

# Austrian DataLAB and Services



## Contact and links:

Webpage: <https://webportal.dev.austrianopensciencecloud.org/>  
(will be continuously updated with training material)

Slack: <https://adls-workspace.slack.com/join/signup>

Email: [adls@uibk.ac.at](mailto:adls@uibk.ac.at)

LinkedIn: <https://linkedin.com/in/croedig>



Funding Agency: BMBWF + participating universities

Start: 06.2020

End: 12.2024

Who should benefit: all Austrian universities

Which parts are effected: research, teaching, ZIDs

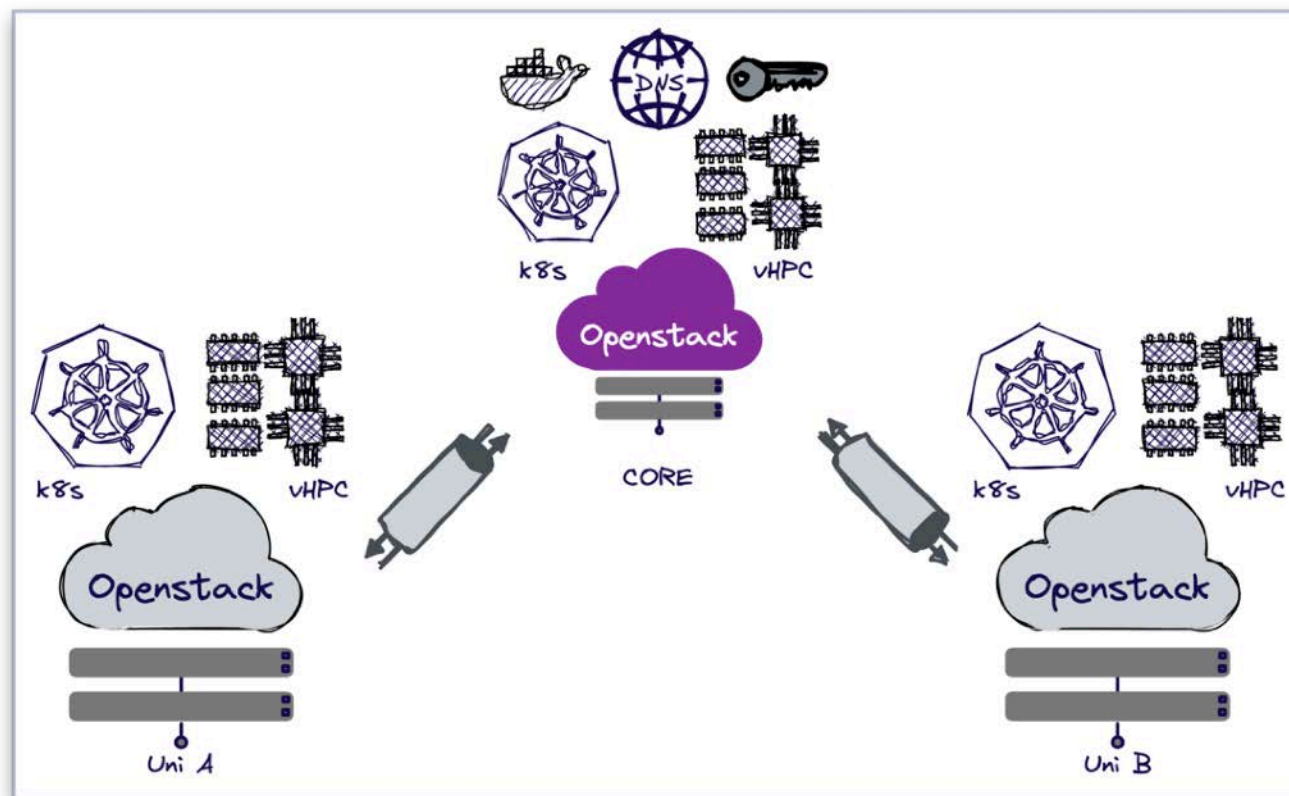
Complexity: high - on the technical side of things

PI: self organized scrum team (formal TU Wien)

Partners:



- inter-university collaboration on cloud infrastructure

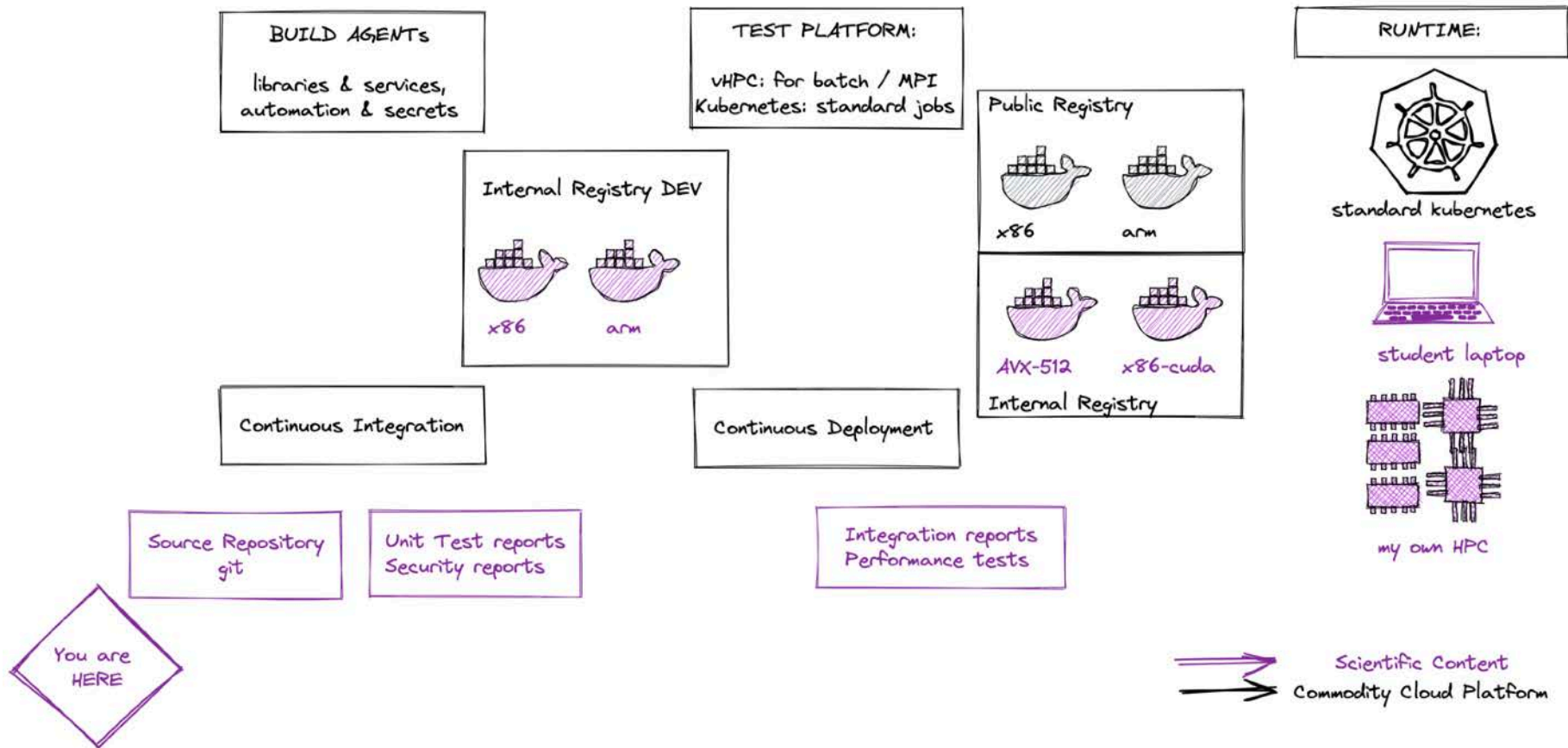


We look for answers to:

What technologies? How to connect them? How to make it secure?  
How to share data/users? How to include LMS? How to federate everything? etc.

- inter-university collaboration on cloud infrastructure
- lowering the learning curve for compute tasks in research and teaching

# Goal - learning curve

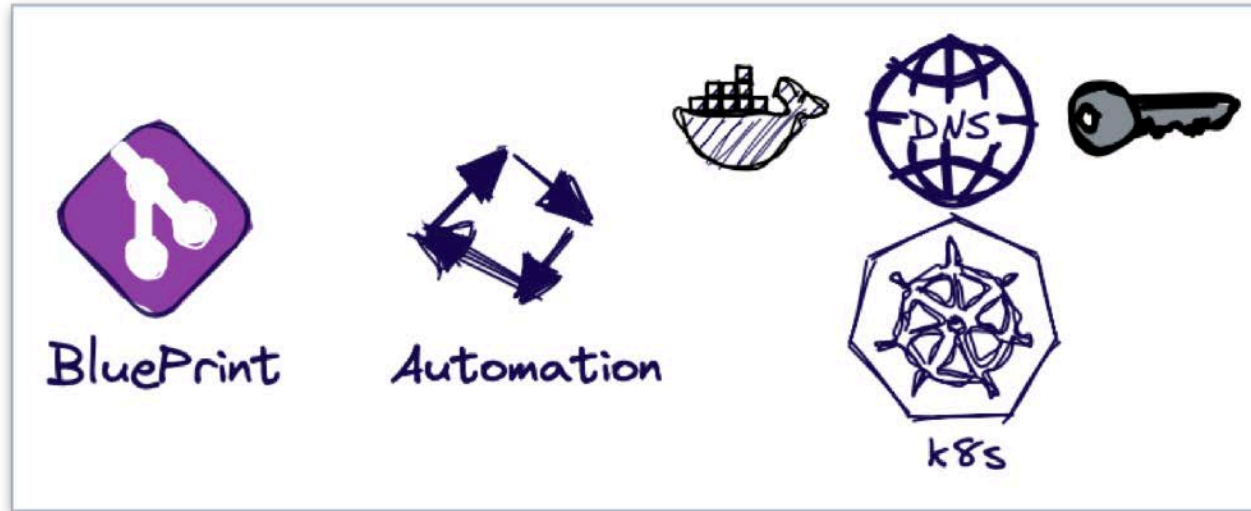


We start drawing a map:

hide the „this should just work“ stuff and provide: building manuals, blueprints, workshops, training, different access for different skillsets,...

- inter-university collaboration on cloud infrastructure
- lowering the learning curve for compute tasks in research and teaching
- secure exchange and collaboration on data, applications and results

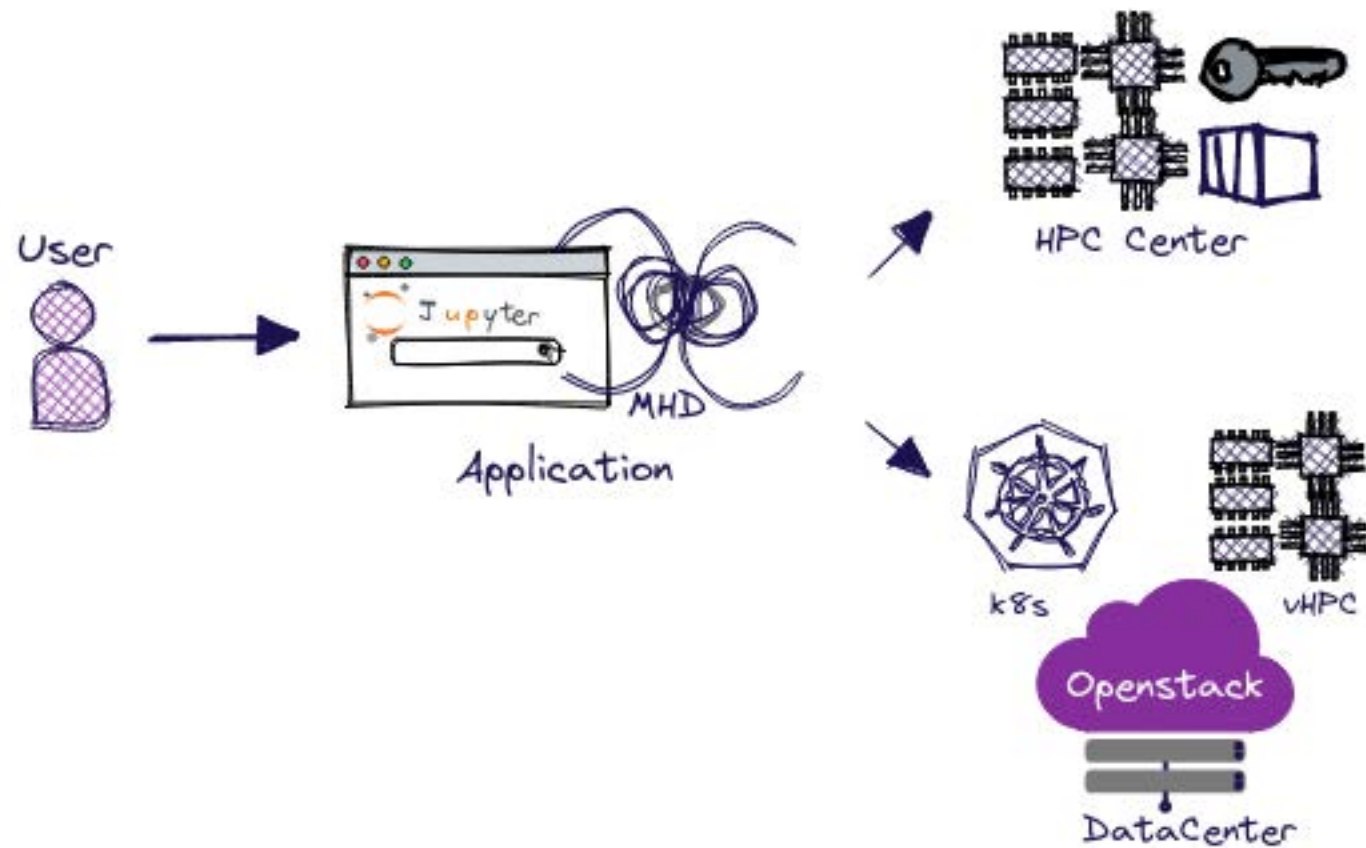




How can I share my program, data, algorithm?

Enable an interactive version of a research paper including data and algorithms, you could also allow people to run. Make sure your results survive.

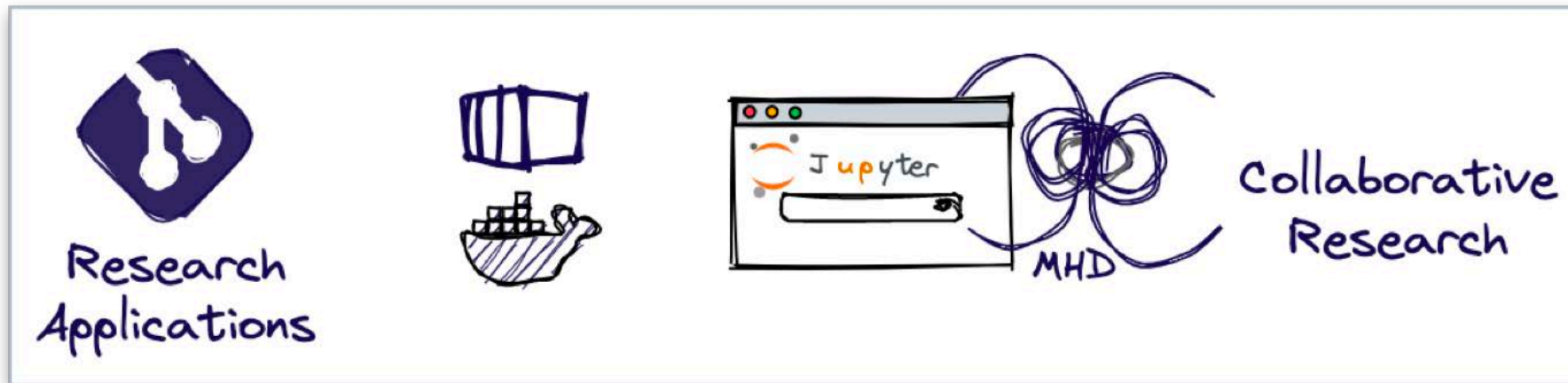
- inter-university collaboration on cloud infrastructure
- lowering the learning curve for compute tasks in research and teaching
- secure exchange and collaboration on data, applications and results
- interactive access to High Performance Computing Systems



## Create a virtual HPC and connect to real HPCs

test interfaces for browser based access, provide resources for interactive use in workflows, create a test environment (for admins), configurations,...

- inter-university collaboration on cloud infrastructure
- lowering the learning curve for compute tasks in research and teaching
- secure exchange and collaboration on data, applications and results
- interactive access to High Performance Computing Systems
- collaborative development and sharing at Austrian academic institutions

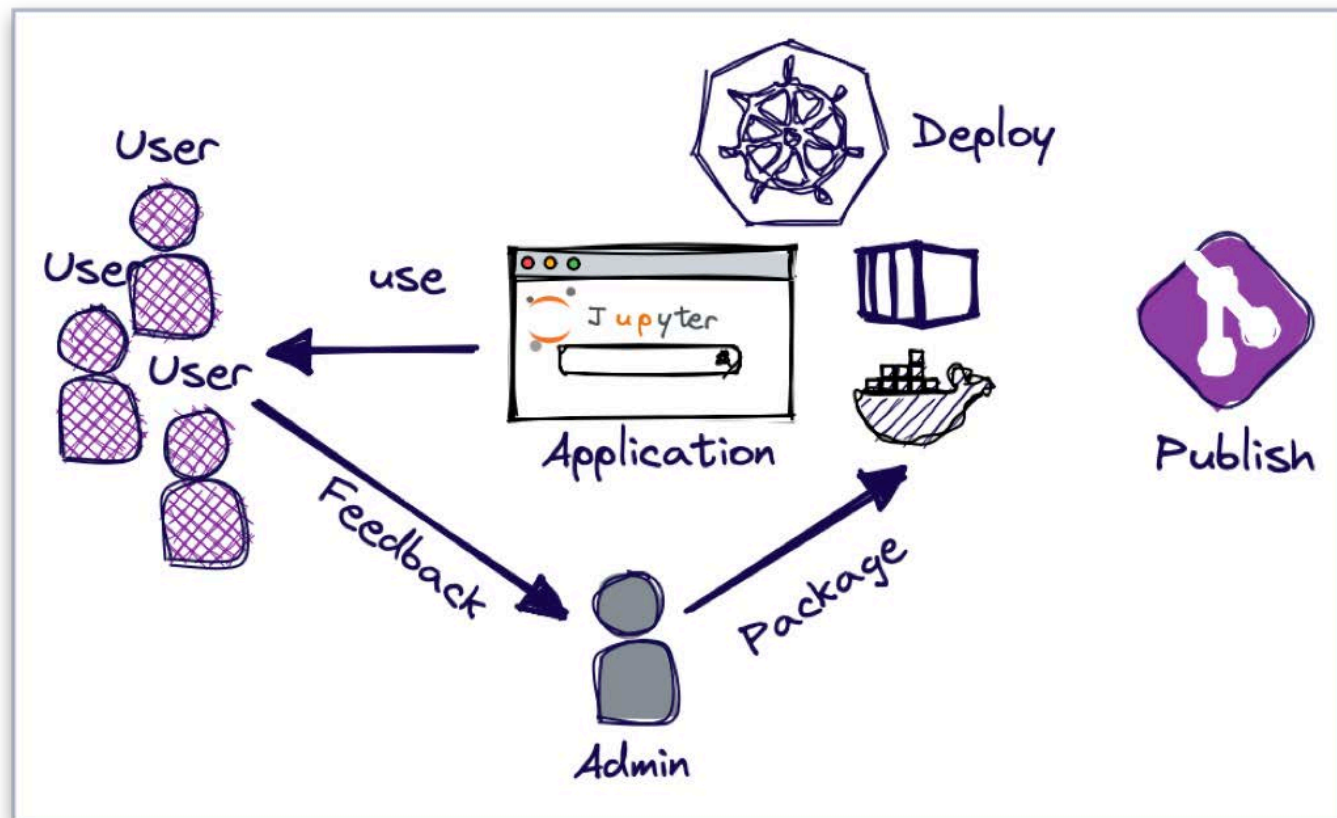


## Enable cooperative development in research

develop a common code base in your research community and gain visibility,  
benefit from a reproducible environment and interoperability

- inter-university collaboration on cloud infrastructure
- lowering the learning curve for compute tasks in research and teaching
- secure exchange and collaboration on data, applications and results
- interactive access to High Performance Computing Systems
- collaborative development and sharing at Austrian academic institutions
- Toolkit for building and hosting interactive teaching materials

# Goal - interactive teaching environment

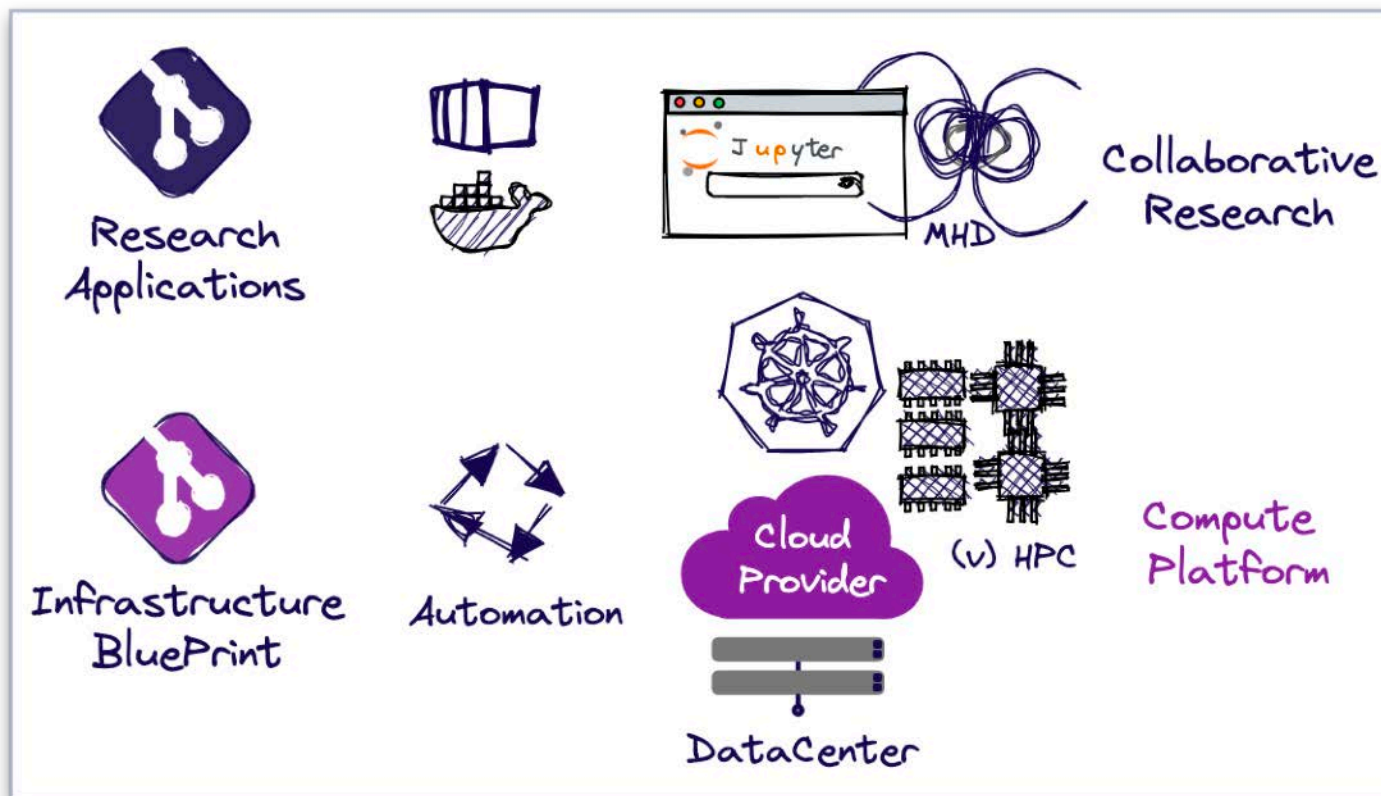


## State of the art teaching

allow for a reproducible environment, benefit from defined standards, give students an immersive experience, bring your research in the classroom,

- inter-university collaboration on cloud infrastructure
- lowering the learning curve for compute tasks in research and teaching
- secure exchange and collaboration on data, applications and results
- interactive access to High Performance Computing Systems
- collaborative development and sharing at Austrian academic institutions
- Toolkit for building and hosting interactive teaching materials
- Establish an open community for users and maintainers to contribute and support each other

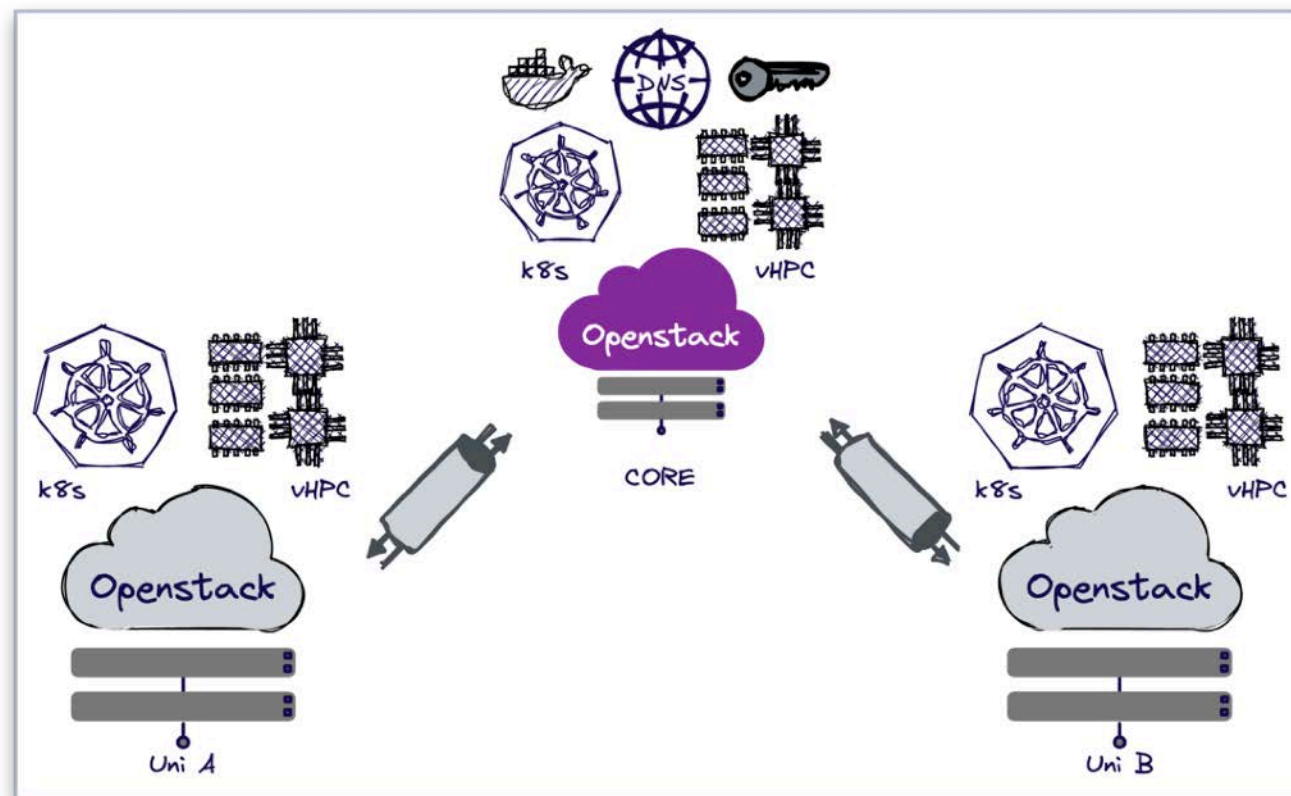




## Establish a self supporting community

benefit and give back to the community, gain visibility, share your ideas and results, gain new insights, use state of the art technologies

- inter-university collaboration on cloud infrastructure
- lowering the learning curve for compute tasks in research and teaching
- secure exchange and collaboration on data, applications and results
- interactive access to High Performance Computing Systems
- collaborative development and sharing at Austrian academic institutions
- Toolkit for building and hosting interactive teaching materials
- Establish an open community for users and maintainers to contribute and support each other



Interested? Get in touch!

We are hiring ;)

# An Austrian wide science cloud: Why it makes sense?

**O'REILLY** TEAMS ▾ INDIVIDUALS FEATURES ▾ BLOG CONTENT SPONSORSHIP 🔍

### Executive summary

- Roughly 90% of the respondents indicated that their organizations are using the cloud. That's a small increase over last year's 88%.
- The response to the survey was global; all continents (save Antarctica) were represented. Compared to last year, there was a much higher percentage of respondents from Europe (33%, as opposed to 11%) and a lower percentage from North America (42%).
- In every industry, at least 75% of the respondents work for organizations using the cloud. The most proactive industries are retail & ecommerce, finance & banking, and software.
- Amazon Web Services (AWS) (62%), Microsoft Azure (48%), and Google Cloud (33%) are still the big three, though Amazon's market share has

**30%**  
We're currently  
cloud native.

**20%**  
We'll be cloud  
native within  
3 or more years.

**11%**

We'll be cloud native  
within 2 years.

**6%**

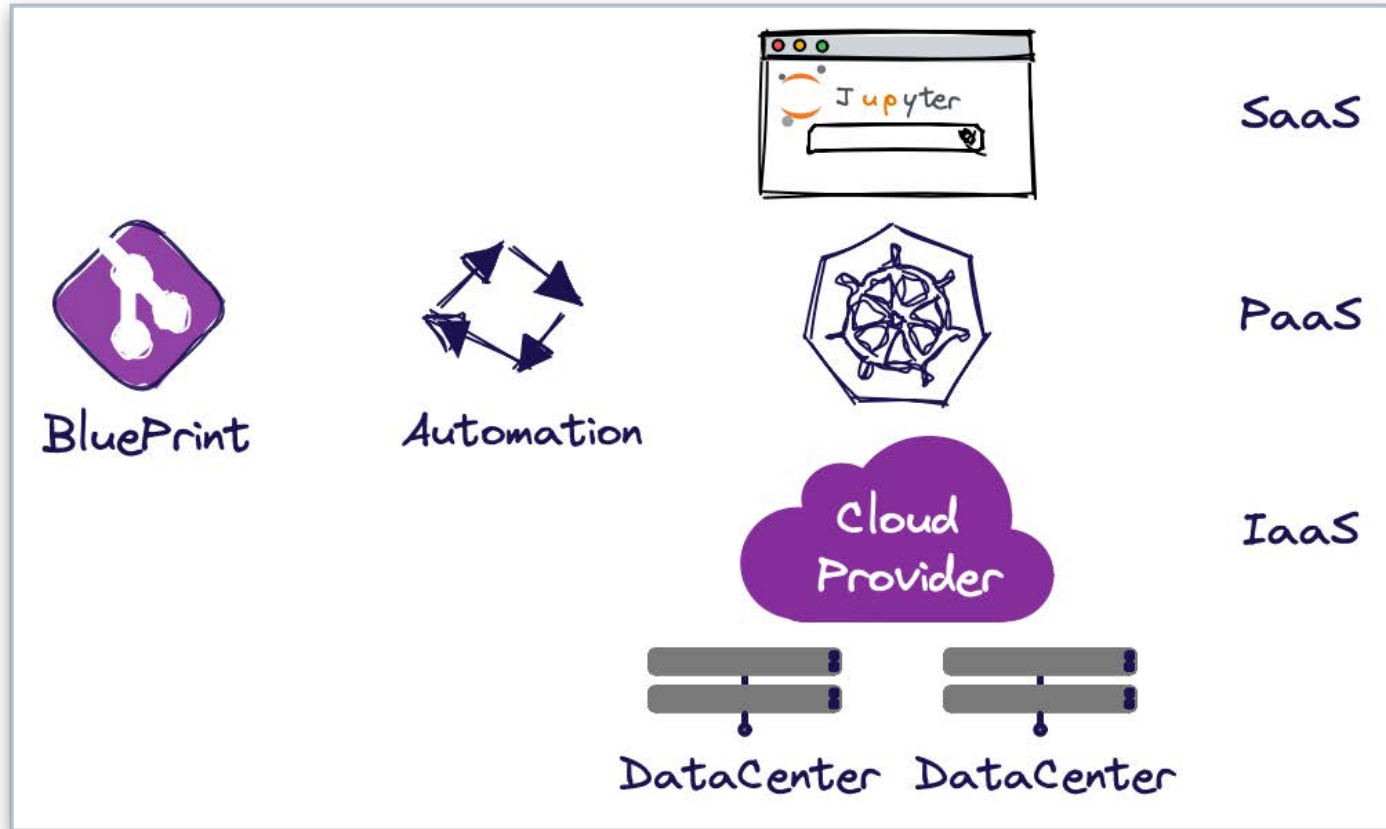
We'll be cloud native  
within 1 year.

**33%**  
We have no  
plans to go  
cloud native  
at this time.

## Is cloud computing taking over?

Cloud makes sense if: you can share, standardize & decouple

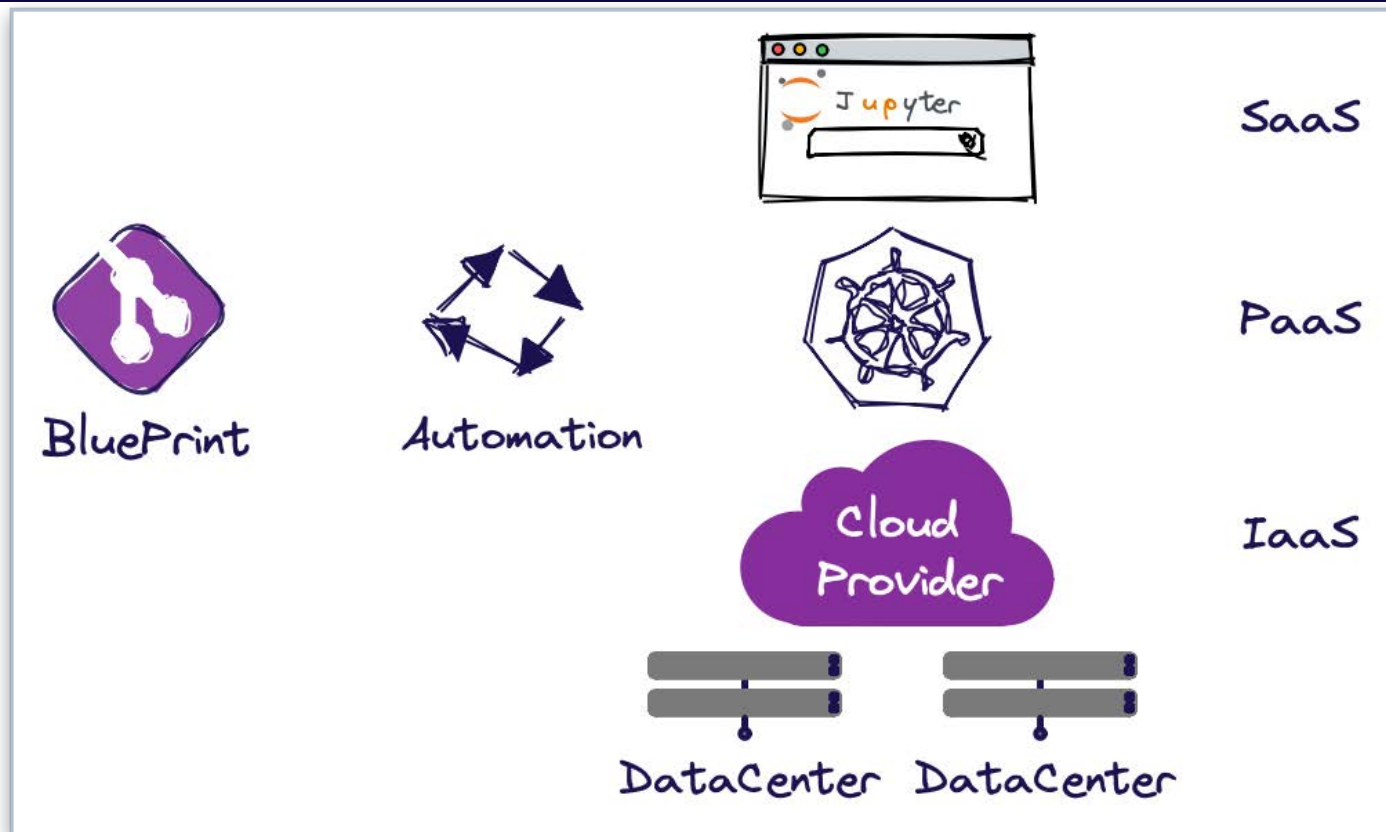
→ The more bespoke your requirements, the harder/costlier



## What do we mean by Open Science Cloud?

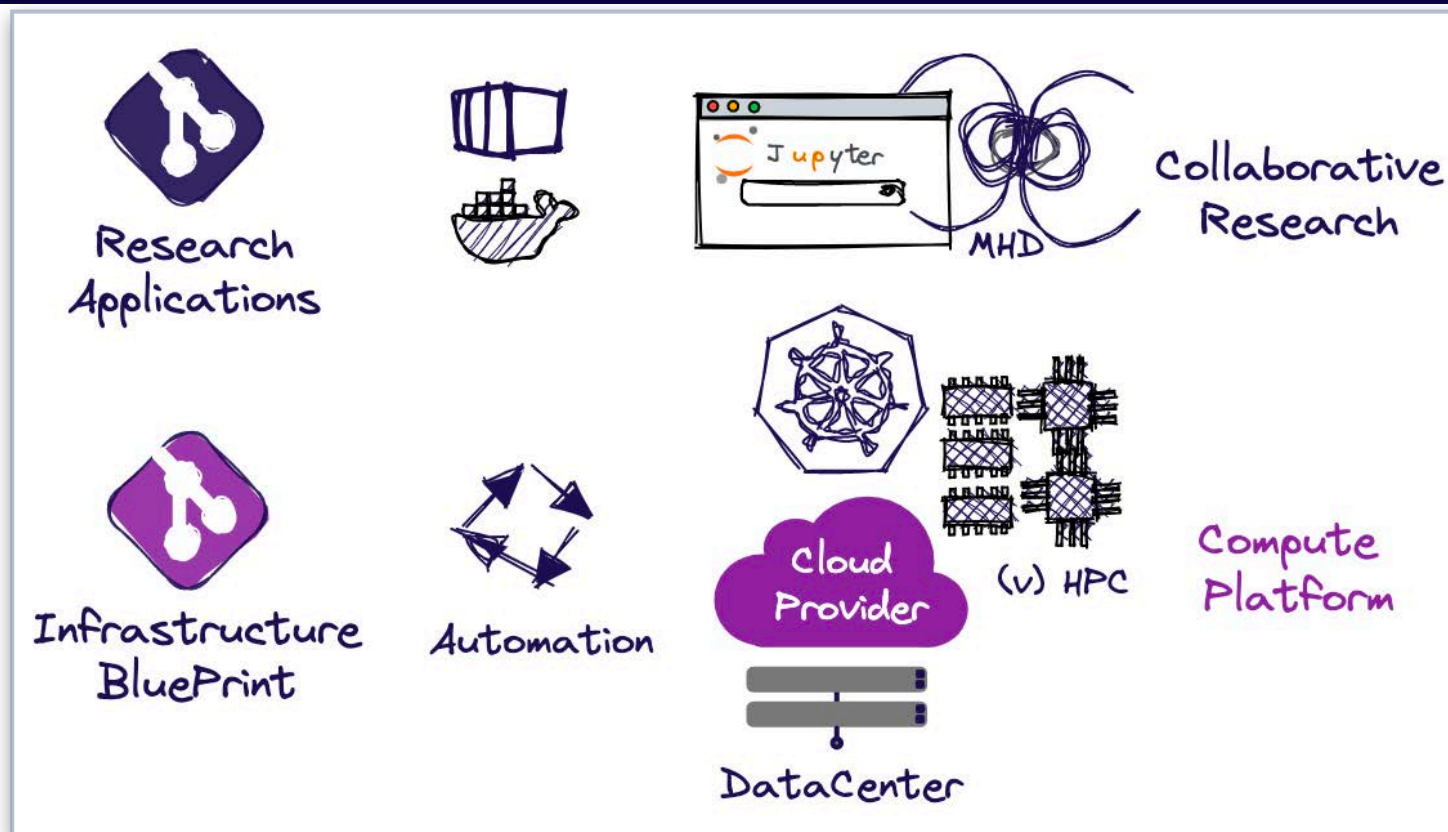
Definition: A distributed virtual runtime, where Infrastructure, Platforms and Applications can be elastically consumed 'as a service', while the customer provides the content.

Key paradigm features: automation, standard APIs, self-service, shared ownership, community driven



## What kind of services are we talking about?

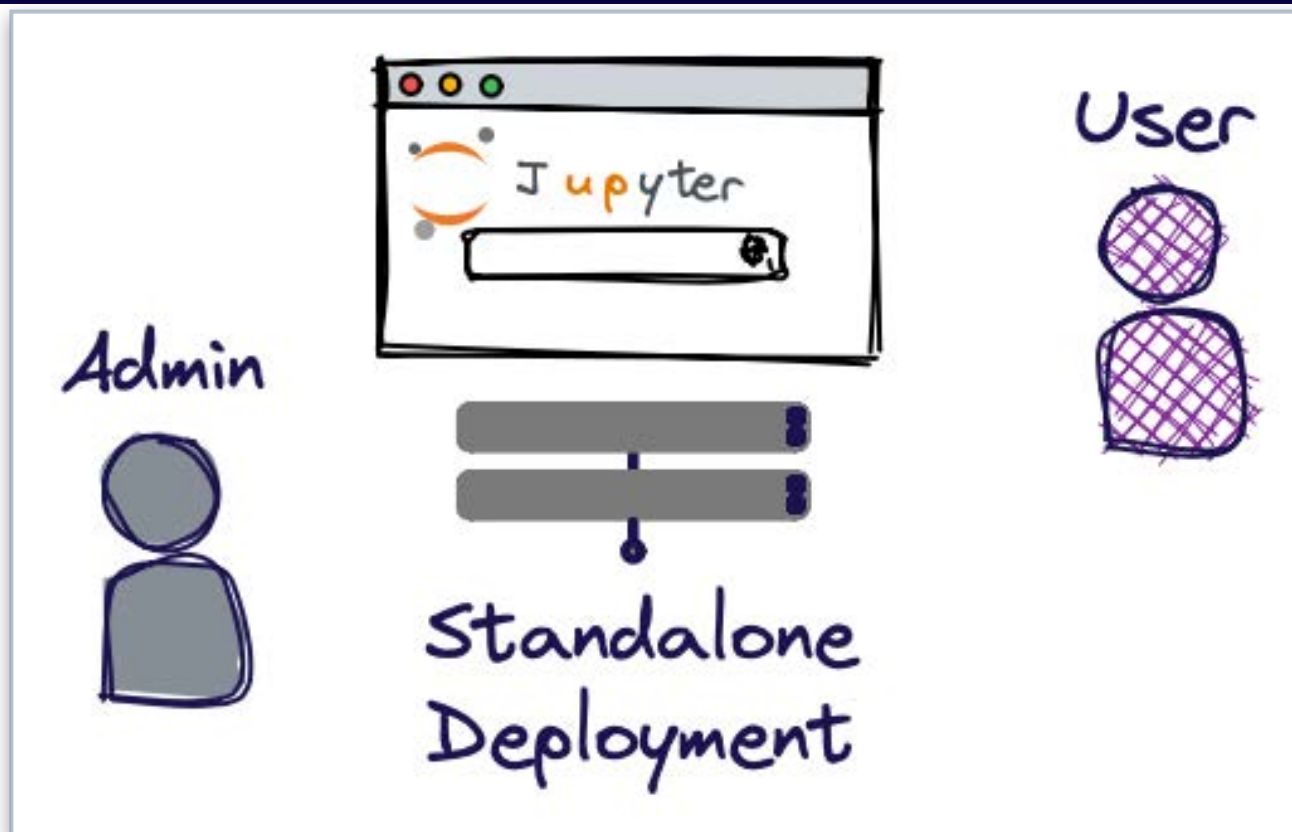
Definition: The CNCF defines cloud-native computing as the use of open source software, as well as technologies such as [containers](#), [microservices](#) and [service mesh](#), to develop and deploy scalable applications on [cloud computing](#) platforms



## Interlude: Each layer has its own blueprints

To guarantee interoperability, there are standards (like "contracts") between each layer -> enables a distributed community to contribute at all layers

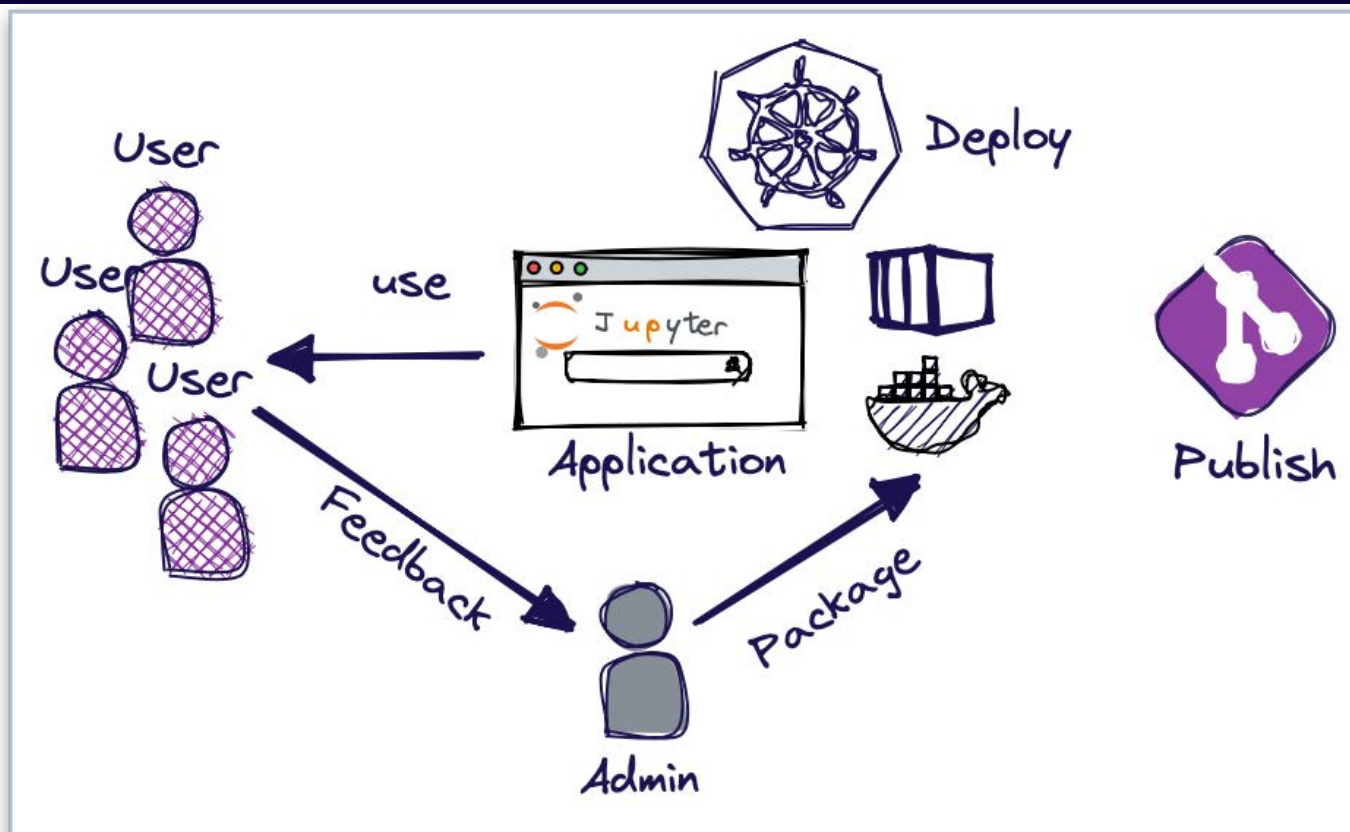




UseCase I: a teaching lab in Jupyter

User Feedback: All universities wish/have Jupyter labs of some form

-> How to make this scale & holistically improve (even to open source upstream)?



Process of making it "cattle":

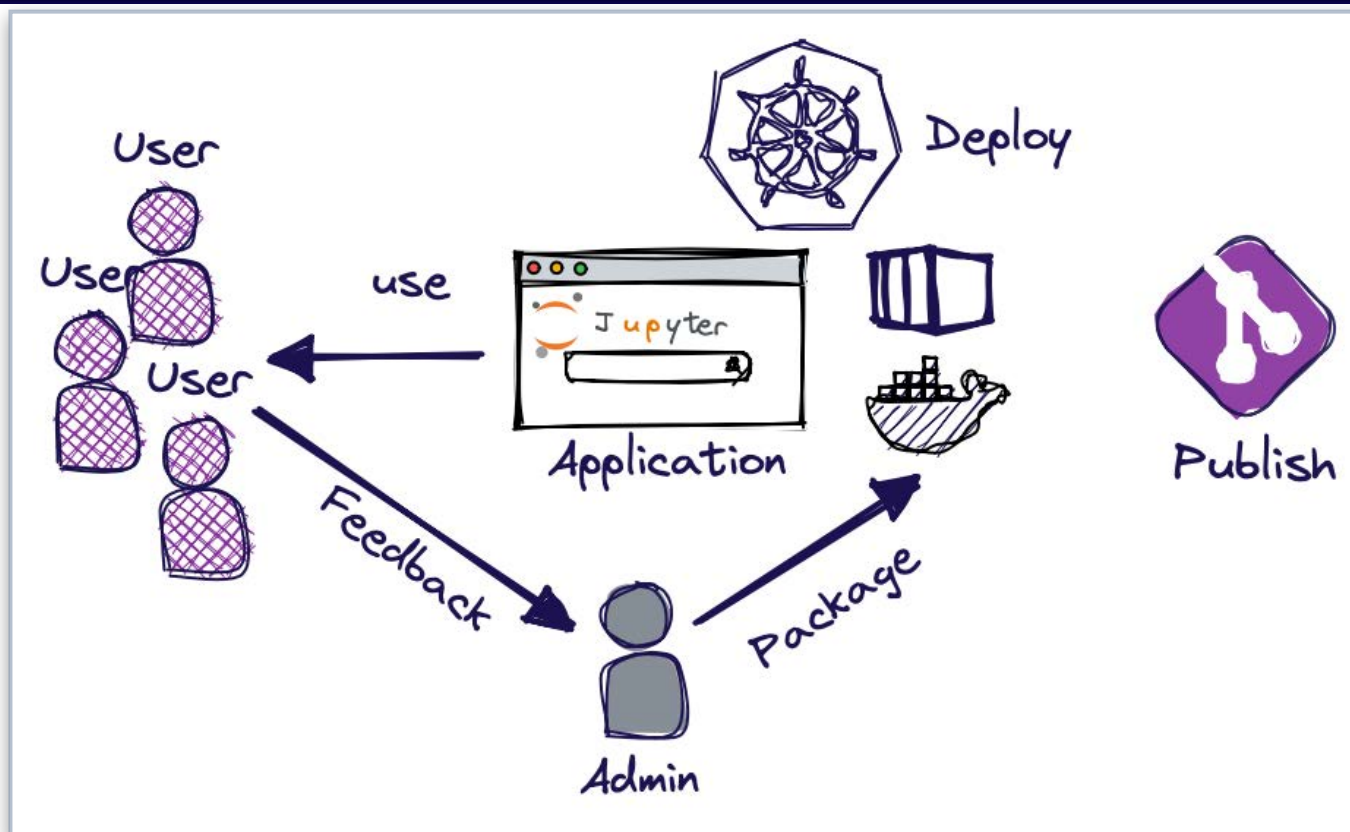
1. Extract common functionality
2. Package container
3. Share & release
4. Decide runtime platform
5. Standardize deployment
6. Publish deployment package for all to use
7. Improve common functionality
8. Contribute back to open source

Plus:

1. Enable users to integrate their custom content
2. Enable user lifecycle

## UseCase I: a teaching lab in Jupyter

-> standard Jupiter platform + custom lectures + federated users  
can be used by anyone who uses same standard runtime platform

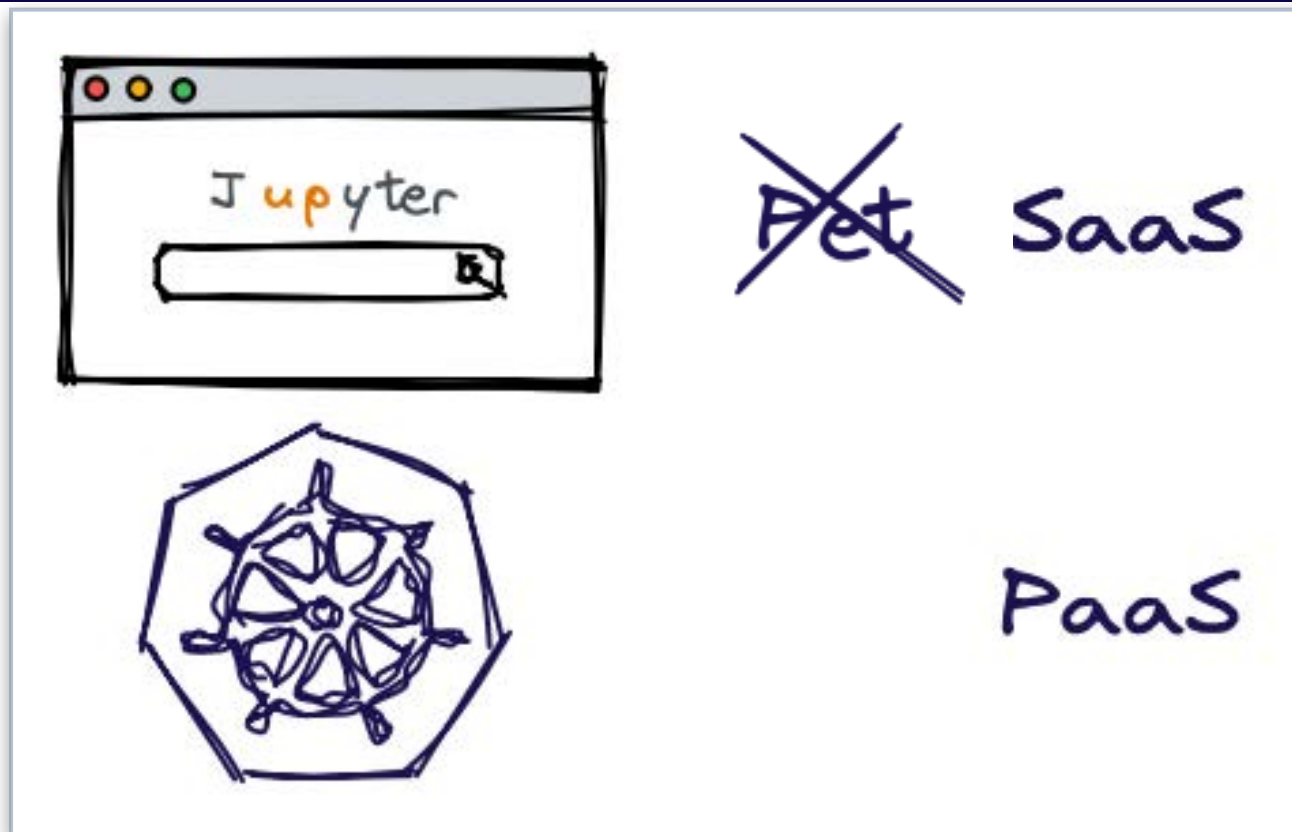


### Services:

1. Build & release services
2. Container registries
3. User federation, IdP
4. Lecture content management
5. (LMS/Moodle Integration)
6. (Application Development)

## UseCase I: a teaching lab in Jupyter

-> standard Jupiter platform + custom lectures + federated users  
can be used by anyone who uses same standard runtime platform



## UseCase II: Container runtime (multi tenant)

User Feedback: We have this application in a container, we want to run it somewhere

→ upkeep and configuration of a multi-tenant container orchestrator highly nontrivial

## What is kubernetes (k8s) ?



**Tabitha Sable**

@TabbySable

Replying to [@krisnova](#)

A friendly robot that uses control theory to manifest our hopes and dreams, so long as those dreams can be expressed in YAML.

7:44 PM · Oct 21, 2021 · Twitter for iPhone

2 Retweets 2 Quote Tweets 24 Likes



Howdy!  
Welcome to Rancher

Log in with AzureAD

Use a local user



local

adls-jupyter x



Cluster



Workload



Apps & Marketplace



Charts

Installed Apps

1

Repositories

4

Recent Operations

0

Service Discovery



Storage



Monitoring



Logging



## Charts

jupyter



All Categories



Filter



jupyterhub

Multi-user Jupyter installation

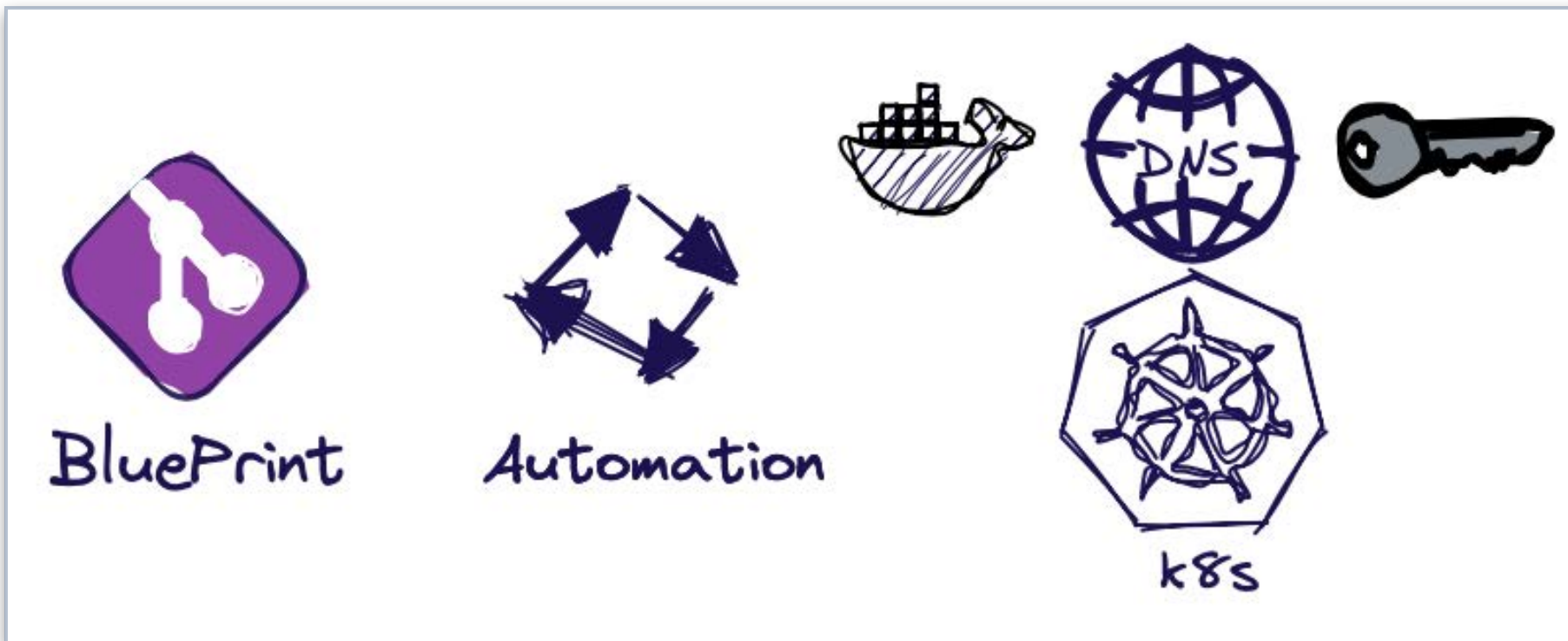
Linux only



pebble

This Helm chart bootstraps Pebble: an ACME server (like...

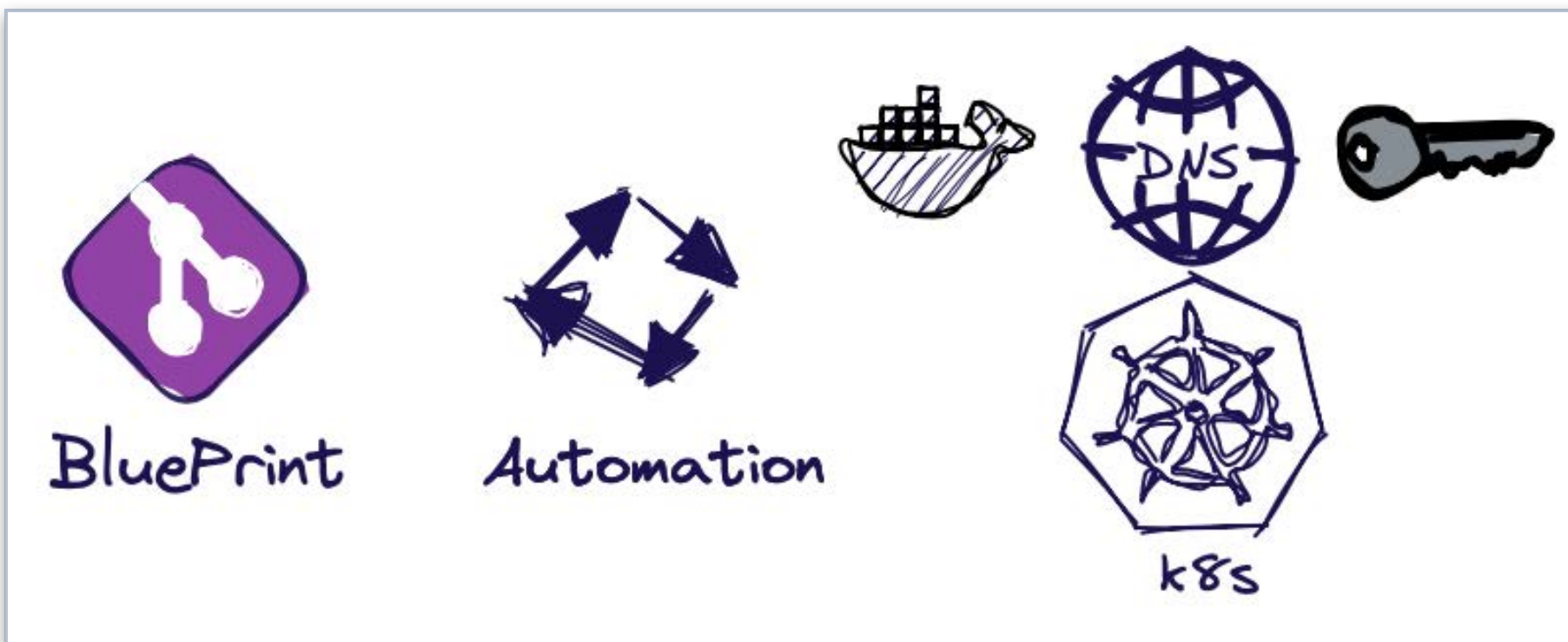
Linux only



## UseCase II: Container runtime (multi-tenant)

Two types of templates: (1) Kubernetes deployments to provide people with k8s

(2) deployments of user apps ON TOP of kubernetes



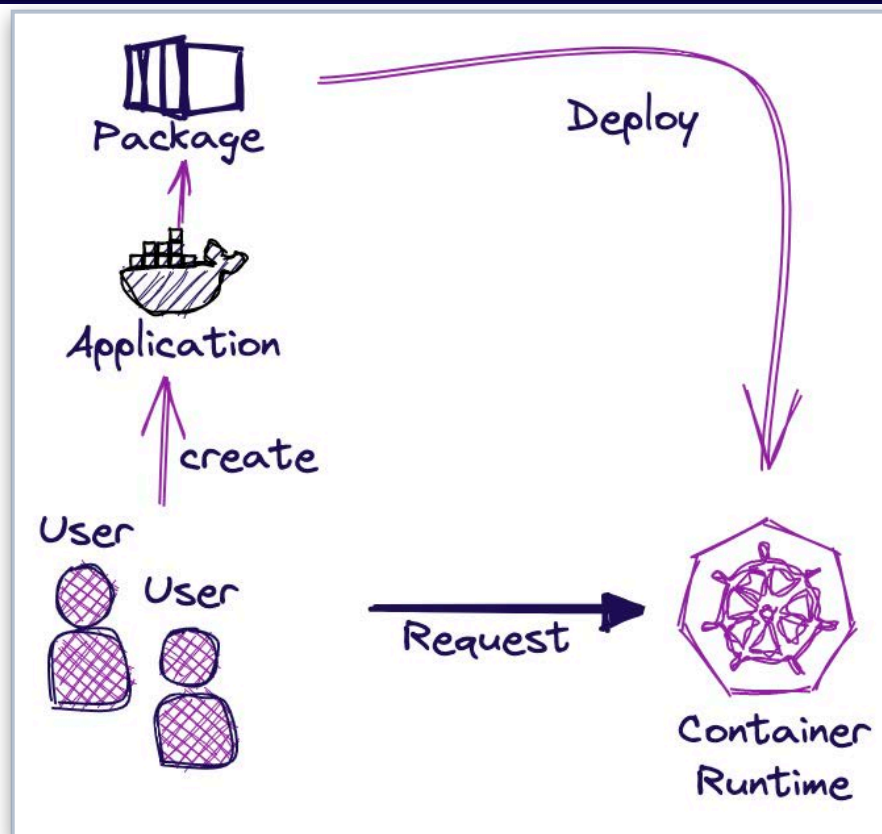
## UseCase II: Container runtime (multi-tenant)

Consolidate various uni's feature requirements, plus:

network, observability, users, security →

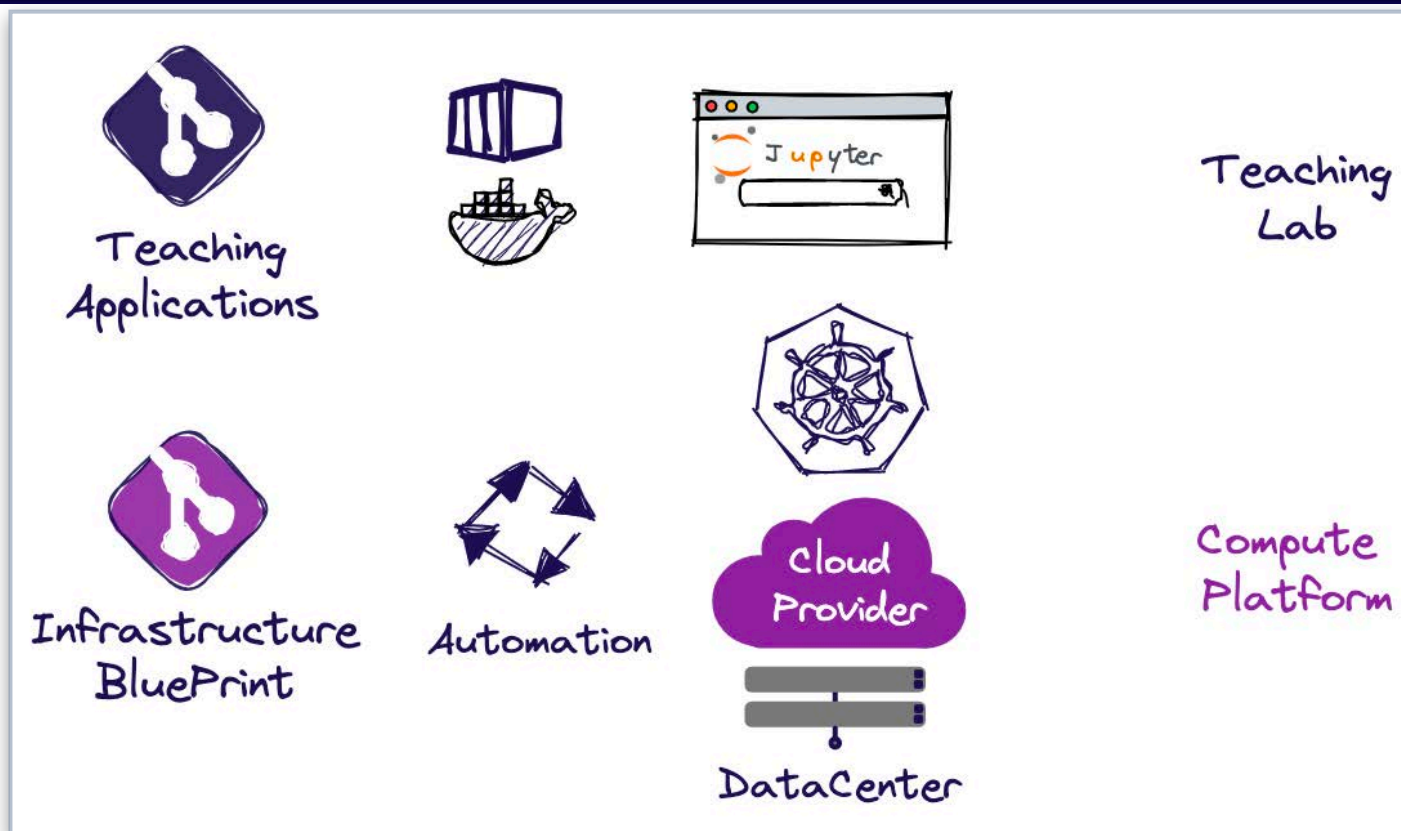
establish templates to add apps





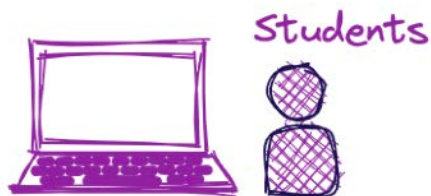
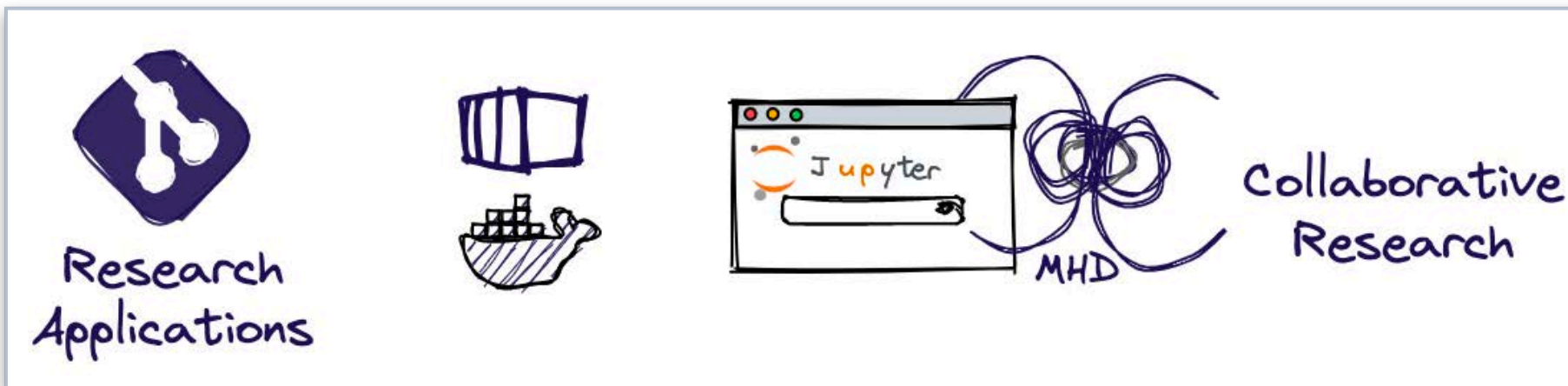
## UseCase II: Container runtime for standard apps

If users follow the standards on how to publish their application, they can request a runtime for it. Since the runtime is standardized, it can be used on various locations.



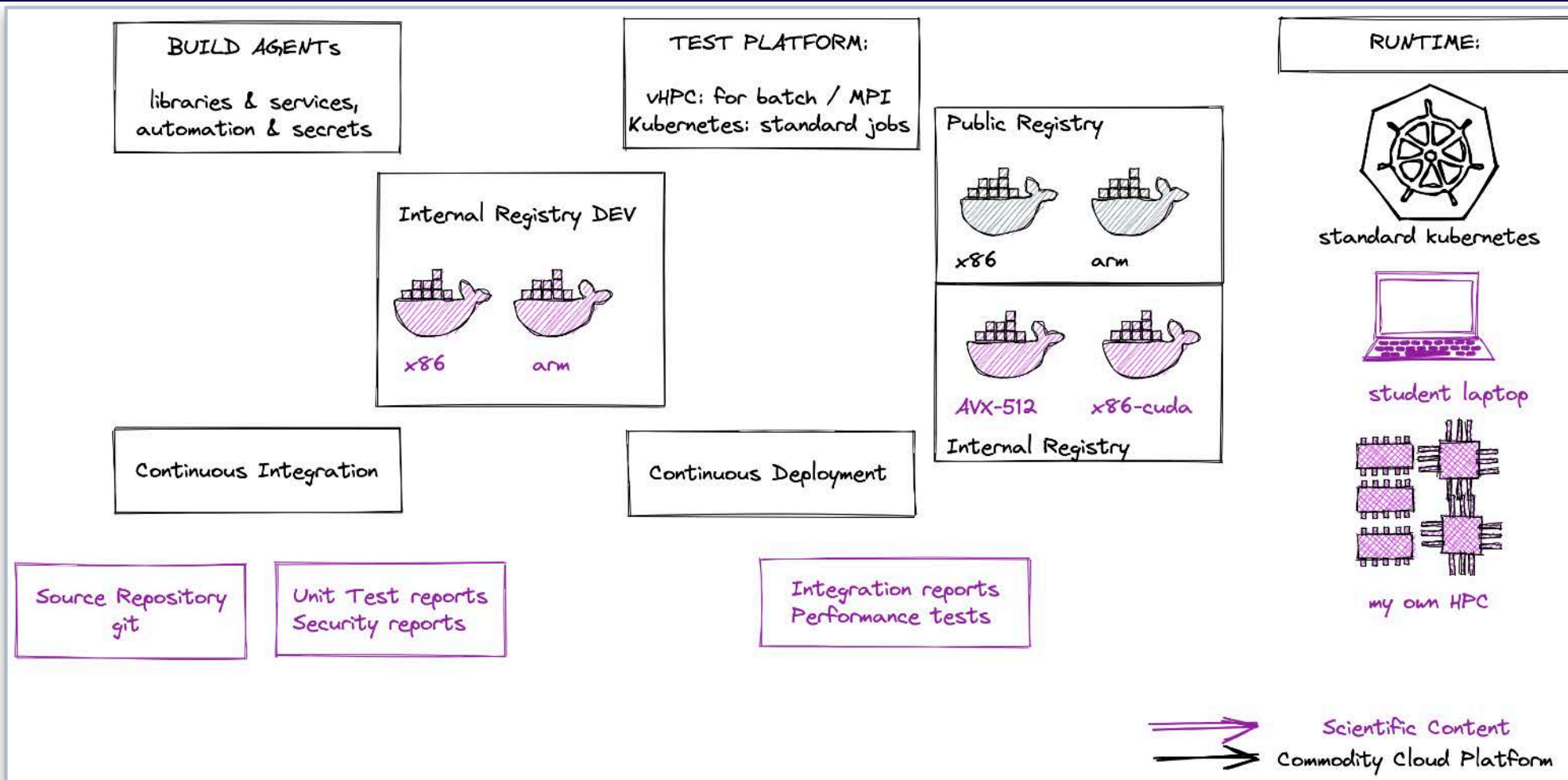
## UseCase II: Container runtime for standard apps

Our example Jupyter SaaS can now be hosted on this Container runtime and used by anyone, who can deploy our PaaS BluePrint

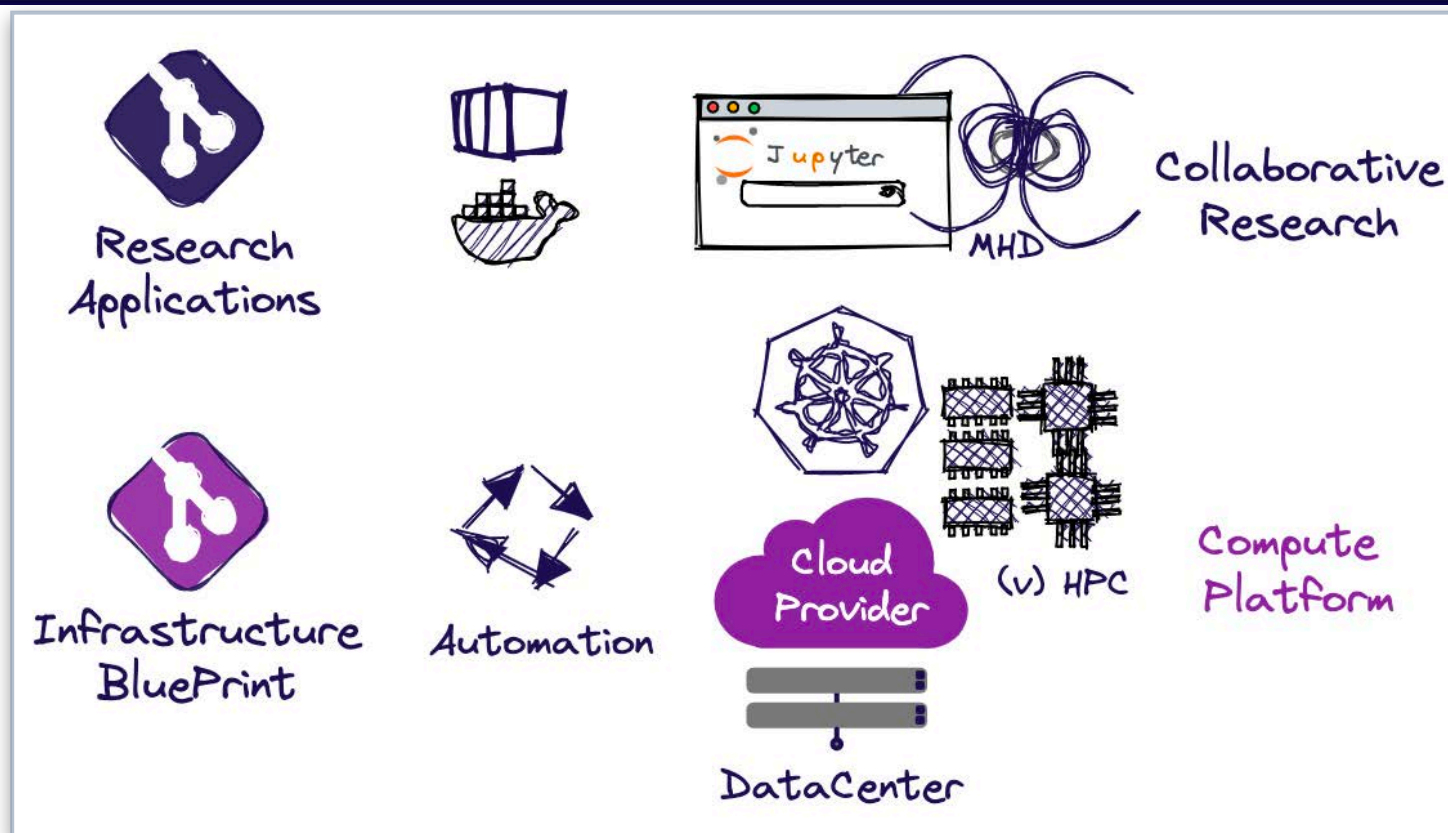


## UseCase III: Research software packaging (incl HPC)

Feedback: researchers would like to distribute easy to run packages of HPC applications (e.g. mpi), especially for onboarding students or new team members



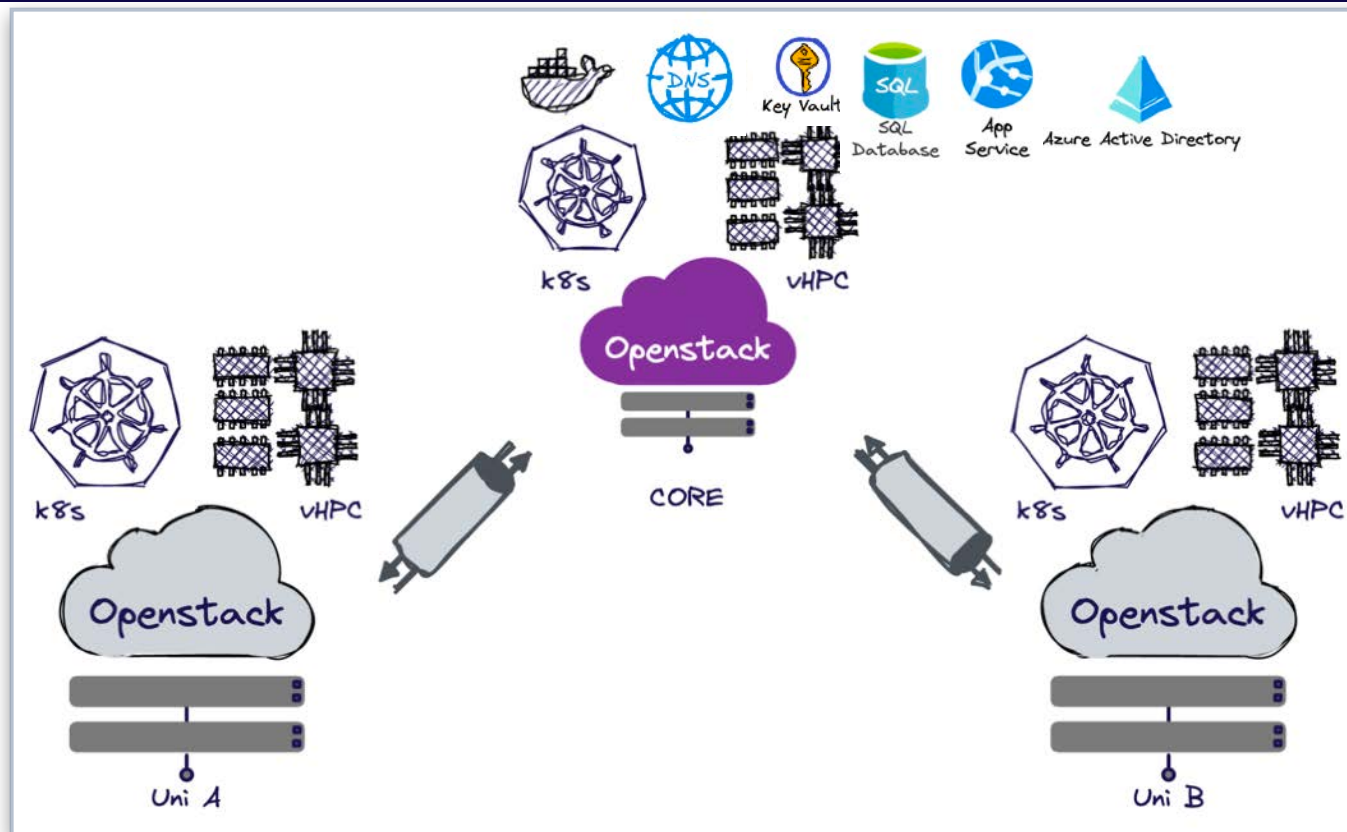
Use Case III: Research software packaging (incl HPC)  
Enable the reproducibility and sharing of research software



Vision: find ways to combine HPC and cloud native

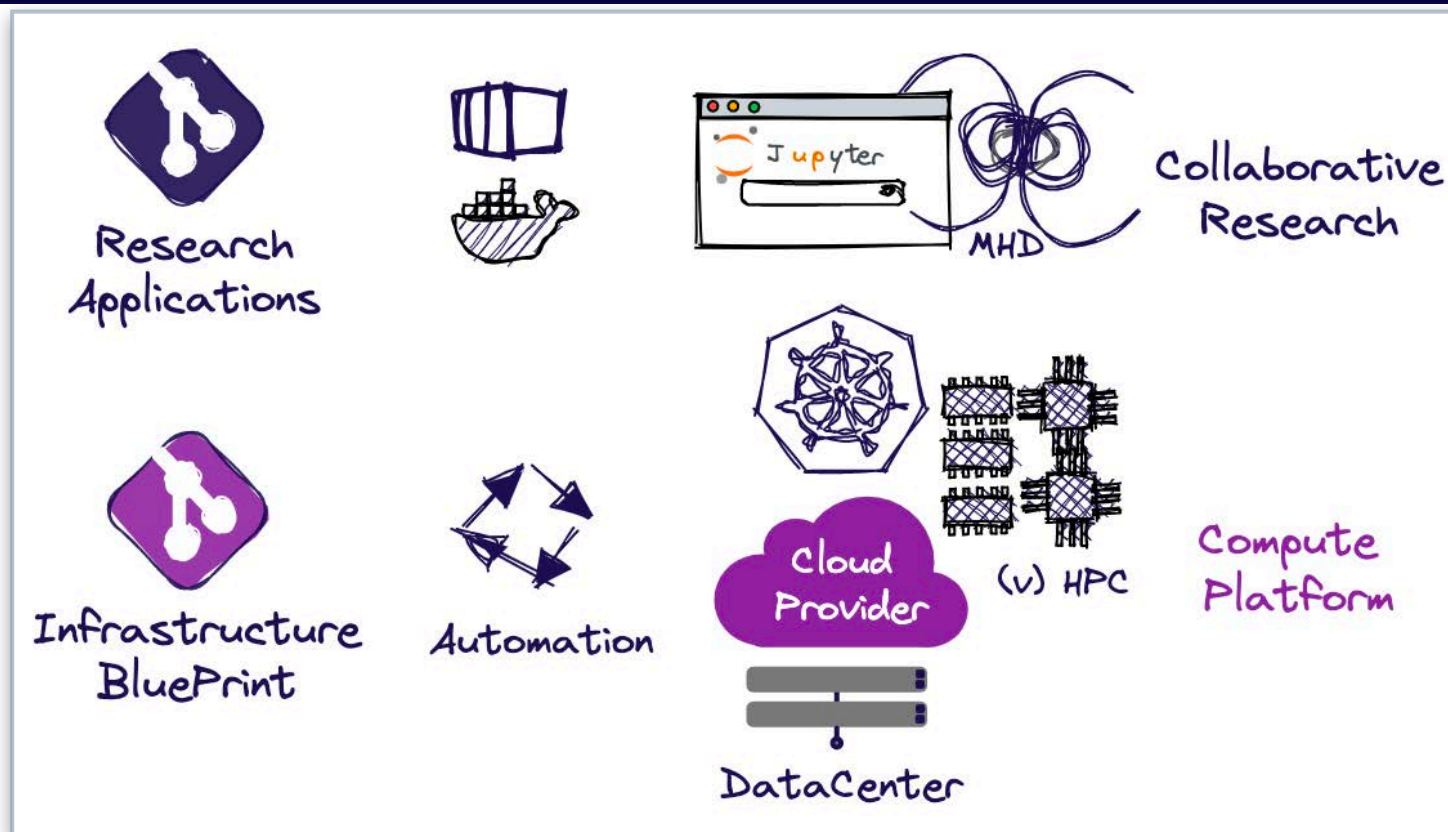
Leverage ease-of-use of cloud-native and integrate it with classical HPC systems

-> early stage of prototypes worldwide, collaborations ongoing



## Multicloud: how to combine public/private cloud

Consume cost efficiently from public providers for specialized services, use "metal" we already have it often makes no sense to build and maintain pure commodity services -> buy before build



Summary: Uni's differentiate by content, ZIDs consolidate

Let engineers enable standard platforms and accelerate researchers to do their research

# TALK 2 | Cloud Native Labs for Teaching and Research



# Jupyter with Kubernetes on Openstack

What and for whom is Jupyter?

Why do we need Jupyter to teach our students?

What is Kubernetes and why is it a good choice for Jupyter?

What is Openstack and why is it a good choice for Kubernetes?

How can researchers/universities benefit from this in the long run?

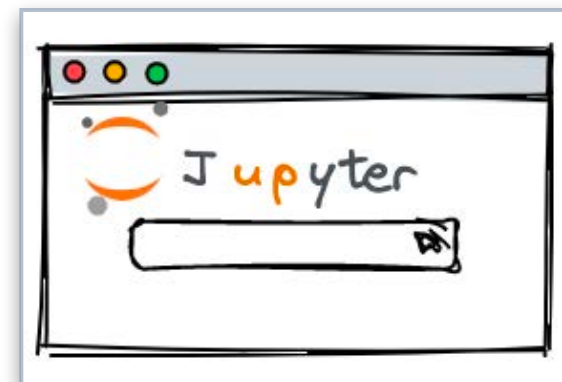
What is already working? - What do we still need for this to work?

## What is Jupyter?

- open standards & web services
- web-based, multi-user, interactive computing environment
- across all programming languages

## For whom is Jupyter?

- classrooms, research labs and companies doing:
  - data science, scientific computing,
  - computational journalism, machine learning, etc.
- EXECUTE, VISUALISE & SHARE code and data in a web-based environment



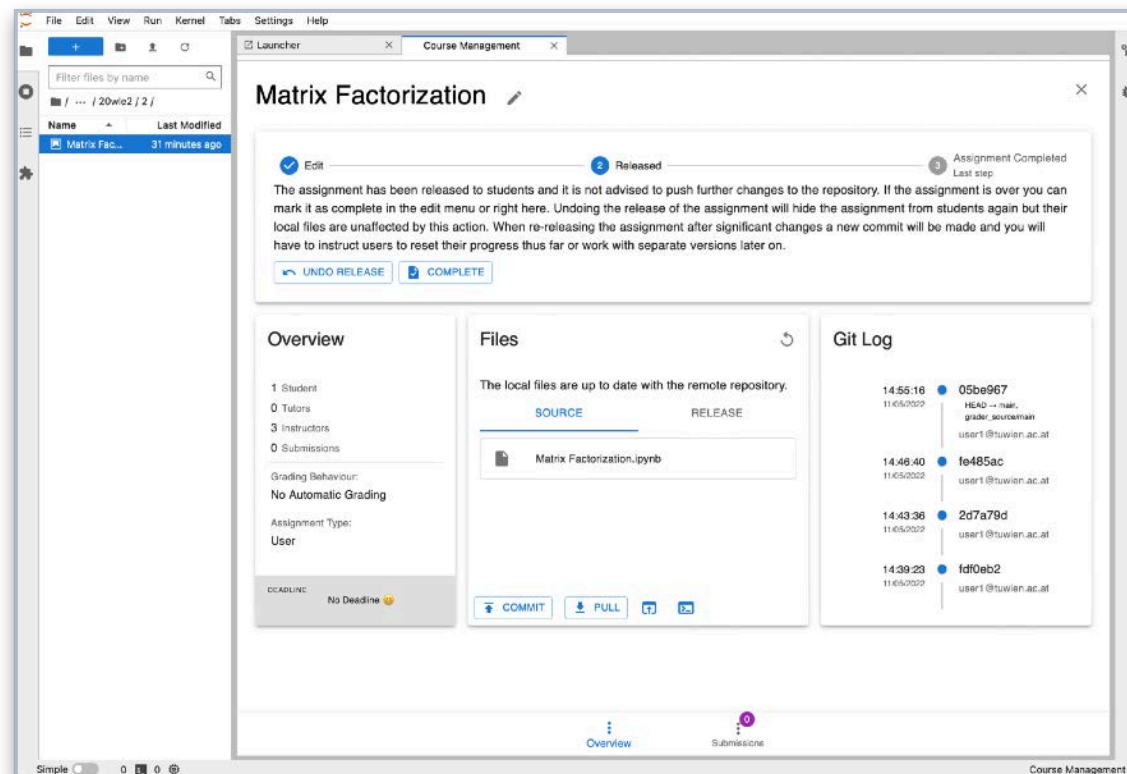
## Why do we need Jupyter to teach our students?

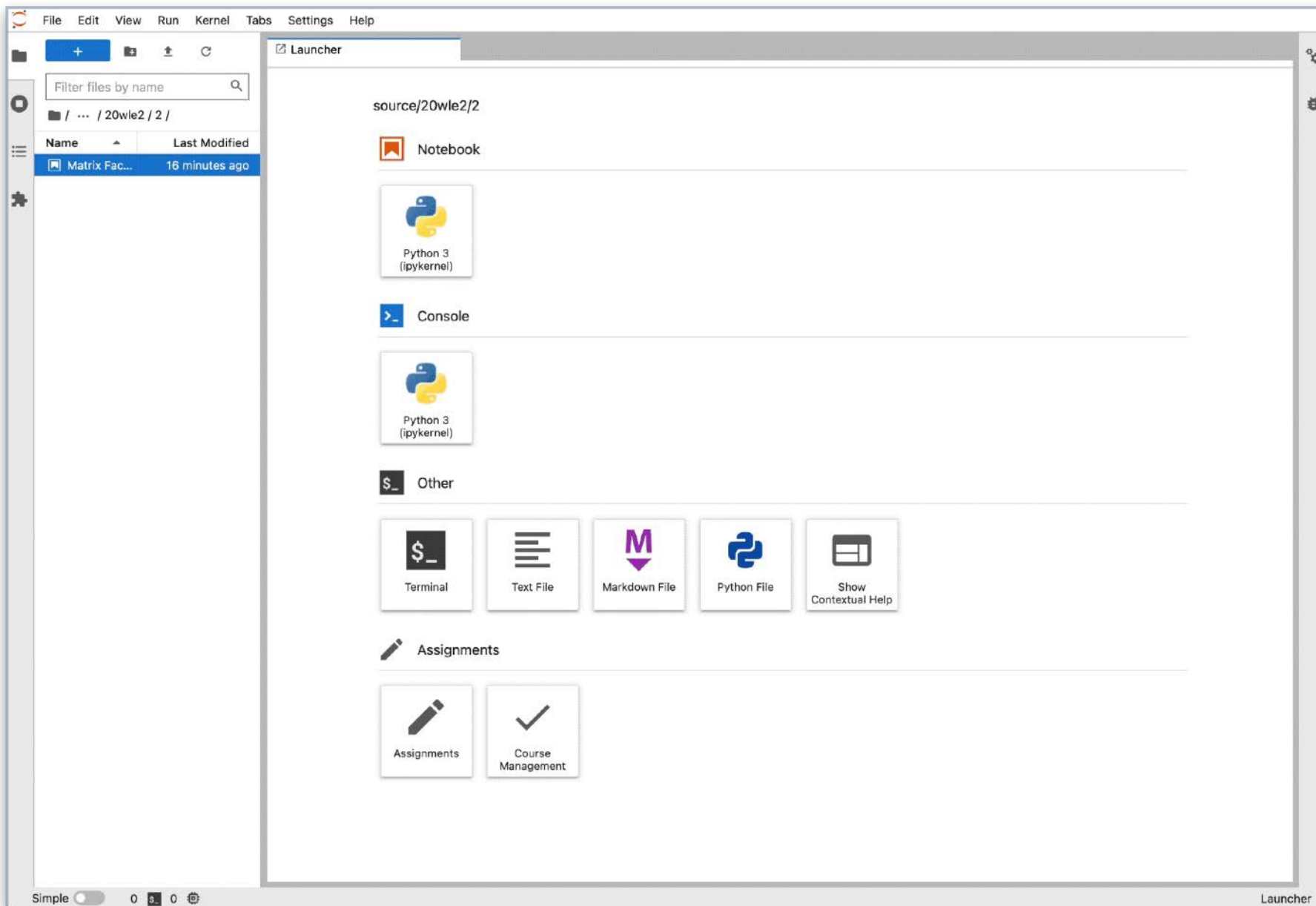
- combine text with code snippets & outputs
- tables, graphs and maps on a single page

Jupyter => open standards & web services  
for interactive computing across all programming languages

JupyterLab => web-based interactive development environment  
for notebooks, code, data science, scientific computing,  
computational journalism, and machine learning

JupyterHub => multi-user version  
for companies, classrooms and research labs

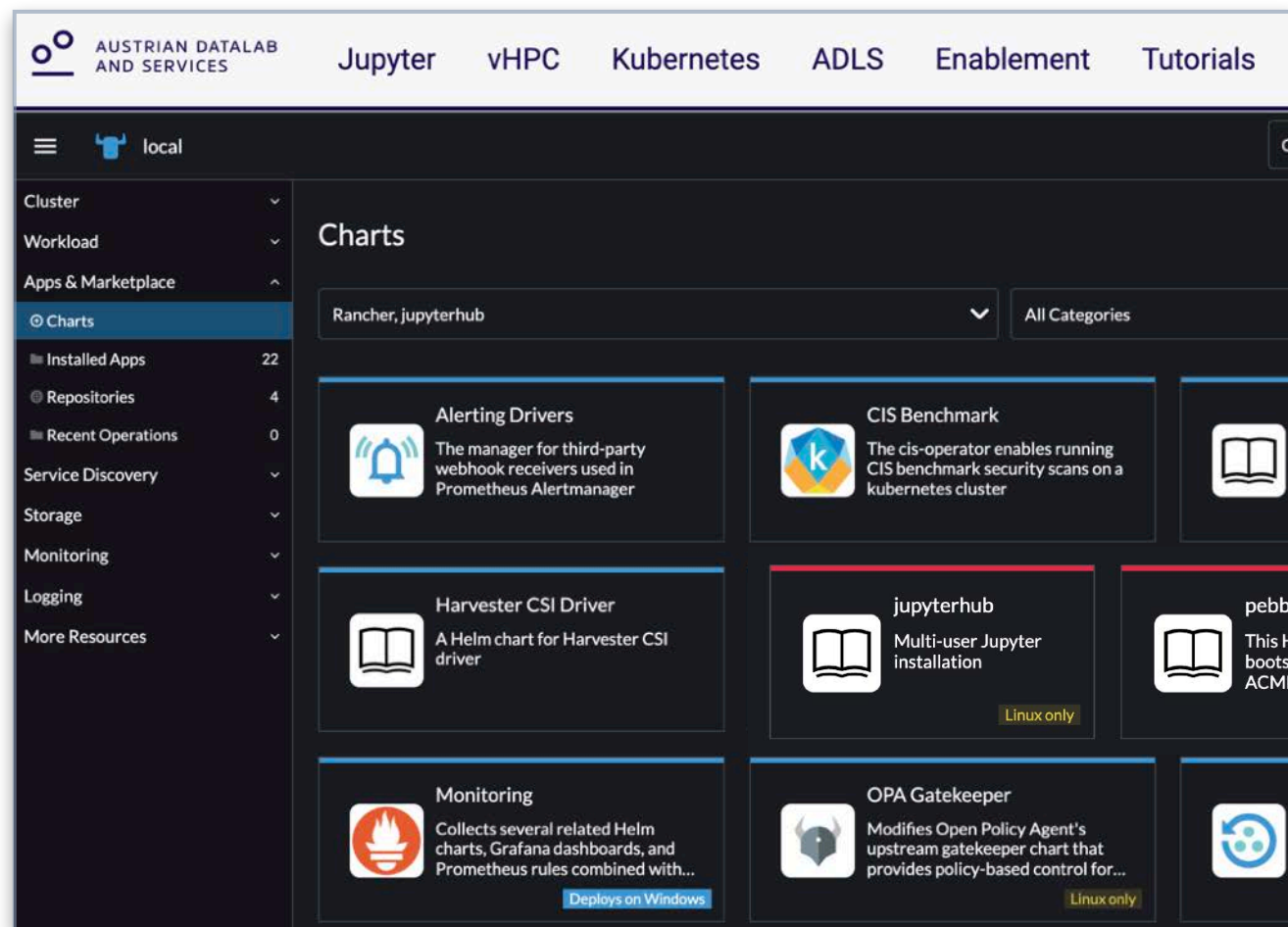




- Manage Assignments
- Grade Automatically
- Inform Students

## What is Kubernetes | k8s ?

- open-source system
- deployment automation
- scaling and management
- for containerized applications



The screenshot shows the 'Kubernetes' section of the Austrian Datalab and Services dashboard. The top navigation bar includes 'Jupyter', 'vHPC', 'Kubernetes', 'ADLS', 'Enablement', and 'Tutorials'. The main content area is titled 'Charts' and displays a grid of available Helm charts for installation. The charts listed include:

- Alerting Drivers**: The manager for third-party webhook receivers used in Prometheus Alertmanager.
- CIS Benchmark**: The cis-operator enables running CIS benchmark security scans on a kubernetes cluster.
- Harvester CSI Driver**: A Helm chart for Harvester CSI driver.
- jupyterhub**: Multi-user Jupyter installation (Linux only).
- Monitoring**: Collects several related Helm charts, Grafana dashboards, and Prometheus rules combined with... (Deploys on Windows).
- OPA Gatekeeper**: Modifies Open Policy Agent's upstream gatekeeper chart that provides policy-based control for... (Linux only).

## What is Kubernetes | k8s ?



The screenshot displays the Kubernetes Cluster Dashboard for a cluster named 'adls-jupyter'. The interface includes a sidebar with navigation options such as Cluster, Workload, and Apps & Marketplace. The main content area shows the Cluster Dashboard with a summary of resources and capacity.

**Cluster Dashboard**

Provider: RKE2    Kubernetes Version: v1.22.9    Created: 3.2 days ago    [Add Cluster Badge](#)

Resource	Count
Total Resources	168
Nodes	9
Deployments	32

**Capacity**

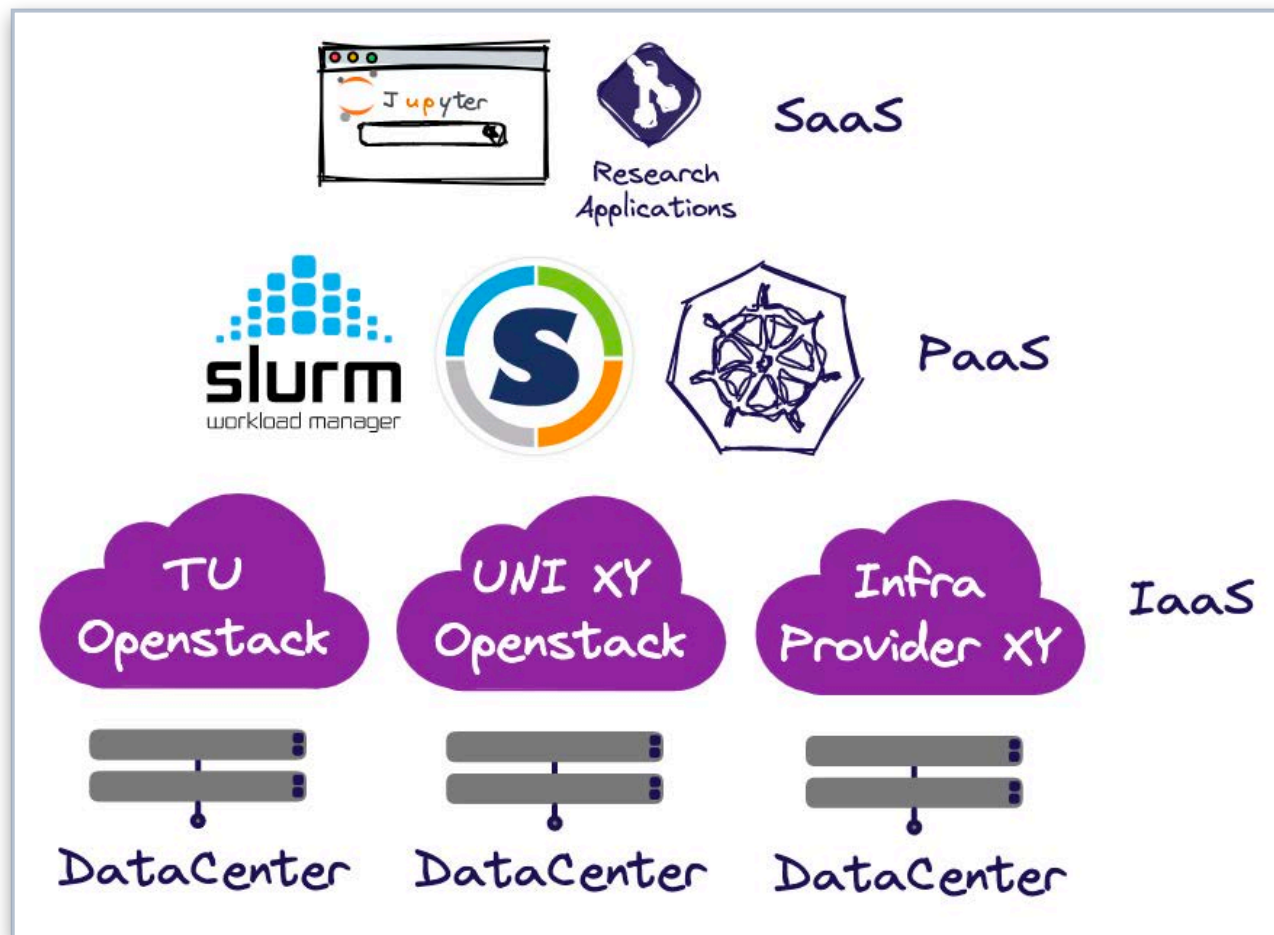
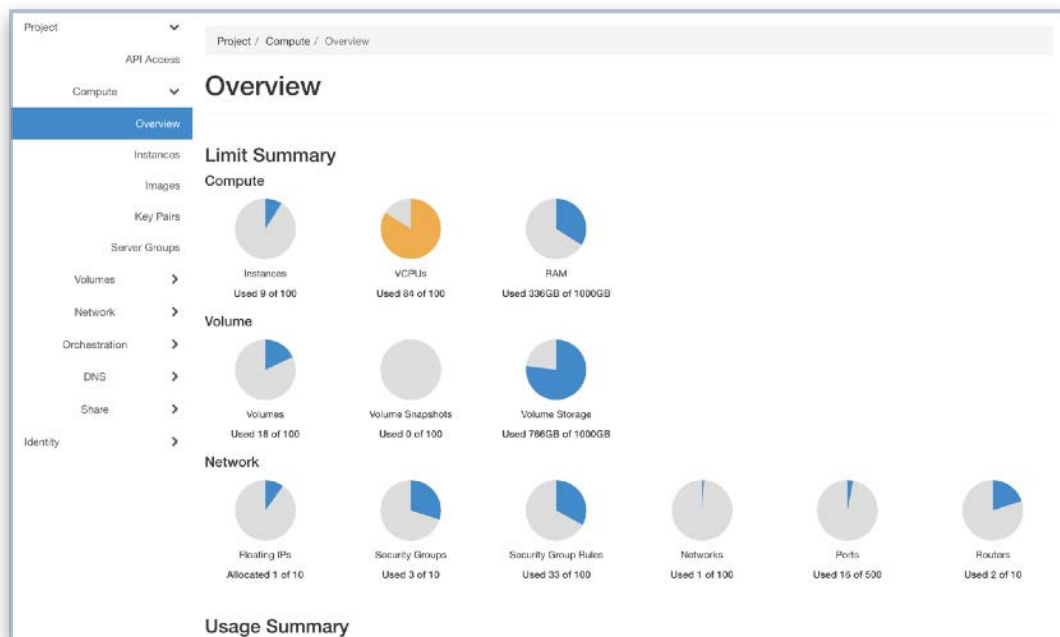
Resource	Used	Limit	Percentage
Pods	171 / 660	660	25.91%
Cores	2.25 / 72	72	3.12%
Memory	28 / 283 GiB	283 GiB	9.89%

Cluster Tools

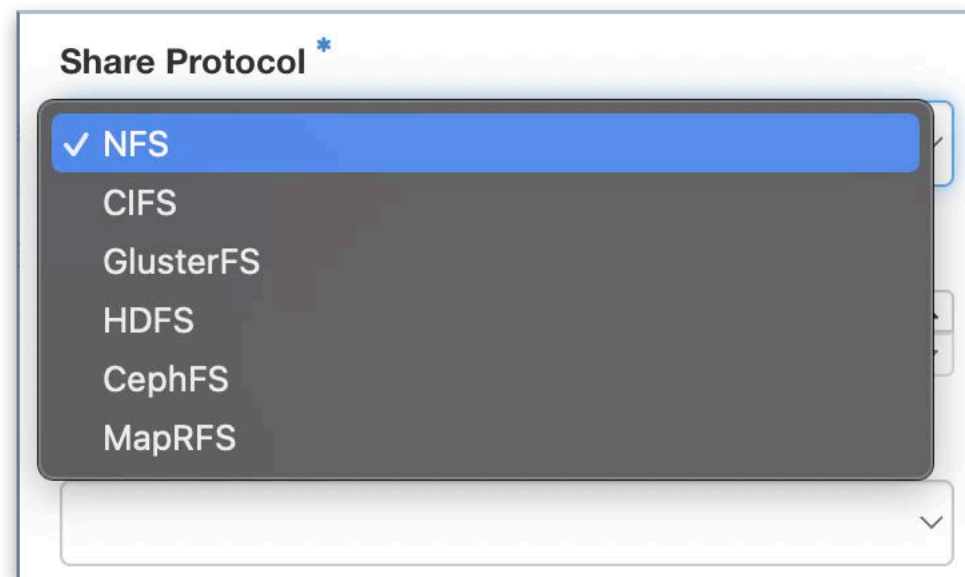
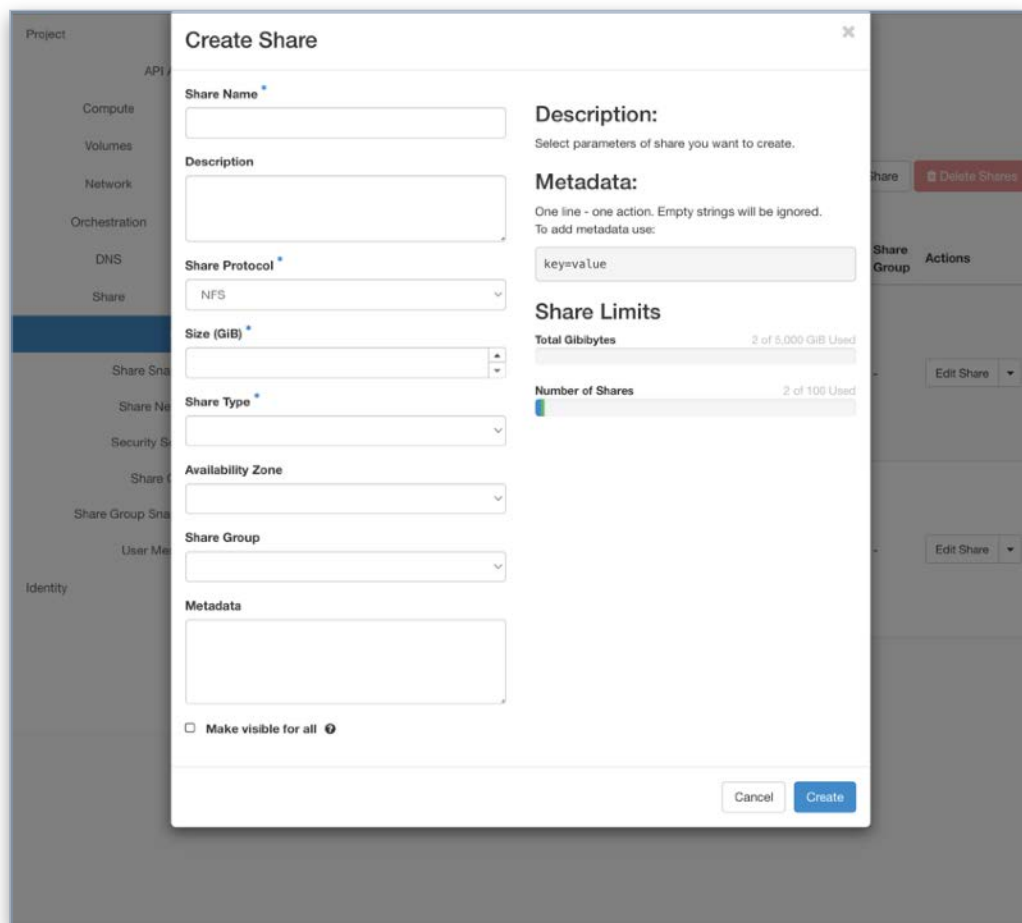
v2.6.4

## What is Openstack?

Open Source Cloud Software to provide common services for cloud infrastructure



## What is Openstack?





## Benefits

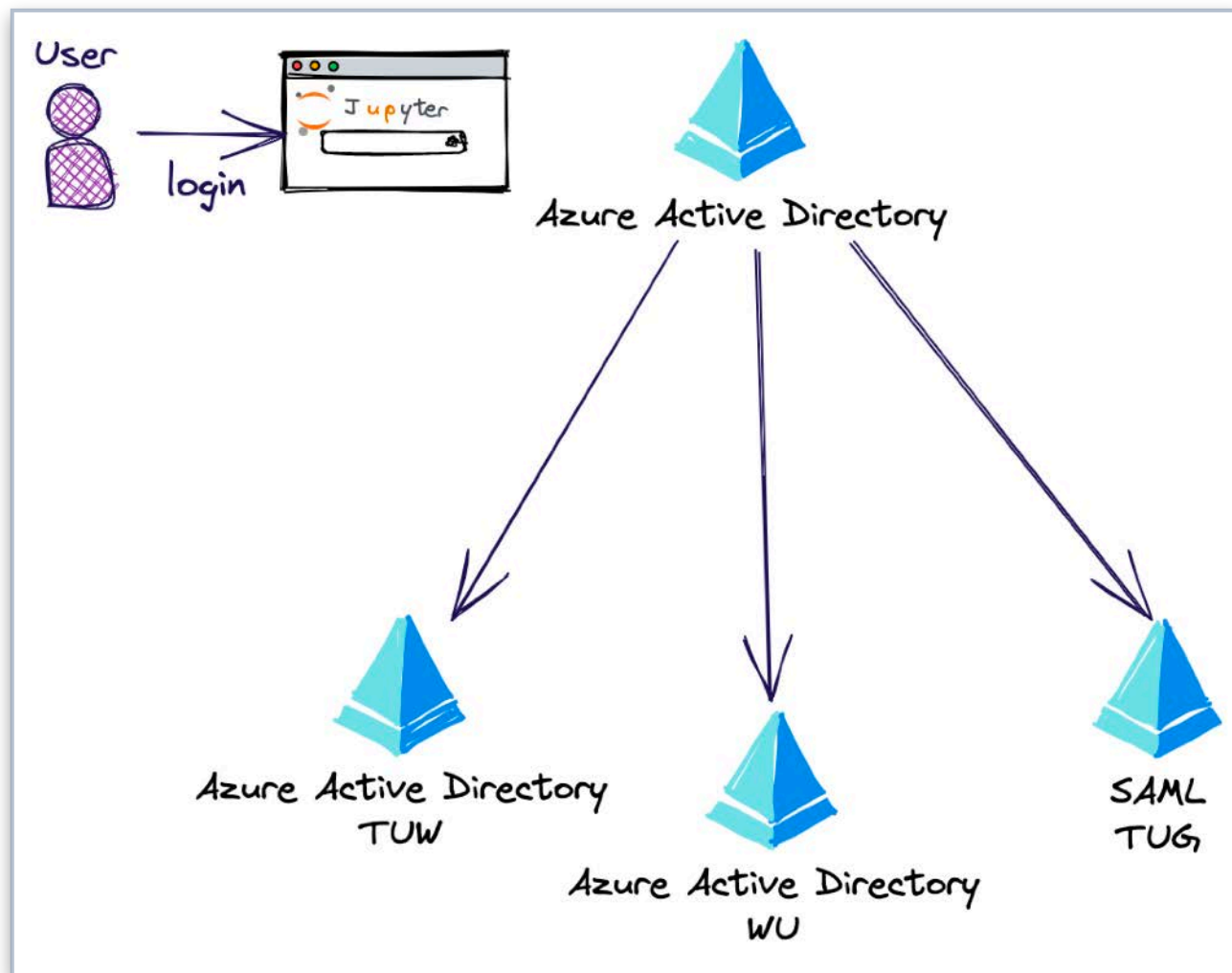
Interactive Access | Jupyter

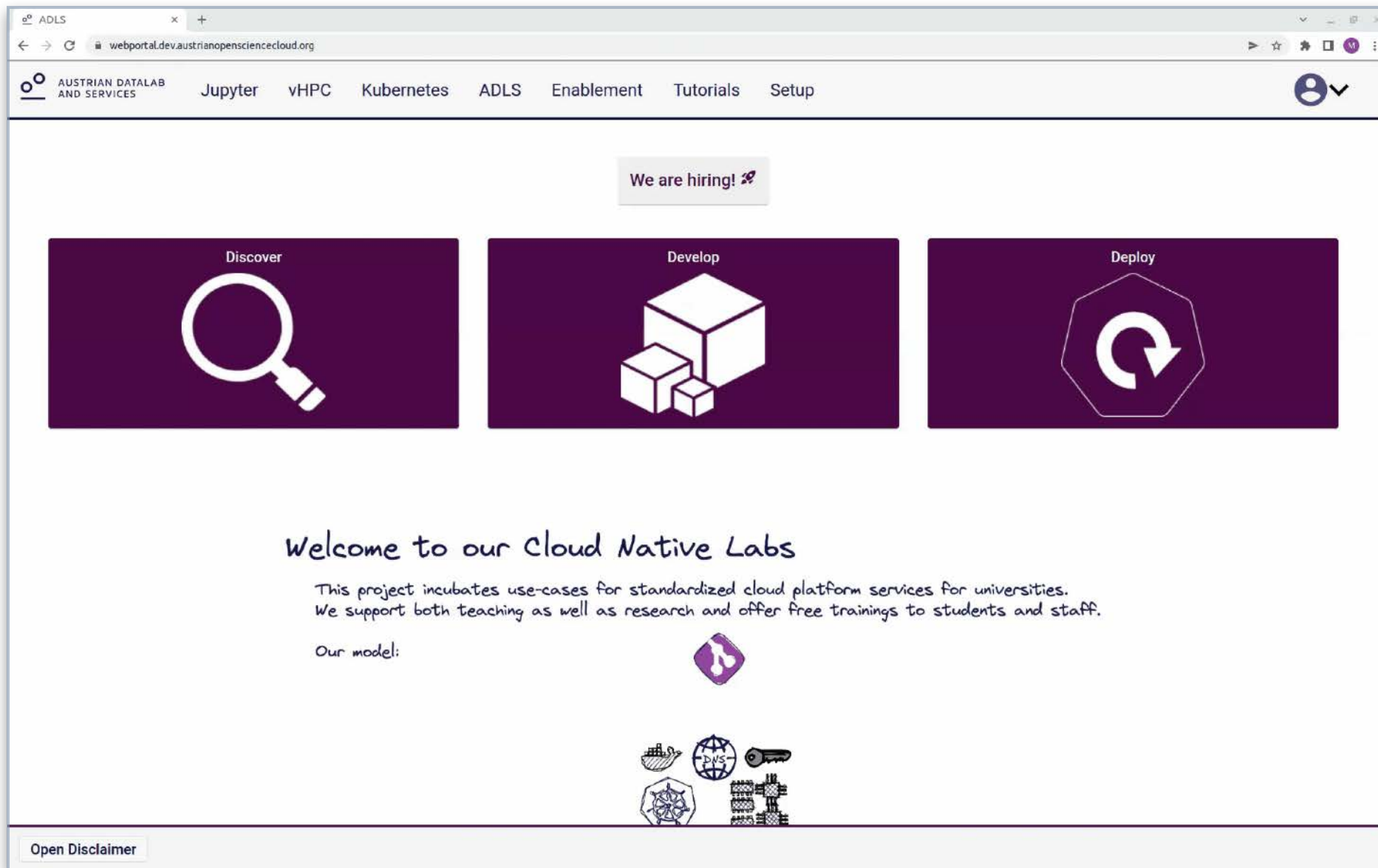
Dynamic, Scalable & Flexible | Kubernetes & OpenStack

Open-source Community | everyone can participate | TRUE Collaboration

## Federation

- between Universities
- no need for separate accounts
- Granting Consent at signup
- Multi Factor Authentication | MFA
- Webportal -> discoverability





The screenshot shows a web browser window with the URL `webportal.dev.austrianopensciencecloud.org`. The page features a navigation menu with links for Jupyter, vHPC, Kubernetes, ADLS, Enablement, Tutorials, and Setup. A user profile icon is visible in the top right corner. The main content area includes a "We are hiring!" notification, three large buttons labeled "Discover", "Develop", and "Deploy" with corresponding icons, and a "Welcome to our Cloud Native Labs" section. The welcome message states: "This project incubates use-cases for standardized cloud platform services for universities. We support both teaching as well as research and offer free trainings to students and staff. Our model:" followed by a collection of icons representing various services and infrastructure. An "Open Disclaimer" button is located at the bottom left of the page.

## Automation Tools

Ansible | Terraform | Operators | Helm | Public Images

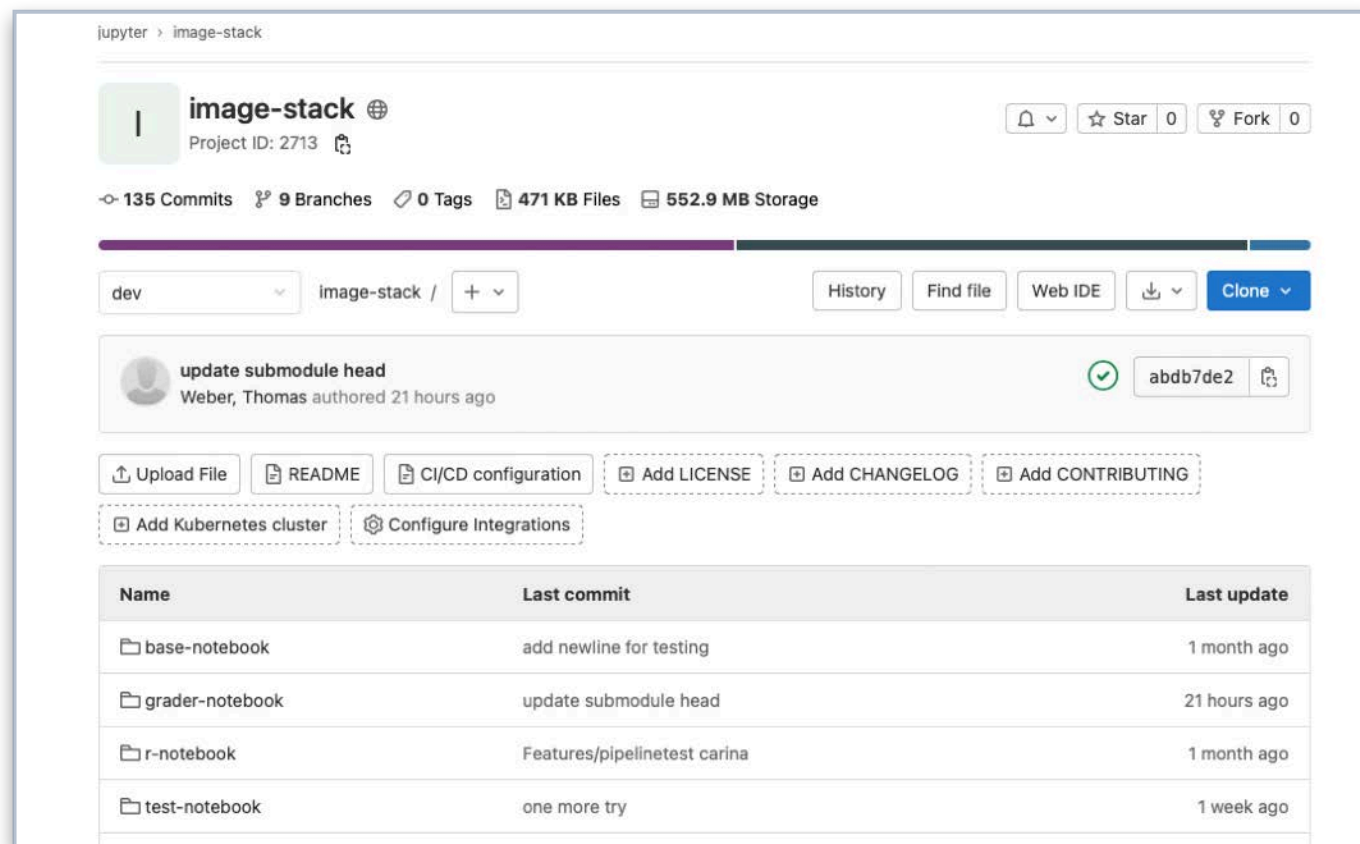
## Public image-stack for all Users

- For UseCase: a teaching lab in Jupyter
- Community contributions
- Take something from the image stack, add to it/modify existing images
- Focus on teaching, don't worry about underlying layers
- <https://gitlab.tuwien.ac.at/jupyter/image-stack>



## Public image-stack for all Users

- Users can create Merge Requests for features, or adapt existing ones
- Pipeline handles build, scanning
- Users can add tests for their code
- Admins verify contribution and can trigger publishing to registries - security measure
- Easy handling for users

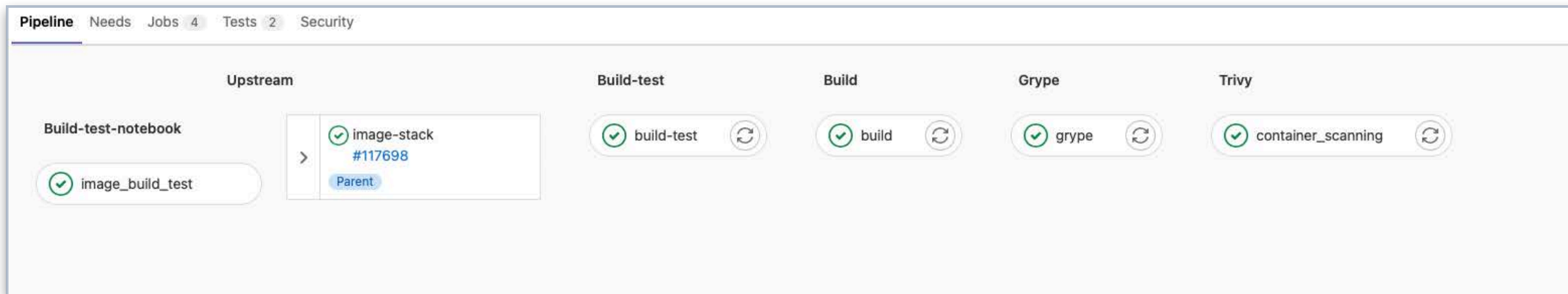


The screenshot shows a Git repository interface for 'image-stack'. The repository is public and has 135 commits, 9 branches, 0 tags, 471 KB files, and 552.9 MB storage. The current branch is 'dev'. A recent commit by Thomas Weber, titled 'update submodule head', is shown with a green checkmark and commit hash 'abdb7de2'. Below the commit, there are buttons for 'Upload File', 'README', 'CI/CD configuration', 'Add LICENSE', 'Add CHANGELOG', 'Add CONTRIBUTING', 'Add Kubernetes cluster', and 'Configure Integrations'. A table lists the repository's contents:

Name	Last commit	Last update
base-notebook	add newline for testing	1 month ago
grader-notebook	update submodule head	21 hours ago
r-notebook	Features/pipelinetest carina	1 month ago
test-notebook	one more try	1 week ago

## Pipeline Output

- Checking the outcome helps to fix something/make modifications
- It visualises what your pipeline is doing



The screenshot displays a pipeline execution interface with the following components:

- Navigation:** Pipeline (selected), Needs, Jobs 4, Tests 2, Security
- Upstream:**
  - Build-test-notebook
    - image\_build\_test (status: success)
  - image-stack #117698 (status: success, Parent)
- Build-test:** build-test (status: success)
- Build:** build (status: success)
- Grype:** grype (status: success)
- Trivy:** container\_scanning (status: success)

## Pipeline Output

- Security Scans
- Alerts

Pipeline Needs Jobs 6 Tests 2 Security

< build-test

2 tests 1 failures 0 errors 50% success rate 1.00ms

**Tests**

Suite	Name	Filename	Status	Duration	Details
test_sample	test_basic2		<span style="color: red;">✘</span>	0.00ms	<a href="#">View details</a>
test_sample	test_basic		<span style="color: green;">✔</span>	1.00ms	<a href="#">View details</a>

Pipeline Needs Jobs 6 Tests 2 **Security**

**Scan details** [Hide details](#)

Container Scanning	1 vulnerability	<a href="#">Download results</a>
Dependency Scanning	0 vulnerabilities	<a href="#">Download results</a>

Severity:  Tool:  [Hide dismissed](#)

<input type="checkbox"/>	Severity	Vulnerability	Identifier	Tool	<input type="checkbox"/>
<input type="checkbox"/>	Unknown	CVE-2018-25032 in zlib-1.2.11-r3 adlsregistrysbx.azurecr.io/jupyter/test-notebook:5766393c8e23d16c194de0202060ea77b0 8f1d67	CVE-2018-25032	Container Scanning GitLab	<input type="checkbox"/>

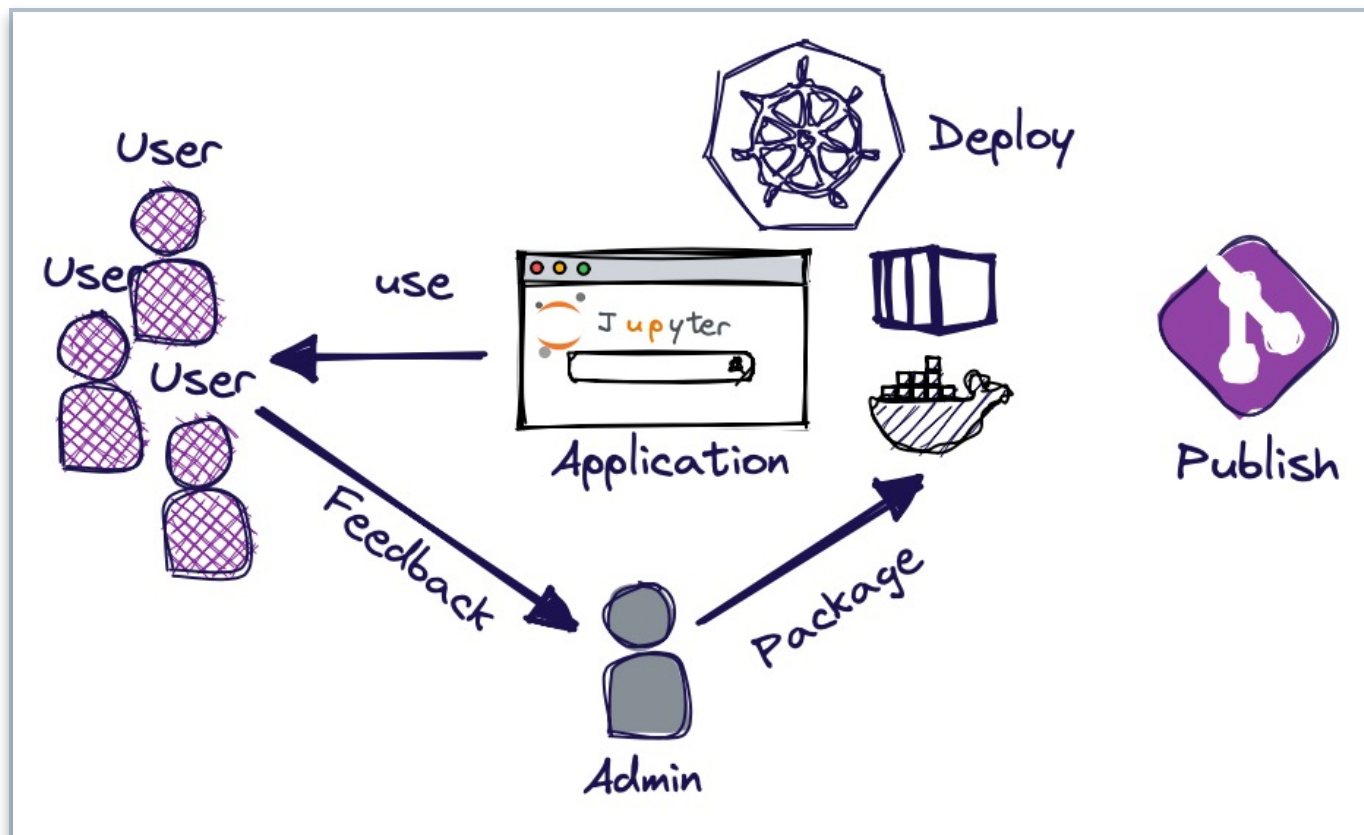


## What still needs to be addressed?

- Liberal access to resources or quota management
- Infrastructure Connection between Universities needs to be established
- Storage/Data sharing between Universities needs to be solved
- Open Source Contribution/Community Guidelines need to be created
- Security implications need to be addressed further
- Observability needs to be improved
- Jupyter Environments still need improvements

Interested? Get in touch!  
We are hiring ;)

# Training and Materials



## Training:

1. Public tech demo, each last Wednesday of the month
2. Hackathons (quarterly) and/or Workshops
3. Tutorials incl short Videos on our Webportal
4. (planned for SoSe 2023) Vorlesung: "Hands-on Cloud Native" on topics like Containers, Kubernetes, Cloud Architecture and Security

## Support:

1. Slack channel\*
2. Issues in gitlab

\*request invite

## Training, Classes & Support ADLS

Monthly demos, Hackathons and an open Chat  
planned 2023: Vorlesung "Hands-on Cloud native"

## Welcome to your Kubernetes Beginner Tutorial

This is aimed at beginners to play with the main components of a k8s deployment. While most components are generic to any k8s, some specific annotations are used.

At the end of this tutorial you will be able to:

- understand and use k8s manifests for the most common components
- understand how to find logs
- understand how to dig through the network
- understand how manifests are different from helm charts
- understand the difference between a statefulset and a deployment
- inspect a secret
- understand the basics of RBAC (Role Based Access Control)
- generate a certificate using cert manager (or by hand, if someone has the patience)
- work with `kubectl` , RancherUI and Lens

### Tutorial Part I: Learn how a k8s-"deployment" works by running Pac-Man

Pac-Man the classic arcade game - modified/upgraded for our Rancher cluster so you may have fun modifying its components.