



EGI Cloud Compute & EGI Cloud Container Compute

Enol Fernández
EGI Foundation



eosc-hub.eu



[@EOSC_eu](https://twitter.com/EOSC_eu)

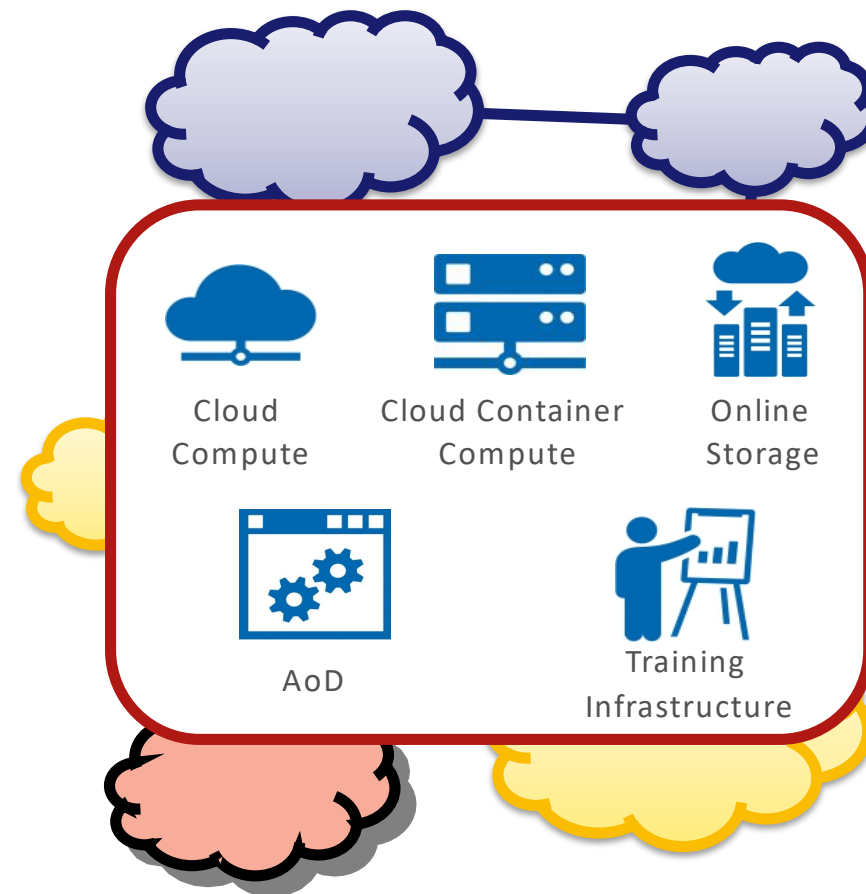
Dissemination level: Public/Confidential *If confidential, please define:*

Disclosing Party: (those disclosing confidential information)

Recipient Party: (to whom this information is disclosed, default: project consortium)



- Multi-cloud IaaS with Single Sign-On via Check-in
 - Technology agnostic, supports OpenStack, OpenNebula and Synnefo
- Extra features
 - Virtual Appliance catalogue
 - Unified GUI dashboard
 - Centralised accounting
 - Resource discovery
 - SLA monitoring





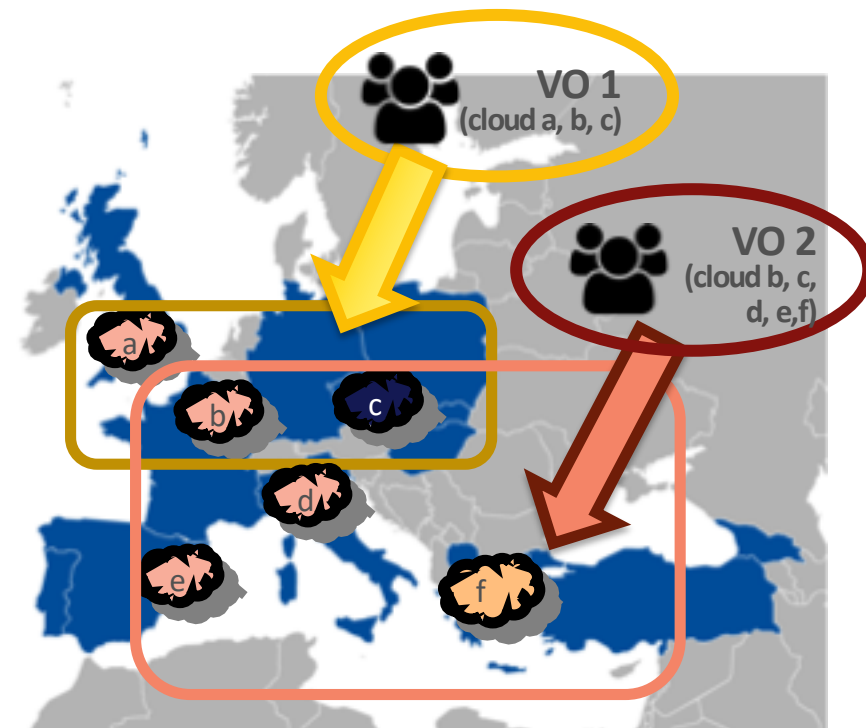
20 resource centres

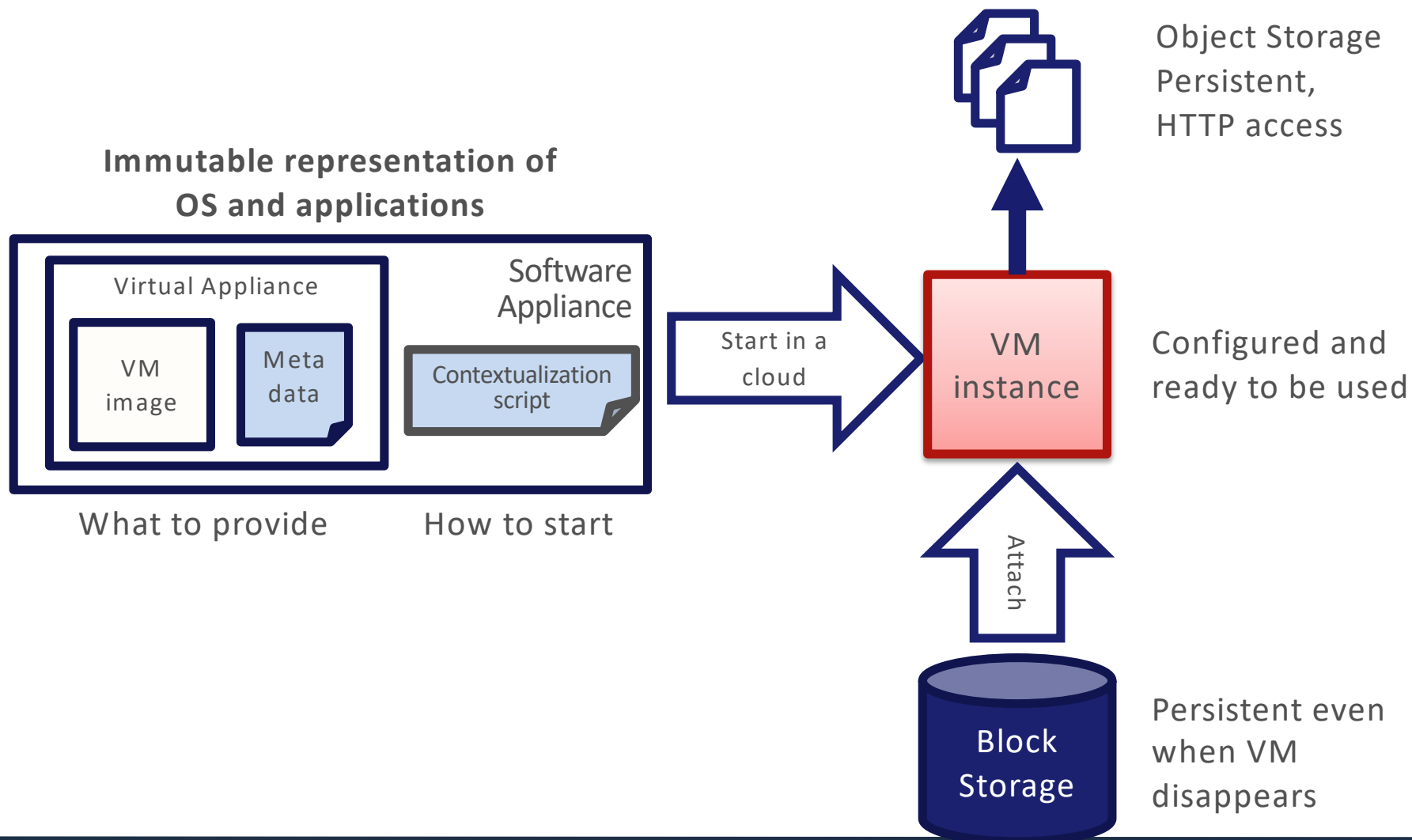
- 15 OpenStack
- 4 OpenNebula
- 1 Synnefo

5 centres under integration

2 centres expressed interest on joining

- Run Virtual Machines on demand on EGI's Cloud Federation
 - Similar to AWS EC2/EBS or GCP Compute Engine
- Access is VO-based: VO = group of users + providers supporting the VO
 - Community-specific VOs – e.g. CHIPSTER, EISCAT, etc.
 - Training VO = training.egi.eu
 - Generic VOs – e.g. fedcloud.egi.eu
- Diverse providers with common:
 - AuthN and AuthZ
 - VM Image catalogue
 - Information discovery
 - Accounting
 - Monitoring
 - GUI dashboard



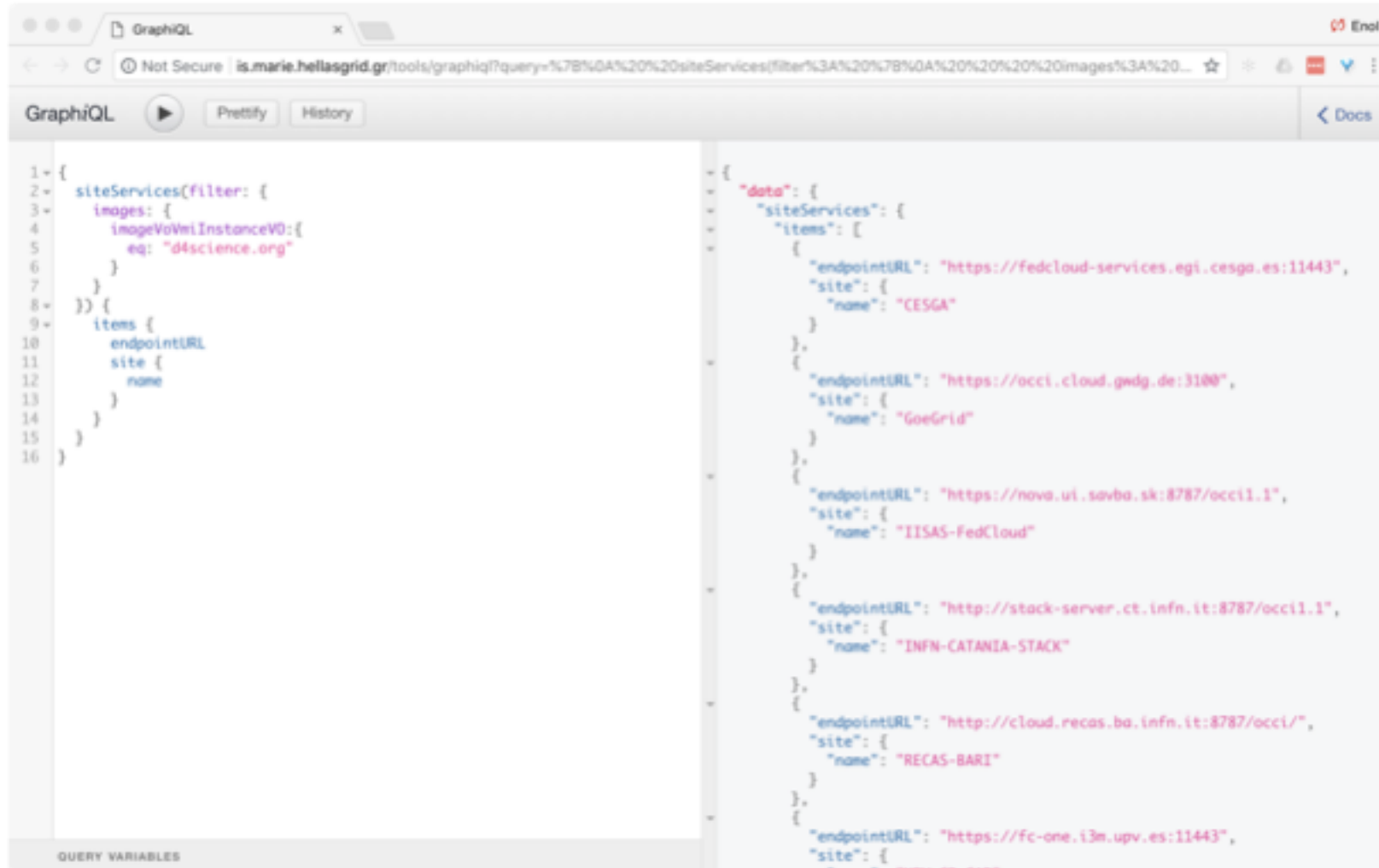




The screenshot shows a web browser displaying the AppDB interface. The main content is a table of virtual machines (VMs) with the following columns: Memory, Disk, Logical/Physical CPUs, Connectivity In/Out, and OS Family. The table is filtered by Site: IN2P3-IREN (FR) and Image: ver.1.7 Ubuntu 14.04 / x86_64 / VirtualBox. The table contains 15 rows of VM data.

Memory	Disk	Logical/Physical CPUs	Connectivity In/Out	OS Family
512000	160 GB	48/48	yes/yes	linux
262144	320 GB	16/16	yes/yes	linux
262144	50 GB	16/16	yes/yes	linux
131072	50 GB	16/16	yes/yes	linux
131072	80 GB	16/16	yes/yes	linux
131072	320 GB	16/16	yes/yes	linux
65536	80 GB	32/32	yes/yes	linux
65536	80 GB	16/16	yes/yes	linux
32768	320 GB	16/16	yes/yes	linux
16384	160 GB	8/8	yes/yes	linux
8192	80 GB	4/4	yes/yes	linux
4096	20 GB	2/2	yes/yes	linux
4096	40 GB	2/2	yes/yes	linux
4096	20 GB	1/1	yes/yes	linux
2048	20 GB	1/1	yes/yes	linux
2048	50 GB	1/1	yes/yes	linux
512	1 GB	1/1	yes/yes	linux

More information at https://docs.google.com/presentation/d/19Yh3kNxl01DfcrDgOf12w-KOW57rd_OnYP2iGp9Kq2Y/edit?ts=5a2ab515#slide=id.p



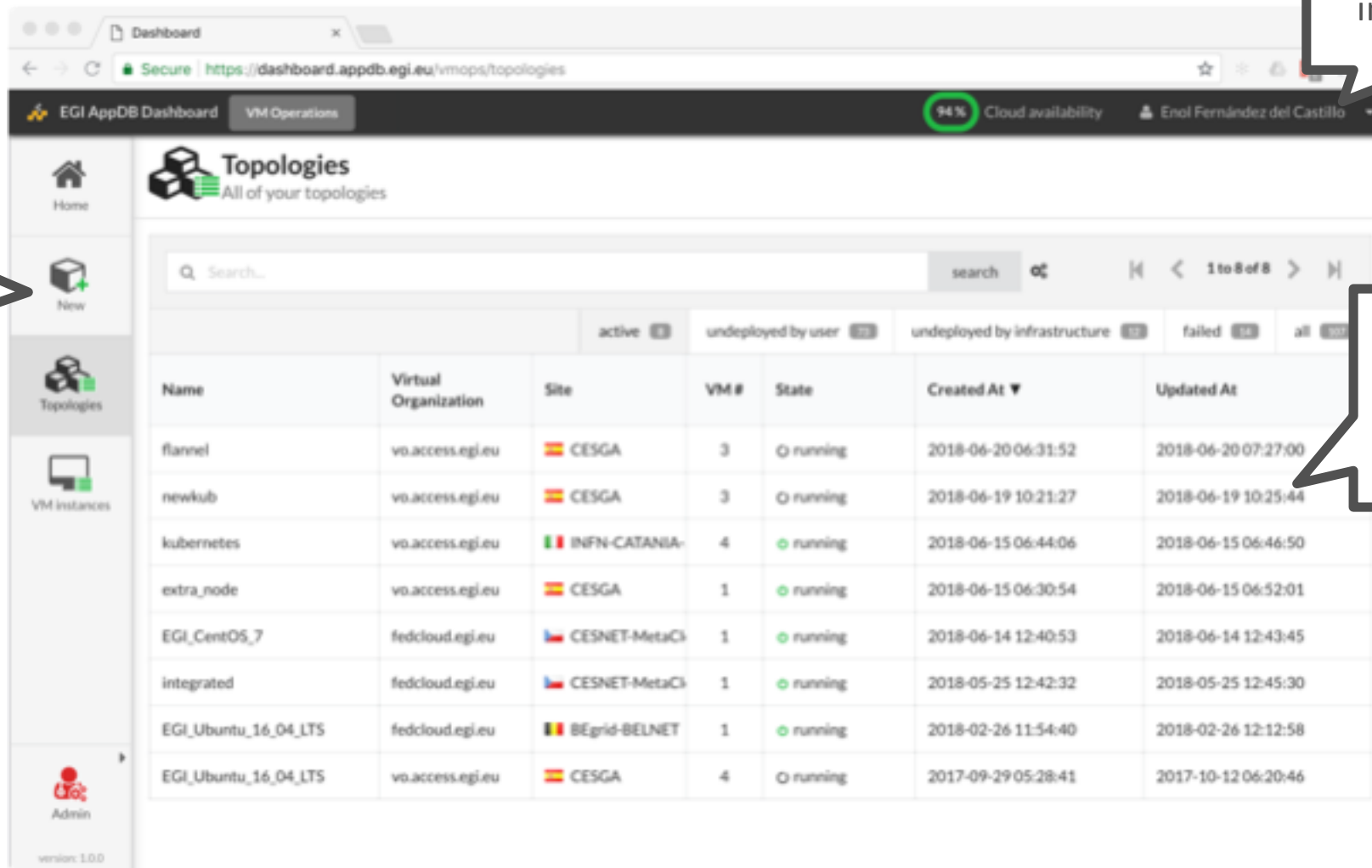
The screenshot shows a GraphQL IDE interface. On the left, a query is written in a light blue font on a white background. The query filters site services by the imageVoVmeInstanceVD field, specifically looking for 'd4science.org'. It requests the endpointURL and site name for each item. On the right, the JSON response is shown in a light red font on a white background. The response is a list of site services, each with an endpointURL and a site name. The sites listed are CESGA, GoeGrid, IISAS-FedCloud, INFN-CATANIA-STACK, RECAS-BARI, and Fc-one.

```
1- {
2-   siteServices(filter: {
3-     images: {
4-       imageVoVmeInstanceVD: {
5-         eq: "d4science.org"
6-       }
7-     }
8-   }) {
9-     items {
10-      endpointURL
11-      site {
12-        name
13-      }
14-    }
15-  }
16- }
```

```
{
  "data": {
    "siteServices": {
      "items": [
        {
          "endpointURL": "https://fedcloud-services.egi.cesga.es:11443",
          "site": {
            "name": "CESGA"
          }
        },
        {
          "endpointURL": "https://occi.cloud.gwdg.de:3100",
          "site": {
            "name": "GoeGrid"
          }
        },
        {
          "endpointURL": "https://nova.ui.savba.sk:8787/occi1.1",
          "site": {
            "name": "IISAS-FedCloud"
          }
        },
        {
          "endpointURL": "http://stack-server.ct.infn.it:8787/occi1.1",
          "site": {
            "name": "INFN-CATANIA-STACK"
          }
        },
        {
          "endpointURL": "http://cloud.recas.ba.infn.it:8787/occi/",
          "site": {
            "name": "RECAS-BARI"
          }
        },
        {
          "endpointURL": "https://fc-one.i3m.upv.es:11443",
          "site": {
            "name": "Fc-one"
          }
        }
      ]
    }
  }
}
```

Complete
Check-in
integration

Wizard-like
creation of
VMs



The screenshot shows the EGI AppDB Dashboard with the following components:

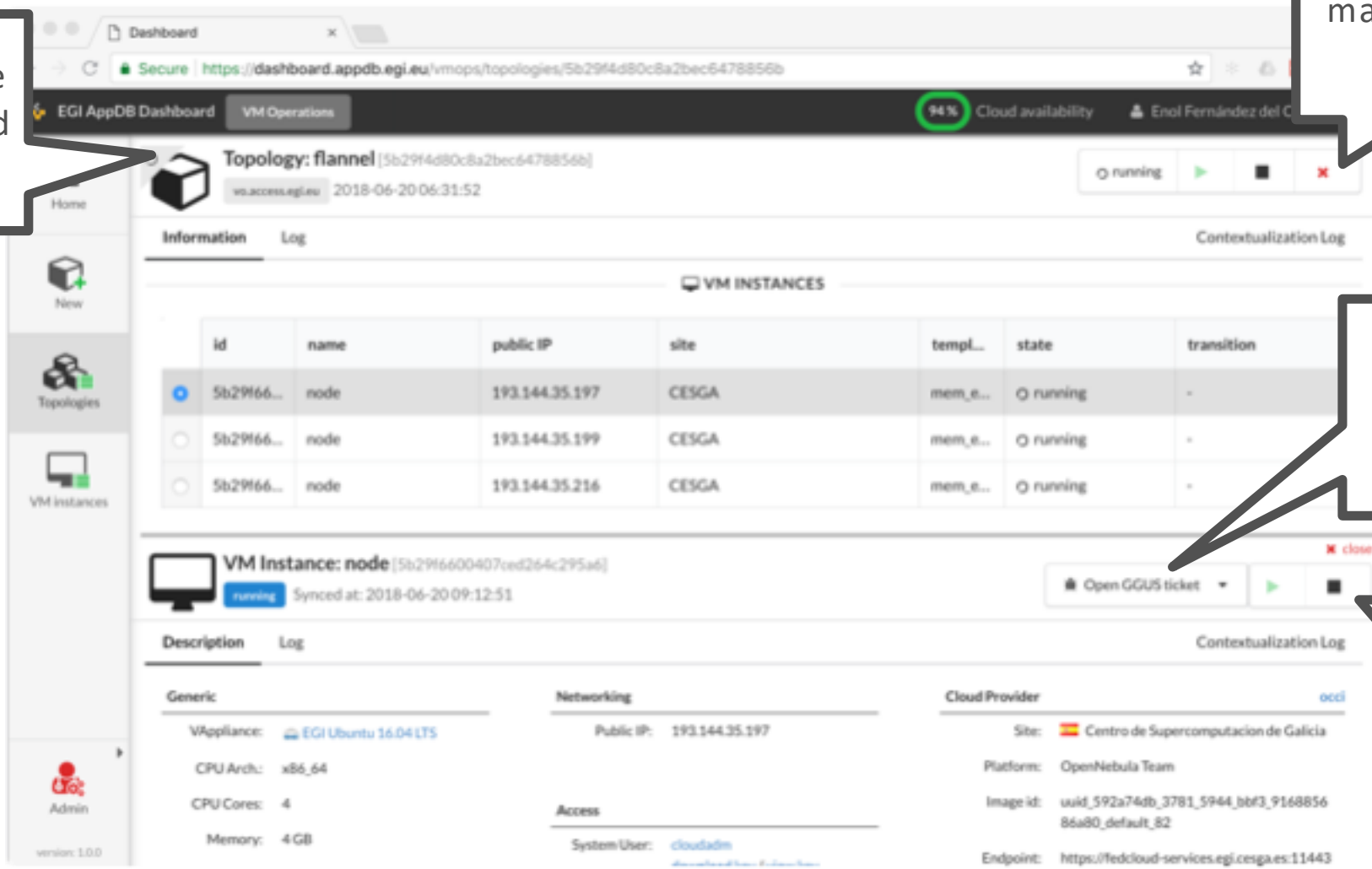
- Header:** "EGI AppDB Dashboard" and "VM Operations". A green circle highlights "94% Cloud availability". The user "Enol Fernández del Castillo" is logged in.
- Left Sidebar:** Navigation menu with "Home", "New", "Topologies", "VM instances", and "Admin".
- Main Content:** "Topologies" section with a search bar and a table of VM instances.

Name	Virtual Organization	Site	VM#	State	Created At	Updated At
flannel	vo.access.egi.eu	ESG	3	running	2018-06-20 06:31:52	2018-06-20 07:27:00
newkub	vo.access.egi.eu	ESG	3	running	2018-06-19 10:21:27	2018-06-19 10:25:44
kubernetes	vo.access.egi.eu	INFN-CATANIA	4	running	2018-06-15 06:44:06	2018-06-15 06:46:50
extra_node	vo.access.egi.eu	ESG	1	running	2018-06-15 06:30:54	2018-06-15 06:52:01
EGI_CentOS_7	fedcloud.egi.eu	CESNET-MetaCl	1	running	2018-06-14 12:40:53	2018-06-14 12:43:45
integrated	fedcloud.egi.eu	CESNET-MetaCl	1	running	2018-05-25 12:42:32	2018-05-25 12:45:30
EGI_Ubuntu_16_04_LTS	fedcloud.egi.eu	BEgrid-BELNET	1	running	2018-02-26 11:54:40	2018-02-26 12:12:58
EGI_Ubuntu_16_04_LTS	vo.access.egi.eu	ESG	4	running	2017-09-29 05:28:41	2017-10-12 06:20:46

Single
dashboard for
all providers

Topologies are a set of related VMs

Global management of VMs



The screenshot shows the EGI AppDB VM Operations dashboard. At the top, there's a navigation bar with 'EGI AppDB Dashboard' and 'VM Operations'. A '94% Cloud availability' indicator is visible. The main content area displays a 'Topology: flannel' with a 'running' status. Below this is a table of 'VM INSTANCES' with the following data:

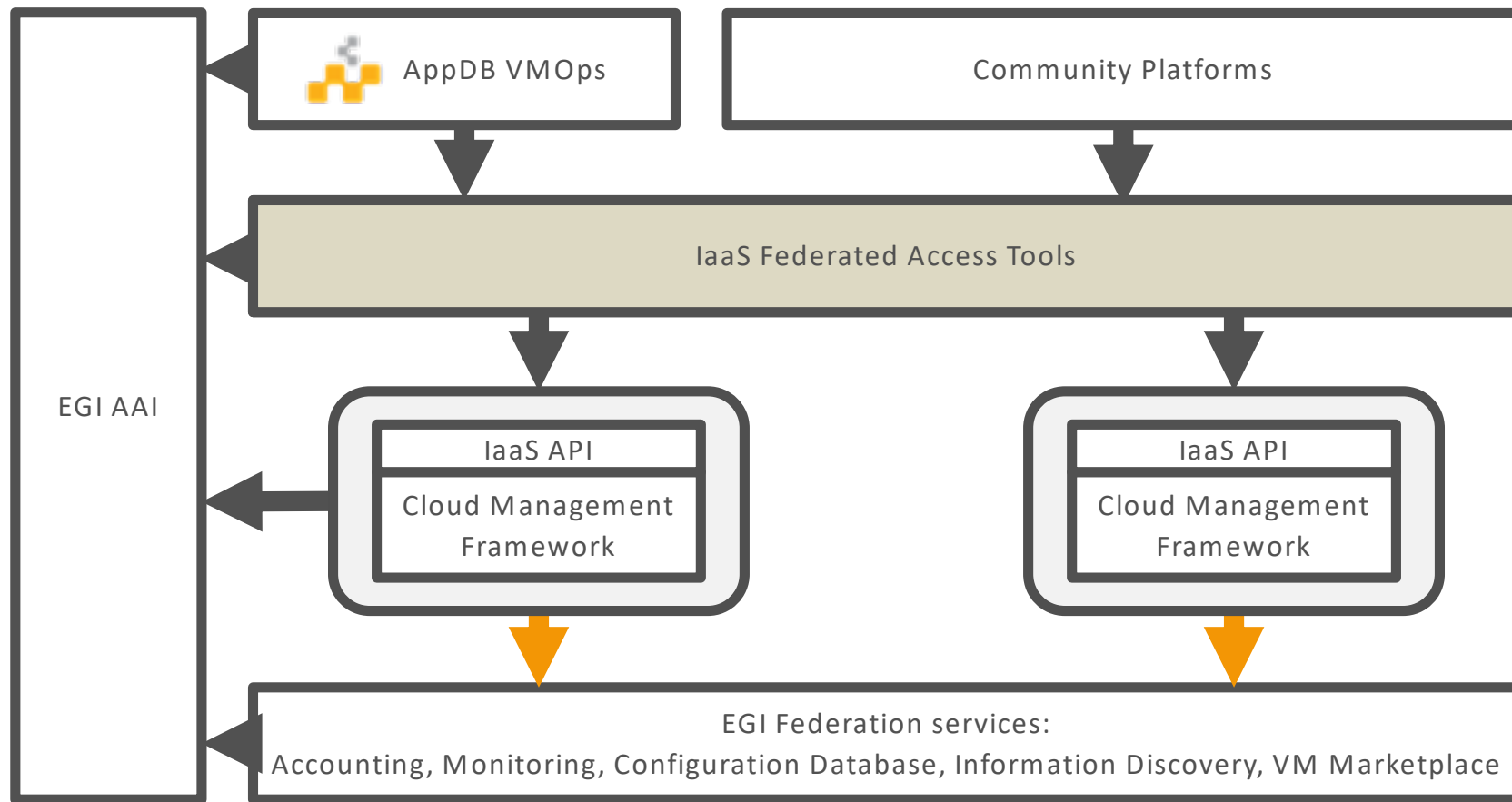
id	name	public IP	site	templ...	state	transition
5b29f66...	node	193.144.35.197	CESGA	mem_e...	running	-
5b29f66...	node	193.144.35.199	CESGA	mem_e...	running	-
5b29f66...	node	193.144.35.216	CESGA	mem_e...	running	-

Below the table, a 'VM Instance: node' is detailed with a 'running' status and a 'Synced at' timestamp. A 'Description' section provides details:

- Generic:** VAppliance: EGI Ubuntu 16.04 LTS, CPU Arch: x86_64, CPU Cores: 4, Memory: 4 GB
- Networking:** Public IP: 193.144.35.197
- Cloud Provider:** Site: Centro de Supercomputacion de Galicia, Platform: OpenNebula Team, Image id: uuid_592a74db_3781_5944_bbf3_9168856_86a80_default_82, Endpoint: https://fedcloud-services.egi.cesga.es:11443
- Access:** System User: cloudadm

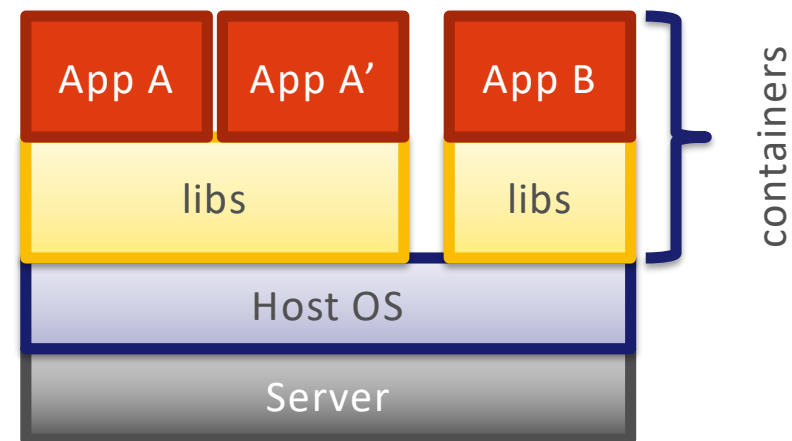
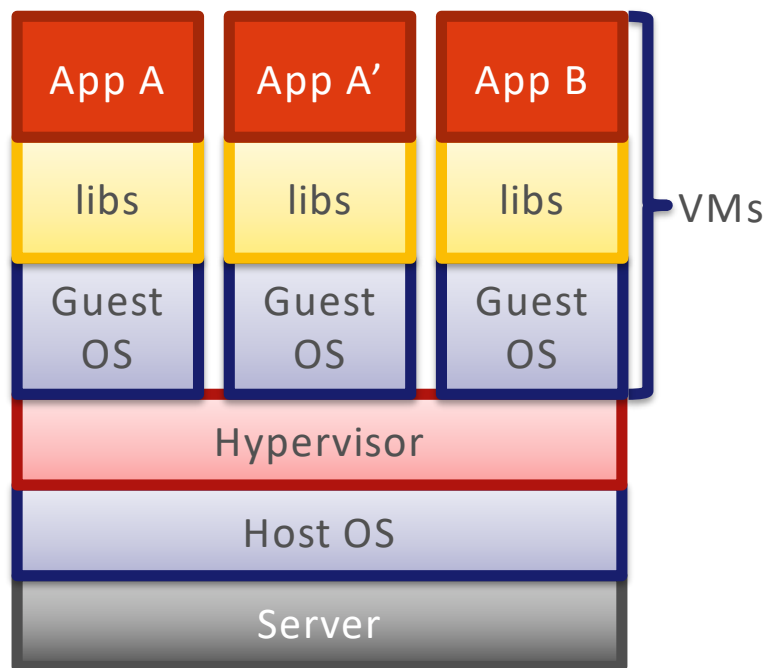
GGUS integration

Individual management of VMs



- ◉ EGI Federated Cloud no longer mandates a single API for every provider
 - OCCl still widely supported but sites are moving native APIs (mainly OpenStack!)
- ◉ Tools to deal with heterogeneity:
 - IaaS orchestration tools with support for multiple APIs:
 - Infrastructure Manager, Terraform, OCCOPUS, ...
 - https://wiki.egi.eu/wiki/Federated_Cloud_IaaS_Orchestration
 - IaaS libraries with support for multiple APIs:
 - libcloud, jclouds,...
 - See guide on migrating from OCCl to IM on EGI's wiki:
https://wiki.egi.eu/wiki/Federated_Cloud_OCCl_to_IM_Migration

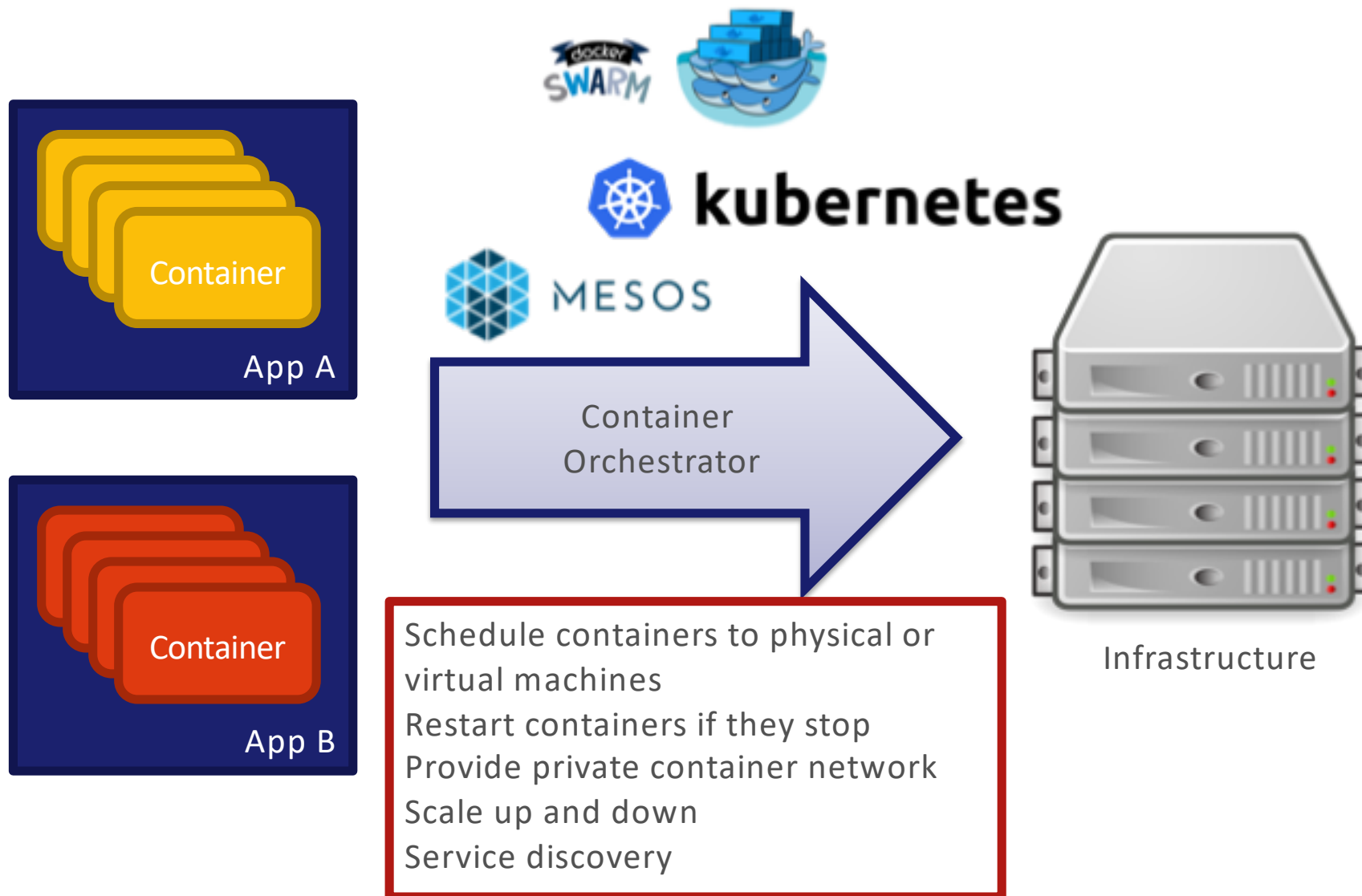
- Containers provide virtualisation at the OS level
 - Same kernel, isolated user-space
 - Faster deployment, less overhead, easier migration...



- “Open-platform for building, shipping and running distributed applications”



- Docker commoditizes containers
 - Hides and automates container management process
 - One-command-line deployment of applications
 - Easy to move from development to production
 - Provides ecosystem to create and share images



- Run containers on top of EGI Cloud Compute VMs
- 2 (+ 1) options:
 - Single node: start the EGI Docker VM and run containers directly (or with docker compose)
 - Kubernetes: start a cluster of VMs and create a Kubernetes cluster to run your containers
 - Start the cluster using IM + Ansible
 - Working on: auto-scaling with EC3, Check-in integration at Kubernetes level
 - udocker: run containers as jobs in the EGI HTC service
- https://wiki.egi.eu/wiki/Federated_Cloud_Containers

- Kubernetes is an *open-source platform for automating deployment, scaling, and operations of application containers across clusters of hosts, providing container-centric infrastructure.*
- Some concepts:
 - *Pod*: group of one or more containers, shared storage and options to run the containers
 - *Deployment* maintains the desired count of Pods all the time
 - *Service*: logical set of Pods and a policy by which to access them.
 - Exposed to the exterior of the Kubernetes cluster via mapping of ports and or Load Balancing
 - *Job*: A *job* creates one or more pods and ensures that a specified number of them successfully terminate.


```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: frontend
spec:
  selector:
    matchLabels:
      app: guestbook
      tier: frontend
  replicas: 3
  template:
    metadata:
      labels:
        app: guestbook
        tier: frontend
    spec:
      containers:
      - name: php-redis
        image: gcr.io/google-samples/gb-frontend:v4
        resources:
          requests:
            cpu: 100m
            memory: 100Mi
        env:
        - name: GET_HOSTS_FROM
          value: dns
      ports:
      - containerPort: 80
```

```
apiVersion: v1
kind: Service
metadata:
  name: frontend
  labels:
    app: guestbook
    tier: frontend
spec:
  # comment or delete the following line if you want to use a LoadBalancer
  type: NodePort
ports:
- port: 80
  selector:
    app: guestbook
    tier: frontend
---
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: frontend
spec:
  rules:
  - host: frontend.test.fedcloud.eu
    http:
      paths:
      - backend:
          serviceName: frontend
          servicePort: 80
```

- Provides Kubernetes v1.10
- Major differences with other offerings:
 - LoadBalancer ServiceType:
 - A NGINX ingress configured by default ready to be used offering similar functionality
 - Expandable with auto-configuration of Let's Encrypt certificates
 - Dynamic provision of volumes for PersistentVolumeClaims
 - No block-storage directly available
 - NFS-based volumes available instead

- EGI Cloud Compute currently relies on legacy X.509 + VOMS proxies for access to resources
 - For users without certificates:
 - PUSP with user-personalised proxies from robot certificate
 - RCAuth Online CA to obtain personal proxies from EGI Check-in identities
- Now rolling-out production providers with native OpenID Connect support
 - 2 sites now available, more coming
 - No need for certificates at all!

**Thank you
for your attention!**

Questions?



EOOSC-hub

 eosc-hub.eu  [@EOOSC_eu](https://twitter.com/EOOSC_eu)