
# Karvdash: A complete software stack for performing data science on Kubernetes

Antony Chazapis chazapis_at_ics.forth.gr

FOSSCOMM 2021, 13-14/11/2021

# The problem

**Large scale computing is extremely difficult for people who actually need it**

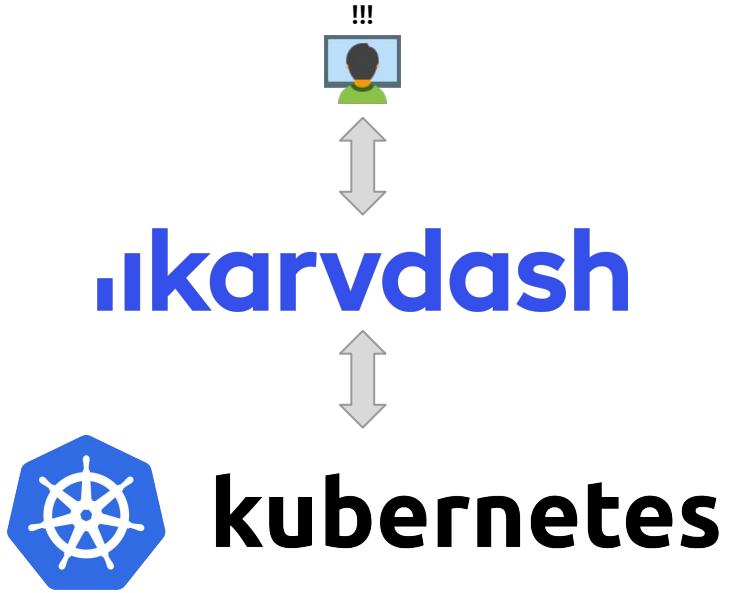
- Because of complexity
  - Time wasted in devops
  - Time wasted in rewritting the application for a specific environment
  - Steep learning curve
  - Vendor lock-in
- Because it's expensive
  - It is difficult to estimate requirements
  - It is difficult to estimate cloud-provider costs


**Scaling out to the cloud?**

# The solution

**Provide a human interface to do actual work that is agnostic of infrastructure**

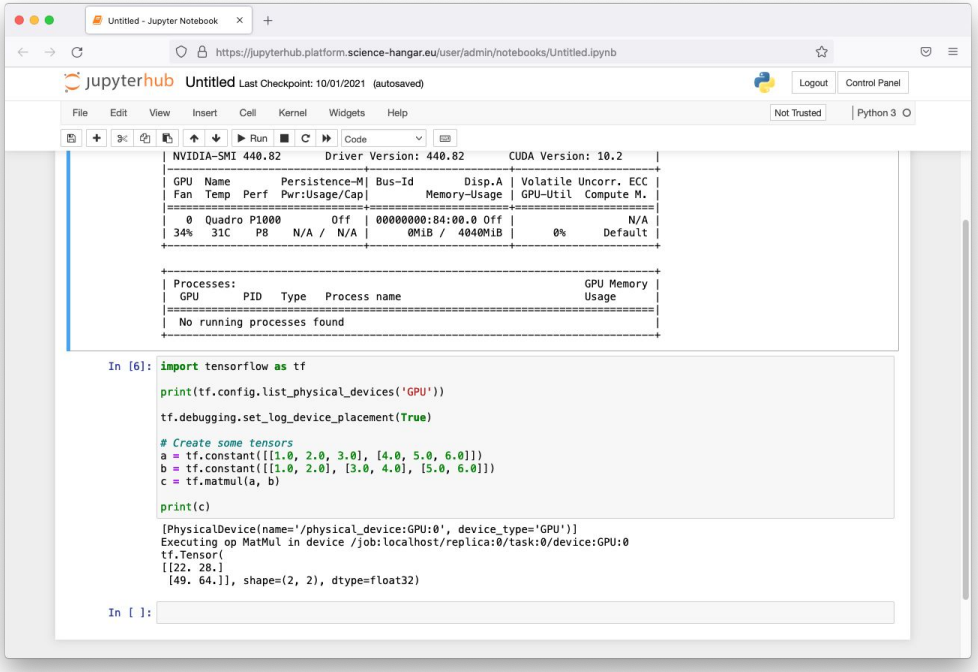
- Hide the complexity
  - No devops
  - Minimal rewritting of the application, mostly packaging in containers
  - Well-known and standard tools, like notebooks and workflows to streamline the process
- Make it affordable
  - Run on any infrastructure (small or large scale: laptop or Cloud)
  - Autoscale to match requirements




**Karvdash provides a user-friendly environment on top of Kubernetes**

# Notebook interface

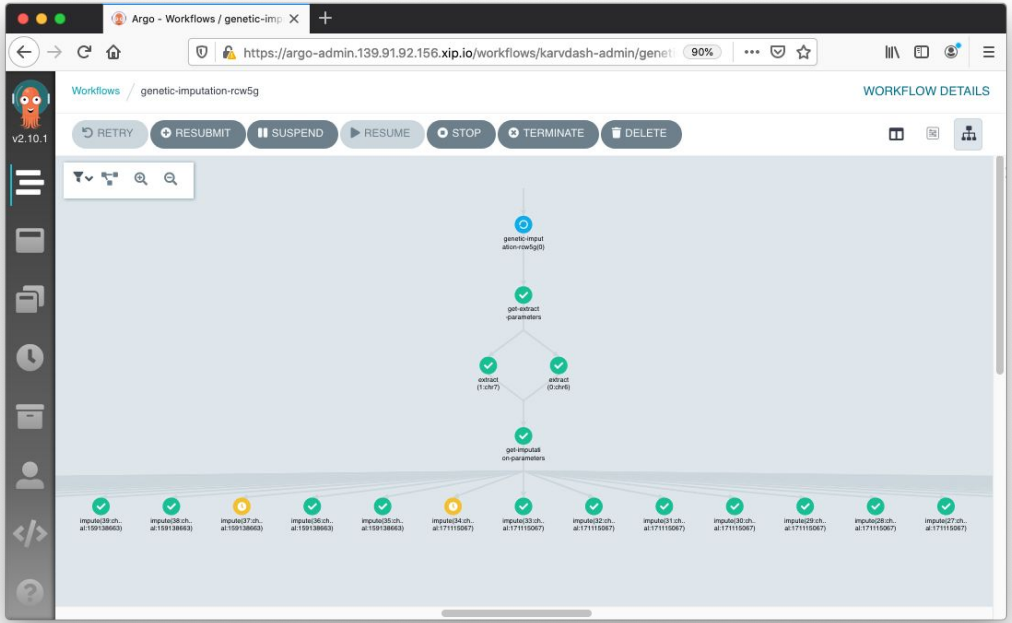
- Combine paragraphs with different execution environments (shell, Python, etc.)
- Describe the configuration and write the code in the same context
- Portable, redistributable format
- Save the notebook with the data outputs for data provenance




**Notebook example**

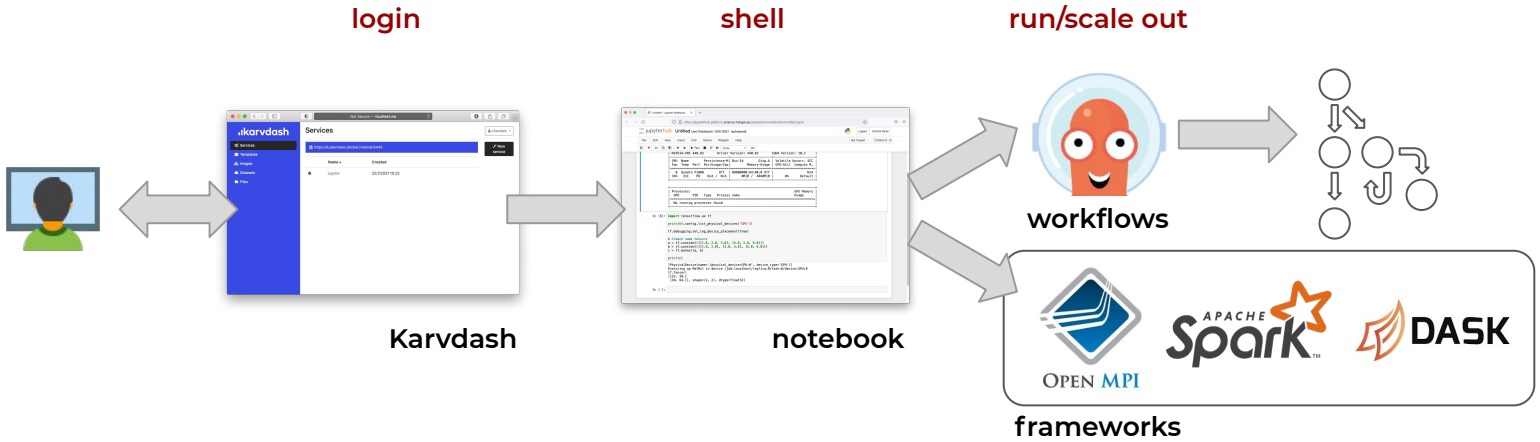
# Workflows

- Describe execution as a DAG
- Extended features, such as loops, conditionals, recursion, etc.
- Each step is encapsulated in a container
- Scale out to all Kubernetes nodes
- Web-based interface
- Notebook integration




**A running workflow with multiple parallel steps**

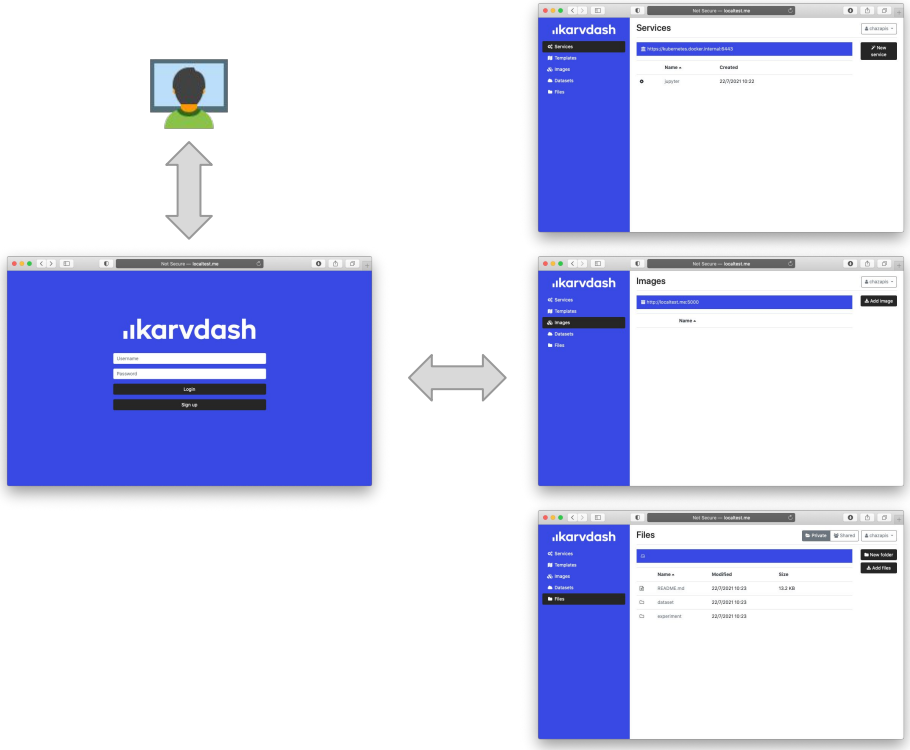
# Overall layout and usage workflow




**Clear separation of hardware infrastructure and software stack, allows to scale the hardware independently of the software**

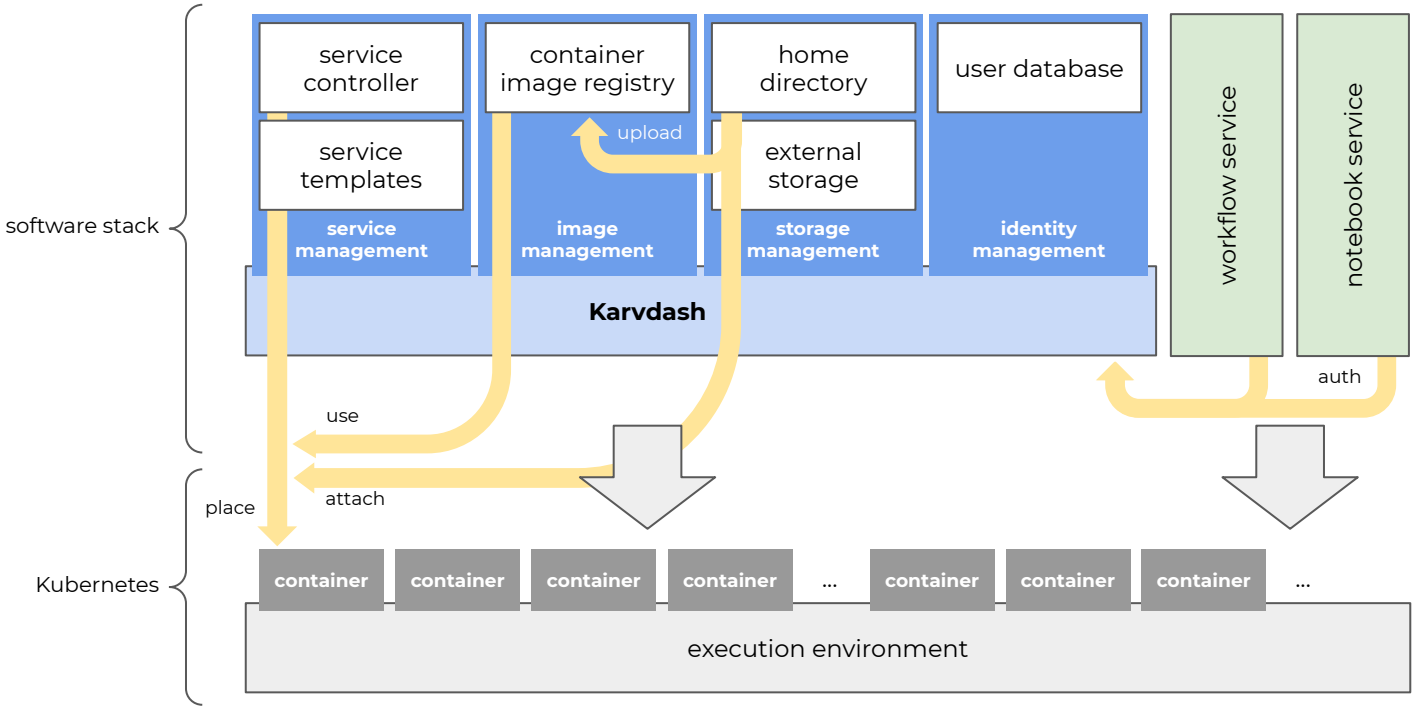
# Karvdash features

- User management and a centralized login point
- Service management with customizable templates
- Private Docker registry interface
- File collections (automatically bound to all containers)
- External Cloud dataset connectivity (S3)
- Isolation of execution per user in separate namespaces
- External service provisioning over HTTPS
- User authentication for external services (OAuth 2.0/OIDC)
- API for integration with the notebook environment




**Karvdash provides users service, container, and file management features**

# Internal components



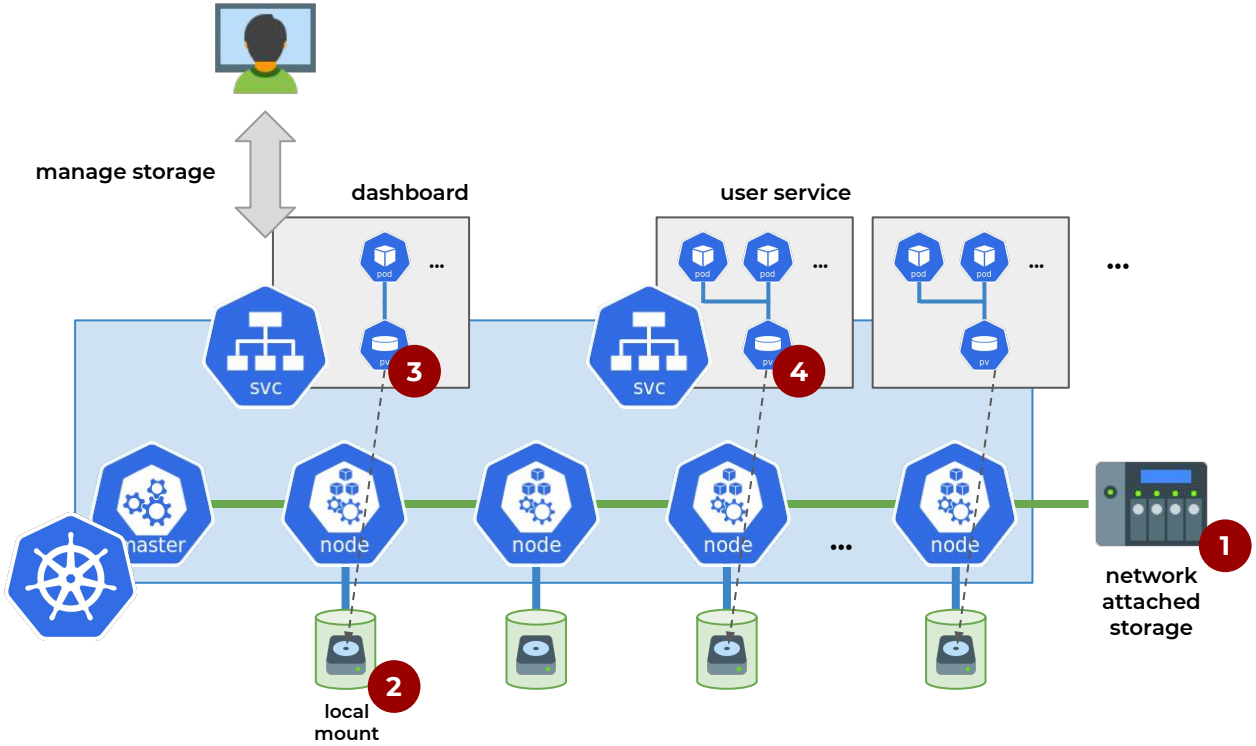
# Service templates

```
kind: Deployment
apiVersion: apps/v1
metadata:
  name: $NAME
spec:
  ...
  template:
   ...
    spec:
      containers:
      - name: $NAME
        image: postgres:12.3
        ports:
        - containerPort: 5432
        env:
        - name: POSTGRES_PASSWORD
          value: $PASSWORD
        - name: PGDATA
          value: ${PRIVATE_DIR}/.postgresql/data
```

```
kind: Template
name: PostgreSQL
description: Database server
singleton: yes
datasets: no
variables:
- name: NAME
  default: postgresql
- name: PRIVATE_DIR
  default: /private
- name: PASSWORD
  default: postgres
  help: Password for postgres user
```

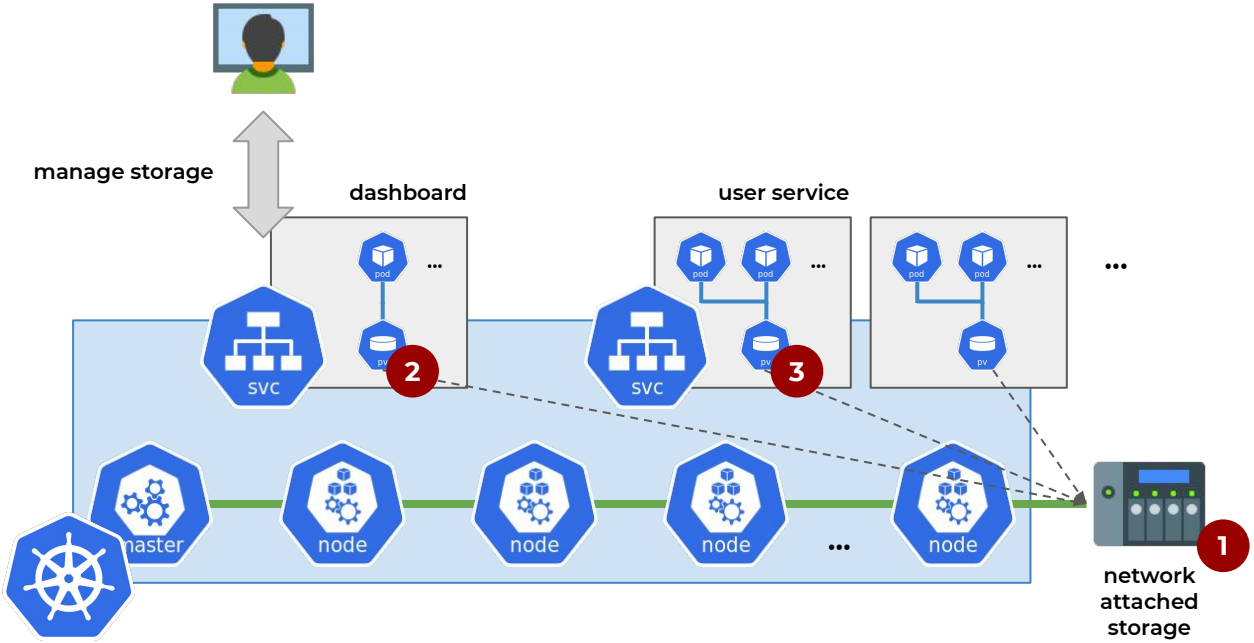
# Storage handling with local storage

1. network attached storage
2. local mount (all nodes)
3. mount in dashboard (using HostPath PV/PVC)
4. mount in containers (using per-user HostPath PV/PVC)



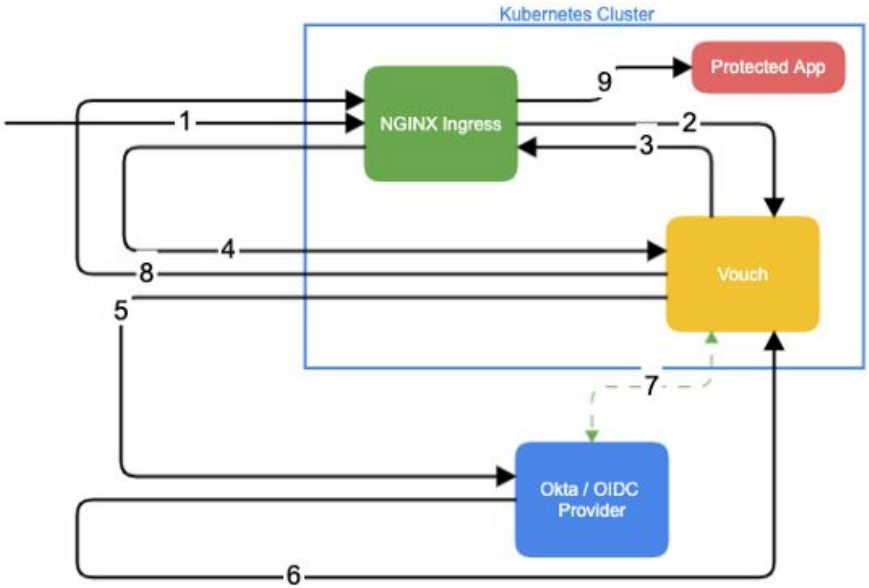
# Storage handling with NFS

1. network attached storage
2. mount in dashboard (using NFS PV/PVC)
3. mount in containers (using per-user NFS PV/PVC)



# Identity services

- Karvdash is an OAuth 2.0/OIDC-compatible identity provider
- We use Vouch identity proxy to authenticate users into their own services
- External services can use Karvdash for user authentication
- Karvdash is deployed side-by-side with Argo and JupyterHub




Source:
https://medium.com/@msvechla/attribute-based-access-control-with-oidc-and-nginx-ingress-controller-in-kubernetes-169481ff2f22

# Development

Visit: https://github.com/CARV-ICS-FORTH/karvdash

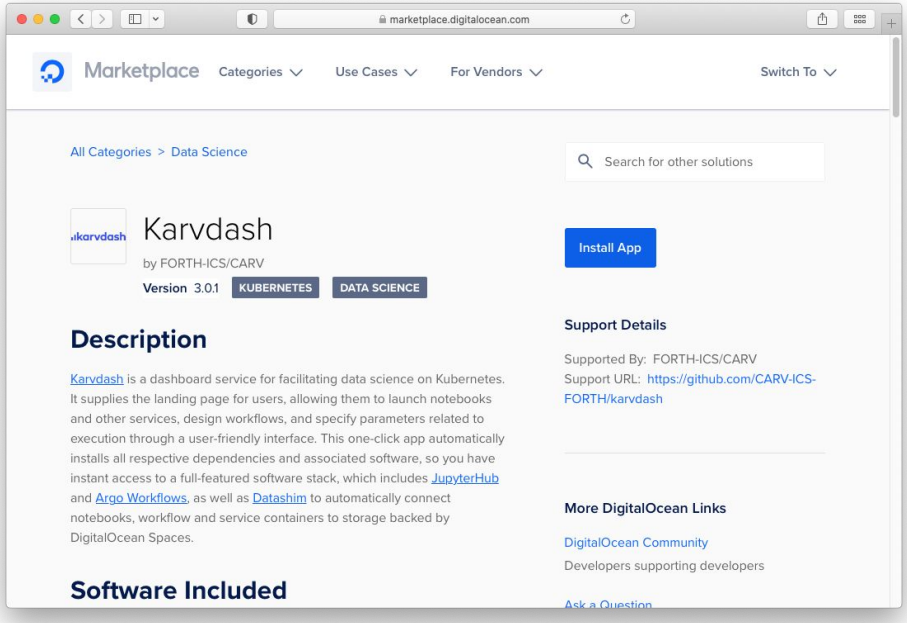
Some details:

- Written in Django (Python)
- Documentation available (user and technical)
- Easy local development
- Helm chart and images for both amd64 and arm64 available
- Automated packaging and release of docs, container images, Python package, and Helm chart


Check our other projects: https://github.com/CARV-ICS-FORTH

Funded as part of the EVOLVE H2020 project:
https://evolve-h2020.eu

# Easy evaluation

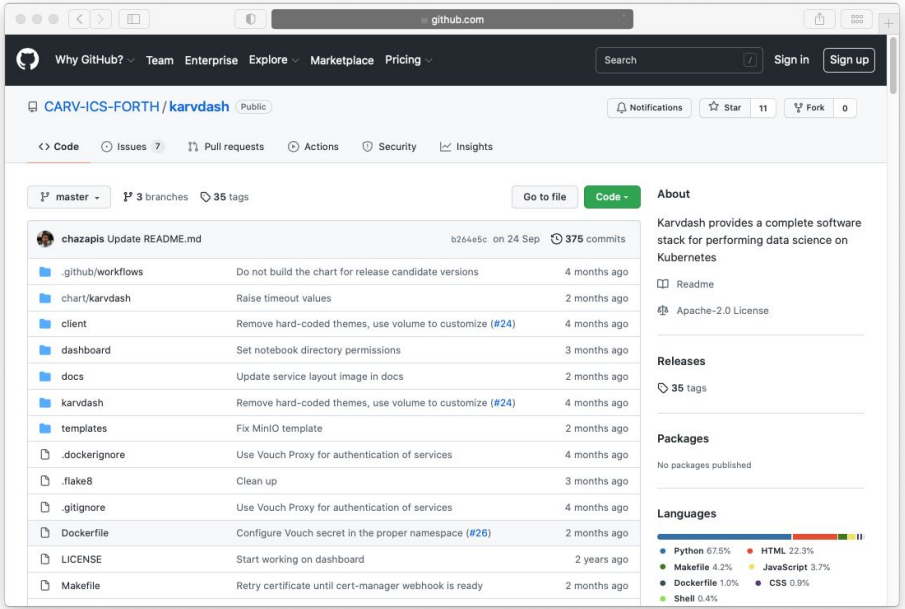



**Try it out in the Cloud!**

Full stack deployment available (with Argo and JupyterHub)

at: https://marketplace.digitalocean.com/apps/karvdash

# Thank you!




https://github.com/CARV-ICS-FORTH/karvdash