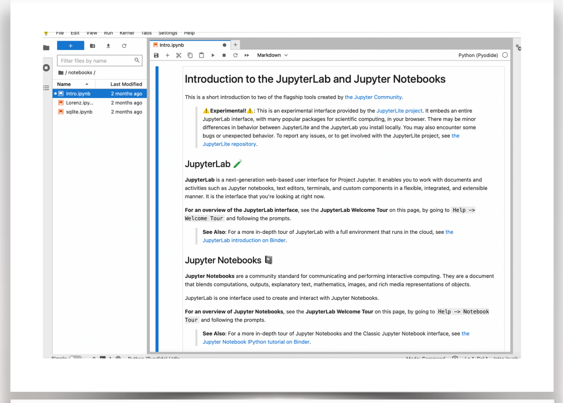
In “fine-grained” mode, each Spark task inside the Sp
(and other frameworks) to share cores at a very fine g
but it comes with an additional overhead in launching
interactive queries or serving web requests.

Client Mode

Starting with Spark 2.4.0, it is possible to run Spark applications on Kubernete
the driver can run inside a pod or on a physical host. When running an applica
following factors:

Client Mode Networking

Spark executors must be able to connect to the Spark driver over a hostname and a port that is routable from the Spark executors. The specific network configuration that will be required for Spark to work in client mode will vary per setup. If you run your driver inside a Kubernetes pod, you can use a headless service to allow your driver pod to be routable from the executors by a stable hostname. When deploying your headless service, ensure that the service’s label selector will only match the driver pod and no other pods; it is recommended to assign your driver pod a sufficiently unique label and to use that label in the label selector of the headless service. Specify the driver’s hostname via `spark.driver.host` and your spark driver’s port to `spark.driver.port`.



Name	Last Modified
Intro.ipynb	2 months ago
Lorenz.ipy...	2 months ago
sqlite.ipynb	2 months ago

Introduction to the JupyterLab and Jupyter Notebooks

This is a short introduction to two of the flagship tools created by the Jupyter Community.

> ⚠️ **Experimental!** ⚠️: This is an experimental interface provided by the JupyterLite project. It embeds an entire JupyterLab interface, with many popular packages for scientific computing, in your browser. There may be minor differences in behavior between JupyterLite and the JupyterLab you install locally. You may also encounter some bugs or unexpected behavior. To report any issues, or to get involved with the JupyterLite project, see the JupyterLite repository.

JupyterLab 🧪

JupyterLab is a next-generation web-based user interface for Project Jupyter. It enables you to work with documents and activities such as Jupyter notebooks, text editors, terminals, and custom components in a flexible, integrated, and extensible manner. It is the interface that you're looking at right now.

For an overview of the JupyterLab interface, see the **JupyterLab Welcome Tour** on this page, by going to `Help -> Welcome Tour` and following the prompts.

> **See Also**: For a more in-depth tour of JupyterLab with a full environment that runs in the cloud, see the JupyterLab introduction on Binder.

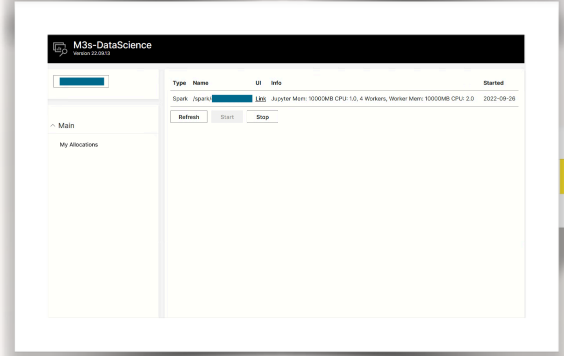
Jupyter Notebooks 📓

Jupyter Notebooks are a community standard for communicating and performing interactive computing. They are a document that blends computations, outputs, explanatory text, mathematics, images, and rich media representations of objects.

JupyterLab is one interface used to create and interact with Jupyter Notebooks.

For an overview of Jupyter Notebooks, see the **JupyterLab Welcome Tour** on this page, by going to `Help -> Notebook Tour` and following the prompts.

> **See Also**: For a more in-depth tour of Jupyter Notebooks and the Classic Jupyter Notebook interface, see the Jupyter Notebook IPython tutorial on Binder.

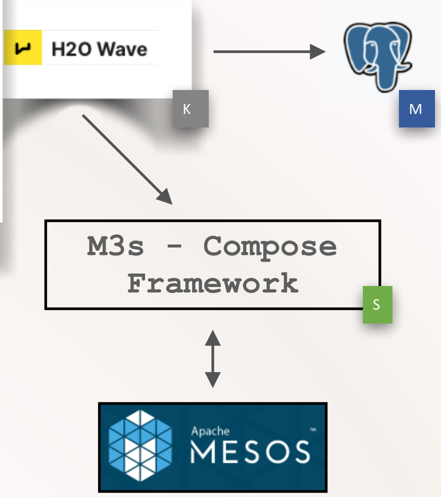


M3s-DataScience
Version 22.09.13

[redacted]

Type	Name	UI	Info	Started
Spark	/spark/[redacted]	Link	Jupyter Mem: 10000MB CPU: 1.0, 4 Workers, Worker Mem: 10000MB CPU: 2.0	2022-09-26

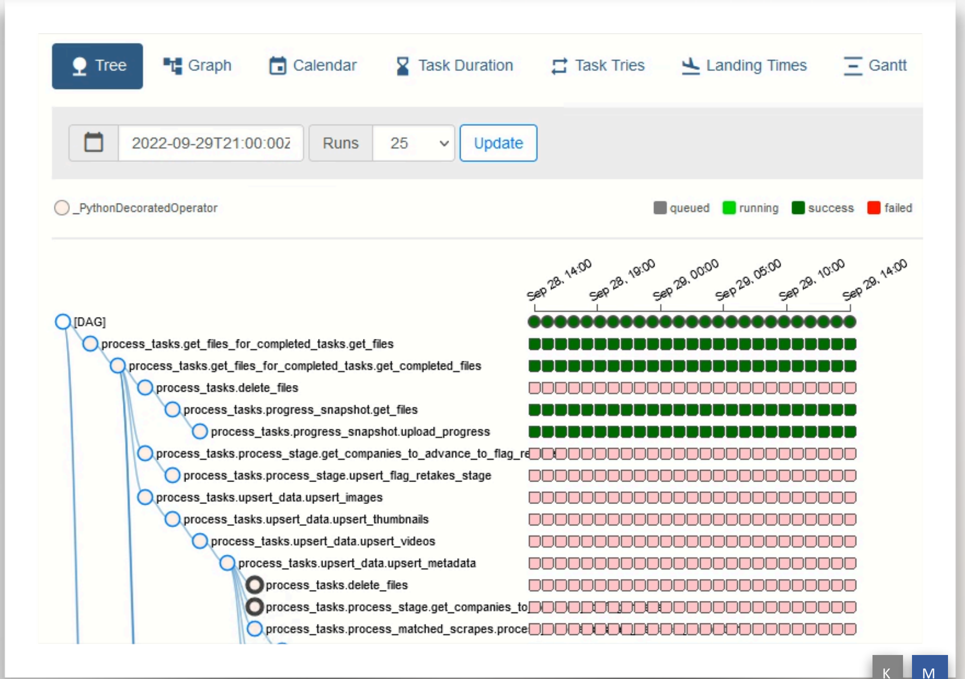
resource-conscious ✓



Apachecon 2022 - Marcel Neuhausler - Institutional Shareholder Services

#BrilliantTogether

Pipeline - Apache Airflow, Apache NiFi



Tree | Graph | Calendar | Task Duration | Task Tries | Landing Times | Gantt

2022-09-29T21:00:002 | Runs 25 | Update

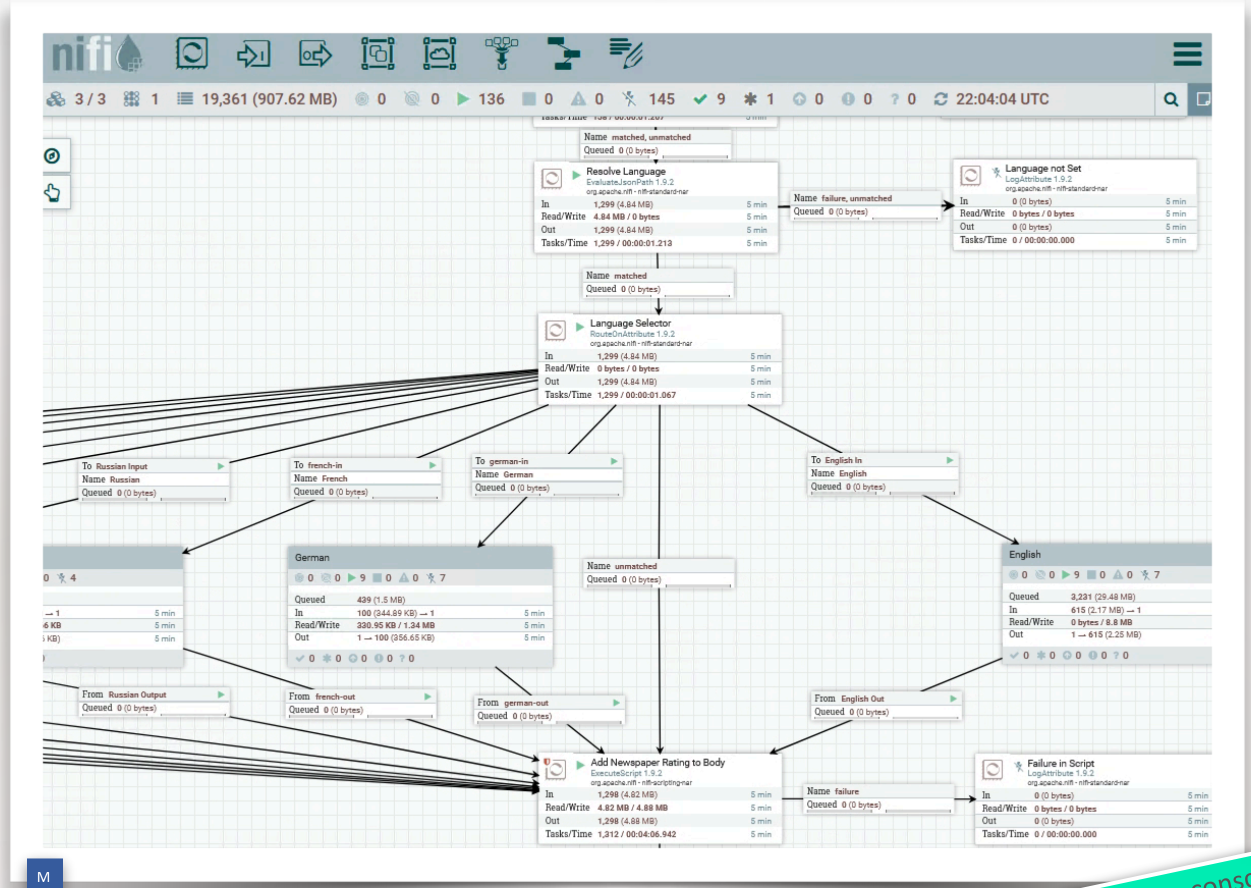
○ _PythonDecoratedOperator

■ queued ■ running ■ success ■ failed

```
[DAG]
process_tasks.get_files_for_completed_tasks.get_files
  process_tasks.get_files_for_completed_tasks.get_completed_files
    process_tasks.delete_files
      process_tasks.progress_snapshot.get_files
        process_tasks.progress_snapshot.upload_progress
    process_tasks.process_stage.get_companies_to_advance_to_flag_re
      process_tasks.process_stage.upsert_flag_retakes_stage
    process_tasks.upsert_data.upsert_images
      process_tasks.upsert_data.upsert_thumbnails
        process_tasks.upsert_data.upsert_videos
          process_tasks.upsert_data.upsert_metadata
            process_tasks.delete_files
            process_tasks.process_stage.get_companies_to
            process_tasks.process_matched_scrapes.proce
```

Helm Chart for Apache Airflow

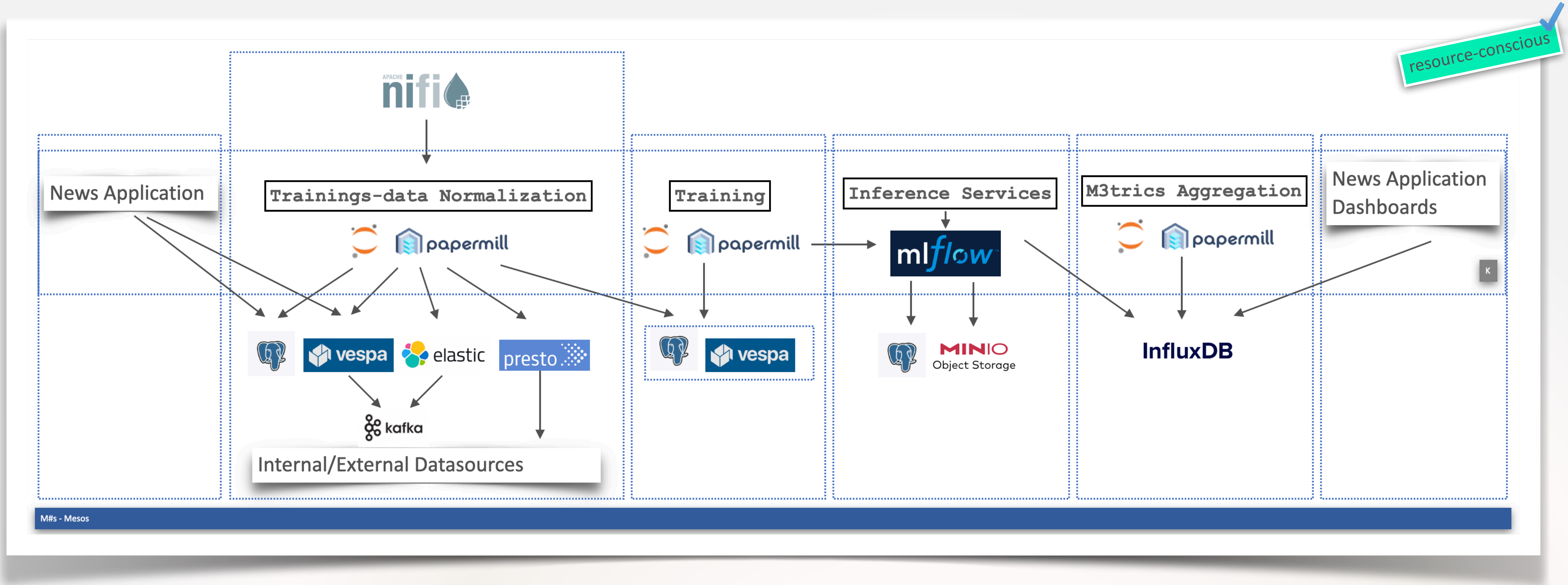
https://nifi.apache.org/
https://airflow.apache.org/



```
3 / 3   1   19,361 (907.62 MB)   0   0   136   0   0   145   9   1   0   0   0   22:04:04 UTC

Name  matched, unmatched
Queued  0 (0 bytes)

Resolve Language
EvaluateJsonPath 1.9.2
org.apache.nifi - nifi-standard-nar
In          1,299 (4.84 MB)          5 min
Read/Write  4.84 MB / 0 bytes        5 min
Out         1,299 (4.84 MB)          5 min
Tasks/Time  1,299 / 00:00:01.213     5 min

Name  failure, unmatched
Queued  0 (0 bytes)

Language not Set
LogAttribute 1.9.2
org.apache.nifi - nifi-standard-nar
In          0 (0 bytes)              5 min
Read/Write  0 bytes / 0 bytes        5 min
Out         0 (0 bytes)              5 min
Tasks/Time  0 / 00:00:00.000         5 min

Name  matched
Queued  0 (0 bytes)

Language Selector
RouteOnAttribute 1.9.2
org.apache.nifi - nifi-standard-nar
In          1,299 (4.84 MB)          5 min
Read/Write  0 bytes / 0 bytes        5 min
Out         1,299 (4.84 MB)          5 min
Tasks/Time  1,299 / 00:00:01.067     5 min

To  Russian Input      To  french-in      To  german-in      To  English In
Name  Russian          Name  French       Name  German       Name  English
Queued  0 (0 bytes)    Queued  0 (0 bytes) Queued  0 (0 bytes) Queued  0 (0 bytes)

German
0  0  9  0  0  7
Queued      439 (1.5 MB)
In          100 (344.89 KB) → 1          5 min
Read/Write  330.95 KB / 1.34 MB          5 min
Out         1 → 100 (356.65 KB)          5 min
0  0  0  0  0

Name  unmatched
Queued  0 (0 bytes)

English
0  0  9  0  0  7
Queued      3,231 (29.48 MB)
In          615 (2.17 MB) → 1
Read/Write  0 bytes / 8.8 MB
Out         1 → 615 (2.25 MB)
0  0  0  0  0

From  Russian Output   From  french-out   From  german-out   From  English Out
Queued  0 (0 bytes)    Queued  0 (0 bytes) Queued  0 (0 bytes) Queued  0 (0 bytes)

Add Newspaper Rating to Body
ExecuteScript 1.9.2
org.apache.nifi - nifi-scripting-nar
In          1,298 (4.82 MB)          5 min
Read/Write  4.82 MB / 4.88 MB        5 min
Out         1,298 (4.88 MB)          5 min
Tasks/Time  1,312 / 00:04:06.942     5 min

Name  failure
Queued  0 (0 bytes)

Failure in Script
LogAttribute 1.9.2
org.apache.nifi - nifi-standard-nar
In          0 (0 bytes)              5 min
Read/Write  0 bytes / 0 bytes        5 min
Out         0 (0 bytes)              5 min
Tasks/Time  0 / 00:00:00.000         5 min
```

resource-conscious

ML Platform, MLOps - Example News Processing



https://nifi.apache.org/
https://github.com/nteract/papermill

https://vespa.ai/
https://www.elastic.co/elasticsearch/

https://prestodb.io/
https://mlflow.org/

https://www.influxdata.com/
https://jupyter.org/

https://www.postgresql.org/
https://kafka.apache.org/

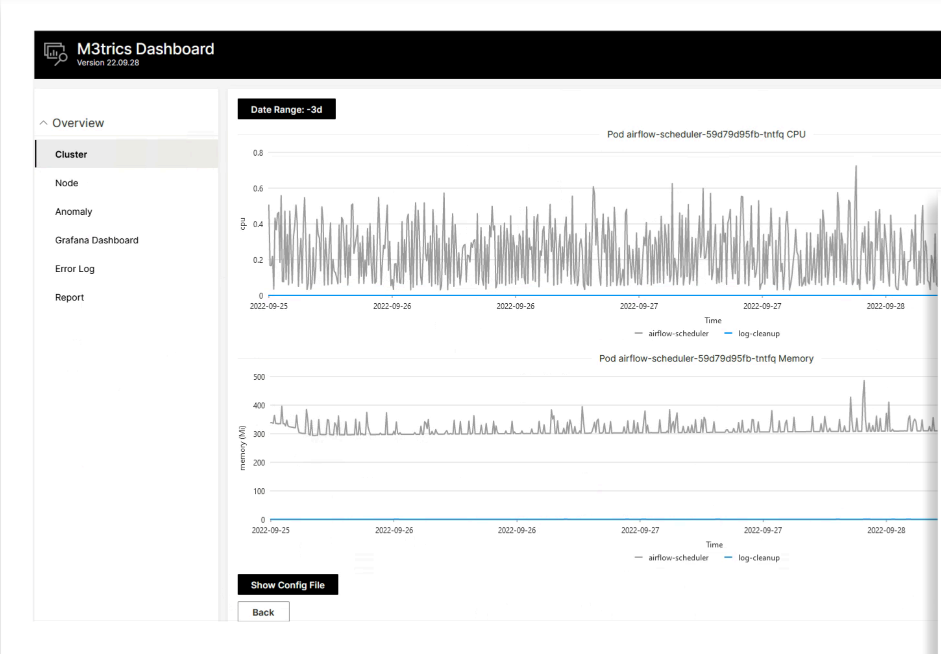
M3trics

H2O Wave resource-conscious ✓

Grafana Labs resource-conscious ✓

InfluxDB resource-conscious ✓

Prometheus



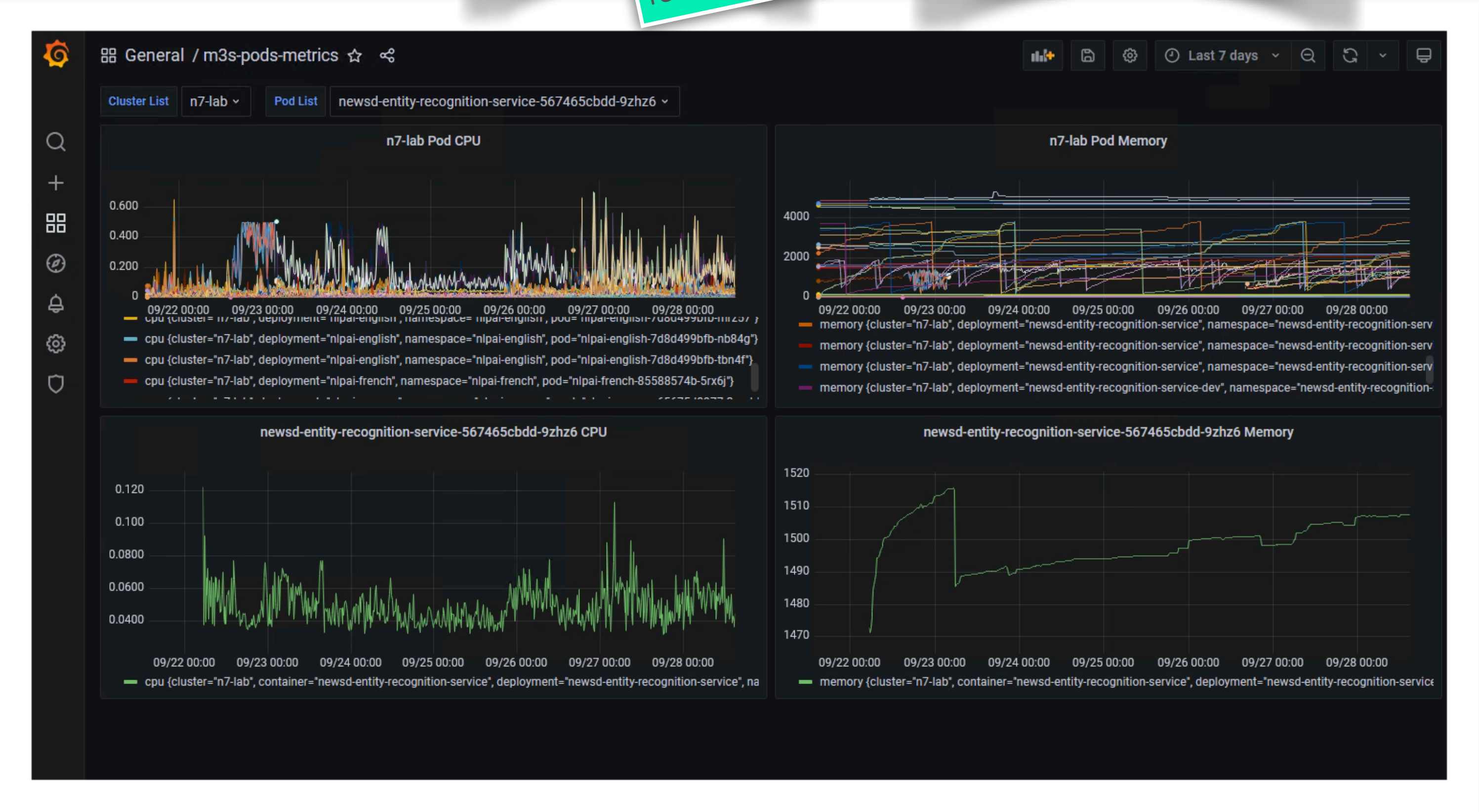
M3trics Dashboard
Version 22.09.28

Overview
- Cluster
- Node
- Anomaly
- Grafana Dashboard
- Error Log
- Report

Date Range: -3d

Show Config File

Back



General / m3s-pods-metrics

Last 7 days

Cluster List: n7-lab

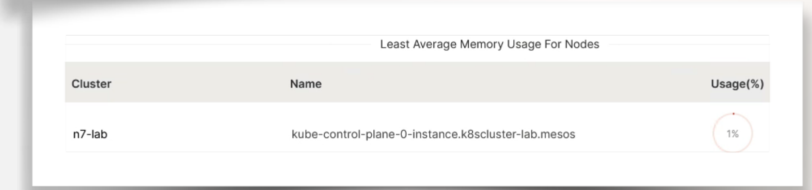
Pod List: newsd-entity-recognition-service-567465cbdd-9zhz6

- cpu {cluster="n7-lab", deployment="nlpai-english", namespace="nlpai-english", pod="nlpai-english-7d8d499bfb-nb84g"}
- cpu {cluster="n7-lab", deployment="nlpai-english", namespace="nlpai-english", pod="nlpai-english-7d8d499bfb-tbn4f"}
- cpu {cluster="n7-lab", deployment="nlpai-french", namespace="nlpai-french", pod="nlpai-french-85588574b-5rx6j"}

- memory {cluster="n7-lab", deployment="newsd-entity-recognition-service", namespace="newsd-entity-recognition-serv
- memory {cluster="n7-lab", deployment="newsd-entity-recognition-service", namespace="newsd-entity-recognition-serv
- memory {cluster="n7-lab", deployment="newsd-entity-recognition-service", namespace="newsd-entity-recognition-serv
- memory {cluster="n7-lab", deployment="newsd-entity-recognition-service-dev", namespace="newsd-entity-recognition-

- cpu {cluster="n7-lab", container="newsd-entity-recognition-service", deployment="newsd-entity-recognition-service", na

- memory {cluster="n7-lab", container="newsd-entity-recognition-service", deployment="newsd-entity-recognition-service



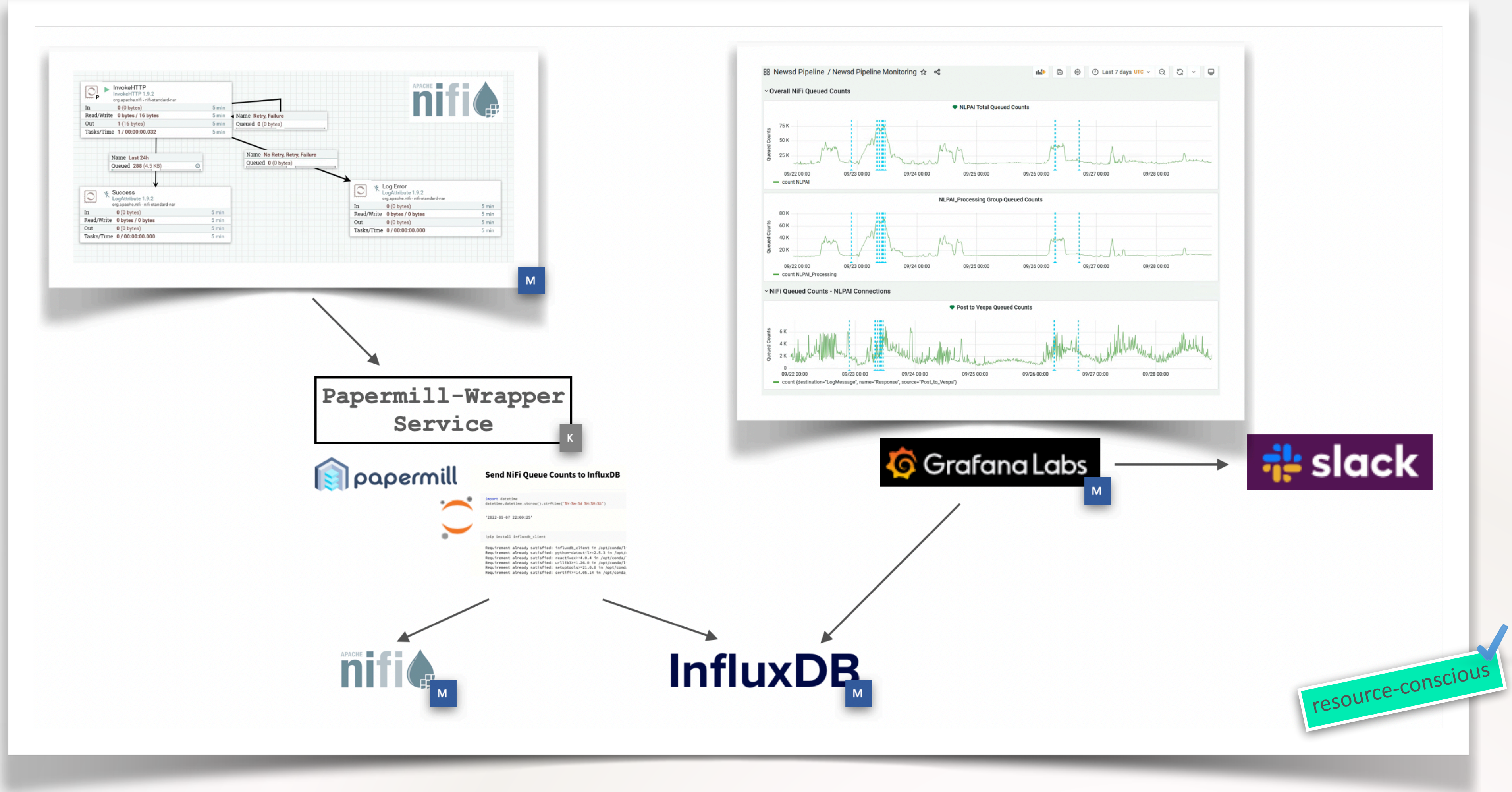
Least Average Memory Usage For Nodes

Cluster	Name	Usage(%)
n7-lab	kube-control-plane-0-instance.k8scluster-lab.mesos	1%

https://wave.h2o.ai/
https://grafana.com/

https://prometheus.io/
https://www.influxdata.com/

M3trics - Example Metrics Collection



InvokeHTTP
InvokeHTTP 1.9.2
org.apache.nifi - nifi-standard-nar
In 0 (0 bytes) 5 min
Read/Write 0 bytes / 16 bytes 5 min
Out 1 (16 bytes) 5 min
Tasks/Time 1 / 00:00:00.032 5 min

Name Retry, Failure
Queued 0 (0 bytes)

Name Last 24h
Queued 288 (4.5 KB)

Name No Retry, Retry, Failure
Queued 0 (0 bytes)

Success
LogAttribute 1.9.2
org.apache.nifi - nifi-standard-nar
In 0 (0 bytes) 5 min
Read/Write 0 bytes / 0 bytes 5 min
Out 0 (0 bytes) 5 min
Tasks/Time 0 / 00:00:00.000 5 min

Log Error
LogAttribute 1.9.2
org.apache.nifi - nifi-standard-nar
In 0 (0 bytes) 5 min
Read/Write 0 bytes / 0 bytes 5 min
Out 0 (0 bytes) 5 min
Tasks/Time 0 / 00:00:00.000 5 min

Send NiFi Queue Counts to InfluxDB

```
import datetime
datetime.datetime.utcnow().strftime('%Y-%m-%d %H:%M:%S')
```

```
'2022-09-07 22:00:25'
```

```
!pip install influxdb_client
```

```
Requirement already satisfied: influxdb_client in /opt/conda/l
Requirement already satisfied: python-dateutil>=2.5.3 in /opt/
Requirement already satisfied: reactivex>=4.0.4 in /opt/conda/
Requirement already satisfied: urllib3>=1.26.0 in /opt/conda/l
Requirement already satisfied: setuptools>=21.0.0 in /opt/cond
Requirement already satisfied: certifi>=14.05.14 in /opt/conda
```

Newsd Pipeline / Newsd Pipeline Monitoring

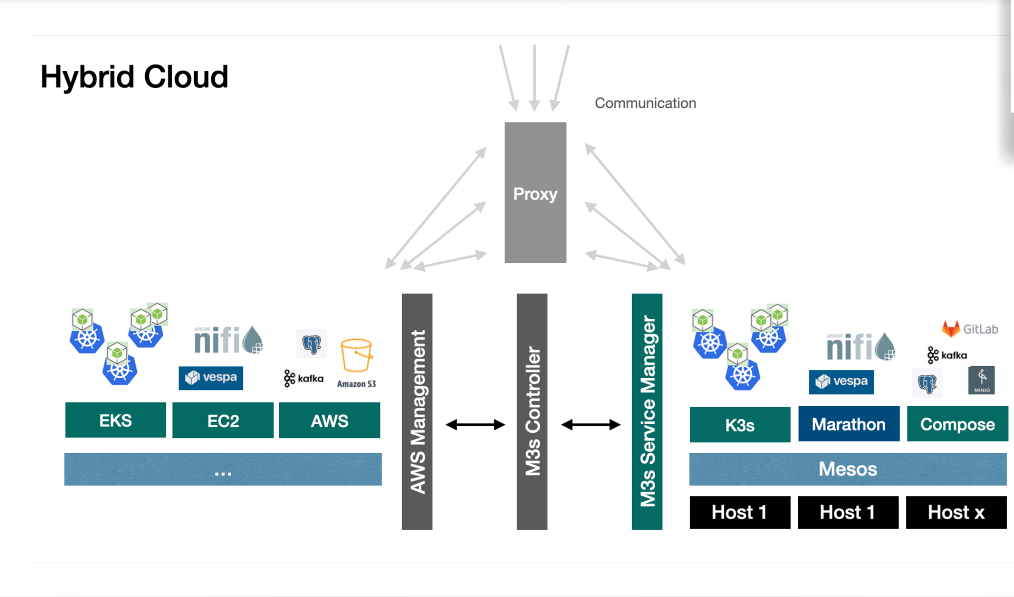
Last 7 days UTC

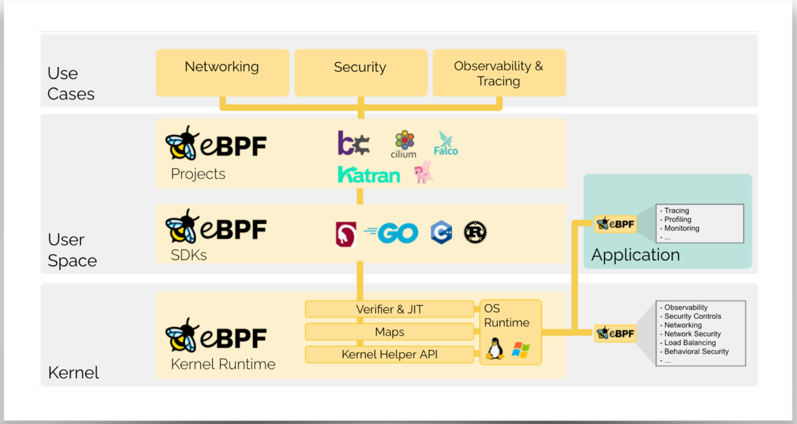
APACHE nifi
M
Papermill-Wrapper Service
K
papermill
Overall NiFi Queued Counts
NLPAI Total Queued Counts
Queued Counts
75 K
50 K
25 K
09/22 00:00
09/23 00:00
09/24 00:00
09/25 00:00
09/26 00:00
09/27 00:00
09/28 00:00
count NLPAI
NLPAI_Processing Group Queued Counts
80 K
60 K
40 K
20 K
count NLPAI_Processing
NiFi Queued Counts - NLPAI Connections
Post to Vespa Queued Counts
6 K
4 K
2 K
0
count (destination="LogMessage", name="Response", source="Post_to_Vespa")
Grafana Labs
M
slack
APACHE nifi
M
InfluxDB
M
resource-conscious

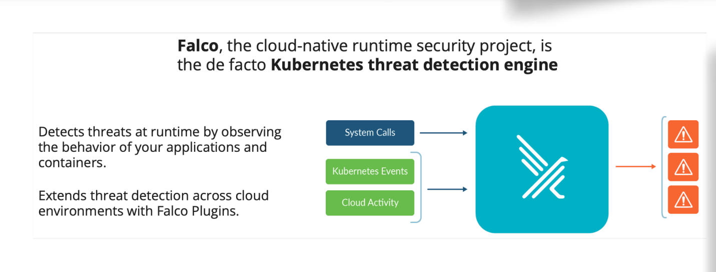
https://github.com/nteract/papermill
https://slack.com/

M3s Future

Data-Science / ML applied to Infrastructure





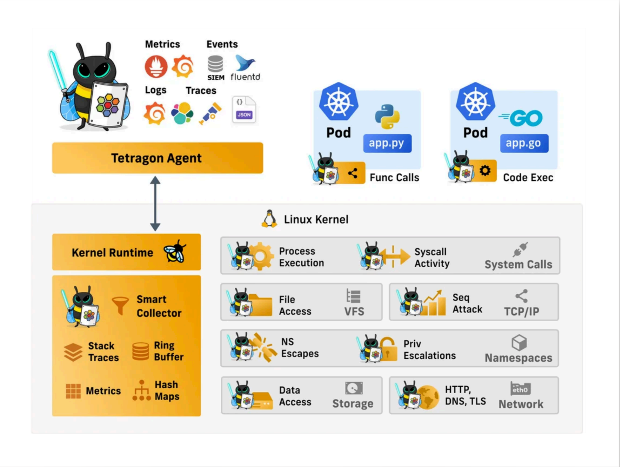


Falco, the cloud-native runtime security project, is the de facto **Kubernetes threat detection engine**

Detects threats at runtime by observing the behavior of your applications and containers.

Extends threat detection across cloud environments with Falco Plugins.

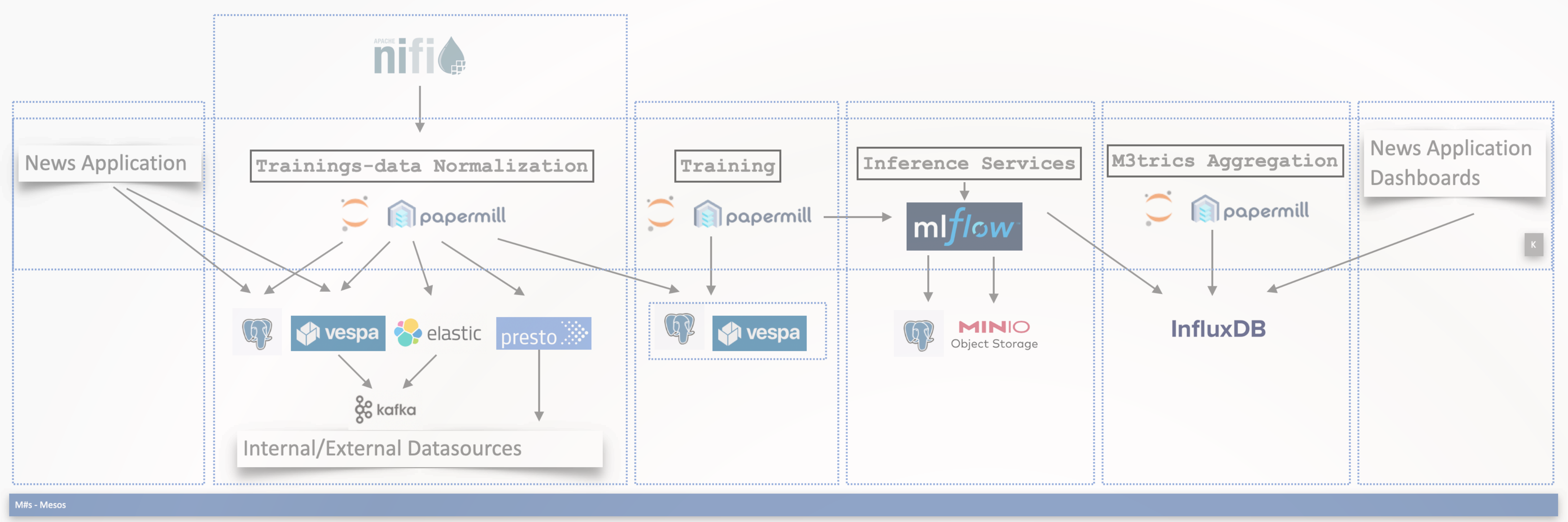
System Calls
Kubernetes Events
Cloud Activity



https://mesos.apache.org/
https://ebpf.io/
https://github.com/cilium/tetragon
https://falco.org/
https://firecracker-microvm.github.io/
https://webassembly.org/

Summary

- ML to Production ✓.. it was an interesting journey .. and there is more to come.
- Brilliant SREs and Data Infrastructure Engineers can build exciting tools and platforms.
- “Data Inspired Continuous Exploration” - Innovation enabled by Apache Mesos, and many other resource-conscious tools and applications. Thanks to all the Committers and Contributors!


APACHE nifi
News Application
Trainings-data Normalization
papermill
Training
papermill
Inference Services
mlflow
M3trics Aggregation
papermill
News Application Dashboards
vespa
elastic
presto
vespa
MINIO Object Storage
InfluxDB
kafka
Internal/External Datasources
M#s - Mesos