The Platform-as-a-Service paradigm meets ATLAS: developing an automated analysis workflow on the newly established INFN CLOUD

Caterina Marcon[1,*], *Leonardo* Carminati[1], *David* Rebatto[1], and *Ruggero* Turra[1]

[1]INFN Milano

Abstract. The Worldwide LHC Computing Grid (WLCG) is a large-scale collaboration which gathers computing resources from more than 170 computing centers worldwide. To fulfill the requirements of new applications and to improve the long-term sustainability of the grid middleware, newly available solutions are being investigated. Like open-source and commercial players, the HEP community has also recognized the benefits of integrating cloud technologies into the legacy, grid-based workflows.

Since March 2021, INFN has entered the field of cloud computing establishing the INFN CLOUD infrastructure. This platform supports scientific computing, software development and training, and serves as an extension of local resources. Among available services, virtual machines, Docker-based deployments, HTCondor (deployed on Kubernetes) or general-purpose Kubernetes clusters can be deployed.

An ongoing R&D activity within the ATLAS experiment has the long-term objective to define an operation model which is efficient, versatile and scalable in terms of costs and computing power. As a part of this larger effort, this study investigates the feasibility of an automated, cloud-based data analysis workflow for the ATLAS experiment using INFN CLOUD resources. The scope of this research has been defined in a new INFN R&D project: the INfn Cloud based Atlas aNalysis faciliTy, or INCANT.

The long-term objective of INCANT is to provide a cloud-based system to support data preparation and data analysis. As a first project milestone, a proof-of-concept has been developed. A Kubernetes cluster equipped with 7 nodes (total 28 vCPU, 56 GB of RAM and 700 GB of non-shared block storage) hosts an HTCondor cluster, federated with INFN's IAM authentication platform, running in specialized Kubernetes pods. HTCondor worker nodes have direct access to CVMFS and EOS (via XRootD) for provisioning software and data, respectively. They are also connected to a NFS shared drive which can optionally be backed by an S3-compatible 2 TB storage. Jobs are submitted to the HTCondor cluster from a satellite, Dockerized submit node which is also federated with INFN's IAM and connected to the same data and software resources. This proof-of-concept is being tested with actual analysis workflows.

[*]e-mail: caterina.marcon@mi.infn.it

1 Introduction

Historically, the ATLAS experiment [1] has been relying on a complex and distributed computing infrastructure: the Worldwide LHC Computing Grid (WLCG), powered by almost a million CPU cores, with an exabyte-size storage capacity, all distributed in different sites worldwide.

With the upcoming HL-LHC era, the ATLAS infrastructure will face an unprecedented challenge in terms of storage and computing power. The collaboration is considering integrating various alternative computing resources into the existing ecosystem, including cloud-based technologies for dynamic, flexible and cost-effective resource provisioning [2].

This work is a pioneering study, part of the research and development activities promoted by ATLAS to explore the cloud's technological landscape with the aim of defining a cost-effective, time efficient, scalable and flexible operating model.

1.1 INFN CLOUD infrastructure

The INCANT project described in this article has been developed on the newly established INFN CLOUD infrastructure [3] which is in production since March 2021 and offers a set of cloud computing services to the INFN community. It is based on a core backbone connecting the large data centers of CNAF and Bari, and on several smaller federated sites that offer opportunistic resources. All centers are interconnected at very high bandwidth through the GARR network [4].

Resources are orchestrated by *OpenStack* and available in two different operation models:

1. Platform-as-a-Service (PaaS), where users are responsible for maintaining their data and applications;

2. Software-as-a-Service (SaaS), where the provider is responsible for the full virtualization and application stack, and the user is responsible only for the data processing.

The INFN CLOUD infrastructure comprises a set of services, available via a user-friendly Dashboard or via command line utilities, capable of fulfilling a broad range of requirements of a diversified audience of users:

1. compute services such as virtual machines, docker-compose instances and different types of clusters such as Apache Mesos, Kubernetes and HTCondor [5–7];

2. ad-hoc solutions for data analysis, such as Elasticsearch and Kibana, Spark and Jupyter clusters, Rstudio;

3. a collection of ready-to-use machine learning solutions based on the python ecosystem;

4. data management and data storage services;

5. highly customizable environments, with optional GPU support, for in-browser code development in different programming languages;

6. POSIX clients and REST APIs for direct access to data.

These services are managed at the infrastructure level by *OpenStack* based on a set of predefined TOSCA templates [8] and can be enabled or disabled individually on a per-project basis.

All PaaS and SaaS services are accessible via INFN's federated authentication and authorization INDIGO-IAM system, fully compliant with European Open Science Cloud (EOSC) and industry standards.

1.2 Long-term objectives of INCANT

The high-level purpose of the project is to investigate the possibility of implementing two distinct analysis workflows:

1. create a batch-like system capable of processing structured and complex data into flat n-tuples compatible with analysis workflows for result extraction or toys generation. Input and output data must be read/written to and from CERN's EOS storage service [9] or other distributed storage locations accessible via XRootD and webDAV protocols and, in general, suitable for POSIX operations (either directly or via a client);

2. develop an interactive analysis platform similar to Jupyter's Notebook-as-a-Service. Usually, this step follows the previous one: once flat n-tuples are produced, the actual analysis can be performed (e.g. with python scripts using *uproot* and *pandas*).

This article outlines the development phase of the first workflow which aims to create a ready-to-use analysis facility, shared among multiple users of the same experiment.

2 The INCANT proof-of-concept

INCANT extends the existing PaaS components available on the INFN infrastructure. In this paradigm, users maintain both their data and analysis applications, while providers deal with the underlying infrastructure maintenance. In Figure 1, the typical layers of a cloud infrastructure are outlined and the PaaS-specific responsibilities are identified. CERN and ATLAS software for user and data management has been added at the Middleware and Runtime layers, while leaving the top user layers unchanged.

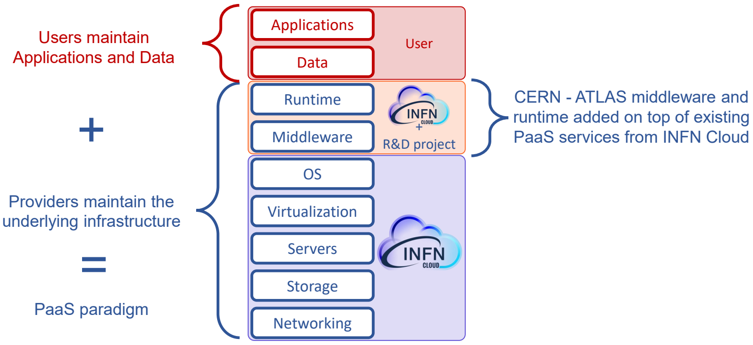
Users maintain Applications and Data
+
Providers maintain the underlying infrastructure
=
PaaS paradigm
Applications
Data
User
Runtime
Middleware
INFN CLOUD + R&D project
OS
Virtualization
Servers
Storage
Networking
INFN CLOUD
CERN - ATLAS middleware and runtime added on top of existing PaaS services from INFN Cloud

Figure 1: PaaS paradigm identifying the activities of the INCANT R&D effort.

2.1 INCANT building blocks

A typical ATLAS analysis workflow relies at least on three fundamental services:

1. the CernVM FileSystem (CVMFS) software distribution service [10];

2. the EOS Open Storage (EOS), to store and share LHC data (real and simulated);

3. a federated authentication and authorization infrastructure to ensure secure access to data. This can be based on the Kerberos protocol or on VOMS certificates.

These services are natively provided within the CERN infrastructure but they can be made available externally. The typical usage pattern is a native installation of the required clients on the host, but supported operating systems are limited and, in view of a cloud-based infrastructure, a template-based virtual machine (VM) provisioning of on-demand resources limits the scalability of the infrastructure and the opportunity to dynamically allocating the optimal amount of computing power for each given workload. On the other hand, Docker containers impose less restrictive requirements on the host systems and allow an easier integration with industry-standard solutions, which are designed for optimal resource management.

For this reason, a set of ATLAS-enabled Docker images, capable of providing the three essential services, have been integrated with a Dockerized HTCondor environment and deployed to a Kubernetes cluster running on VMs drawn from the INCANT resource pool.

2.2 The R&D resource pool

The entire resource pool available to INCANT is summarized in Table 1. The scale of the pool is intended for an audience of 5 concurrent test users.

Table 1: INCANT R&D resource pool

vCPU	92
POSIX Storage	1000 GB
RAM	168 GB
External storage (compatible with S3)	2048 GB

INCANT, via the INFN CLOUD dashboard has access to a predefined set of applications that can be instantiated from the pool, such as pure Kubernetes clusters, HTCondor clusters deployed on Kubernetes, general purpose Virtual Machines (with Ubuntu 18.04, Ubuntu 20.04 or CentOS 7) and an S3-compatible storage. The scale of these applications is configurable by the developer, even though full control on the TOSCA templates is not available.

2.3 Running HTCondor on top of a Kubernetes cluster

From the whole INCANT resource pool (Table 1), a set of 8 VMs was drawn (each with 4 vCPU, 8 GB of RAM and 100 GB of block storage).

A Kubernetes cluster (Figure 2), equipped with 7 nodes (total 28 vCPU, 56 GB of RAM and 700 GB of non-shared block storage) has been created via the INFN CLOUD Dashboard. The cluster template is pre-configured to use Ubuntu 20.04 as the operating system on the underlying host VMs.

The Kubernetes instance hosts an HTCondor cluster, federated with INFN's IAM authentication platform, running in specialized Kubernetes pods. Basic monitoring services for the Kubernetes cluster's lifecycle, such as a Grafana dashboard with Prometheus, are configured and provided by default. The developer, when creating a new cluster instance, has the opportunity to select the size of the VMs hosting the Kubernetes master and slave nodes and the Docker image to use for the HTCondor worker nodes.

All the HTCondor components run as Docker containers and are based on CentOS 7 images; their lifecycle is managed by Kubernetes deployments. Each deployment can control several replicas; for this proof-of-concept, a single central manager and a single scheduler are configured and a total of 7 worker node replicas are instantiated (Figure 3).

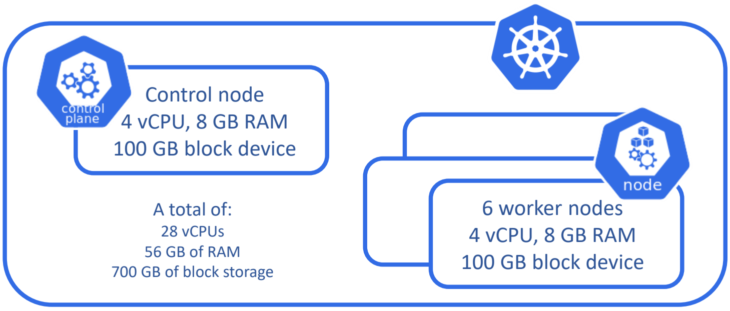

Figure 2: Kubernetes cluster layout for the INCANT proof-of-concept.

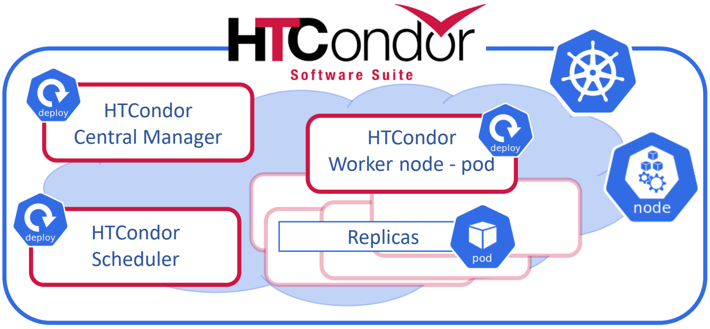

Figure 3: Layout of the HTCondor cluster running on the Kubernetes instance.

By design, this layout does not include the HTCondor submit node, which is instead intended to run as a satellite Docker container (Figure 4) to allow remote submission of jobs. This layout allows great flexibility but imposes additional requirements on the submit node installation, as it also needs an interface to the INFN and CERN user authentication and authorization infrastructures and to all the required software and data provisioning services. The submit Docker container, based on CentOS 7, has been deployed to an additional VM running Ubuntu 20.04. This choice is intended to test the functionality in heterogeneous systems.

The network configuration allowing the communication between the Kubernetes/HTCondor cluster and the satellite submit node is pre-configured by default via the TOSCA template and cannot be customized by the developer.

2.4 Merging CERN and INFN resources

As discussed in Section 2.1, ATLAS-specific requirements need to be integrated with the HTCondor cluster in order to allow software and data access to users.

The default Docker images of the HTCondor worker and scheduler nodes have been extended to include (Figure 5 (left)): the configuration for the CVMFS software distribution system, Kerberos and VOMS clients to allow user authentication to XRootD-enabled resources (either via Kerberos tokens or X509 certificates) and NFS clients to access a cluster-wide shared filesystem (see Section 2.5). It is important to stress that CVMFS and NFS are

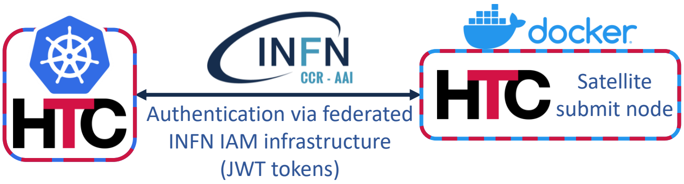
INFN
CCR - AAI
Authentication via federated INFN IAM infrastructure (JWT tokens)
docker
HTC
HTC
Satellite submit node

Figure 4: Detached HTCondor submit node running in a remote Docker container.

accessed as POSIX drives and the automatic handling of these folders is done by AutoFS, which in turn requires the Docker containers to run in privileged mode. The XRootD servers, although suitable for a POSIX usage, are usually operated in a REST-like fashion, based on atomic query-response interactions.

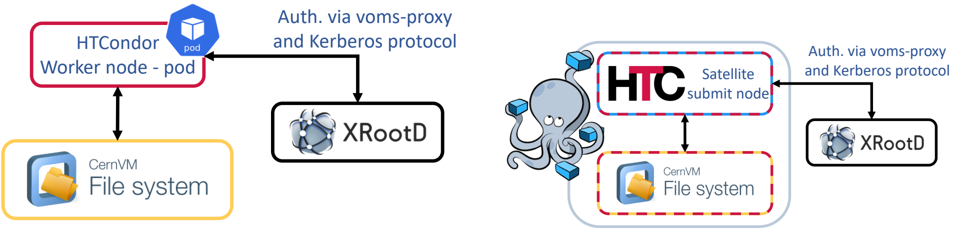

Figure 5: Integration with CERN services: worker node (left), submit node (right).

The same requirements hold for the submit node, but the implementation has been achieved in a different way. The Docker image of the submit node, initially developed with INFN CLOUD, has been extended to add the Kerberos and VOMS authentication backend.

An additional decoupling, to improve security and robustness, has been implemented for the CVMFS client. Instead of adding the AutoFS dependencies directly to the submit image, the official `cvmfs/service:2.10.1-1` Docker image has been used (Figure 5 (right)).

As outlined in Figure 6, the CVMFS container connects to the CVMFS servers and mounts the `/cvmfs` path onto the underlying host as a standard Docker bind mount. The path is then mounted back into the submit container also as a standard bind mount. This interaction is orchestrated by Docker Compose.

This layout ensures a better separation between the CVMFS privileged container and the non-privileged container intended for users' activities. In addition, the requirement of a native CVMFS installation is removed, allowing a more flexible, cloud-friendly, management.

2.5 External storage integration

While the integration with CERN storage systems, such as CVMFS and EOS, can be achieved in several ways with existing drivers and components, the choice of a suitable shared filesystem for the internal INCANT metabolism (users' home folders, temporary folders, etc.) has

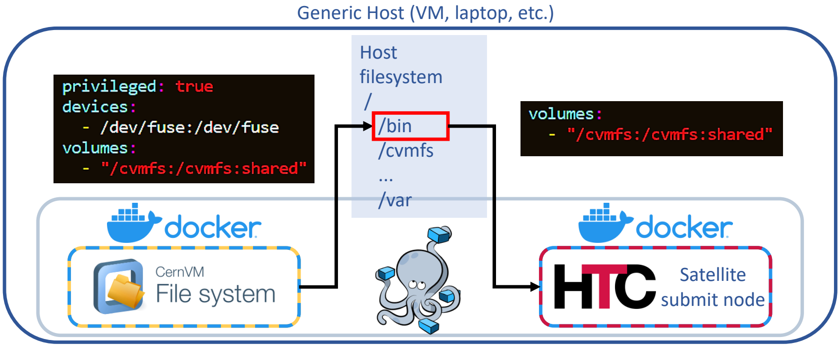

Figure 6: Double-container mount strategy of the CVMFS filesystem.

not been finalized yet. Several constraints must be considered and those already identified are outlined below (Figure 7):

1. purpose: a shared storage intended for long-term archiving has different requirements from a low-latency drive used for intense computing;
2. ease of access and configuration: nodes running on the Kubernetes cluster don't allow interactive access and must not require constant supervision. Therefore authentication workflows must be self-sustaining and, if token-based, capable of handling token refresh automatically, either on a schedule or lazily upon users' requests;
3. possibility of remote access: the submit node is outside the Kubernetes/HTCondor fence but must nevertheless be able to access the shared resources seamlessly. This entails that it is necessary to secure the connection of the submit node to the INFN CLOUD resources.

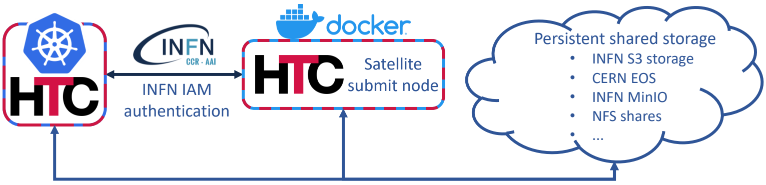

Figure 7: High-level layout of the external storage integration.

As a first step, the following configuration is being tested:

1. a VM with Ubuntu 20.04, the *storage controller*, has been configured as a NFS server;
2. the storage controller is also equipped with RClone [11], which is connected to a 2 TB, S3-compatible bucket, hosted on INFN resources. The S3 storage system also provides a web interface;
3. all cluster nodes, as well as the remote submit node, can natively mount the NFS share as a POSIX drive by means of standard Linux libraries.

This architecture has advantages in terms of ease of configuration, as NFS is a well-known protocol available on the large majority of Linux distributions. The main drawback, as anticipated, is the need for a Docker container running with privileged mode enabled and poses, therefore, an additional security concern.

In addition, the S3 bucket connected via the RClone client is better suited for high-latency storage service and is therefore more appropriate for archiving files, rather than for jobs with high rates of read/write operations.

The functionality of the HTCondor cluster when operating without a shared filesystem is also being evaluated, given the particular layout of the Kubernetes/HTCondor cluster operating with a detached submit node.

3 Conclusions and outlook

The increasing need for computing resources foreseen for the HL-LHC era is accelerating the deployment of analysis workflows on distributed and cloud computing systems. In 2021 INFN joined this effort with the new INFN CLOUD infrastructure.

The new R&D project INCANT extends the existing PaaS with ATLAS data analysis capabilities to support batch and interactive workflows, ensuring an optimized use of dynamically allocated resources.

So far, a working prototype supporting batch-like workflows has been implemented and its main components are identified: a Kubernetes cluster hosting an HTCondor cluster with a detached submit node. Investigations are ongoing to identify an optimal storage backend.

Actual analysis workflows are being tested and the prototype will soon be shared with a restricted pool of selected test users, in order to identify the most critical aspects and further refine the development roadmap.

Acknowledgments

This R&D project is funded by INFN Milano, INFN CLOUD and supported by ATLAS Italia Calcolo.

References

[1] ATLAS Collaboration, JINST **3**, S08003 (2008), doi:10.1088/1748-0221/3/08/s08003

[2] P. Calafiura, J. Catmore, D. Costanzo, A. Di Girolamo, Tech. rep., CERN (2020), CERN-LHCC-2020-015; LHCC-G-178, https://cds.cern.ch/record/2729668

[3] *INFN CLOUD Infrastructure*, https://www.cloud.infn.it/

[4] *GARR Consortium*, https://www.garr.it/en/

[5] *Docker*, https://www.docker.com/

[6] *Kubernetes documentation*, https://kubernetes.io/docs/home/

[7] *HTCondor*, https://htcondor.org/documentation/htcondor.html

[8] *TOSCA - OASIS Topology and Orchestration Specification for Cloud Applications* (2021), https://wiki.oasis-open.org/tosca/FrontPage

[9] *EOS Open Storage*, https://eos-web.web.cern.ch/eos-web/

[10] *CVMFS - The CernVM File System*, https://cernvm.cern.ch/fs/

[11] *Rclone - Synchronization for cloud storage*, https://rclone.org/