The DODAS Experience on the EGI Federated Cloud

Daniele Spiga[1,*] *Enol Fernandez*[2], *Vincenzo Spinoso*[3], *Diego Ciangottini*[1] , *Mirco Tracolli*[1,4], *Giacinto Donvito*[3], *Marica Antonacci*[3] , *Davide Salomoni*[5], *Andrea Ceccanti*[5], *Doina Cristina Duma*[5], *Luciano Gaido*[6]

[1]INFN Sezione di Perugia, Via Alessandro Pascoli 23c, 06123 Perugia (ITALY)
[2]EGI Foundation Science Park 140, 1098 XG Amsterdam (NETHERLANDS)
[3]INFN Sezione di Bari, Via Giovanni Amendola, 173, 70126 Bari (ITALY)
[4]Università degli studi di Firenze, Viale Morgagni, 67/a - 50134 Firenze (ITALY)
[5]INFN-CNAF, Viale Carlo Berti Pichat, 6/2, 40127 Bologna (ITALY)
[6]INFN Sezione di Torino, Via Pietro Giuria, 1, 10125 (ITALY)

Abstract. The EGI Cloud Compute service offers a multi-cloud IaaS federation that brings together research clouds as a scalable computing platform for research accessible with OpenID Connect Federated Identity. The federation is not limited to single sign-on, it also introduces features to facilitate the portability of applications across providers: i) a common VM image catalogue VM image replication to ensure these images will be available at providers whenever needed; ii) a GraphQL information discovery API to understand the capacities and capabilities available at each provider; and iii) integration with orchestration tools (such as Infrastructure Manager) to abstract the federation and facilitate using heterogeneous providers. EGI also monitors the correct function of every provider and collects usage information across all the infrastructure. DODAS (Dynamic On Demand Analysis Service) is an open-source Platform-as-a-Service tool, which allows to deploy software applications over heterogeneous and hybrid clouds. DODAS is one of the so-called Thematic Services of the EOSC-hub project and it instantiates on-demand container-based clusters offering a high level of abstraction to users, allowing to exploit distributed cloud infrastructures with a very limited knowledge of the underlying technologies.This work presents a comprehensive overview of DODAS integration with EGI Cloud Federation, reporting the experience of the integration with CMS Experiment submission infrastructure system.

1 Introduction

The EGI Federated Cloud [1] consists in a network of both private and public clouds integrating resources together to provide a scalable computing platform for research.

[*] Corresponding author: spiga@infn.it

It consists of a federation of over 200 computing and data centres spread across Europe. It relies on 47 Countries, 71000 users, 12 Integrated e-Infrastructures, and provides more than 1 million cores through the High Throughput Compute service, 450PB of Online Storage, 380PB of Archive Storage. The EGI Federated Cloud is continuously evolving and including new providers through a certification process provided by EGI Operations. Currently, there are 21 providers in production, and 2 new providers finalising the integration process.

Heterogeneous cloud providers are federated using a single authentication and authorisation software layer, running on top of the local cloud; in this way, workloads can be easily moved across multiple providers that are using different technologies, bringing computing to data. Resource centres of the federated infrastructure operate a Cloud Management Framework (CMF) according to their own preferences. In particular, EGI provides ready-to-use software components to enable the federation for OpenStack and OpenNebula. These components rely on public APIs and use EGI Check-in, the EGI Authentication and Authorisation Infrastructure based on the OpenID Connect standard protocol, for authentication.

In this contribution we describe how we integrated EGI FedCloud and DODAS (Dynamic On Demand Analysis Service) [2], an open-source Platform-as-a-Service tool implementing the infrastructure as a code paradigm, which allows to deploy software applications over heterogeneous and hybrid clouds. In Sec.2 we describe the EGI service portfolio and the foreseen strategy to bring in new scientific community pilots. In the Sec.3, we provide a brief architectural overview of DODAS. In Sec.4, we detail the integration strategy with EGI compute resources and we describe the setup of the pilot testbed. Finally, Sec.5 summarizes the obtained results.

2 EGI service and portfolio and community pilots

EGI delivers its federated computing and storage capacity through a set of services listed in Fig. 1. These services include an IaaS cloud to run compute or data-intensive tasks and host online services in virtual machines or docker containers. It supports high-throughput data analysis to run compute-intensive tasks for producing and analysing large datasets and store/retrieve research data efficiently across multiple service providers. Moreover higher level platforms like Notebooks allow users to perform interactive computations from a web browser. The computing services are complemented with storage services that enable access to online and data, archiving the data and moving the data across the infrastructure. All the services rely on EGI Check-in [3] to enable a single sign-on experience on the infrastructure.

For the activity reported in this paper, we rely on the EGI Cloud Compute and the Check-in authentication service. The EGI Cloud Compute Service consists of an IaaS federation layer that provides resource discovery, central Virtual Machine (VM) image catalogue, usage accounting and monitoring. In particular, users can instantiate VMs on the providers from a set images available on a central catalogue implemented in the AppDB's Cloud Marketplace [4]. Images are synchronised to the federated providers periodically using the HEPiX image lists format. Usage of resources is gathered centrally using EGI Accounting repository and made available for visualisation on the EGI Accounting portal.
The baseline solutions, available today, for users and community platforms to build on top of the EGI IaaS are:

- Directly using the IaaS APIs to manage individual resources. This option is recommended for pre-existing use cases with requirements on specific APIs.

- Using the AppDB VMOps dashboard, a web-based GUI that simplifies the management of VMs on any provider of the EGI infrastructure. AppDB VMOps in turn relies on the Infrastructure Manager.
- Using IaaS Federated Access Tools that allow managing the complexity of dealing with different providers in a uniform way. These tools include:
    - IaaS provisioning systems that allow to define infrastructure as code and manage and combine resources from different providers, thus enabling the portability of application deployments between them (e.g. IM or Terraform);
    - cloud brokers, that provide matchmaking for workloads to available providers (e.g. the INDIGO-DataCloud Orchestrator [5]).

2.1 Bring new pilots to the EGI Federated Cloud

Whenever a new use case expresses interest in the EGI services, EGI's support team follows a well-defined process that allows users to port their applications into the infrastructure shown in Fig.2 . After the initial contact, EGI helps users to identify the best strategies to integrate the application and to define a complete work-plan. To allow a fast access to the infrastructure, EGI has reserved a set of resources on every provider for prototyping activities. This allocation is accessible through a dedicated Virtual Organisation (VO) which any user with an account on EGI Check-in can join for a 6-month period. This allocation offers opportunistic, non-guaranteed resources for piloting activities guided by the EGI support team. Once the user successfully completes its integration work and expresses interest to start to use the service in production, the support team provides assistance to establish a Service Level Agreement (SLA) backed up by Operational Level Agreements (OLA) with the commitment from a set of providers of the federation to support the use case with resources. The services offered in the SLA are accessed through a new production VO enabled in EGI Check-in.

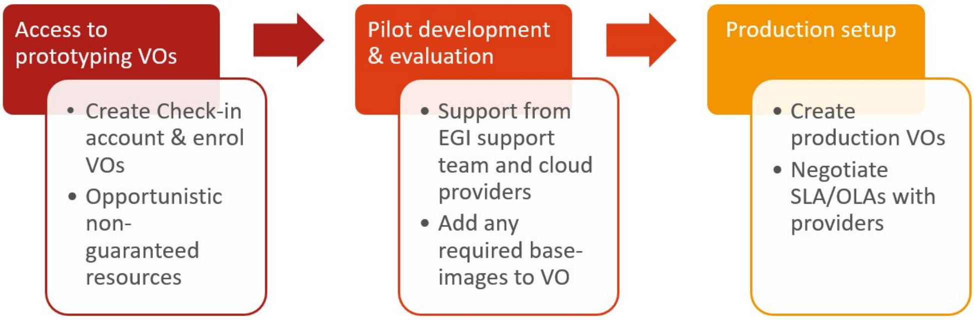
Access to prototyping VOs
• Create Check-in account & enrol VOs
• Opportunistic non-guaranteed resources
Pilot development & evaluation
• Support from EGI support team and cloud providers
• Add any required base-images to VO
Production setup
• Create production VOs
• Negotiate SLA/OLAs with providers

Fig. 1. The schema summarizes the EGI support process for the inclusion of new use cases.

3 DODAS overview

DODAS is a Thematic Service of the EOSC-hub EU project [6]. It is an open source toolkit originally developed within the INDIGO-DataCloud EU project [7] in order to integrate and to support the computing system of the CMS [8] experiment at CERN with the primary objective to opportunistically exploit Cloud resources. Since then, DODAS evolved both in terms of functionalities offered and in terms of scientific communities supported, becoming nowadays an experiment agnostic cloud enabler for scientists seeking to easily exploit distributed and heterogeneous clouds to process, manipulate or to generate simulated data. One of the major drivers of DODAS is to reduce as much as possible the learning curve, as well as the operational costs of managing community specific services, while running on Cloud. As such DODAS guarantees the deployment of complex and intricate setups on "any cloud provider" with almost zero effort and, to this end, the adopted strategy has been to implement the infrastructure as code paradigm based on a templating engine to specify high-level requirements.

3.1 Architecture of DODAS: brief overview

From the architectural perspective DODAS is built on top of a set of core components that have the role of abstracting the underlying infrastructure (the IaaS), providing a common layer for the Authentication and Authorization, including the support for legacy authentication mechanisms. The templating engine is based on TOSCA [9] which is used by the PaaS abstraction layer. Namely the core services are :

- Infrastructure Manager [10]
- INDIGO IAM [11]
- PaaS Orchestrator

The value of DODAS lies in the integration of the end user workflow, possibly including end-users and/or community specific services, together with on-demand container orchestrator following a high level templating approach. At the time of writing there are a number of scientific communities exploiting DODAS such as, CMS, AMS[12], Fermi-LAT[13], Virgo[14], at different levels of maturity.

From the applications perspectives there are a number of generic implementations supporting several use cases, namely

- Big Data Pre-Post processing
    - HDFS, Spark
- Exploitation of Machine Learning
    - Training facility, Inference engine
- Batch System as a Service
    - HTCondor batch system, HTCondor federation solutions
- Data Caching as a Service
    - XRootd federation on demand

3.2 DODAS and container orchestration

As introduced, DODAS manages container based applications and thus it relies on container orchestration for application deployment and management. Originally the architecture was built on top of Mesos and Marathon but, at the time of writing, all the applications mentioned in Sec.3.1 have been migrated to Kubernetes. The latter has been integrated in the DODAS workflow in order to guarantee two main aspects as detailed below:

- The creation of a plain Kubernetes cluster through a configurable Ansible role that takes care of all the needed steps from the initialization with "kubeadm" tool until the deployment of the kubernetes dashboard or metric server.
- The deployment of the applications that leverages the templating capabilities of Helm[15]. A full integration of Helm with TOSCA is provided although the

capability to run Helm on top of any pre-existing kubernetes instance has also been preserved. The applications are structured in such a way that, through the very same base template structure, different flavors of the same cluster can be deployed. For instance one can activate a certain type of shared filesystem to be used by putting a flag at Helm configuration level (so called "Helm values"). In addition multiple applications can be combined as needed with the Helm dependency system, where the child application will wait for the parent to be completely deployed before starting its own installation.

The Helm charts integration in the TOSCA template has been possible thanks to the usage of Ansible roles which take care of compiling Helm values only when the cluster has been automatically created and thus all the parametrized information are known. All the produced charts are documented following the best practices adopted by official projects in the Helm, so that anyone interested can easily fix or add features to the existing charts.

For the use case described in this article DODAS was used to generate a WLCG [16] site on top of a Kubernetes cluster, for the CMS experiment. This is used as a benchmark use case for the exploitation of EGI FedCloud.

4 DODAS and EGI FedCloud integration

The Fig. 2. shows how DODAS has been adopted as a PaaS federation layer on top of EGI Cloud IaaS federation. As mentioned, the actual integration relies on two services offered by EGI Portfolio: the EGI Cloud Compute and the Check-in.

The first step was the integration of the Authentication and Authorization flow at three levels: access to DODAS core service; access to CMS specific computing services (namely the global pool); access to the IaaSes belonging to the EGI Federation. Two distinct authorization services are involved: the EGI Check-in and the INDIGO-IAM. Both services provide Single Sign-On through EduGAIN, social media and other institutional or community-managed identity providers, Harmonised authorisation information, aggregated from multiple sources, based on the VO concept and Industry Standard OpenID Connect technology allowing web and non-web access to services.

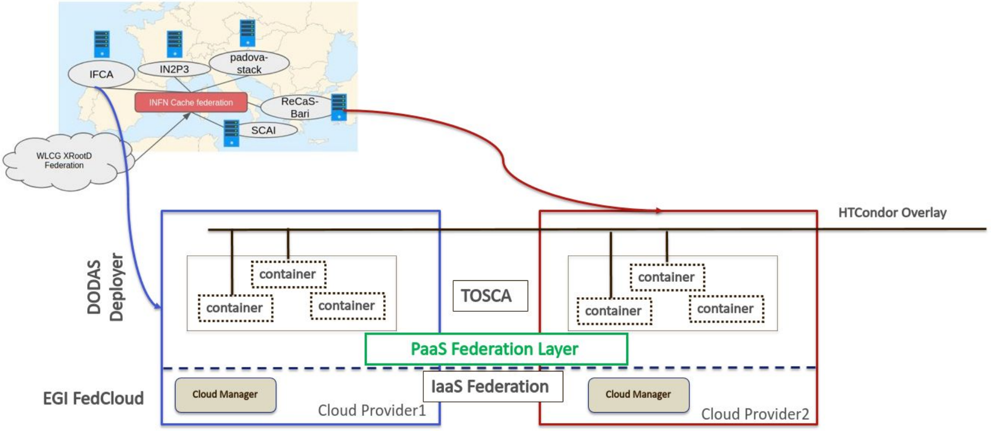

Fig. 2. The figure shows the topology of the 4 sites used in the testbed. Below the schema represents how DODAS integrates with Egi FedCloud. Finally the overlay HTCondor on top of the DODAS deployed sites is represented.

Due to the fact that DODAS integrates nicely with IAM for user authentication, authorization and token translation needs and already grants access to EGI Check-in identities we decided to use it as harmonisation solution. The IAM is then used to manage CMS domain-specific security to cope with HTCondor security via GSI as explained in the next section Sec.4.1.
The second step has been to instruct the DODAS core services to interact with public APIs of the IaaSes belonging to the EGI Federation. This, in turn, made it possible to automate the provisioning of the virtual resources and, in afterwards, to automatically configure the cluster including the applications on top of it. As mentioned, Kubernetes was used as orchestrator to execute the WLCG site for CMS over EGI providers.

4.1 The implemented testbed

The pilot testbed has been built exploiting computing resources from 4 distinct IaaS providers belonging to EGI FedCloud namely: IFCA-LCG2 (Spain), INFN-PADOVA-STACK (Italy), RECAS-BARI (Italy) and IN2P3-IRES (France)**.** All the 4 selected IaaSes have been accessed by DODAS to provision and setup the CMS virtual site, automatically integrated with the Workload Management system of the Experiment.

The CMS Workload Management system is built over a Global Pool by meaning a single HTCondor pool covering all GRID computing processing resources [17]. The production system of CMS relies on GlideinWMS [18] which has the knowledge of all the sites and thus has the responsibility to dispatch glideins (the pilots) which then connect back to the global pool and fetch payload upon successful matchmaking. The DODAS strategy is to implement a model that de facto bypasses the resource provisioning glidein WMS based.

From a technical point of view each provider has been accessed by DODAS deployer in order to automatically instantiate and configure both the Kubernetes and the application. The latter consists of the set of services required by a regular CMS Site namely: CVMFS, squid proxy, Json Web Token - X.509 certificates cache service, renewal mechanisms and, of course properly configured HTCondor startd daemon. These daemons auto-register to the global-pool using X.509 certificate provided by INDIGO-IAM as stated above. The auto-started HTCondor demons are implemented as Kubernetes pods referred to as WorkerNodes pods. Jobs pulled from the global pool are then executed within the WorkerNode pods. Finally regarding data ingestion, the virtual infrastructure is configured as storage-less sites, as such input is managed through the XrootD [19] federation of CMS while the output is copied to the remote location via srm protocol, according to the job configuration. For the described pilot the actual data access happened through a XCache system deployed at INFN [20] acting as an intermediate layer for the remote data access.

Concluding, the obtained result was to get automatically configured four independent Kubernetes clusters executing four twisted setups for all the auxiliary services. HTCondor daemon was configured to use the very same logical site name: *T3_IT_Opportunisitc_dodas*. This allows CMS to see the 4 distributed clusters as a single logical entity.

5 Results

The CMS virtual infrastructure, implemented through DODAS over the EGI Federated Cloud providers, has been successfully tested and integrated in the CMS computing system. CMS monitoring system collects all the monitoring information and shows them as under a single site named T3_IT_Opportunistic_dodas. In order to finalize the integration several runs of tests were performed starting with a single cluster first and then testing the addition of a cluster each time, to validate the complete setup. Several days-long tests were done and the longer one lasted weeks and had all the providers running together. An average of 400 running jobs were continuously executed during the two weeks. The EGI Cloud accounting system registered the usage of the DODAS deployment on the 4 different providers as shown in Fig. 3. Peaks in the figure are related to initial tests of the deployments on the providers, due to the granularity of the accounting data, the number of allocated CPUs is accumulated over a single day even if these are not used concurrently. Once the test phase ended, DODAS was using approximately 400 concurrent CPUs for executing CMS jobs (matching the average 400 jobs).

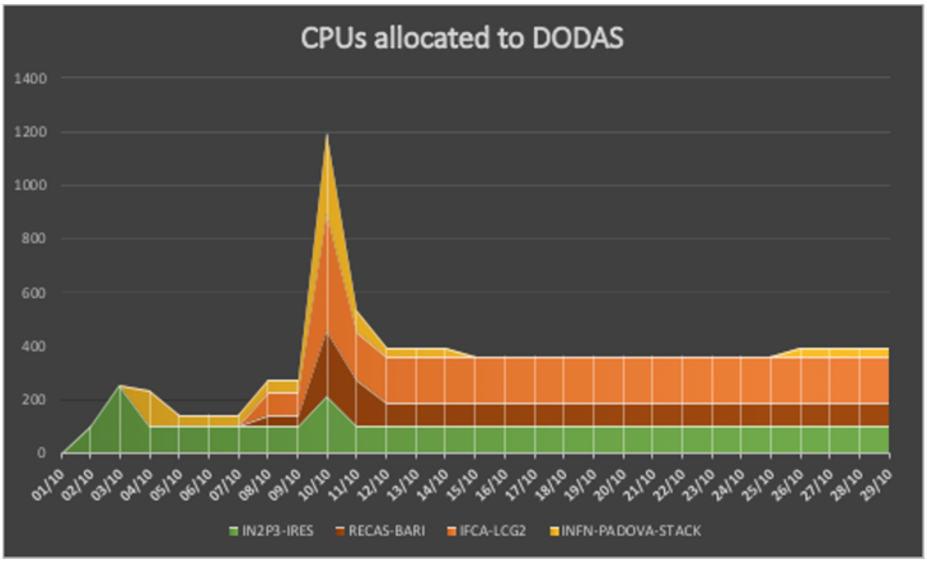

Fig. 3. CPUs allocated to DODAS during the tests.

6 Conclusions and future directions

DODAS deployer has been used to exploit EGI compute resources; a successful test was done implementing a virtual site over 4 distinct providers. CMS Analysis Workflow was used as a demonstrator and the full integration was successful. To move to a production setup further activities must be done such as the consolidation of the data ingestion flow, perhaps integrating another layer of cache close to each instance of the cluster. Another aspect we’ll take into consideration will be to improve the accounting and the monitoring: although there is detailed accounting information on the resources usage at each provider, the complete information is not accessible on the currently available dashboards and understanding the actual consumption of a given use case requires manual extraction of information from accounting repository.

Overall the experience done has been very useful and demonstrated that from a technical perspective there is no show stopper for a more consolidated integration. DODAS can act as an effective PaaS federation layer on top of EGI Cloud Federation. This actually means that all the scientific communities and the use cases supported by DODAS could be deployed over EGI Federated Cloud.

This work has been partially supported by the EOSC-hub EU project G.A 777536.

References

1. E. Fernandez del Castillo, D. Scardaci, A. Lopez Garcia, "The EGI Federated Cloud e-Infrastructure", Procedia Computer Science 68 (2015) 196-205, https://doi.org/10.1016/j.procs.2015.09.235
2. D. Spiga et al. Sep.2019, Exploiting private and commercial clouds to generate on-demand CMS computing facilities with DODAS, https://doi.org/10.1051/epjconf/201921407027
3. EGI-CheckIn project [online] https://wiki.egi.eu/wiki/AAI [Accessed 23 June 2020]
4. EOSC-hub project. [online] Available at https://eosc-hub.eu/ [Accessed 10 March 2020]
5. INDIGO-DataCloud PaaS Orchestrator [software]. Version 2.2.0. Available from https://github.com/indigo-dc/orchestrator/releases/tag/v2.2.0-FINAL
6. https://egi-federated-cloud.readthedocs.io/en/latest/vmi.html
7. Salomoni, D., Campos, I., Gaido, L. et al. INDIGO-DataCloud: a Platform to Facilitate Seamless Access to E-Infrastructures, J Grid Computing (2018) 16: 381. https://doi.org/10.1007/s10723-018-9453-3
8. Chatrchyan S and (CMS Collaboration) 2008 The CMS experiment at the CERN LHC JINST 3 S08004
9. Palma, D., Rutkowski, M., Spatzier T.: TOSCA Simple Profile in YAML Version 1.1. Tech. rep., OASIS Standard. http://docs.oasis-open.org/tosca/TOSCA-Simple-Profile-YAML/v1.1/TOSCA-Simple-Profile-YAML-v1.1.html (2016) [Google Scholar]
10. Caballer, M., Zala, S., García, Á. et al. J Grid Computing (2018) 16: 3. https://doi.org/10.1007/s10723-017-9418-y
11. Andrea Ceccanti, Enrico Vianello, & Marco Caberletti. (2018, May 18). INDIGO Identity and Access Management (IAM) (Version v1.4.0). Zenodo. http://doi.org/10.5281/zenodo.1874791
12. The Alpha Magnetic Spectrometer (AMS) on the International Space Station: Part I – results from the test flight on the space shuttle. Physics Reports, 366(6):331 – 405,2002
13. Atwood W.~B., et al., 2009, ApJ, 697, 1071, THE LARGE AREA TELESCOPE ON THE *FERMI GAMMA-RAY SPACE TELESCOPE* MISSION
14. Acernese F., et al., 2015, CQGra, 32, 024001 Advanced Virgo: a second-generation interferometric gravitational wave detector
15. https://helm.sh/
16. WLCG - Worldwide LHC Computing Grid, http://wlcg.web.cern.ch. [Google Scholar]
17. Yzquierdo, Antonio et al. (2019). Exploring GlideinWMS and HTCondor scalability frontiers for an expanding CMS Global Pool. EPJ Web of Conferences. 214. 03002. 10.1051/epjconf/201921403002.
18. Balcas J et al. 2015 Using the glideinWMS system as a common resource provisioning layer in CMS J. Phys. Conf. Ser 2015 J. Phys.: Conf. Ser. 664 062030
19. xrootd project. [online] Available at https://xrootd.slac.stanford.edu [Accessed 20 Feb. 2020]
20. D. Ciangottini et.al, (2019). “Integration of the Italian cache federation within the CMS computing model.” 014. 10.22323/1.351.0014