# Ultra HA platform: Kubernetes federation

Marco Lorini
Matteo Di Fazio
Alex Barchiesi

GARR

WORKSHOP GARR 2020

NET MAKERS

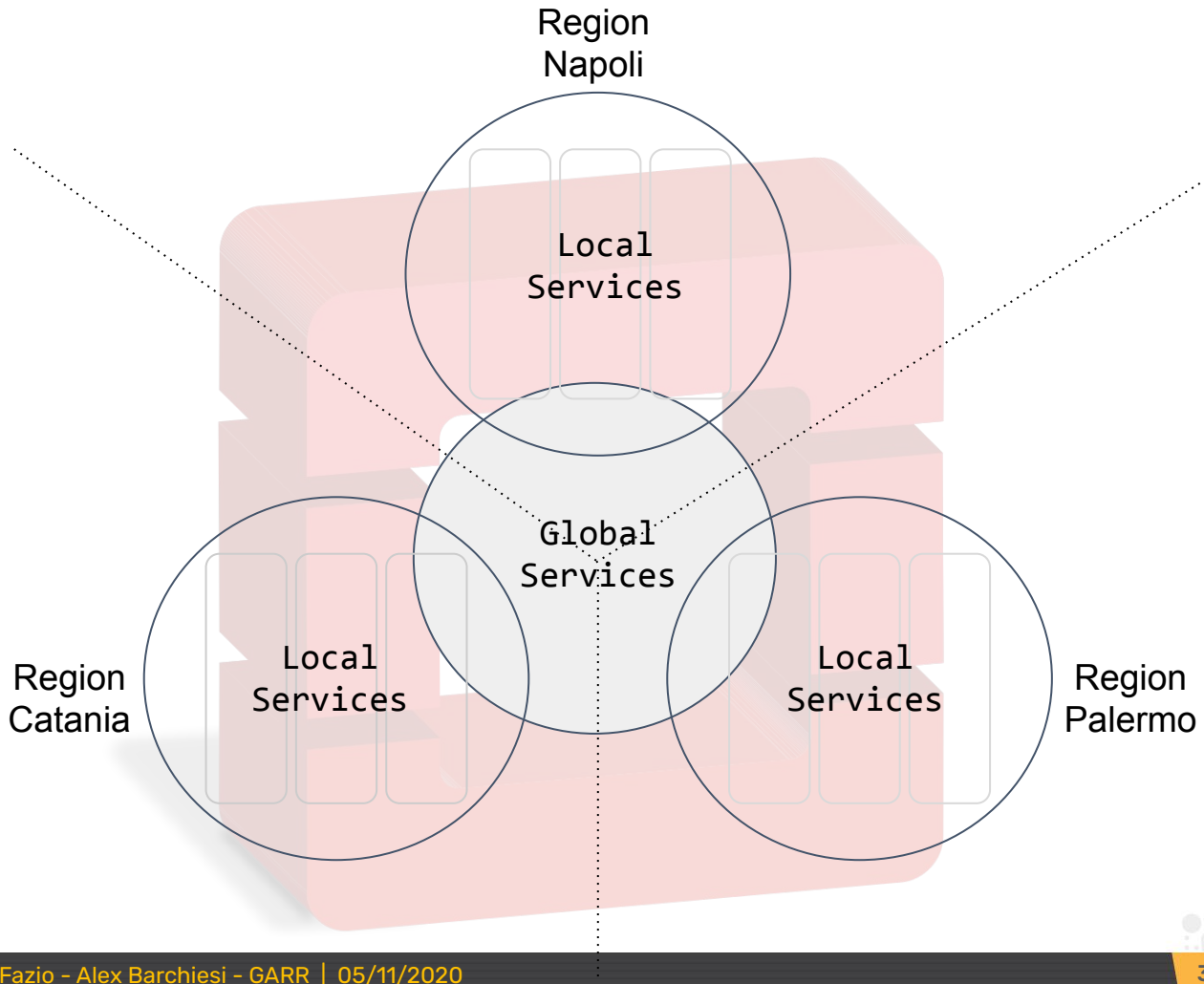
Consortium GARR

# 6000 core

# 10 PB

# 8 GPU

... 8 rack/CSD-modules


Region
Napoli
dns
Local
Services
Global
Services
Region
Catania
Local
Services
Local
Services
Region
Palermo

# Container platform: Kubernetes for users

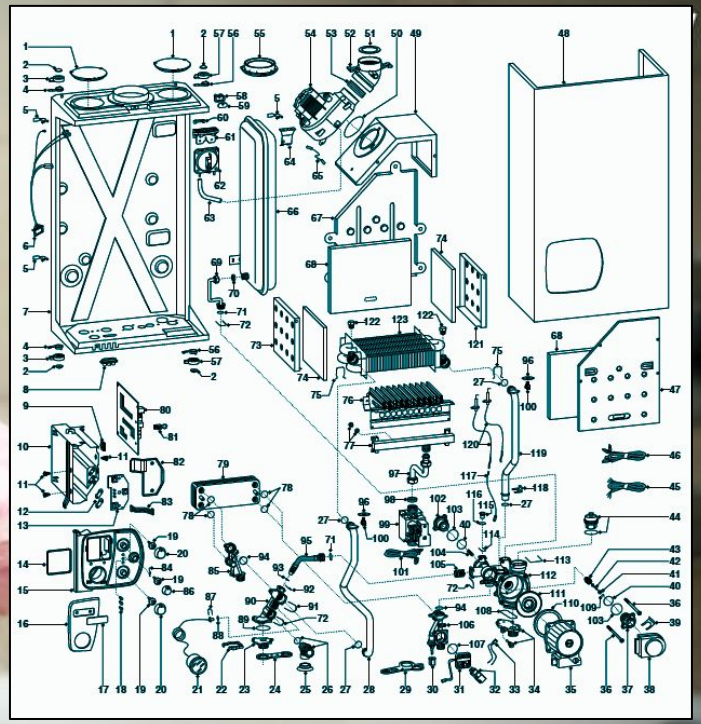
- bare metal cluster 200CPU @ Palermo
- Storage class @ CEPH cluster
- 4 GPU con scheduling per utenti
- HELM charts per package management (ongoing)
- Accesso integrato con I.a.a.S. via application credentials
    - sviluppo fatto in CSD e inserito in main stream 0~S
    - integrato con dashboard 0~S per UX

Container platform (Kubernetes): la comunità la usa principalmente per GPU

- Esempi: Virgo, UniMIB, IRCCS

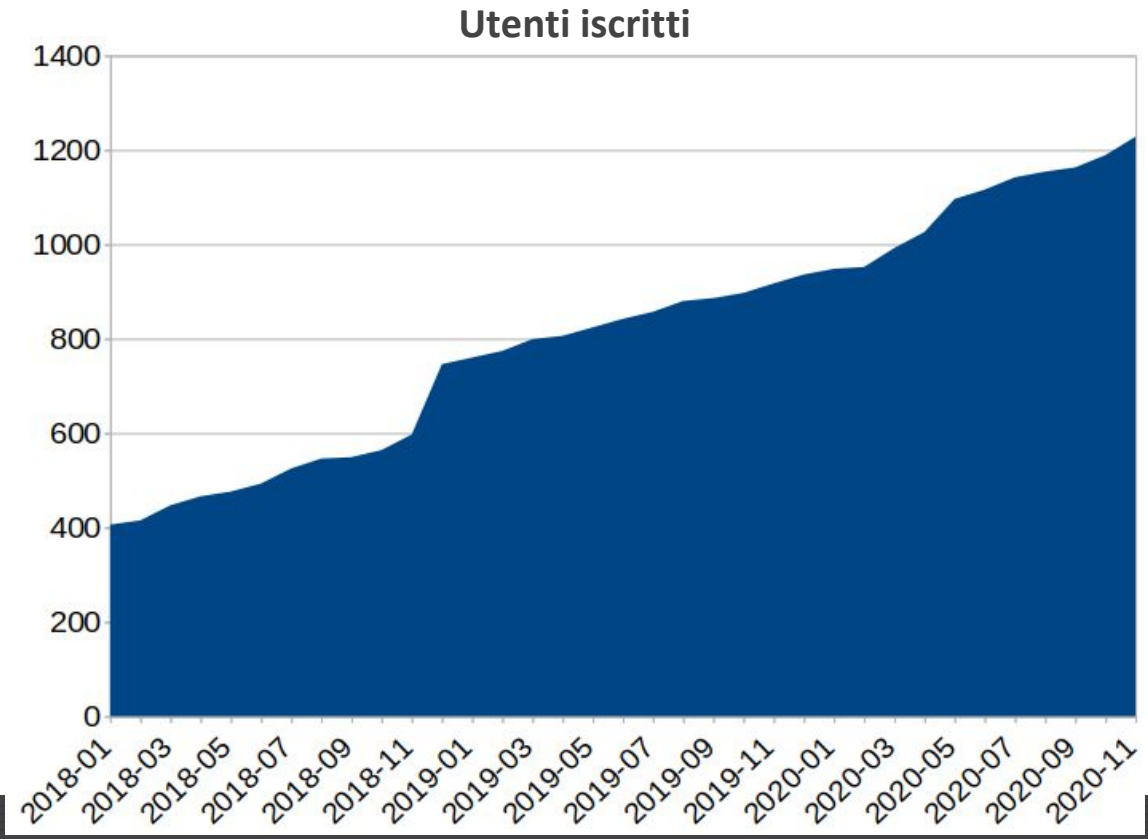
Molte richieste ricevute: large k8s provision, Storage request, backup, business continuity…

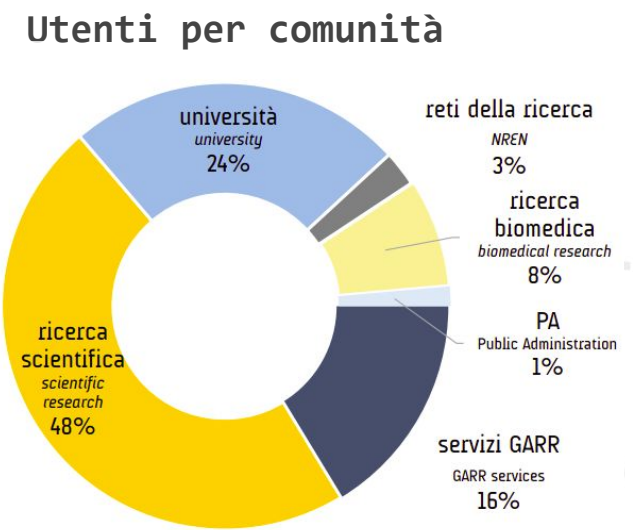
- OK per storage e processamento dati
- Non abbiamo (ancora) soluzioni per Business Continuity e Backup (Nastro ?)

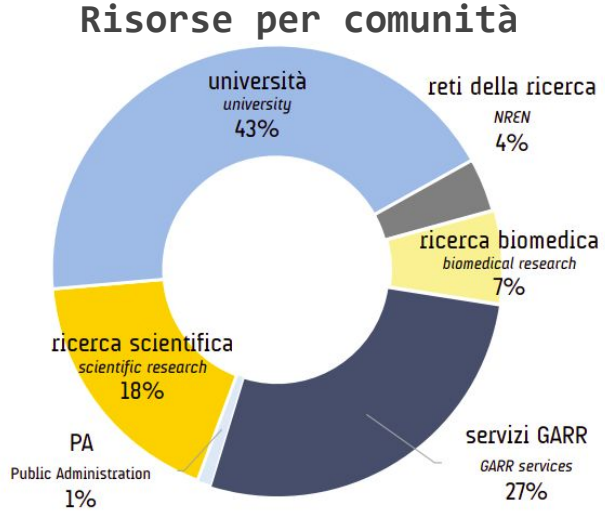


BETTY GILLIS

# Cloud Status

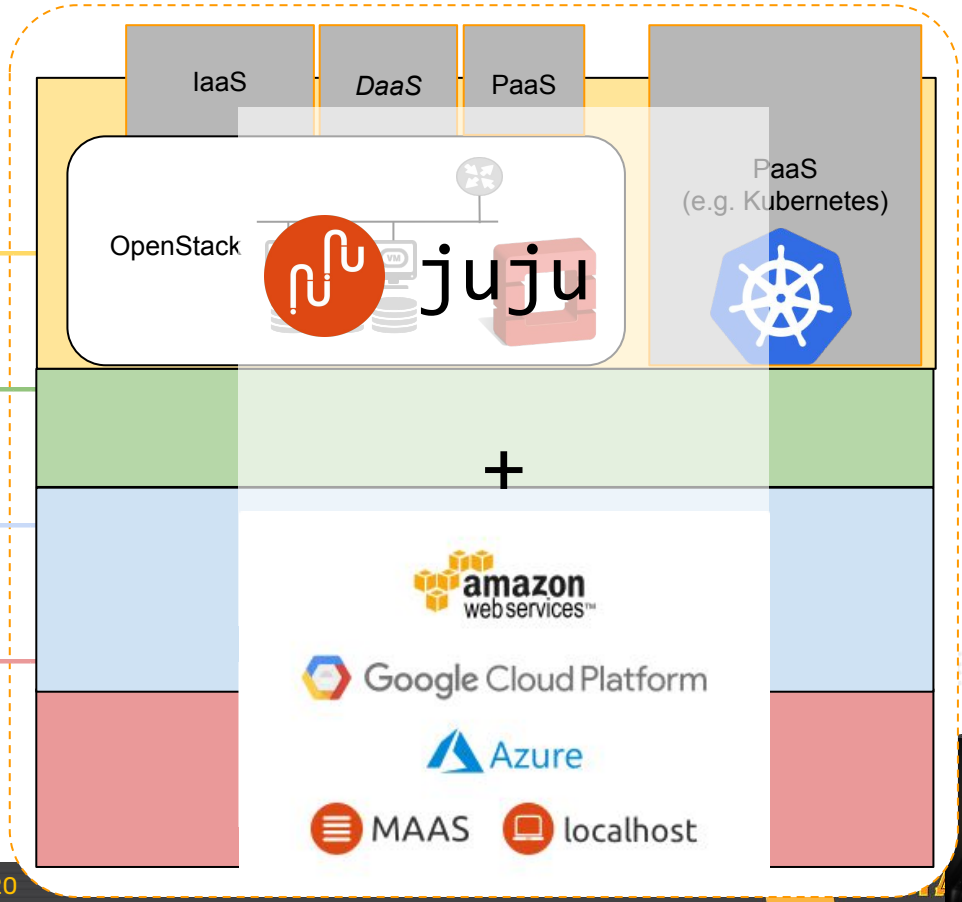


# automation recipe:

- Application Services
- Infrastructure *Virtualization*
- Operating System
- Physical resources




# **Federazione:** modello *from 0 to everything*
(con accesso federato idem)

Prerequisiti per la federazione:

- Hardware montato e “spina inserita”
- I server devono poter accedere alla rete per l’installazione del software
- Rete ILO accessibile via IPMI-over-LAN

Ricetta in tre *ingredienti*:

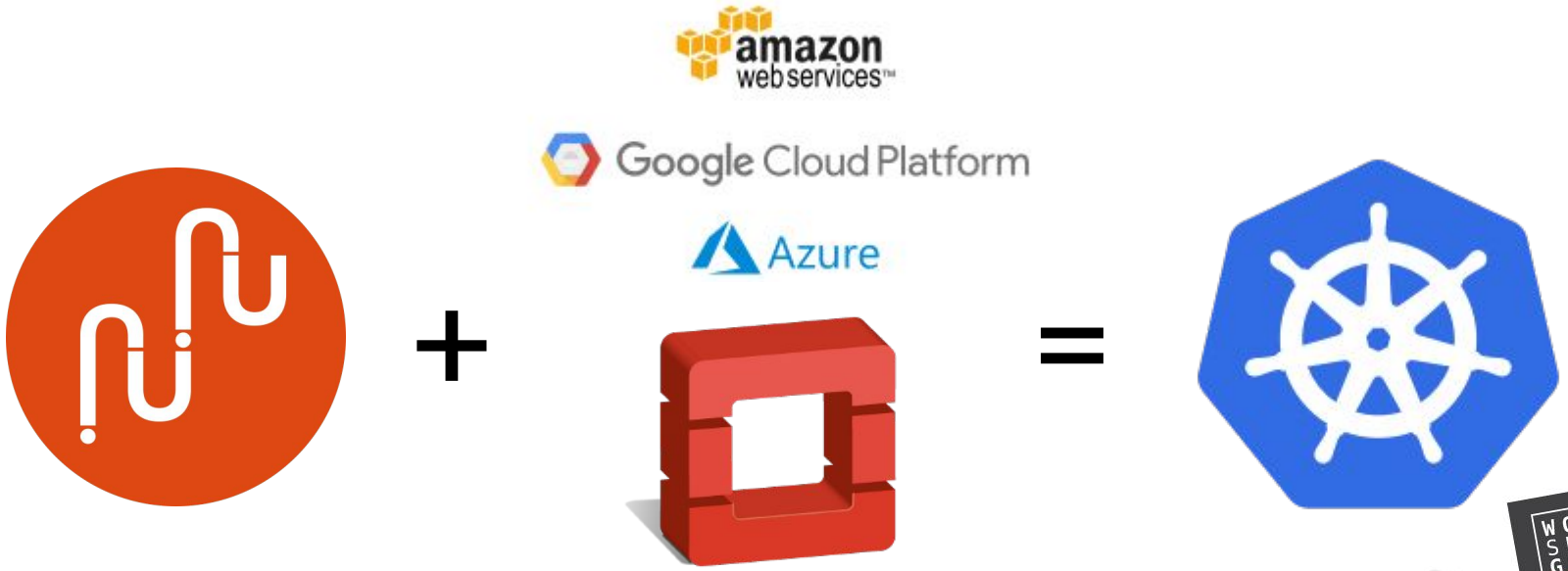
- Installazione e configurazione dei tool di **automazione** (MAAS e Juju);
- deploy e configurazione della regione **OpenStack** (via Juju);
- **Trasferimento credenziali di sistema** ed endpoint URL sulla regione centrale (ad-hoc scripts)

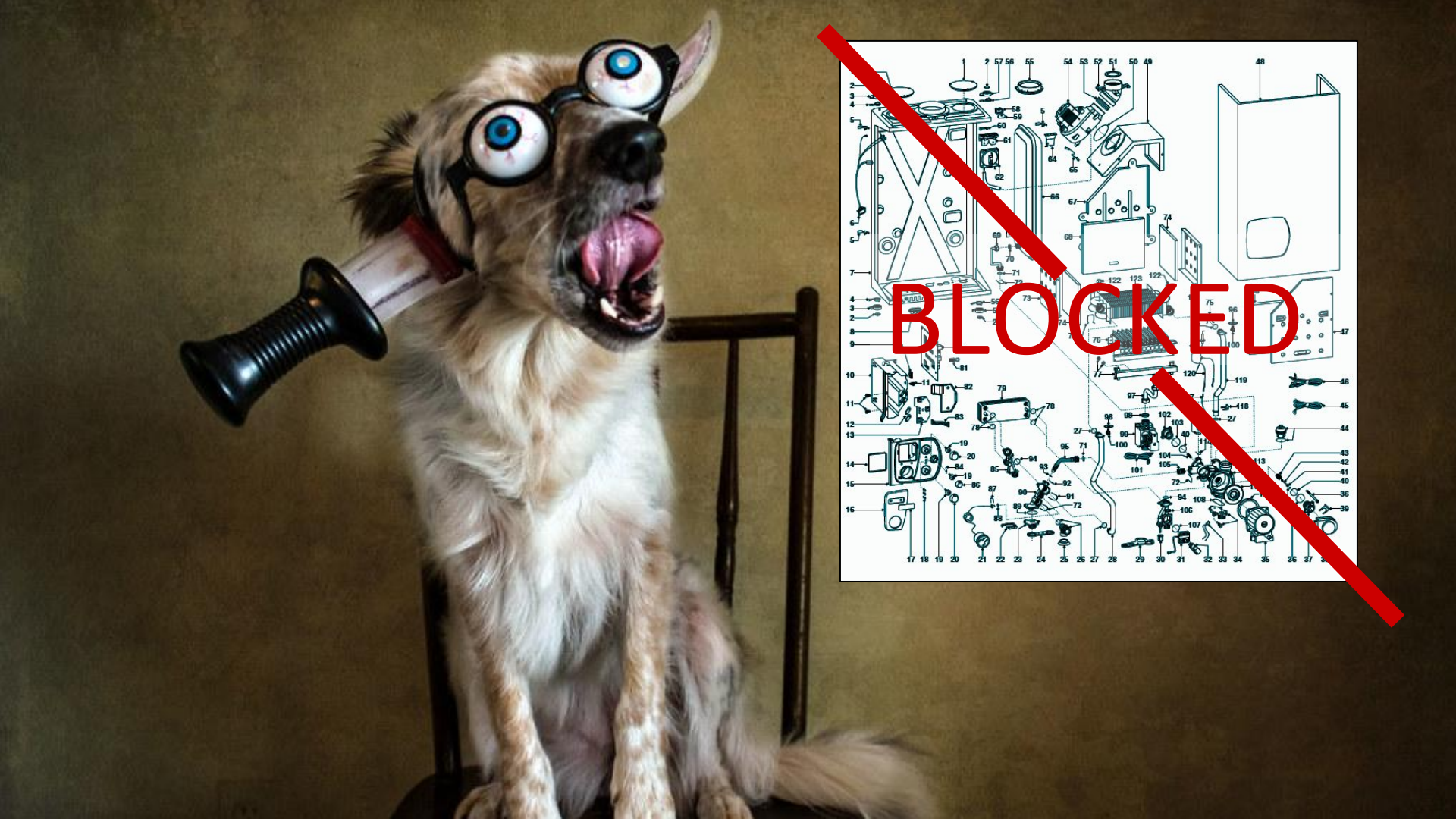
Operations:

- registrazione utenti attraverso credenziali idem
- amministratori locali della regione assegnano le risorse

# Kubernetes as a Service for admins

- cluster on demand o self-service
- OpenStack project backend (or any backend)

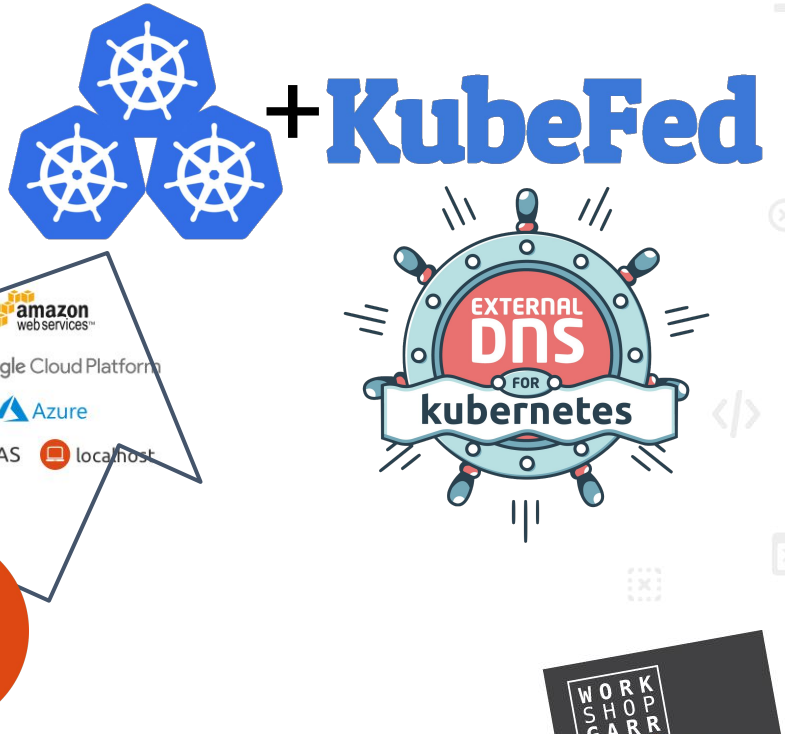



BLOCKED

# *Ultra* HA platform: Kubernetes federation

fornire HA nativa su multisito e/o cloud provider terzi

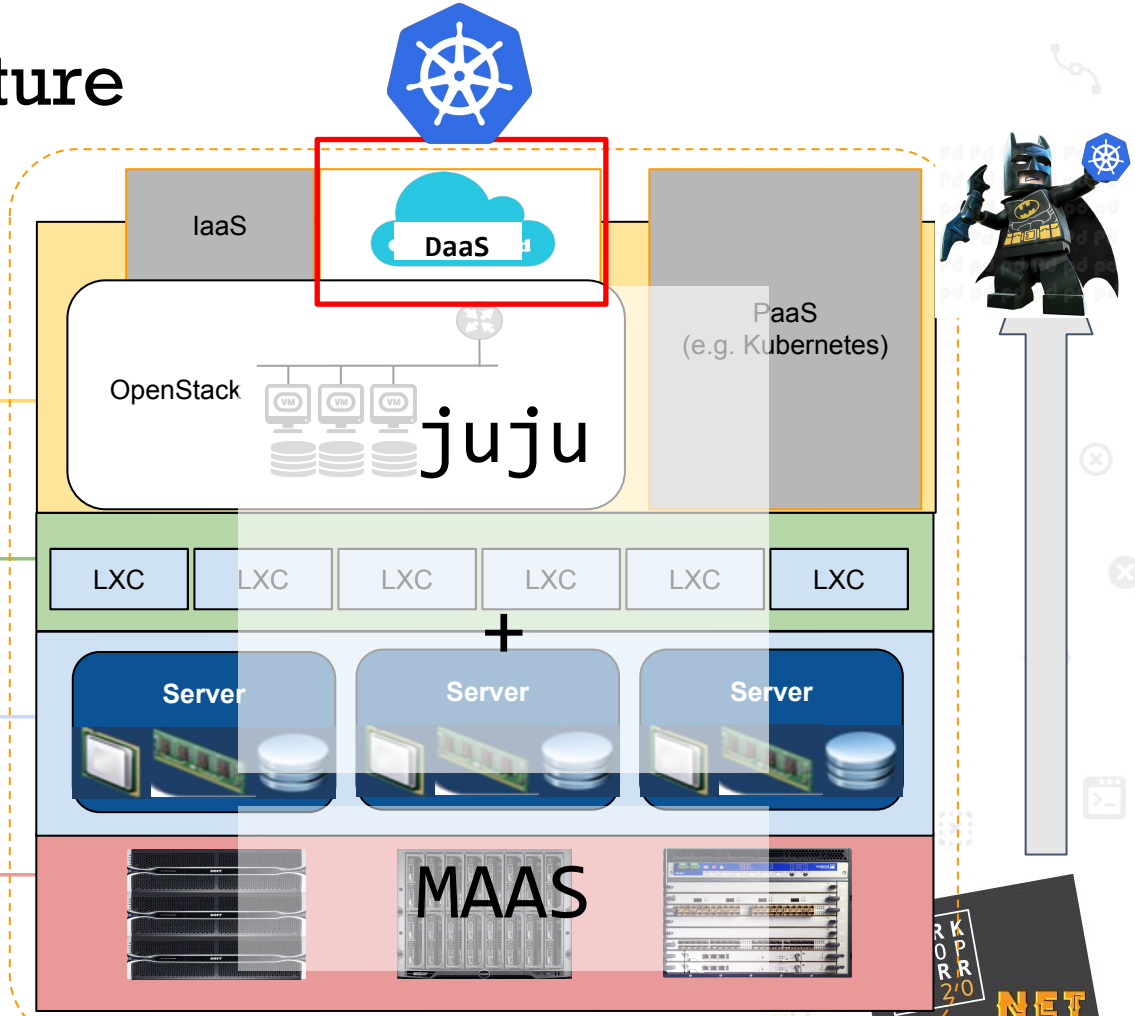
- any undercloud
- Multi-Cluster Ingress DNS
- ~transparent for User
- (allo studio/test)
  - redundancy cluster host
  - Multi-region storage HA



# passaggio token

# GARR Cloud architecture

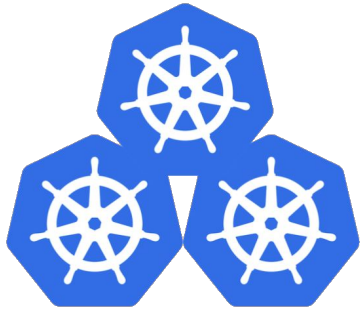
1. Application Services
2. Infrastructure *Virtualization*
3. Operating System
4. Physical resources



# What we asked ourselves

Can we have an infrastructure to run services on that also frees the user from thinking how to achieve a multi-region HA deployment?

# Our answer



KubeFed

# Topics

- Kubernetes Cluster Federation (KubeFed)
  - Actors
  - Propagation mechanism
  - Architecture


- External-DNS
  - Multi-Cluster DNS

# KubeFed

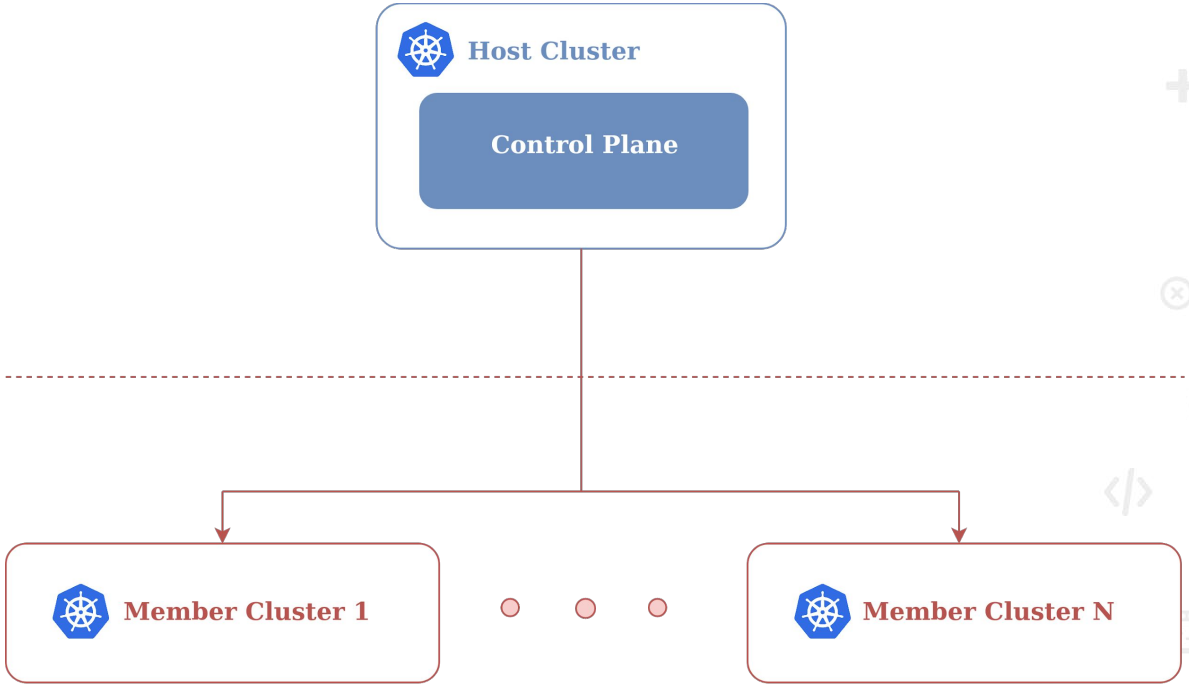
- KubeFed is a tool that allows the coordination and configuration of multiple Kubernetes clusters

- Provides a mechanism for managing multi-region applications and disaster recovery

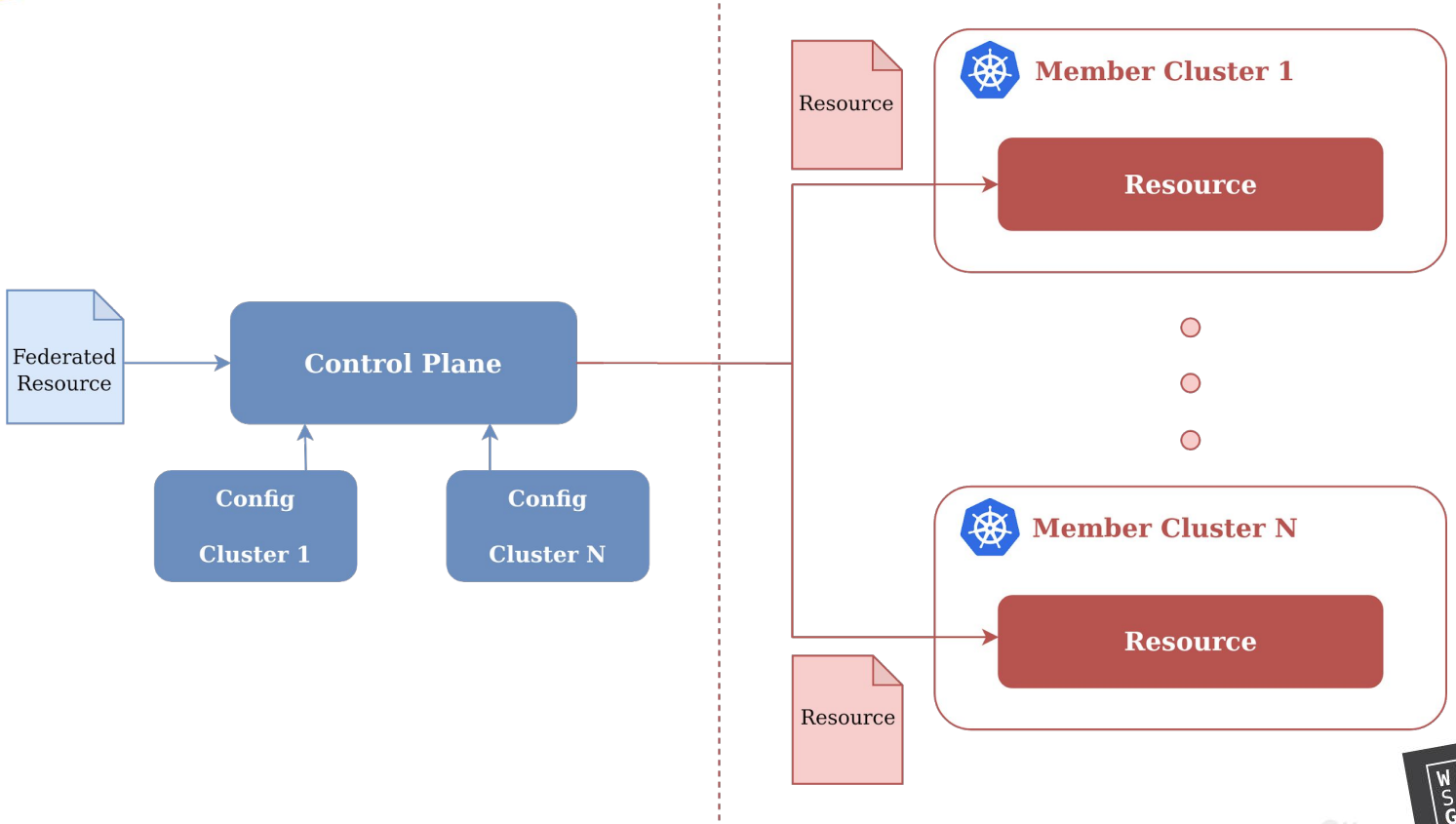
KubeFed

# KubeFed actors

- Host cluster
- Member cluster



# Propagation



# Federated resources

- **Template**: represents the classic definition of the resource

- **Placement**: the set of member clusters on which you intend to federate the resource

- **Overrides**: the set of changes that you want to apply to the resource for specific clusters (optional)

# Federated resources

Type of kind:

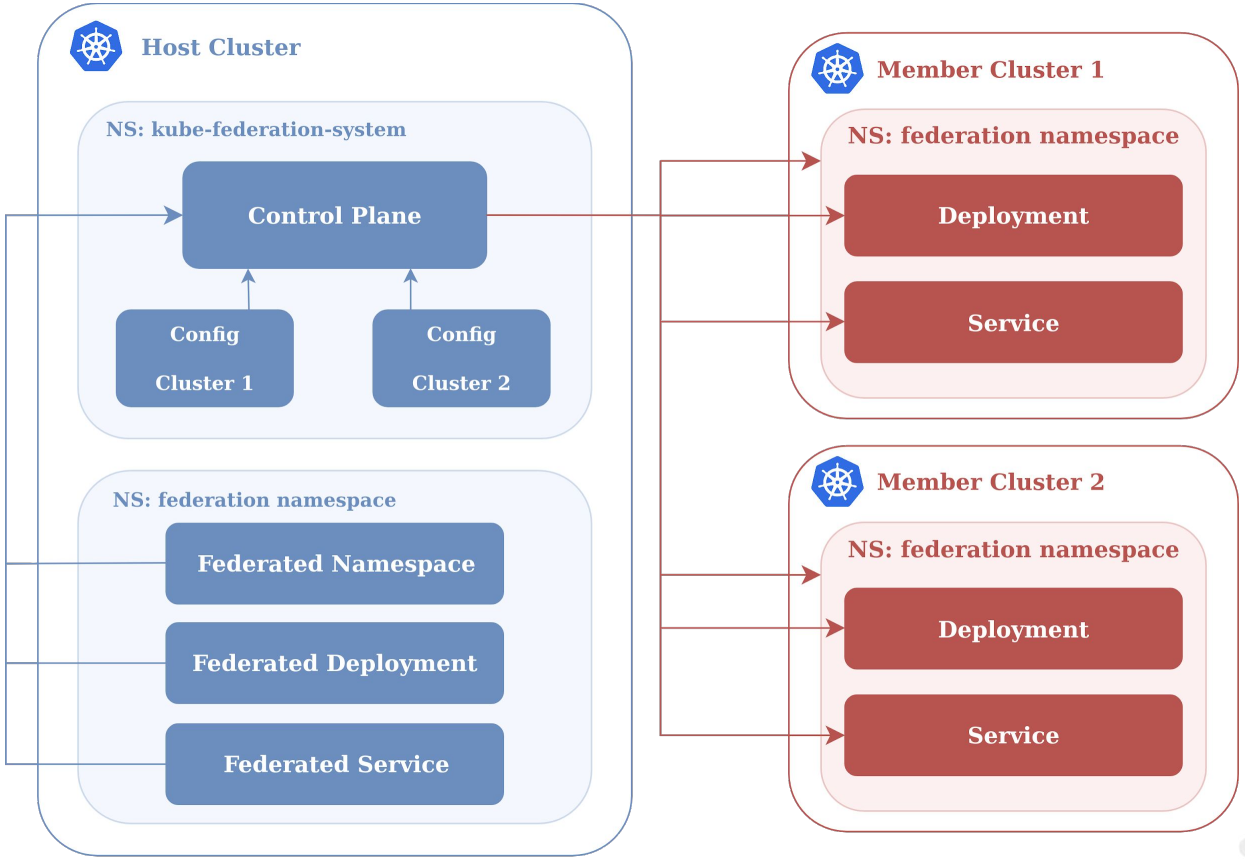
- FederatedNamespace
- FederatedDeployment
- FederatedService
- FederatedIngress
- …

```
# federated resource structure
apiVersion: types.kubefed.io/v1beta1
kind: <federated-resource-type>
metadata:
  name: <federated-resource-name>
  namespace: <federated-namespace>
spec:
  template:
    <resource-spec>
  placement:
    <list-member-cluster>
  overrides:
    <config-changes>
```

# Federated resources: example

```yaml
apiVersion: types.kubefed.io/v1beta1
kind: FederatedDeployment
metadata:
  name: fed-helloworld
  namespace: fed-namespace
spec:
  template:
    metadata:
      name: fed-helloworld
      namespace: fed-namespace
    spec:
      replicas: 2
      selector:
        matchLabels:
          app: helloworld
      template:
        metadata:
          labels:
            app: helloworld
        spec:
          containers:
          - image: docker.io/csdgarr/hello-world:v1
            imagePullPolicy: IfNotPresent
            name: helloworld
  placement:
    clusters:
    - name: member-cluster-1
    - name: member-cluster-2
  overrides:
    - clusterName: member-cluster-2
      clusterOverrides:
      - path: "/spec/replicas"
        value: 3
```

# Architecture



# And now?

How can we automate the management of DNS records for our applications and services?

# ExternalDNS

- ExternalDNS is a tool that makes services deployed on Kubernetes reachable through DNS servers

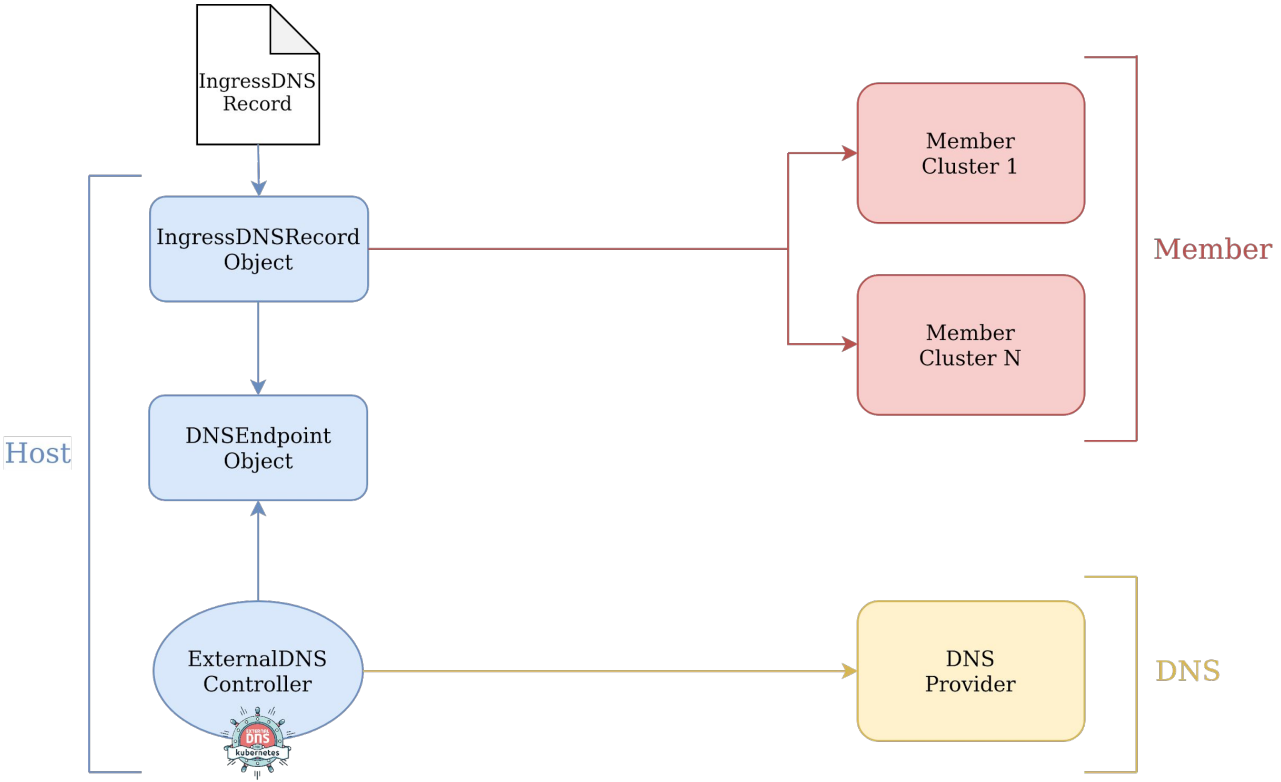
- It retrieves a list of ingress and service resources and creates/updates records (based on the info retrieved) in a DNS server

- How does it retrieve the resources?

# Multi-Cluster DNS

- The mechanism that retrieves informations from service and ingress resources located in multiple Kubernetes clusters
- Used in federation contexts
- Integrates with External DNS

# Multi-Cluster DNS



# IngressDNSRecord

```yaml
# create_ingressdnsrecord.yaml
apiVersion: multiclusterdns.kubefed.io/v1alpha1
kind: IngressDNSRecord
metadata:
  name: <ingress-name>
  namespace: <federated-namespace>
spec:
  hosts:
  - <domain>
  recordTTL: 300
```

# Example IngressDNSRecord Object

```
marco@marco-xps:~$ kubectl describe ingressdnsrecords fed-helloworld-ingress -n fed-namespace
Name:         fed-helloworld-ingress
Namespace:    fed-namespace
Labels:       <none>
Annotations:  <none>
API Version:  multiclusterdns.kubefed.io/v1alpha1
Kind:         IngressDNSRecord
Metadata:
  Creation Timestamp:  2020-04-15T13:35:26Z
  Generation:          1
  Resource Version:    8718306
  Self Link:           /apis/multiclusterdns.kubefed.io/v1alpha1/namespaces/fed-namespace/ingressdnsrecords/fed-helloworld-ingress
  UID:                 0051a39f-0d73-4dbe-af5f-7ba428082eb2
Spec:
  Hosts:
    helloworld.test.global.garrservices.it
  Record TTL:  300
Status:
  Dns:
    Cluster:  fed-cluster-ct1
    Load Balancer:
      Ingress:
        Ip:  90.147.166.11
        Ip:  90.147.167.127
        Ip:  90.147.167.68
    Cluster:  fed-cluster-na
    Load Balancer:
      Ingress:
        Ip:  90.147.152.69
        Ip:  90.147.152.76
        Ip:  90.147.152.83
Events:       <none>
```

# Example DNSEndpoint Object

```
marco@marco-xps:~$ kubectl describe dnsendpoints ingress-fed-helloworld-ingress -n fed-namespace
Name:         ingress-fed-helloworld-ingress
Namespace:    fed-namespace
Labels:       <none>
Annotations:  <none>
API Version:  multiclusterdns.kubefed.io/v1alpha1
Kind:         DNSEndpoint
Metadata:
  Creation Timestamp:  2020-04-15T13:35:26Z
  Generation:          2
  Resource Version:    8718413
  Self Link:           /apis/multiclusterdns.kubefed.io/v1alpha1/namespaces/fed-namespace/dnsendpoints/ingress-fed-helloworld-ingress
  UID:                 c0e12bfa-4059-44a4-9ee7-781afe93a351
Spec:
  Endpoints:
    Dns Name:     helloworld.test.global.garrservices.it
    Record TTL:   300
    Record Type:  A
    Targets:
      90.147.152.69
      90.147.152.76
      90.147.152.83
      90.147.166.11
      90.147.167.127
      90.147.167.68
Status:
  Observed Generation:  2
Events:                 <none>
```

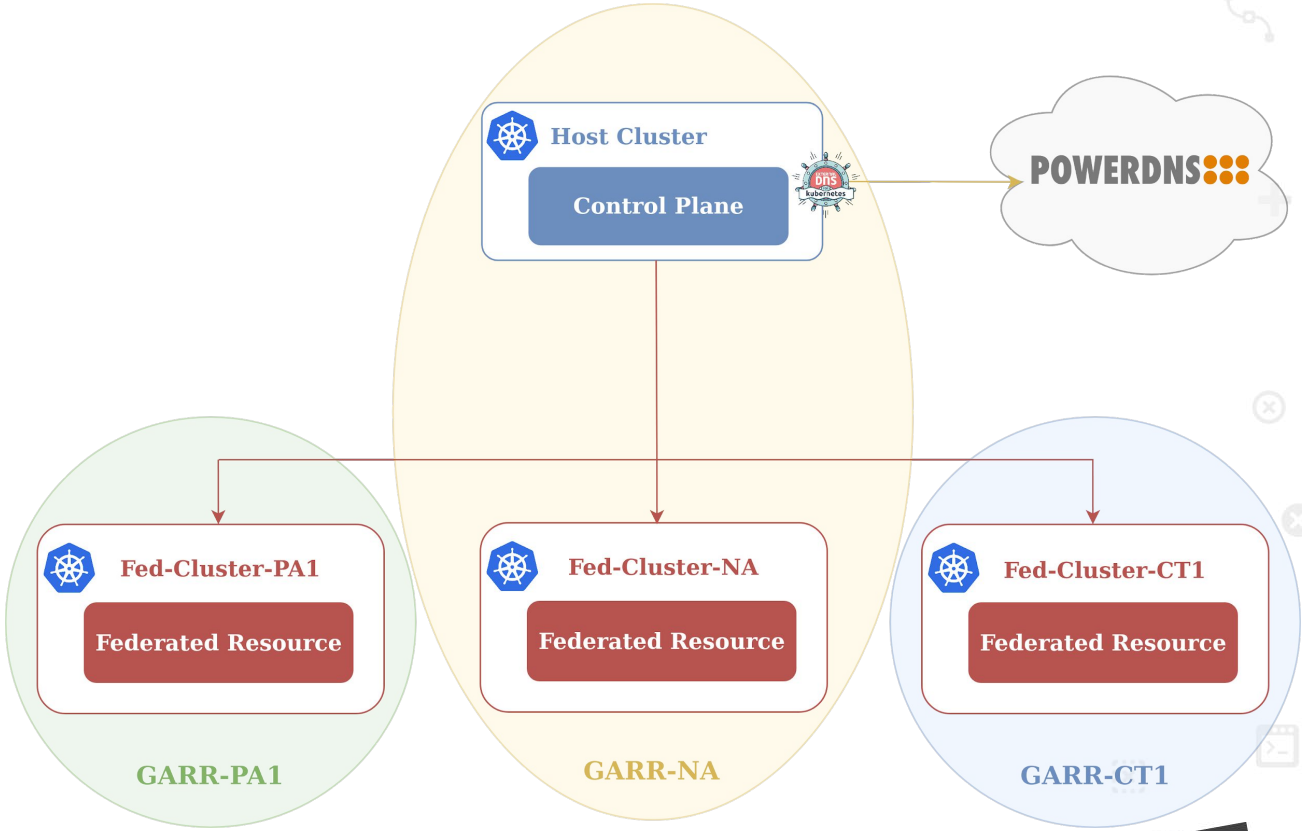
# PowerDNS records

| | | | | | | |
|---|---|---|---|---|---|---|
| cnamehelloworld | TXT | Active | 300 | "heritage=external-dns,external-dns/owner=default,external-dns/resource=crd/fed-namespace/ingress-fed-helloworld-ingress" | Edit | Delete |
| helloworld | A | Active | 300 | 90.147.152.69 | Edit | Delete |
| helloworld | A | Active | 300 | 90.147.152.76 | Edit | Delete |
| helloworld | A | Active | 300 | 90.147.152.83 | Edit | Delete |
| helloworld | A | Active | 300 | 90.147.166.11 | Edit | Delete |
| helloworld | A | Active | 300 | 90.147.167.127 | Edit | Delete |
| helloworld | A | Active | 300 | 90.147.167.68 | Edit | Delete |

POWERDNS

# Our architecture

- Host Cluster (garr-na):
  - runs KubeFed Control Plane
  - runs ExternalDNS configured to interact with our PowerDNS server
- Member Clusters:
  - three clusters in three different regions (garr-ct1, garr-pa1 and garr-na)
  - run applications and services



# Future works

- Ensure a system of redundancy for the cluster host
- Multi-region data redundancy (infrastructure level)

# Link Docs

- https://github.com/kubernetes-sigs/kubefed
- https://github.com/kubernetes-sigs/external-dns
- https://github.com/kubernetes-sigs/kubefed/blob/master/docs/ingressdns-with-externaldns.md

# Link Git

- https://git.garr.it/CSD/public/kubefed

# Q&A

Mentimeter

96 53 88 5

Cloud Federata

https://bbb.meet.garr.it/b/fed-phc-zdk-9b6

# Thanks for your attention!