www.egi.eu
@EGI_eInfra

Managing Docker containers and Kubernetes clusters in the EGI Cloud

Enol Fernández
Cloud Solutions Manager – EGI Foundation

The work of the EGI Foundation
is partly funded by the European Commission under H2020 Framework Programme

Outline

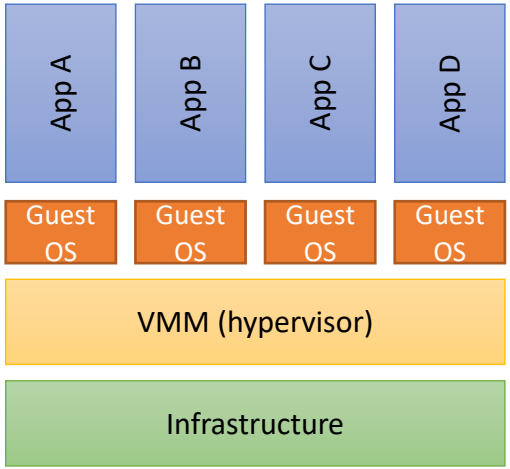
- Containers & container orchestration
  - Docker
  - Kubernetes
- Running containerised applications in EGI
  - udocker
  - Docker VM
  - Kubernetes
- Q & A

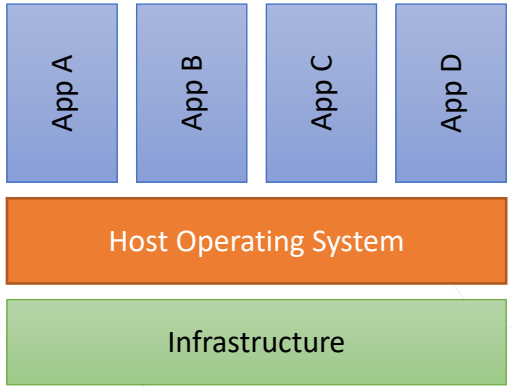
Containers

OS – level virtualisation

Containers

- OS-level virtualisation
  - Isolated runtime built on Linux namespaces and cgroups, without VMM overhead





Why Containers?

- Container ship all application code and dependency on a single package
  - Run the same thing everywhere from devel to production
- Containers share the machine's OS system kernel
  - Little to none overhead, server consolidation
- Containers run isolated from other applications
  - OS provides a sandboxed view of the system where all resources are virtualised

Docker

- “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

Container images

- Images are the read-only template from which containers are started from
- Layers of filesystems built with a *Dockerfile*

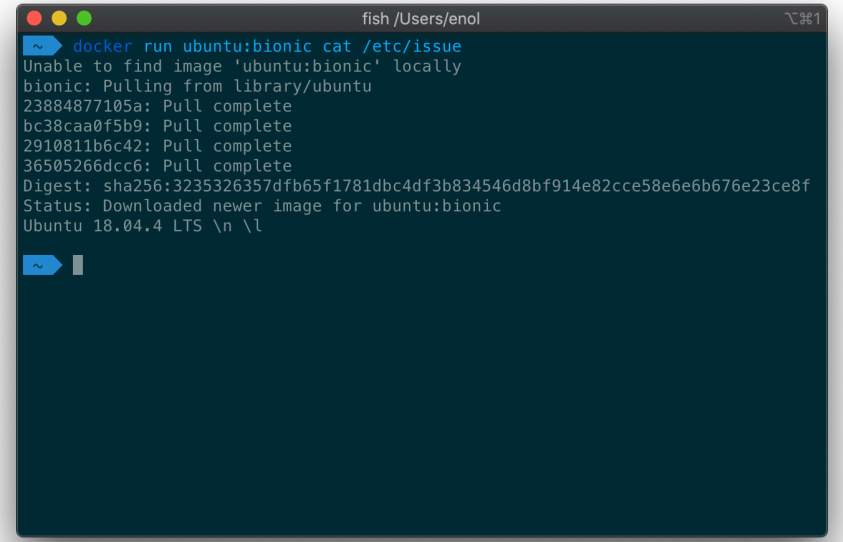
```
# our base image
FROM alpine:3.5

# Install python and pip
RUN apk add --update py2-pip

# install Python modules needed by the Python app
COPY requirements.txt /usr/src/app/
RUN pip install --no-cache-dir -r /usr/src/app/requirements.txt

# copy files required for the app to run
COPY app.py /usr/src/app/
COPY templates/index.html /usr/src/app/templates/

# tell the port number the container should expose
EXPOSE 5000

# run the application
CMD ["python", "/usr/src/app/app.py"]
```

Sample from https://github.com/docker/labs/blob/master/beginner/chapters/webapps.md

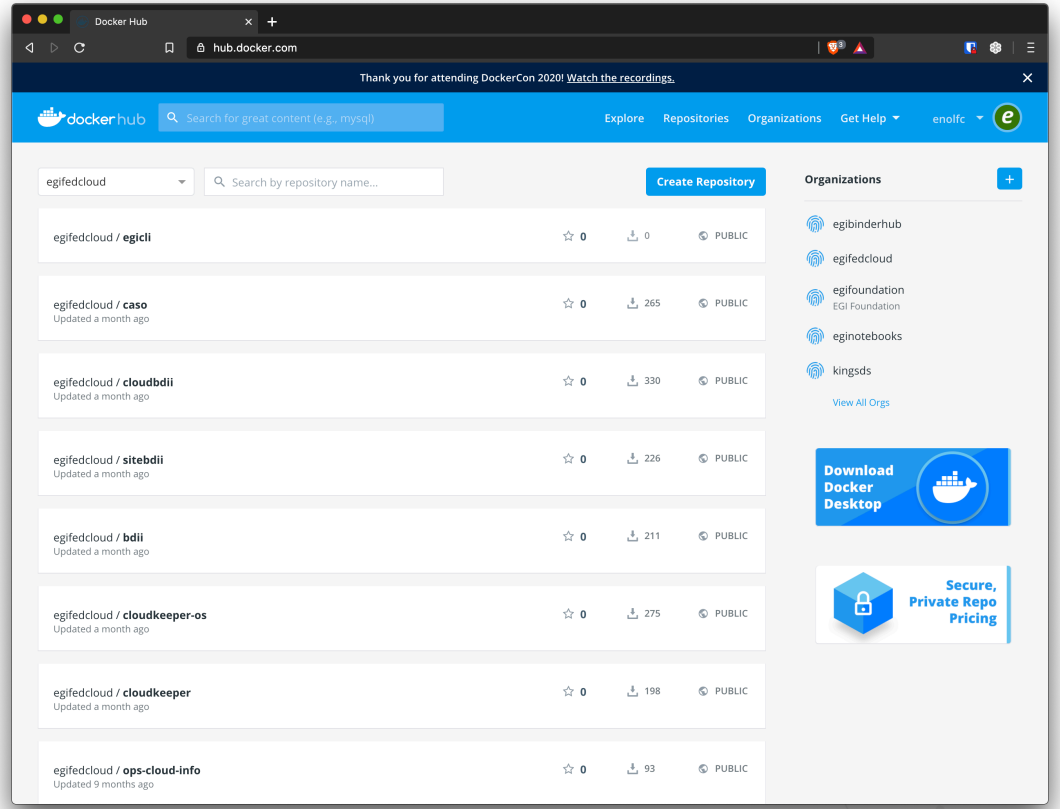
DockerHub / container registries

- Container registry service for sharing applications

- DockerHub is the main registry available
  - Public and private repositories
  - Official repositories
  - Automated builds



DockerHub


Docker Hub
hub.docker.com
Thank you for attending DockerCon 2020! Watch the recordings.
dockerhub
Search for great content (e.g., mysql)
Explore
Repositories
Organizations
Get Help
enolfc
egifedcloud
Search by repository name...
Create Repository
egifedcloud / egicli 0 0 PUBLIC
egifedcloud / caso
Updated a month ago
0 265 PUBLIC
egifedcloud / cloudbdii
Updated a month ago
0 330 PUBLIC
egifedcloud / sitebdii
Updated a month ago
0 226 PUBLIC
egifedcloud / bdii
Updated a month ago
0 211 PUBLIC
egifedcloud / cloudkeeper-os
Updated a month ago
0 275 PUBLIC
egifedcloud / cloudkeeper
Updated a month ago
0 198 PUBLIC
egifedcloud / ops-cloud-info
Updated 9 months ago
0 93 PUBLIC
Organizations
egibinderhub
egifedcloud
egifoundation
EGI Foundation
eginotebooks
kingsds
View All Orgs
Download Docker Desktop
Secure, Private Repo Pricing

Running containers

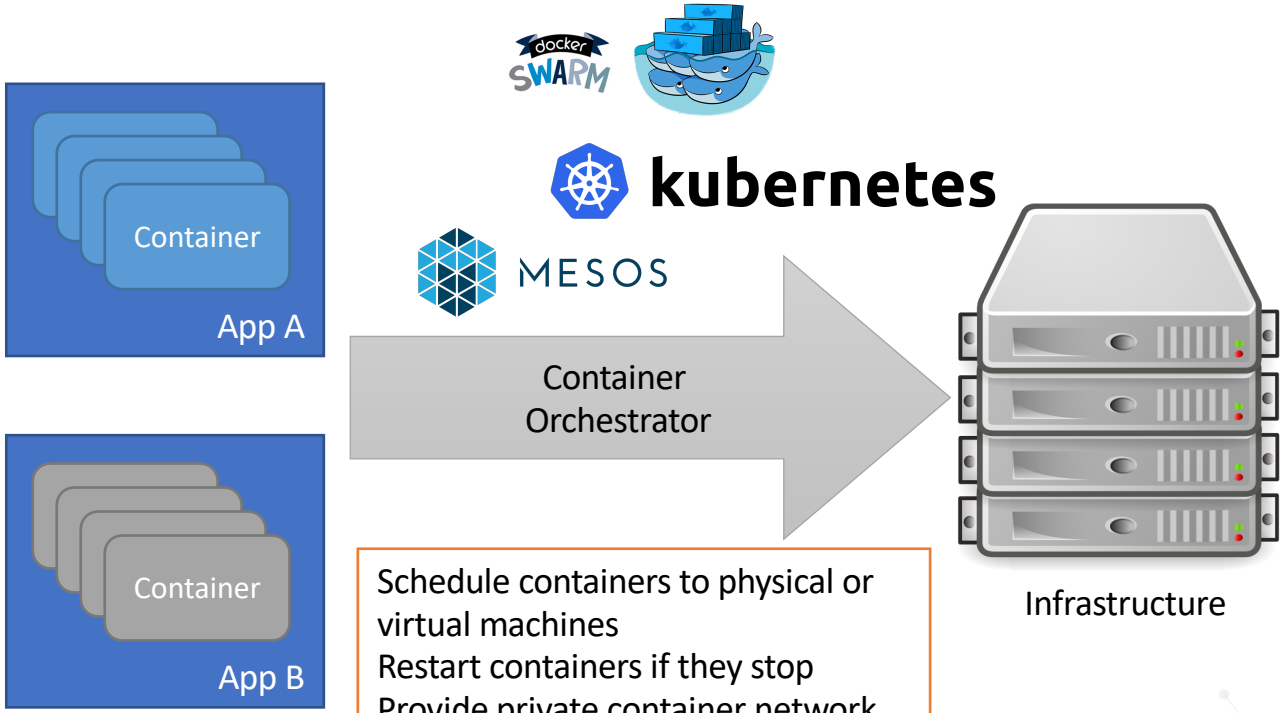
- Just use docker run!
  - Usual practice: 1 container = 1 process
- What if I have to run more than one process/container? → Docker Compose
  - Define multi-container applications
    - What containers to run? (services)
    - How to communicate between them? (network)
    - How to interact with storage? (volumes)
  - Manage the application from a single tool (docker-compose)
- What if 1 machine is not enough? → Container orchestration

Docker-compose

```
version: '2.0'
services:
  web:
    build: .
    ports:
    - "5000:5000"
    volumes:
    - .:/code
    - logvolume01:/var/log
    links:
    - redis
  redis:
    image: redis
volumes:
  logvolume01: {}
```

Sample from https://docs.docker.com/compose/

Container orchestration



Schedule containers to physical or virtual machines
Restart containers if they stop
Provide private container network
Scale up and down
Service discovery

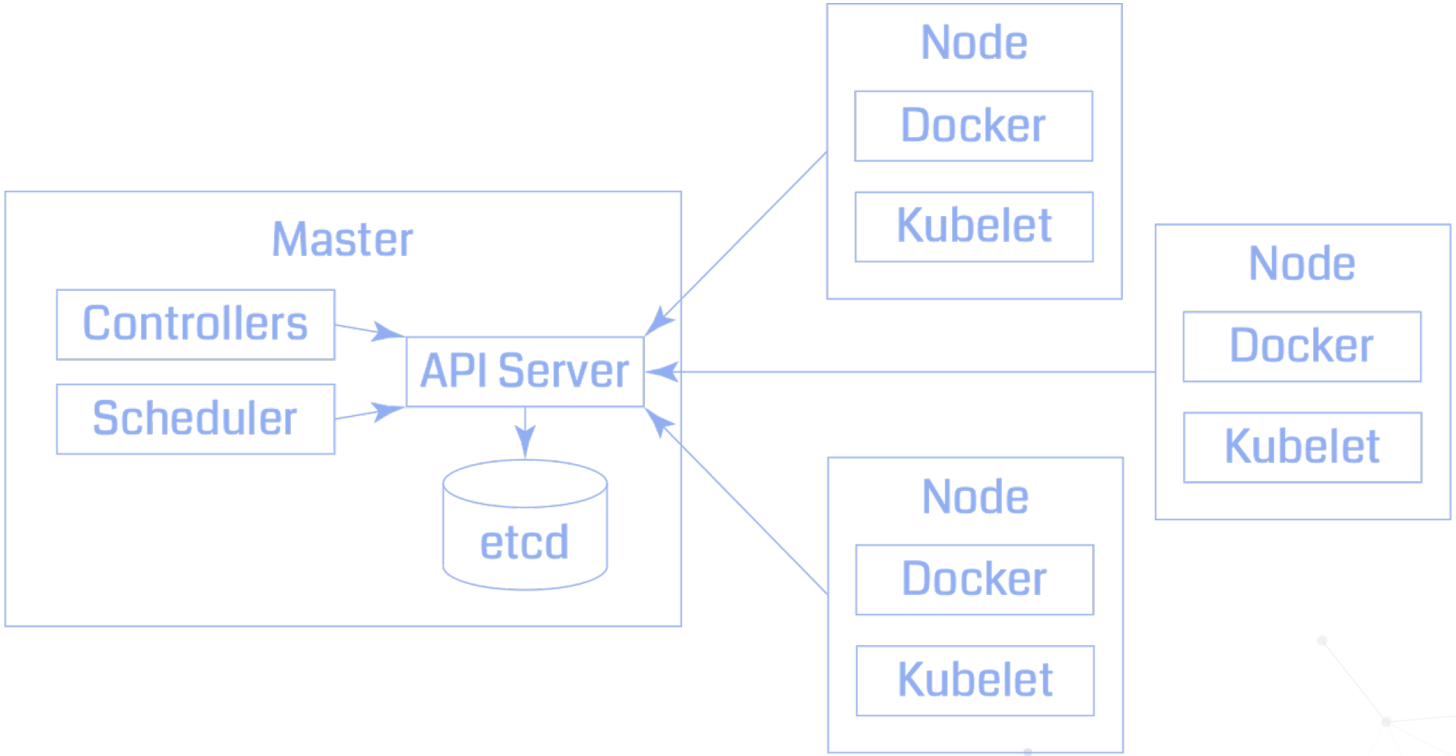
Kubernetes

- Kubernetes is an ***open-source*** *platform for automating deployment, scaling, and operations of application containers across clusters of hosts, providing container-centric infrastructure.*
- Kubernetes:
  - Places containers on nodes
  - Recovers automatically from failure
  - Basic monitoring, logging, health checking
  - Enables containers to find each other

Kubernetes principles

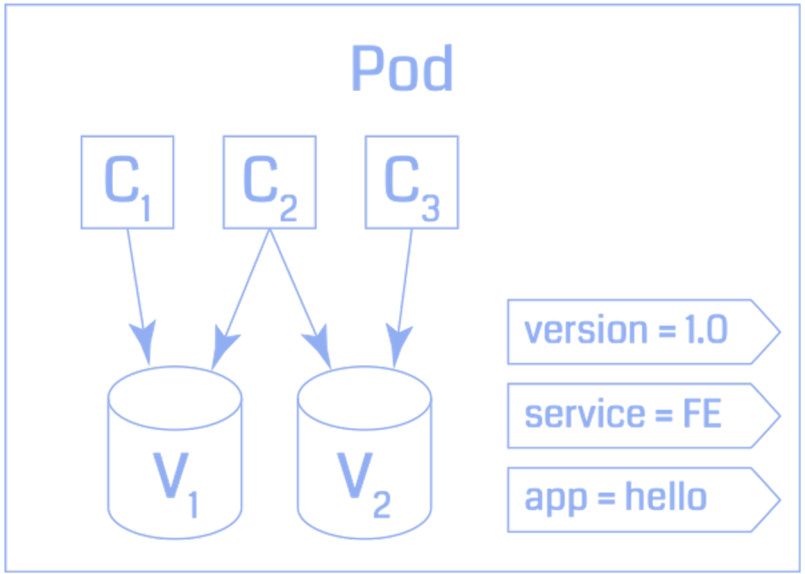
- Declarative > imperative
  - API driven application management
  - Agents monitor endpoints for state changes (real-time)
  - Controllers enforce desired state, self-healing
- No hidden APIs
  - Immutable
  - Modularity
- Workload portability
  - Decouple application development from cluster implementation
  - Cattle > pets

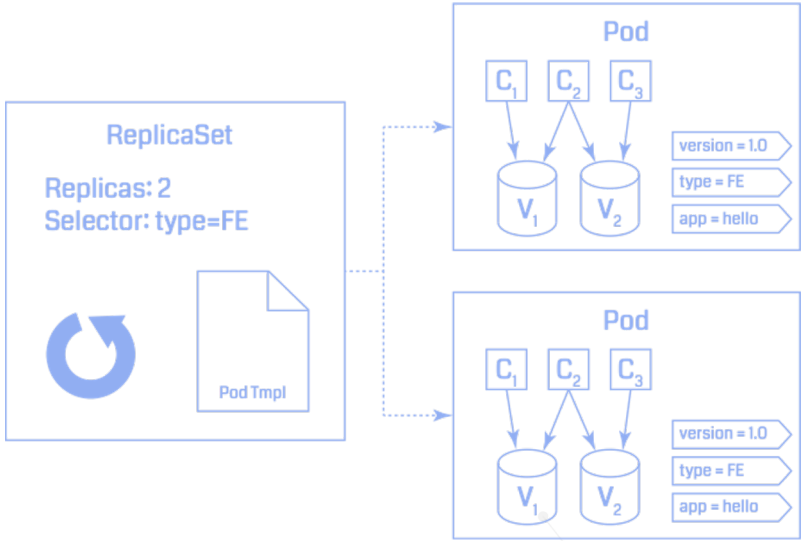
Kubernetes Cluster



Taken from https://github.com/jbeda/slides-kubernetes-101

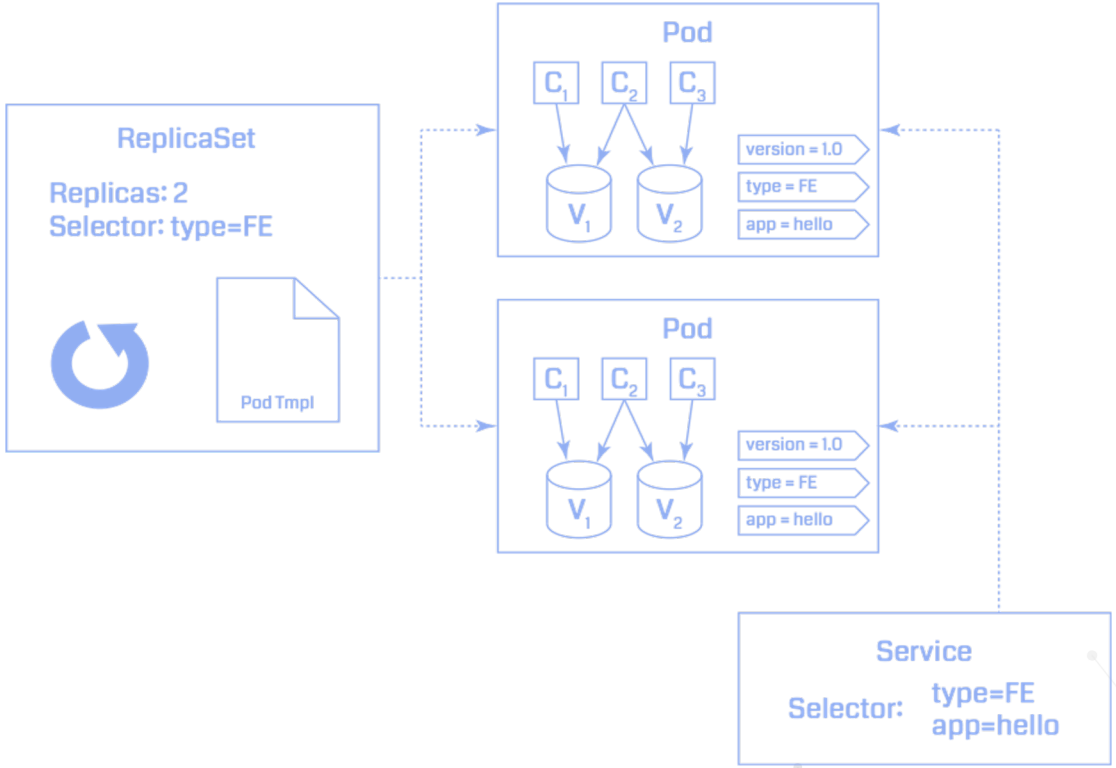
Kubernetes: Pod, ReplicaSets





Taken from https://github.com/jbeda/slides-kubernetes-101

Kubernetes Services



Taken from https://github.com/jbeda/slides-kubernetes-101

Kubernetes: sample

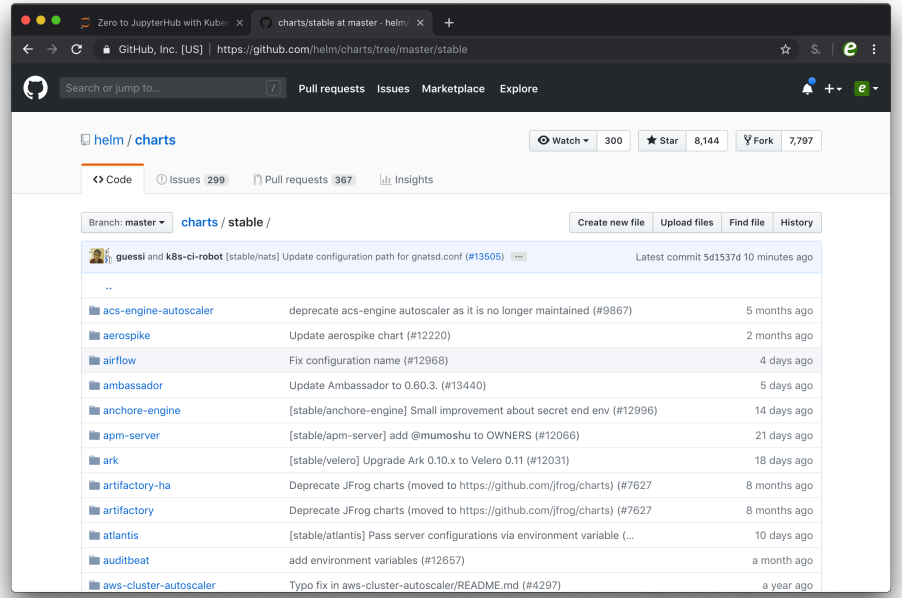
```yaml
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: frontend
  labels:
    app: guestbook
    tier: frontend
spec:
  # modify replicas according to your case
  replicas: 3
  selector:
    matchLabels:
      tier: frontend
  template:
    metadata:
      labels:
        tier: frontend
    spec:
      containers:
      - name: php-redis
        image: gcr.io/google_samples/gb-frontend:v3
```

```yaml
apiVersion: v1
kind: Service
metadata:
  name: my-service
spec:
  selector:
    app: MyApp
  ports:
    - protocol: TCP
      port: 80
      targetPort: 9376
```

Examples taken from kubernetes docs: https://kubernetes.io/docs

Helm

- Helm is a package manager for Kubernetes
- Allows to easily install applications/services easily
- Creates from templates all the k8s objects needed to run the application
- Wide catalogue of *charts* with (normally) sane defaults



Containers on EGI

Containers as jobs

Using HTC for running containers

- Docker requires root access to run your container
  - Need alternative runtimes for running jobs
- Singularity:
  - Container image and runtime specifically designed to run on multi-tenant computational resources
  - Conversion of images from docker (automatic)
  - Requires system administrator to configure
- udocker:
  - run applications encapsulated in docker containers without using docker, without using privileges, without system administrators intervention and without additional system software
  - Run as a normal user, with the normal process controls and accounting, in interactive or batch systems

udocker

- Can be installed on-the-fly as regular user as part of job preamble
  - Requirement: Python
- Provides a docker-like CLI
- Pulls images from Docker Hub
- Modular engine to execute images
- Support for GPGPUs and MPI applications

https://github.com/indigo-dc/udocker

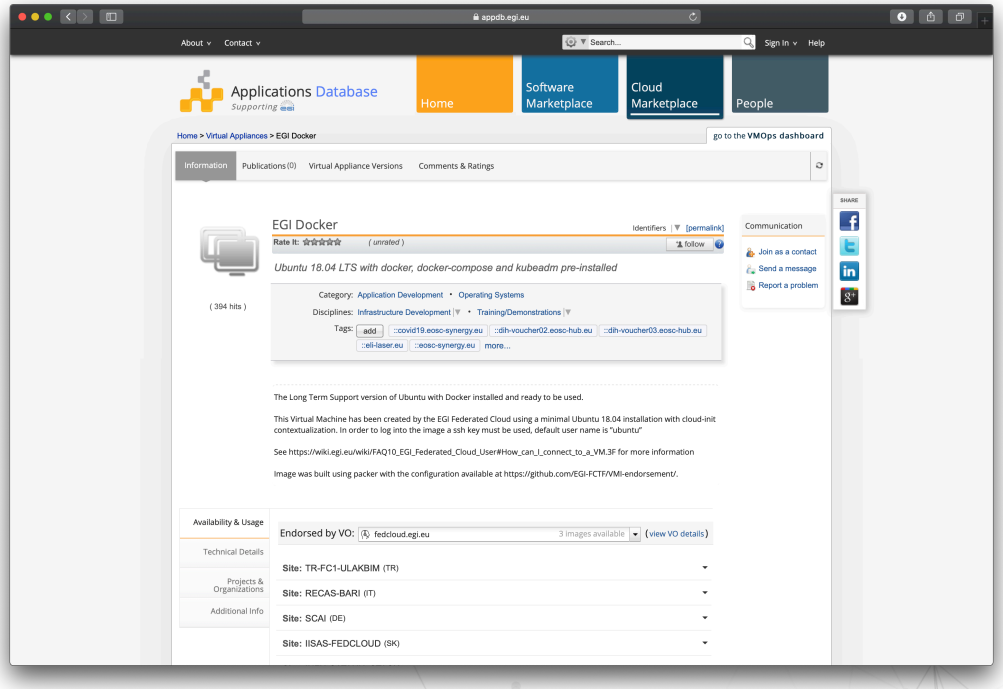
UDOCKER

Containers on EGI Cloud Compute

Run docker on VMs

- Pre-built image with Docker, Docker-compose and Kubernetes installed
- Instantiate at one provider, ssh and:

```
$ docker run …
```

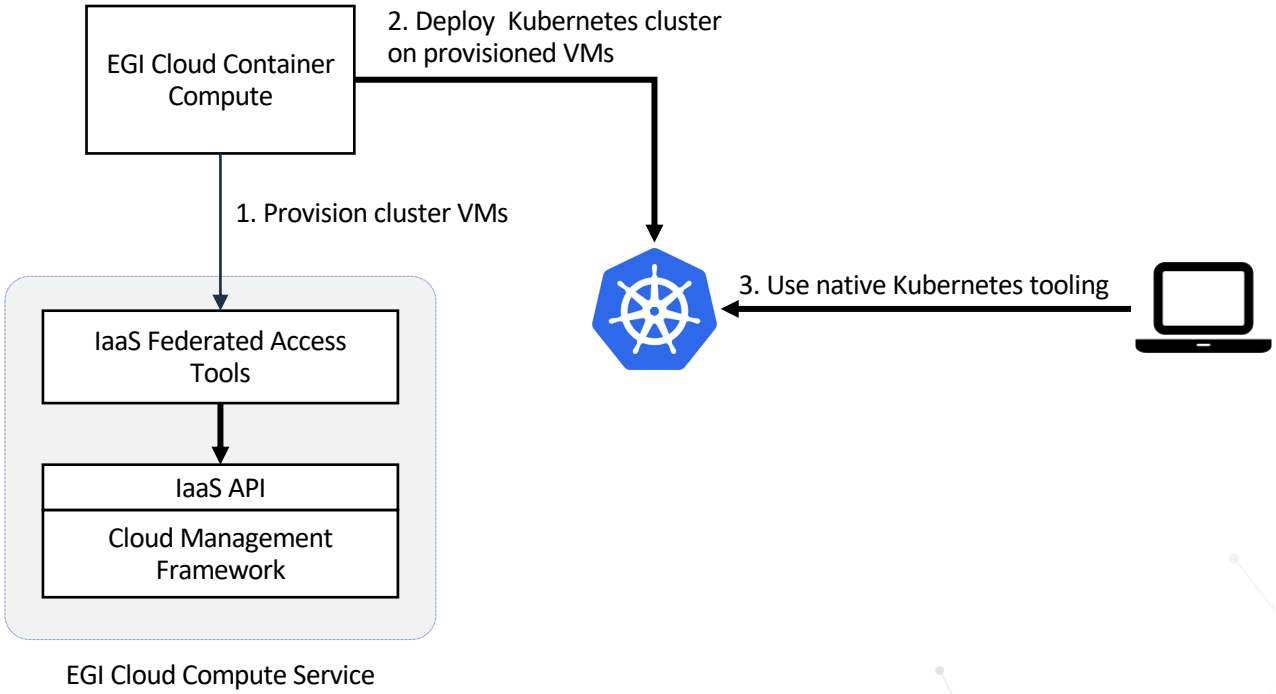


Cloud Container Compute [BETA]

Kubernetes @ EGI providers

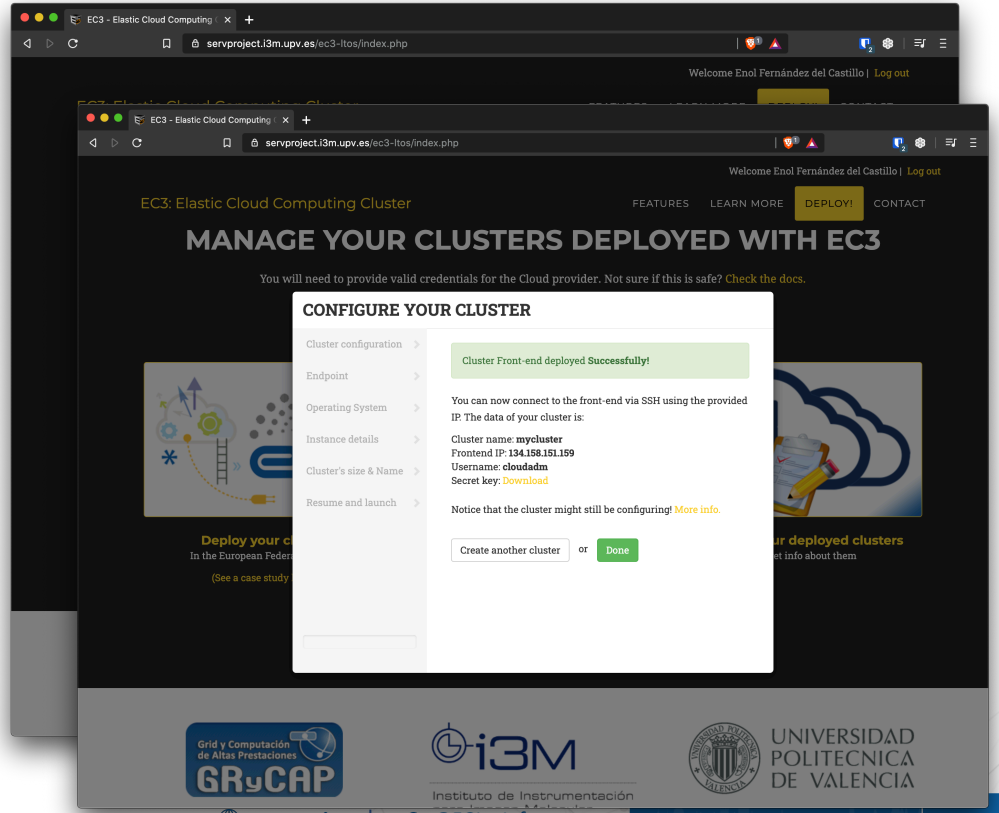
Cloud Container Compute gives you the ability to deploy and scale Docker containers on-demand using Kubernetes technology. The service provides with easy provision of Kubernetes clusters on EGI Cloud Compute resources that can be scaled and upgraded without the overhead of installing, managing and operating the nodes

Cloud Container Compute [BETA]



Deploy with EC3

- EC3: Elastic Cloud Computing Cluster
- Deploys and configures Kubernetes
  - Elasticity – can grow/shrink depending on your load
  - GUI and CLI access
  - Integrated with Check-in and AppDB
- Builds on ansible and kubeadm



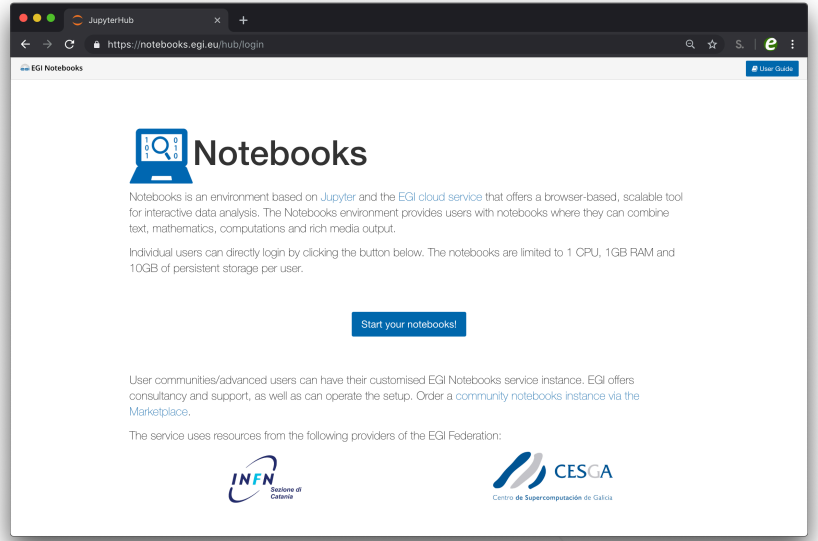
Kubernetes in EGI

- Provides Kubernetes 1.16+
- LoadBalancer ServiceType:
  - A NGINX ingress configured by default ready to be used offering similar functionality
  - Expandable with auto-configuration of Let's Encrypt certificates
- Volumes
  - NFS driver for flexible and shared volumes
  - Cinder (OpenStack volume) driver also available
- Check-in integration
  - Use Check-in tokens for AuthN & AuthZ in the Kubernetes API!
  - https://kubernetes.io/docs/reference/access-authn-authz/authentication/#openid-connect-tokens

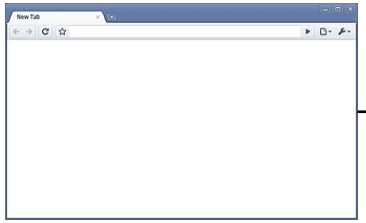
EGI Notebooks

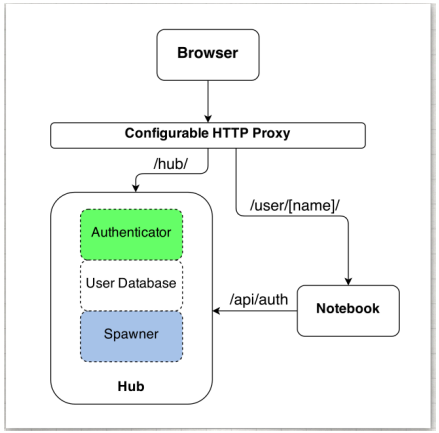
Running on top of Kubernetes

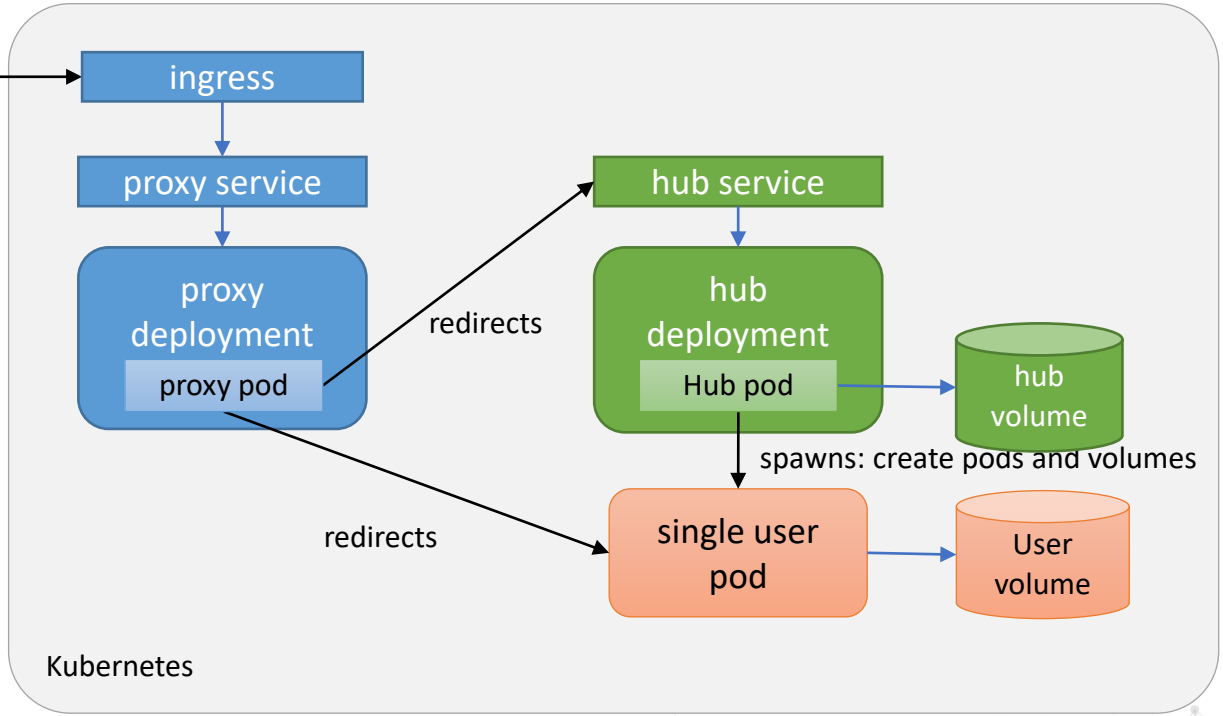
- JupyterHub hosted in the EGI Cloud
  - Offers Jupyter notebooks ‘as Service’
  - One-click solution: login and start using



Kubernetes and JupyterHub







Documentation

- User documentation:
  - https://egi-federated-cloud.readthedocs.io/en/latest/

EGI: Advanced Computing for Research

www.egi.eu
@EGI_eInfra

Thank you
for your attention.

Questions?

This work by the EGI Foundation
is licensed under a Creative Commons Attribution 4.0 International License.

The work of the EGI Foundation
is partly funded by the European Commission under H2020 Framework Programme