

Delivering integrated high-level services at Bari IBiSCo-ReCaS site through the PaaS dashboard and the Orchestration system

Mauro Patano on behalf of: Marica Antonacci, Giovanni Savarese, Ettore Serra, Marco Tangaro

> IBiSCo Workshop Napoli - April 18-19, 2024

Table of Contents

1 Overview

- 2 ReCaS PaaS
- **3** RECAS PaaS Dashboard
- 4 Architecture Details

5 Conclusions



Overview

Cloud Computing

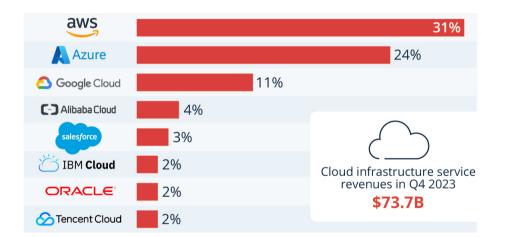
is a de facto standard for modern data centre implementation and a foundation for cloudnative application deployments. It is used by the majority of organisations worldwide.

WorldWide Keyplayers

Amazon Web Services, Microsoft Azure, Google Cloud Platform, Alibaba Cloud, offer a wide service assortment, readily accessible to customers.



World Market



¹Worldwide market share of leading cloud infrastructure service providers in Q 2023. Includes PaaS and IaaS as well as hosted private cloud services. Source: Synergy Research Group

Patano, Antonacci, Savarese, Serra, Tangaro

ReCaS PaaS: Orchestrator and Dashboard

Services

Delivery Modalities

Software as a Service (SAAS), Data as a Service (DAAS), Hardware as a Service (HAAS), Platform as a Service (PAAS), and Infrastructure as a Service (IAAS), provide a heterogeneous set of resources with varying degrees of ease in use and operation.

Data Centers Research Oriented

This model of service supply and access is becoming increasingly important, even in general purpose Data Centers with a significant focus on research, such as ReCaS Bari, or in data center federations like INFN-Cloud, of which ReCaS is a member.



Advantages

Users can access resources in a transparent and easy way, with very short waiting times. Admins and support team can limit the manual operations and configurations needed to support the users in their daily activities.

- Cost Effective
- Time Savings
- Future-Proof
- Increase Security
- Dynamically Scale
- Custom Solutions
- Flexibility



User Needs

Our users

come from different scientific communities (high energy physics, bioinformatics, medical physics, engineering...) and some local small and medium-sized enterprises.

Virtualization

Ever-increasing demand for virtual computer hardware platforms, storage devices, and computer network resources.

Containers

this lightweight virtualization technology dramatically simplifies the distribution and deployment of the software encapsulating the runtime dependencies in a single package.

Container Orchestration

system for automating software deployment, scaling, and management.

Patano, Antonacci, Savarese, Serra, Tangaro

ReCaS PaaS: Orchestrator and Dashboard

The ReCaS PaaS Orchestrator: three facts

- 1 The PaaS Orchestrator allows access to distributed cloud compute and other resources on top of cloud in a transparent and federated way
- 2 Users can easily deploy services without having to worry about where the resources are available and how to create and configure the resources they need
- A set of "pre-cooked" service templates are available through the Orchestrator Web Dashboard: once the user is logged in, she/he can choose different categories of services: from the instantiation of virtual machines, to the automatic installation of softwares like docker, docker-compose, to the deployment of complex architecture such as kubernetes clusters.



Foundations and key enablers

- Develop and integrate open-source and open standard-based components to ensure portability and interoperability
- Adopt a modular micro-service architecture
- Use Infrastructure as Code (IaC), DevOps and containers to reduce manual processes and increase flexibility and portability across environments
- Leverage federated Authentication and Authorization technologies based on OpenID-Connect (supporting also legacy AAI solutions) like ReCaS-Bari-IAM
- Enable flexible service composition and re-use



ReCaS PaaS - Main features

- Enabling the federation of distributed and heterogeneous compute environments: clouds, docker orchestration platforms
 - Further integrations can be easily included implementing new plugins (adapters)
- Smart scheduling → Automatic selection of the provider
 - First level based on the deployment type (Cloud, Docker, Kubernetes)
 - Second level based on SLAs and compute/storage requirements vs provider capabilities
 - Support for specialized hardware (GPU, Infiniband)
 - Data location
 - Best provider selection based on:
 - Resource quotas (SLA)
 - Monitoring data
 - Automatic retry of failed deployments
- Client interfaces for advanced users (REST APIs, CLI, python bindings) and end-users (web dashboard - no skills required)



ReCaS PaaS Platform

The Orchestrator ensures the transparent access and exploitation of the resources. RECAS-IAM (based on INDIGO IAM) is used to manage the users identities and permissions



The Dashboard

	Jocker-compose				
Description: Deploy a virtual machine with docker engine and docker-compose pre-installed. Optionally run a docker compose etched from the specified URL.	e file				
Deployment description					
description					
aneral Services Advanced					
locker_storage_size					
20	0	GB			
ze of the volume to be mounted in /var/lib/docker					
iorts					
Add rule					
orts to open on the machine					
aver					
Select		,			
umber of vCPUs and memory size of the Virtual Machine					
umber of VCPOs and memory size of the Virtual Machine					
umber or VU-VUs and memory size of the Virtual Machine bo you want to run a docker-compose file? res		,			

ReCaS Dashboard Deployments * Advanced * External Links * Admin *

PROJECT 👻 🦂 Mauro Patano 💌

Deployment discription discription Configuration Advanced advin, token Enter your password Rear for accessing Kits dashboard number, of, nodes number, of, nodes 	Description: Deplo	y a single master Kubernetes 1.28.7 cluster	
description Configuration Advanced admin.token Enter you password Anarobe of Kills node kVds notbe full in the Kills node kVds pots full in the Kills node kVds pots Farts to spon on the Kills matter VM moder.fullyor Farts to spon on the Kills matter VM moder.fullyor Farts to spon on the Kills matter VM moder.fullyor Farts to spon on the Kills matter VM moder.fullyor Farts to spon on the Kills matter VM moder.fullyor Farts to spon on the Kills matter VM moder.fullyor Farts to spon on the Kills matter VM moder.fullyor Farts to spon on the Kills matter VM for the Kills matter VM for the Kills matter VM for the Kills for the Kills matter VM for the Kills for the Kills matter VM for the Kills	Deployment descr	tion	
Enter your paraword Enter your paraword Enter your paraword Paraword token for accessing Kills distributed marber, off, nodes			
Enter your password for the state water VM memory size of the Kills maater VM mode, Slaved TM memory size of the Kills maater VM media. The state TM memory size of the Kills maater VM media. The state TM memory size of the Kills maater VM media. The state TM memory size of the Kills maater VM media. The state TM memory size of the Kills maater VM media. The state TM memory size of the Kills maater VM media. The state TM memory size of the Kills maater VM media. The state TM memory size of the Kills maater VM media. The state TM memory size of the Kills maater VM memory	Configuration	tvanced	
Enter your passend Enter your passend Enter your passend Enter for accessing Kills dashboard Enter the second Relation And and Enter the second Relation Enter the second Relati			
tasaarot taken for accessing Kils dashboard umrater dr.nodes 1 Unamber dr.Nodes 1 Unamber dr.Node 1 Un			
Number, of, nodes			•
1 Lumber of Kills node VMs oots Text to spee on the Kills master VM master_floor Sedect Sedect Lumber of vCM and memory size of the Kills master VM oot_floor Sedect-Lumber of vCM and memory size of the Kills master VM oot_floor	assword token for	ccessing K8s dashboard	
Aurober of MBs node VMs optis Add state by CARS and memory size of Ubs KBs master VM master_flow	number_of_nodes		
norts Ket rate Vit to open of the KBs master VM seter. La open of the KBs master VM seter. Seter.	1		
Addrade Vert Is open on the KBs master VM master_flavor 	Number of K8s nod	v Ms	
vis lo poin on the KBs master VM master_flevor Setect amber of CC-Bu and memory size of the KBs master VM mode_flavor	ports		
master_flavor Select Aurobe of CCPs and memory size of the KBs master VM dog flavor	Add rule		
Select Number of VCPM and memory size of the KBs master VM node_flavor	Ports to open on the	K8s master VM	
Select Number of VCPM and memory size of the KBs master VM node_flavor	master 0mm		
Number of vCPUs and memory size of the K8s master VM node_flavor			
		nd memory size of the KBs master VM	
	podo flavor		

Nodes and Services

The system architecture consists of five virtual machines on which the microservices (docker containers) run:

- Node recas-paas, services:
 - orchestrator-dashboard
 - redis
 - nginx
 - vault
- Node recas-orchestrator, service: orchestrator
- Node recas-cmdb, services:
 - slat
 - cpr
 - cmdb
 - couchdb
- Node recas-im, service im
- Node recas-paas-db, service: Mysql db



Key Points I

The Orchestrator interacts with the provider services through:

- the Infrastructure Manager for deploying complex and customized virtual infrastructures on multiple IaaS Cloud backends (Openstack...)
- direct APIs for deploying dockerized workloads on container platforms.

ReCaS Bari IAM manages the authentication through the whole stack:

users are organized in different IAM groups, each group can access a specific set of services from the dashboard.



Key Points II

The catalogue is a graphical representation of the TOSCA templates:

- Each card in the catalogue is associated to one or more templates
- The configuration form allows the user to specify requirements for the deployment in a straightforward way checking the mandatory fields hiding the complexity of TOSCA

A notification system is implemented in the Dashboard:

the user receives an automatic email as soon as the deployment is ready. Then, the details about the deployed service can be accessed through the Dashboard.

The Dashboard is integrated with Hashicorp Vault (Secrets Manager)

to support some functionalities (e.g. ssh key pair management, service credentials store)



Conclusions

ReCaS PaaS facilitates the deployment of complex services both for end users with different levels of technical skill and for maintainers who see most repetitive work completed automatically.

The dashboard allows the deployment of the required services by choosing a set of options available in easy-to-understand forms. The technologies behind the platform (tosca templates, tosca types, OpenID identity and access management, Ansible ...) are transparent to the user.

We will gradually add other "plugins" to enable other features/services.

