EOSC Test Suite
Cloud Benchmarking and Validation

Ignacio Peluaga
Shreyasvi Natraj

EGI Conference - October 2021

Motivation

Modern cloud services moving beyond the traditional categories of IaaS, PaaS and SaaS

Testing and validation should cover the whole stack: from the infrastructure level (CPU, GPU, FPGAs, Storage, Network....) to higher layers (Software)

Commodity services to be validated to multidisciplinary research use cases, across different environments

Need for a tool to automate deployment, benchmarking and validation of cloud services

- Flexibility to onboard commercial clouds, satisfying diverse research workloads
- Ease cloud services comparison
- Rely on open standards and tools; vendor independent
- Validate exit strategies
- Foster wider adoption of cloud services by the research community

Timeline

HELIX NEBULA
THESCIENCECLOUD
hnscicloud.eu

OCRE Open Clouds for Research Environments
ocre-project.eu

ARCHIVER
ARCHIVING AND PRESERVATION FOR RESEARCH ENVIRONMENTS
archiver-project.eu

CloudBank
ngiatlantic.eu/funded-experiments/cloudbank-eu-ngi

EUROPEAN OPEN SCIENCE CLOUD

2018 Concrete application of lessons learned and feedback gathered in **HNSciCloud**

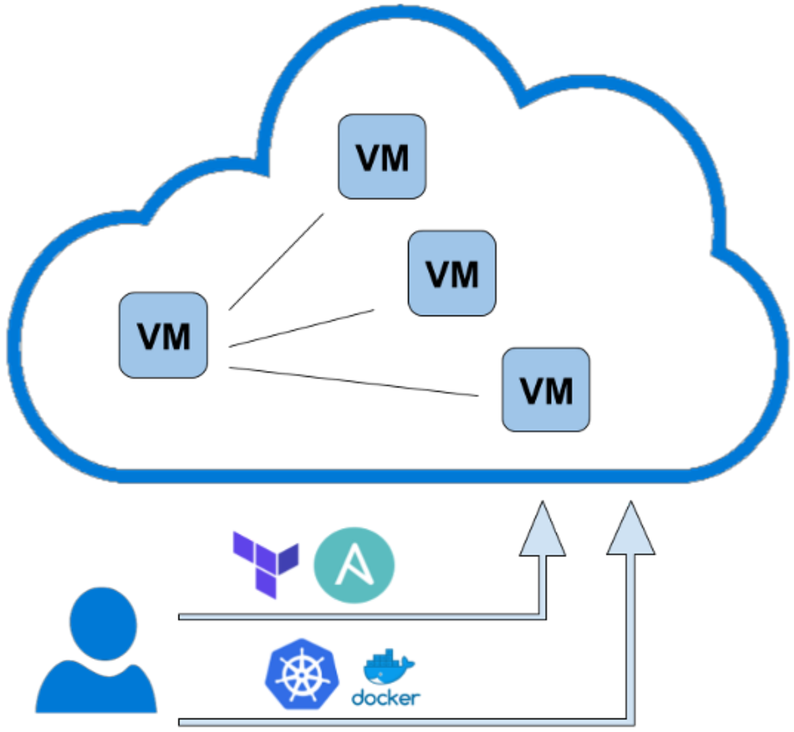
2019 First version of the tool used in the **OCRE** project

2020 Serving as a base for the **ARCHIVER** project test suite. Expanding the tests catalog

2021 Extensively used in **OCRE**. Also used in **CloudBankEU**

To be adopted in the European Open Science Cloud, **EOSC**

In a nutshell



- Written in Python 3
- Resource provisioning with Terraform
- Kubernetes cluster bootstrapping done by Ansible
- Deployment of tests on Docker containers (K8s pods)
- Simple YAML configuration
- Results as JSON, optionally pushed to CERN Cloud's S3
- Dashboard to visualize results instead of raw JSON (internal)

Catalog

Existing tests

- CPU benchmarking using HEP workloads (CERN)
- Network performance measurements with perfSONAR (Univ. of Michigan / CERN)
- CMS jobs emulation to verify ability to run real workflows as in DODAS HTCondor environment (INFN)
- Data Repatriation - Backup data from a commercial cloud provider to Zenodo (CERN)
- Basic S3 endpoint functional tests and checks (CERN)
- Satellite Image analysis and generation using Progressive Growing of GANs and GPUs (UNOSAT)
- Distributed training of GANs for HEP data generation using GPUs (CERN)

Under development

- HPC benchmarks based on openQCD and Grid (CERN)
- Advanced Object Storage performance with COSBench (CERN)
- FAIR Evaluator - “FAIRness” degree of repositories, looking to the ingested datasets and their quality (Uni. Oxford)
- AAI Capabilities

Process

1. User clones the public repository
2. Configuration of the tool by filling in simple YAML files, then start run
3. The test-suite provisions VMs and other resources according to configurations
4. The test-suite bootstraps a Kubernetes cluster with the previously created VMs
5. The test-suite completes K8s resource definition YAML files according to configurations
6. Pods are deployed using Kubernetes' API and the K8s resource definition YAML files
7. Containers on each pod pull and run test-specific images
8. As tests complete, the test-suite harvests the result files from each pods using kubectl
9. Once results are harvested, pods are killed and resources (VMs) optionally destroyed

Experiences

DEMO

Google Cloud

Running on Google Cloud Platform

Configuration and process overview

Custom Workload Deployment

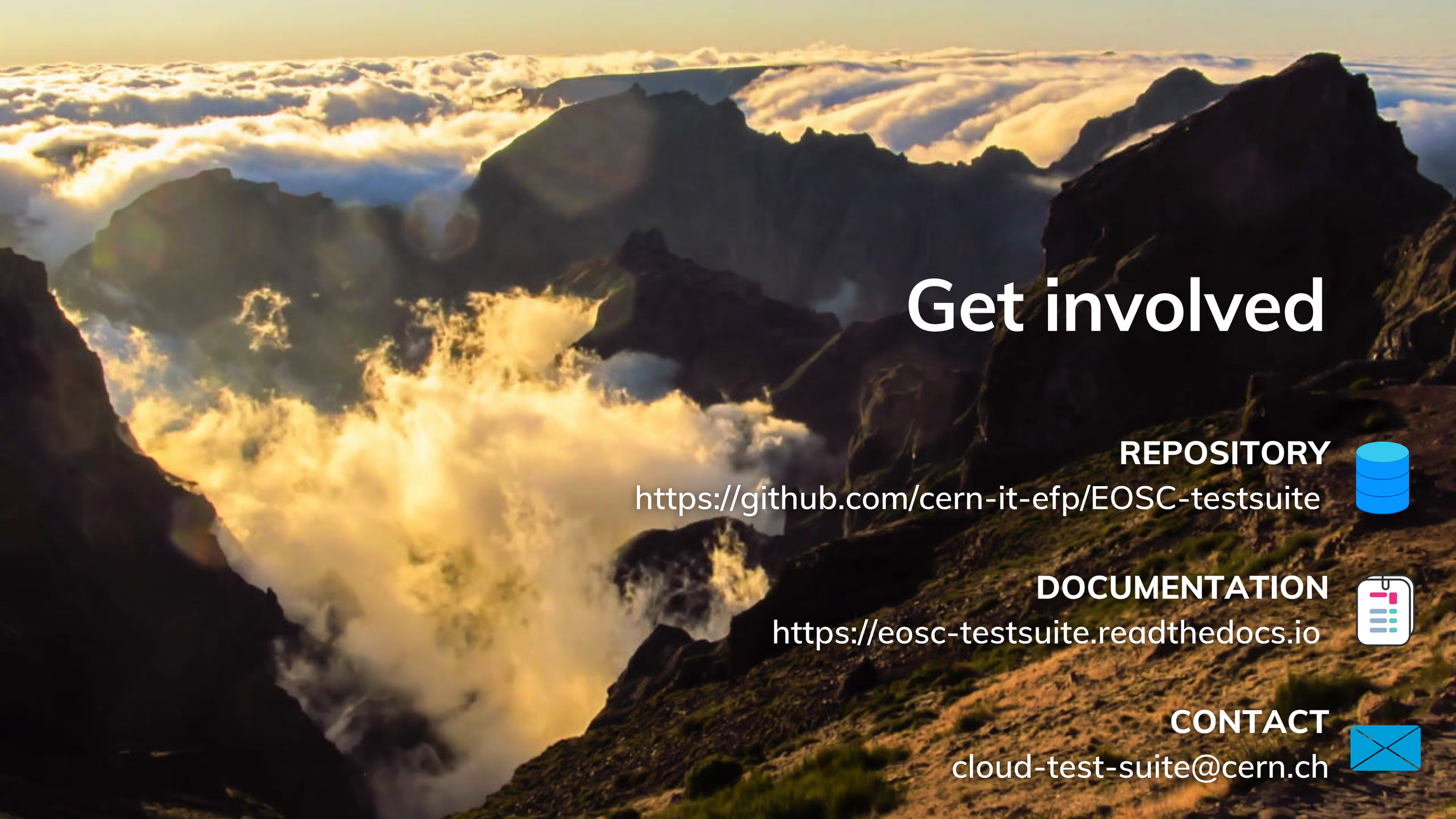
Next Steps

Kubernetes deployment method: Support for Helm charts but also Deployment, Job, etc

Updates and improvements of current tests plus new ones: Storage, Network, Deep Learning, etc.

Increase resilience and flexibility: Retries, logs and verbosity control, transparency, broader settings and configurations, cluster provisionment independence, etc

More clouds: support for more platforms



Get involved

REPOSITORY

https://github.com/cern-it-efp/EOSC-testsuite

DOCUMENTATION

https://eosc-testsuite.readthedocs.io

CONTACT

cloud-test-suite@cern.ch

home.cern