Analyze your data using DODAS generated cluster

D. Ciangottini, D. Spiga
On behalf of the DODAS team
**Disclaimer**

This demo will **focus on the end user point of view** and will be in two parts:

- A first introduction to the core concepts of DODAS
- A demo showing the main features of the system

The main objectives are:

- Showing the **customization possibilities** for community-specific needs
- Illustrating the currently **supported data access patterns**
- Demonstrate the possibility to bring an integrated system for both **batch and interactive analysis**
Introduction

**Dynamic On Demand Analysis Service**: DODAS

- initially **prototyped within INDIGO-DataCloud project** (2017)
  - a primary use case: to **develop a effective solution for dynamic resource provisioning@CMS** (targeting Opportunistic computing)
- Since then it has been **evolved**:
  - In term of **supported use cases and communities**
    - e.g. interactive analysis platforms integrate with existing batch systems
  - In term of **adopted technologies and architecture**
    - Now sit on top of a generic Kubernetes cluster, see later
Fundamentals behind DODAS

- A solution designed with the goal to enable users to create and provision infrastructure deployments, automatically and repeatedly, on “any cloud provider” with almost zero effort.

- As we saw this sharing common objectives with many cloud-native initiative, we converged toward the creation of a software layer to put on top of “any Kubernetes cluster (as matter of fact a standard for container deployment and orchestration)

- Effectively decoupling the stack deployment from the infrastructure provisioning and promoting the re-use of computing solutions in different context

To lower the bar of sys-admins skills for accessing cloud resources
- automate the complete flow

To facilitate the modernization process
- brings the cloud to the scientific communities

To bridge scientific experiment data and distributed computing
- builds composable and portable clusters
Core concepts 1/2

Using, wherever possible, “de-facto” standards

- Templating ➔ Helm Chart
  - DODAS services == k8s Helm Charts

- DODAS chart dashboard allows for service composition via GUI
  - Adopting bitnami kubeapps
  - deployable via helm chart itself
  - OIDC IAM authN/Z supported

Supported K8s provisioning:
- Indigo PaaS orchestrator
- Public Cloud
- Hand-made cluster
- Local machine cluster
Core concepts 2/2

- **AAI:**
  - INDIGO-IAM is used to implement authentication and authorization
  - allows for a federated model (i.e. Egi Check-In)
  - supports capabilities based AuthZ

- **Compute Resources orchestration**
  - Relies on INDIGO Paas Orch, IM

- **Data Access**
  - XrootD is used to implement caching mechanisms for remote I/O optimization
  - S3 storage access supported via Min.io

- **Software Distribution**
  - CVMFS is the adopted service for distributing software and libraries and user configurations across distributed clusters
Lego-style service combination

In summary **all you need is an Helm chart repository** with your software components:

- **DODAS provides you with an initial one** with the most common needs encountered so far:
  - **On-demand JupyterHUB** instance integrated with a batch system via **HTCondor**
  - On demand software repository mount points via **CVMFS**
  - **S3 storage** via Minio
  - **Data Caching via XRootD** software

We will see in a demo how you leverage and combine this solutions to get a software stack tailored on your needs.
Service combination examples

We are currently supporting the following combination of services requested and adopted by our users:

- Cluster of **HTCondor WNs automatically attached to an existing experiment pool** of resources
- **Interactive user analysis facility** via JupyterHUB instance integrated with a dedicated batch system
- Dedicated **HTCondor cluster with software stack mounted via CVMFS**
- Dedicated **HTCondor cluster with software stack mounted via CVMFS and experiment data read via S3 dedicated storage**
What’s in the demo

Today’s Demo By Steps

- Walkthrough the DODAS Catalog
  - to show authN/Z and how to choose your DODAS preferred app

- let’s instantiate a generic batch on demand
  - how to instantiate a DODAS app

- Interacting with instantiated services
  - how to use the app, i.e. launching a job

- Combine htcondor deployment with jupyterhub deployment
  - Interact with condor from jupyterlab

Today we will go through these
It’s demo time...
What if I need my own customization?

- You can **create your own helm chart** combining the existing ones or from scratch
- **Use your own docker image** in an existing template

Also at any time **you can contact the DODAS team** at dodas-support<at>lists.infn.it for guidance on the integration process of your solutions
Cluster requirements

DODAS has a few **kubernetes requirements** that has to be satisfied:

- Automatic certificate management through **Cert-Manager**
- An **nginx ingress controller** (other kind can be easily integrated)
- A **default StorageClass** (many supported cases use Longhorn)

N.B. every one of these are **already provided by many cloud providers out of the box**