EPJ Web of Conferences **214**, 07026 (2019) https://doi.org/10.1051/epjconf/201921407026
CHEP 2018

Advanced features of the CERN OpenStack Cloud

José CASTRO LEÓN[1,*] for the CERN Cloud Infrastructure Team

[1]CERN IT Department

Abstract. The CERN OpenStack cloud has been delivering a wide variety of services to the whole laboratory since it entered in production in 2013. Initially, standard resources such as Virtual Machines and Block Storage were offered. Today, the cloud offering includes advanced features such as Container Orchestration (for Kubernetes, Docker Swarm mode, Mesos/DCOS clusters), File Shares and Bare Metal, and the Cloud team is preparing the addition of Networking and Workflow-as-a-Service components. In this paper, we will describe these advanced features, the OpenStack projects that provide them, as well as some of the main usecases that benefit from them. We will show the ongoing work on those services that will increase functionality, such as container orchestration upgrades and networking features such as private networks and floating IPs.

1 Introduction

The CERN OpenStack cloud delivers Infrastructure as a Service resources for the computing needs of the researchers at CERN. Initially the resources offered were Virtual Machines and Volumes. Since this initial service offering, the cloud team has been adding more services into the portfolio while improving the availability and features of the initial offering. The services that were added follow the new computing IT trends, like containers and software defined networks.

The cloud service is based on the OpenStack cloud software and it has been available for production workloads since July 2013. Since this date it has been upgraded transparently up to the Queens release of OpenStack, keeping the resources offered to the end users untouched. All the servers in the CERN cloud that provide computing resources are running CentOS 7 and they are distributed into two datacentres located in Meyrin (Geneva, Switzerland) and Wigner (Budapest, Hungary). In order to efficiently manage the number of servers of both datacentres, they are grouped first into cells and then we aggregate these groups in a highly scalable architecture. At this moment we operate more than seventy cells. This method allowed us to grow it quickly with the increasing demands of computing power of our clients.

*e-mail: jose.castro.leon@cern.ch

Figure 1. Current resource status of the CERN Cloud Infrastructure Service

Currently in the production environment, we operate 9,117 servers hosting around 39,104 virtual machines that are using around 314,500 cores. Along with these resources, we are also offering other types of resources like File Shares, Bare Metal nodes and Magnum clusters. As can be seen in Figure 2, the initial service offering consisted of Compute and Storage services. These included virtual machines, images and volumes as resources available to be requested by end users.

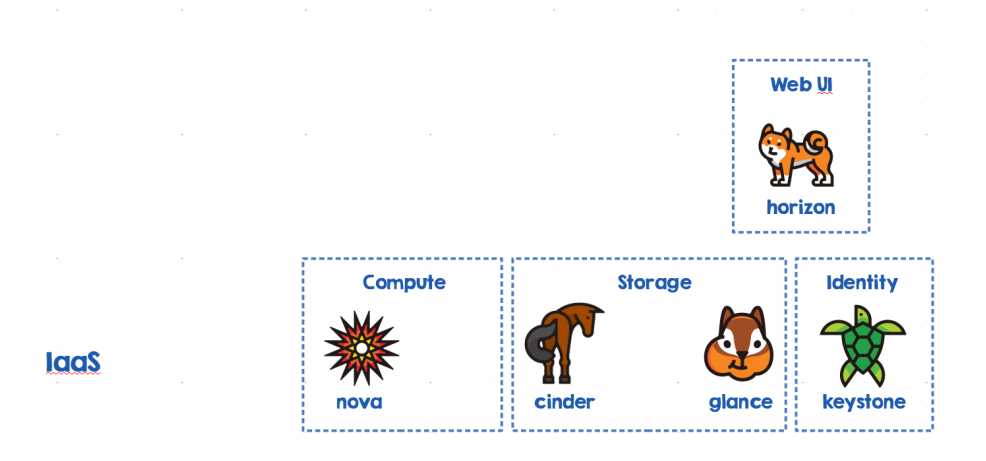

Figure 2. Initial service offering

All these resources are provided by OpenStack projects such as Nova for virtual machines, Cinder for Block Storage and Glance for images. For authentication into the cloud we use the Keystone project and the Web User Interface is provided by the Horizon project.

2 Advanced Services

With the new use cases provided by clients that are following recent IT trends such as containers and software defined networking, we have been adding more types of resources to the service portfolio. The current status of the Cloud Infrastructure Service is shown in Figure 3.

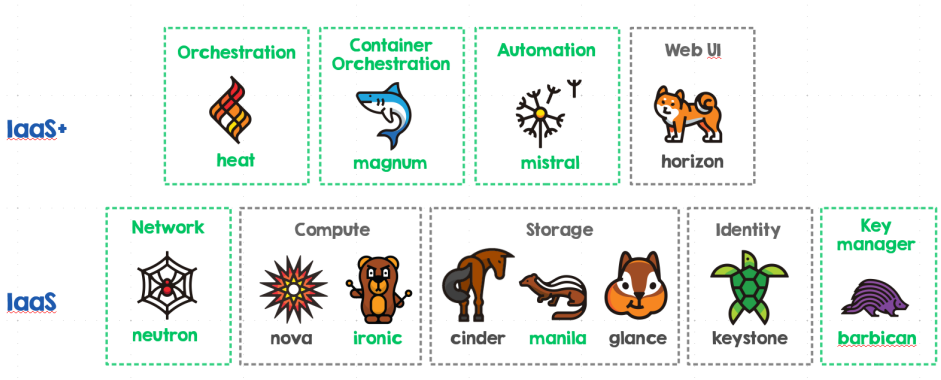

Figure 3. Current service offering

The service was also continuously improved over time. During this process, we have extended the services offered in the compute and storage to include Bare Metal and File Shares. These new services are covered by the Ironic and Manila projects. At the same time, we have also added more resource types like Network, Container Orchestration among others. A more detailed description of the new services and updates on the existing ones is provided in the following sections.

2.1 File Shares as a Service

The File Share as a Service project is provided by the OpenStack Manila project. This project provides a simple way for our users to create and manage file shares. The main difference between the file shares and the volumes provided by the block storage service is that the file shares can be accessed simultaneously by virtual machines while the volumes cannot. The resources provided by this service fulfills one of the most frequent requests from our users.

At the moment this service is being used for the High Performance computing cluster that uses these file shares as the shared storage for its computing workloads. The shares are used for input/output, temporary storage and synchronization amongst other things. The service is also used to provide a replacement for the NFS filers as it simplifies the management of these dedicated machines.

Once the file shares have been created, the user can use the CephFS protocol to access the data. We are working to extend it to NFS through Ganesha[1] that will allow it to cover even more use cases.

2.2 Container Orchestration Engine as a Service

The Container Orchestration Engine provides engines to deploy containers to our users. These engines are Kubernetes, docker-swarm, Mesos and DC/OS. This service is provided by the OpenStack project Magnum. In this service, the user will request a container cluster using one of these engines. Once the deployment has been completed, the user will receive an endpoint and he/she will be able to interact with the specific engine by using the appropriate client tool according to the type of engine deployed.

In order to improve the integration of the applications running there within the CERN ecosystem, we have added some features like kerberos, CVMFS and CephFS access through the Container Storage Interface (CSI)[2]. Kerberos credentials allow the containers to access 3rd party services. CVMFS is required by many of the experiment workloads. CephFS allows our users to have access to file shares provided by Manila in their orchestration engines.

These are some of the applications using the Container Orchestration Engine to run their workloads:

- Service for web access analysis that run their Jupyter notebooks inside a Kubernetes cluster with access to CVMFS. [3]
- Reproducible physics analysis through Recast[4] and REANA[5].
- Spark on Kubernetes to run Spark at scale for analysis. [6]
- Gitlab CI running continuous integrations of the code.
- ATLAS TDAQ for the evaluation of kubernetes as a replacement of their TDAQ management system. [7]

The upcoming work is focused on improving automation of these services by providing means to seamlessly upgrade a current cluster or heal it in case of issues. We are also looking to improve availability by adding multi master support in the kubernetes clusters.

2.3 Software defined networking

The production environment was deployed with an initial component to manage the network called nova-network. Nowadays this deprecated component only allows us to connect the machines to the network. On recent OpenStack releases this basic component has been superseded by the Neutron service. This component allows a richer set of features than the legacy network service such as project networks, floating IPs, security groups, load balancer as a service or firewall as a service. All these features provides more flexibility in terms of network management to our end-users and also to us as operators.

Currently we are focussed on replacing the legacy network component by Neutron. To minimize impact on our users, this replacement is done keeping the same functionality. At the same time we are deploying a software defined network (SDN) in order to investigate and then implement some of these new features in production. This small setup of a SDN is based on Tungsten Fabric[8] and we are testing all the features that the application provides in order to see its applicability in the CERN private Cloud. This setup is deployed in a different OpenStack region, that allows us to offer the new features to selected customers while keeping backwards compatibility to existing ones.

2.4 Bare Metal as a Service

Bare Metal as a Service allows our users to provide physical machines using the same interface as the virtual machine provisioning. This comes as a response to a need for more performant compute resources on our service, since the performance provided by virtual machines is not adequate for some application workloads.

Normally this occurs in applications that require:

- raw CPU performance, such as the apps run on the High Performance Computer cluster.
- low IO latency, like the Storage or Database nodes.

With this service, physical servers can be managed in the same way as Virtual Machines, making it attractive for allocating resources both to ourselves (for hypervisors) and to our users. One special use case is to run containers on Bare Metal removing the virtualization overhead on container deployments and make them more attractive for our end users.

This service treats physical servers as a pool. On commissioning new hardware, the procurement team will allocate resources and add them into the pool. When the machines reach End of Life, they will ask us to free them and then retire them from the pool. This simplifies a lot the management of physical resources. We are currently enrolling all the physical servers of our cloud into this service.

Also, we are adding Hardware Inventory functionality to the system. This feature will provide details about the hardware components installed in the server, delivery date and serial numbers. It will also track changes of each component like firmware updates, CMOS changes or hardware replacement. Once deployed, it will allow us to easily validate new deliveries, simplify retirements and correlate hardware failures with inventory.

2.5 Workflow as a Service

One important aspect of the current infrastructure is automation. Thanks to this element we have been able to simplify and ease maintenance and support operations. This Workflow as a Service component is provided by the OpenStack project Mistral. It allows our users, and also ourselves to automate OpenStack actions triggered by an API call or an event in the infrastructure like creation/deletion or operations on virtual machines.

The service is used for internal procedures to automate complex tasks like project management. On project creation for example, the workflow creates the project and prepares it to be ready to use for the end-user, like setting default quotas or configuring the services appropriately. On the contrary for deletion, the workflow needs to take care of the resources used, and free them in an appropriate order.

We also use this service to increase the resource efficiency by expiring machines used for test and development, keeping enough resources free for other tests or production use cases. In this procedure, we iterate over all the projects that have expiration and process the machines, expiring the ones created in test and development projects that have been running for more than six months, unless the user has decided to extend its usage. Since the policy was enabled, we have recovered 3,000 cores that become available to other applications. This set of workflows allows us to focus on more value added tasks for our end-users, such as providing automation on common operations like snapshotting/restoration of instances among others.

3 Summary

The Cloud Infrastructure Service started in July 2013 with a small set of services that allowed our users to have access to the computer centre, scale their applications and get benefit of the cloud model. This basic offering has been enhanced, improved and extended over the years following technological trends like containers and software defined networks. As a result, our users have more resources to provision, more flexibility to operate them and they will have even more in the future. We have also incorporated new use cases from our customers and integrate them into the CERN ecosystem, making them part of the offering. We also took advantage of these new use cases to reevaluate the current infrastructure and improve it over time.

The services described in this paper are part of a continuous improvement process in the Cloud Infrastructure Service as we look into an environment that is easy to use, scale, manage, and support.

References

[1] GANESHA - http://nfs-ganesha.github.io

[2] CephFS CSI - https://techblog.web.cern.ch/techblog/post/container-storage-cephfs-csi-part1

[3] E. Tejedor, Facilitating collaborative analysis in SWAN, Proceedings of this conference (CHEP2018)

[4] RECAST - http://recast.it

[5] T. Simko, REANA: A System for Reusable Research Data Analyses, Proceedings of this conference (CHEP2018)

[6] P. Kothuri, Apache Spark usage and deployment models for scientific computing, Proceedings of this conference (CHEP2018)

[7] G. Avolio, Evaluating kubernetes as an orchestrator of the high level trigger computing darm of the trigger and data acquisition system of the ATLAS experiment at the Large Hadron Collider, Proceedings of this conference (CHEP2018)

[8] Tungsten Fabric - https://tungsten.io