Building and operating the CrownLabs Service

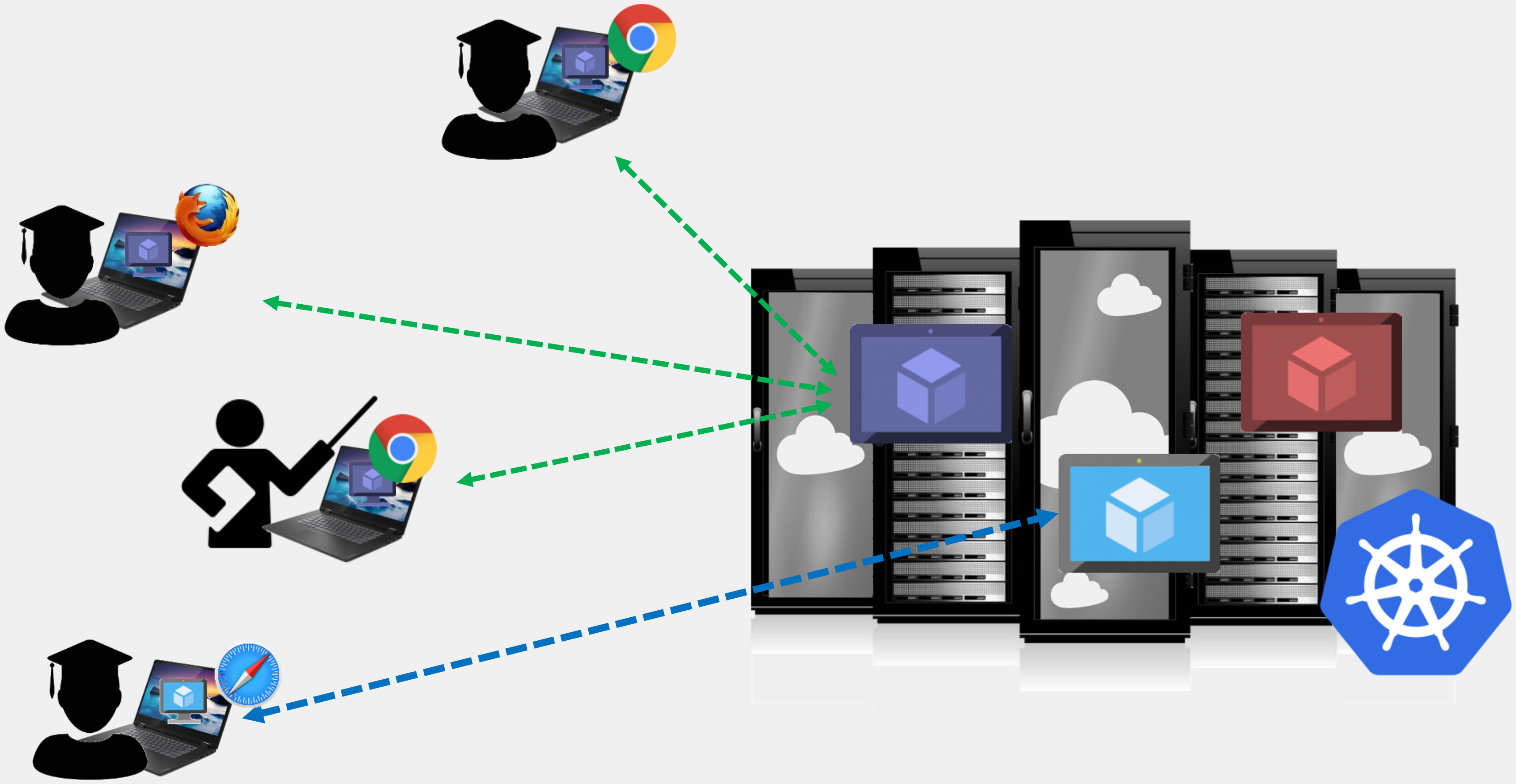
Architecture and lessons learned

Marco Iorio

Politecnico di Torino

Why CrownLabs

A Collaborative Learning Environment



Main Strong Points

Synchronous Collaboration

- Group works and peer support
- Simplified tutoring

Flexibility

- You own your environment
- Available 24/7

Versatility

- Multiple environments
- Tailored setups

Security

- Isolated environments
- Authentication and authorization

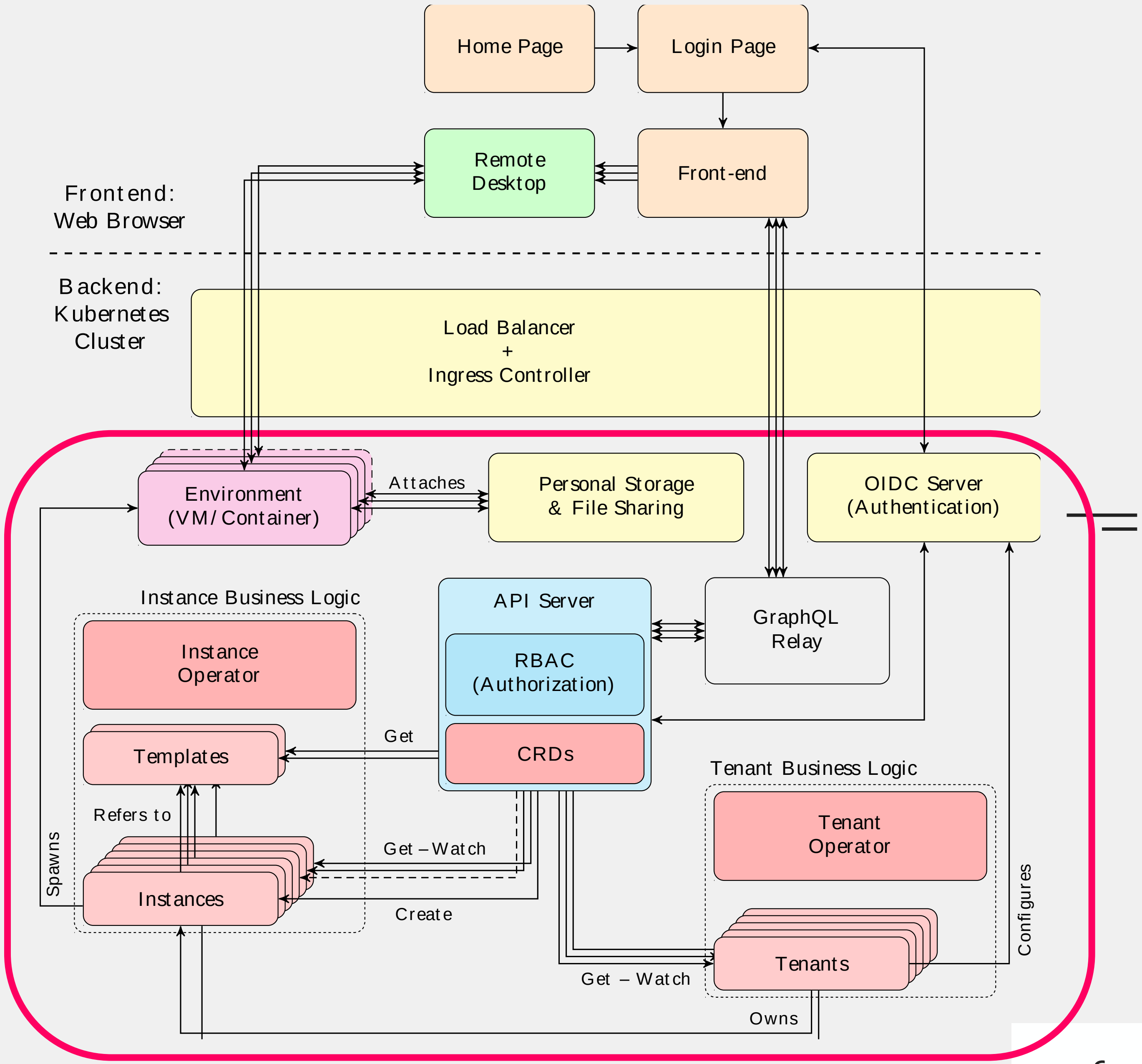
Compliance

- Access to licensed software

How does CrownLabs work?

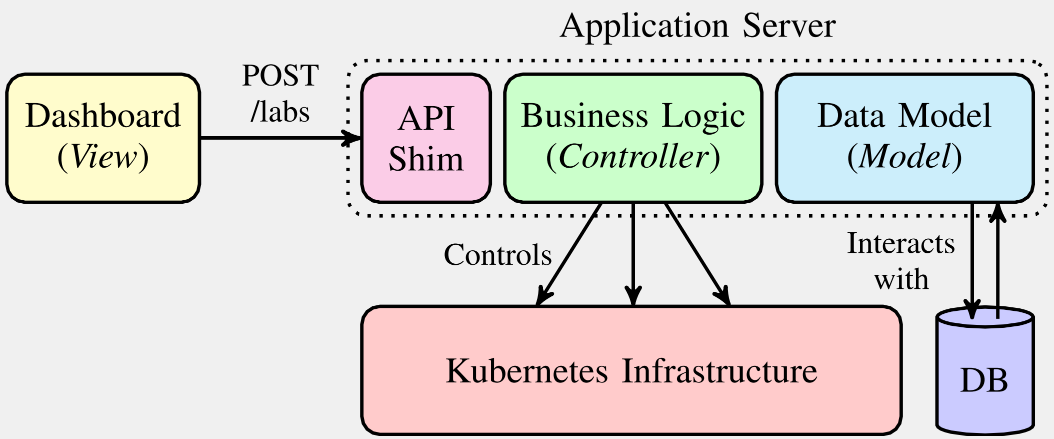
Let’s deep dive!

The CrownLabs Architecture

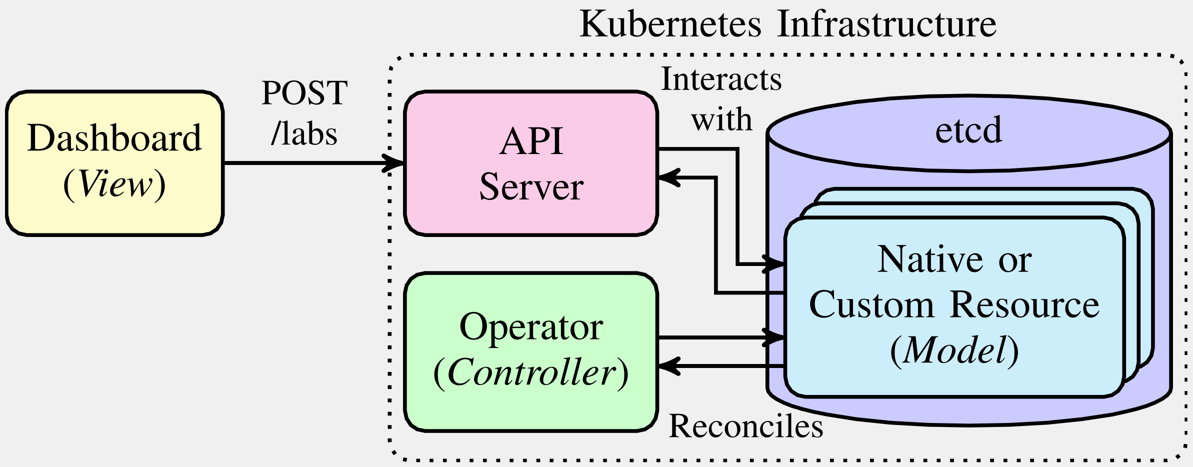


A k8s-powered application backend

The traditional Approach



The CrownLabs Approach



Offloading the API management

Main advantages:

- Easy business logic definition → operators + CRDs
- Reuse of existing features provided by the API server:
  - Authentication, authorization, validation, rate limiting, …
- Reduction of the operational costs

Possible Drawbacks:

- The API server is exposed on the internet → strict authn/authz
- Missing support for transactions → reconcile loop
- Increased frontend complexity → GraphQL relay

Why VMs in 2020?

Virtual Machines 👍

- We needed a working solution in limited time
- We were familiar with VMs, users are familiar with VMs
- Long-running environments, limited dynamicity

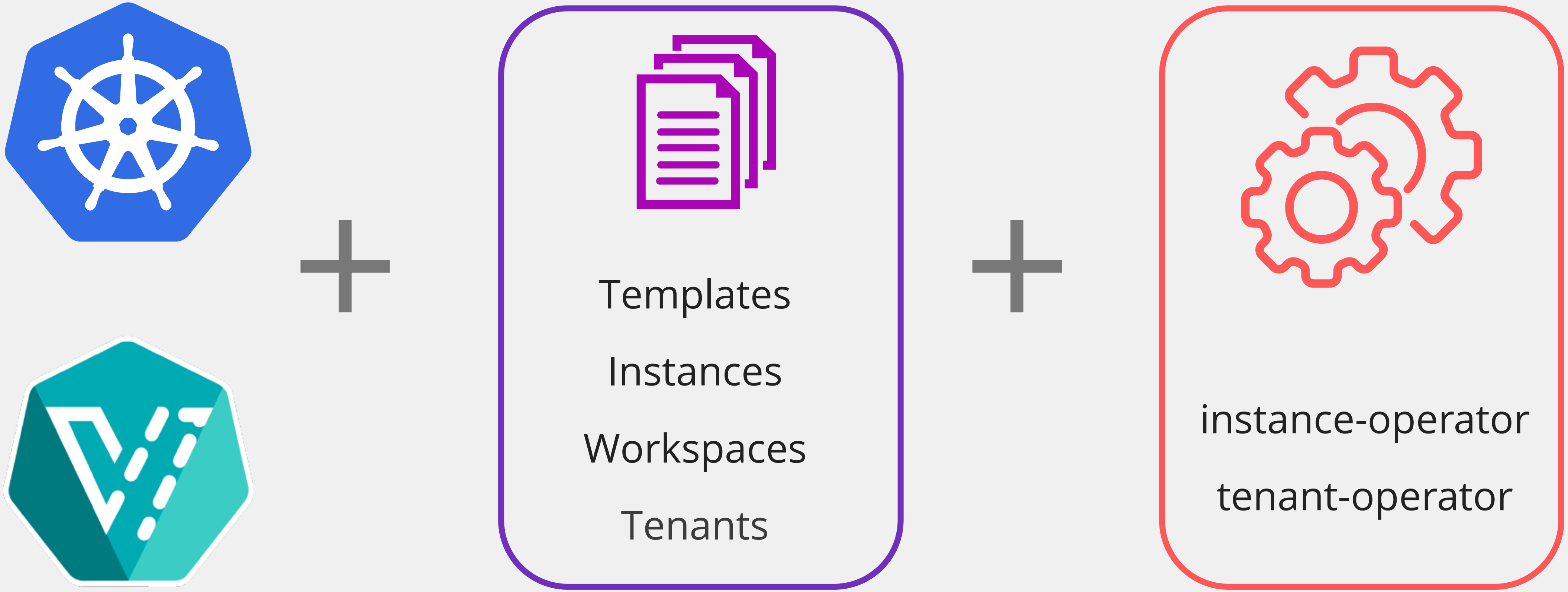
Containers 👎

- Unknown UX of graphical containers: Single application? Full DE?
- Security is more challenging to configure:
  - Cannot afford a container to break one entire node

VMs over k8s: KubeVirt

- Introduces a VirtualMachineInstance resource
- When created, KubeVirt starts a new pod:
  - One container creates a local libvirtd instance
  - A second one wraps a qcow2 disk image file
- The disk images are prepared in advance and stored in a local registry:
  - Vanilla Ubuntu + CrownLabs requirements + additional software
  - Automated with VirtualBox + bash + Ansible

The CrownLabs Backend



+

Templates

Instances

Workspaces

Tenants

+

instance-operator

tenant-operator

Users and Groups management (1)

Workspace Definition

```
apiVersion: crownlabs.polito.it/v1alpha1
kind: Workspace
metadata:
  name: netgroup
spec:
  prettyName: Netgroup Official Workspace
  quota:
    instances: 3
    cpu: 10
    memory: 20G
```

Tenant Definition

```
apiVersion: crownlabs.polito.it/v1alpha2
kind: Tenant
metadata:
  name: marco.iorio
spec:
  firstName: Marco
  lastName: Iorio
  email: marco.iorio@polito.it
  workspaces:
    - name: netgroup
      role: manager
    - ...
publicKeys:
    - ssh-rsa AAAAB3NzaC1yc2E...
```

Information about myself

The workspaces I can access

Users and Groups management (2)

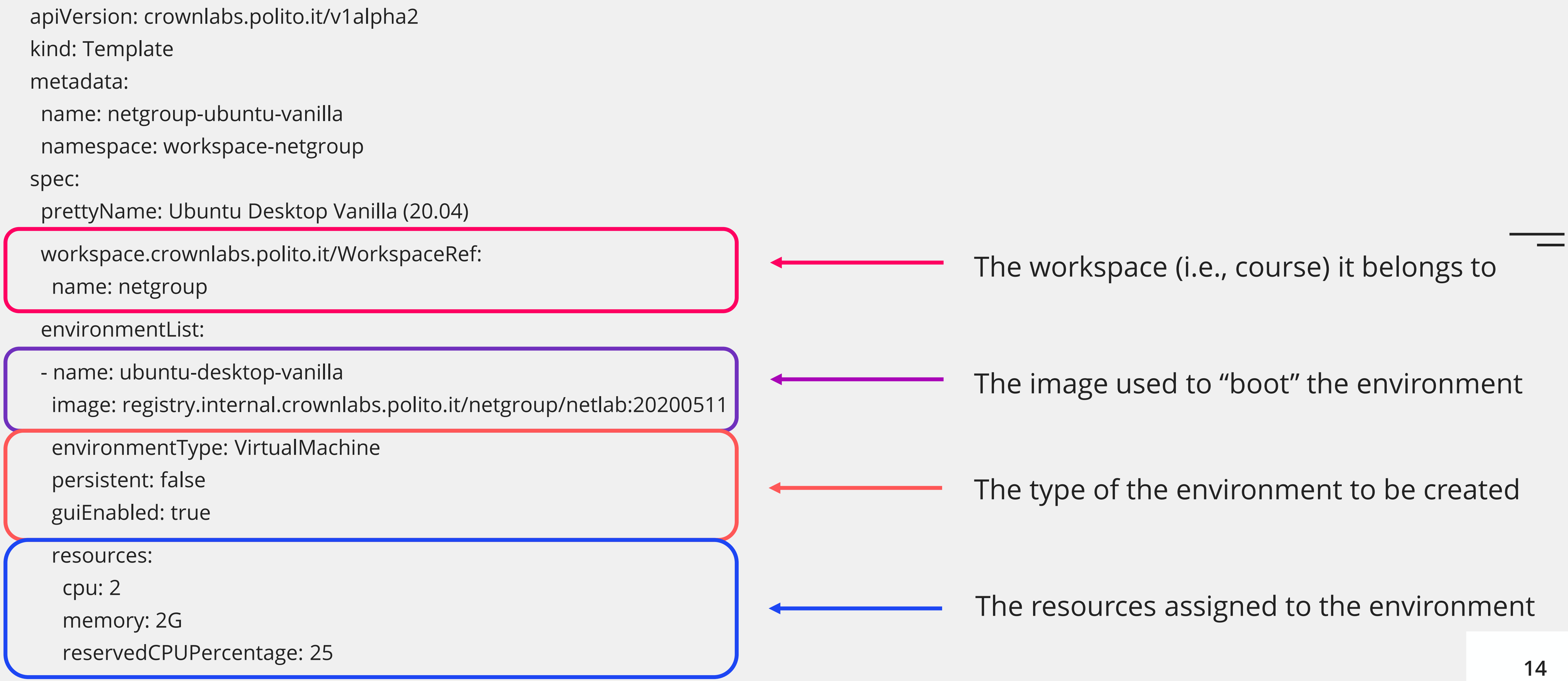
Personal SSO identities provided by an external component (Keycloak)

Security and isolation implemented with k8s mechanisms:

- Each tenant/workspace corresponds to one namespace
- Limited privileges, leveraging RBAC + Custom Admission Webhooks
- Isolation by means of resource quotas and network policies

The configuration is completely automated by the tenant-controller

Template: the model definition



```
apiVersion: crownlabs.polito.it/v1alpha2
kind: Template
metadata:
  name: netgroup-ubuntu-vanilla
  namespace: workspace-netgroup
spec:
  prettyName: Ubuntu Desktop Vanilla (20.04)
  workspace.crownlabs.polito.it/WorkspaceRef:
    name: netgroup
  environmentList:
  - name: ubuntu-desktop-vanilla
    image: registry.internal.crownlabs.polito.it/netgroup/netlab:20200511
    environmentType: VirtualMachine
    persistent: false
    guiEnabled: true
    resources:
      cpu: 2
      memory: 2G
      reservedCPUPercentage: 25
```

The workspace (i.e., course) it belongs to

The image used to “boot” the environment

The type of the environment to be created

The resources assigned to the environment

Instance: the actual environment

```yaml
apiVersion: crownlabs.polito.it/v1alpha2
kind: Instance
metadata:
  name: instance-p85f4
  namespace: tenant-marco-iorio
spec:
  template.crownlabs.polito.it/TemplateRef:
    name: netgroup-ubuntu-vanilla
    namespace: workspace-netgroup
  tenant.crownlabs.polito.it/TenantRef:
    name: marco.iorio
```

The template to instantiate

The creator of the instance

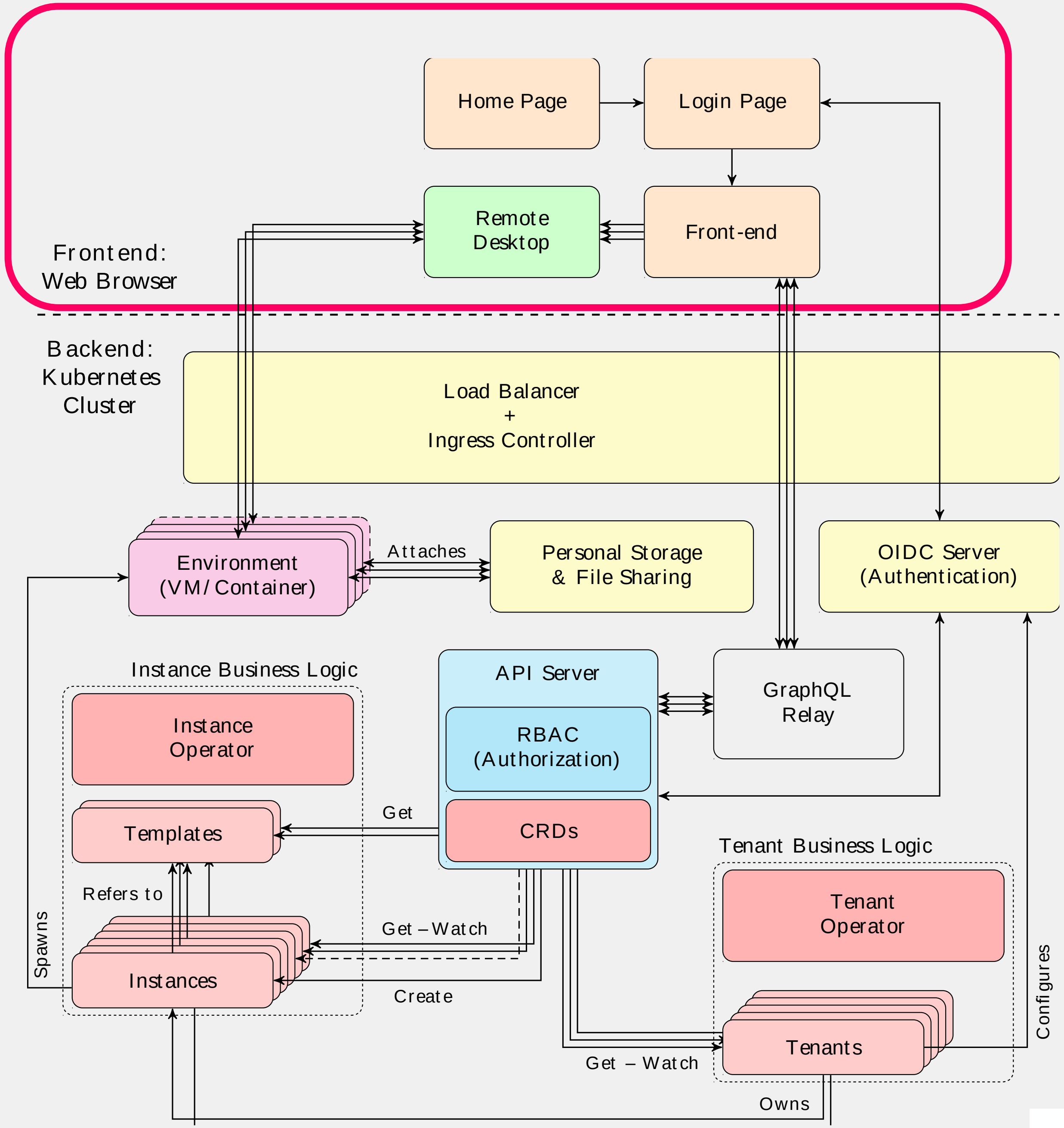
Upon instance creation:

- Create VMI
- Create Service
- Create Ingress
- …
- Update Status

Upon instance deletion:

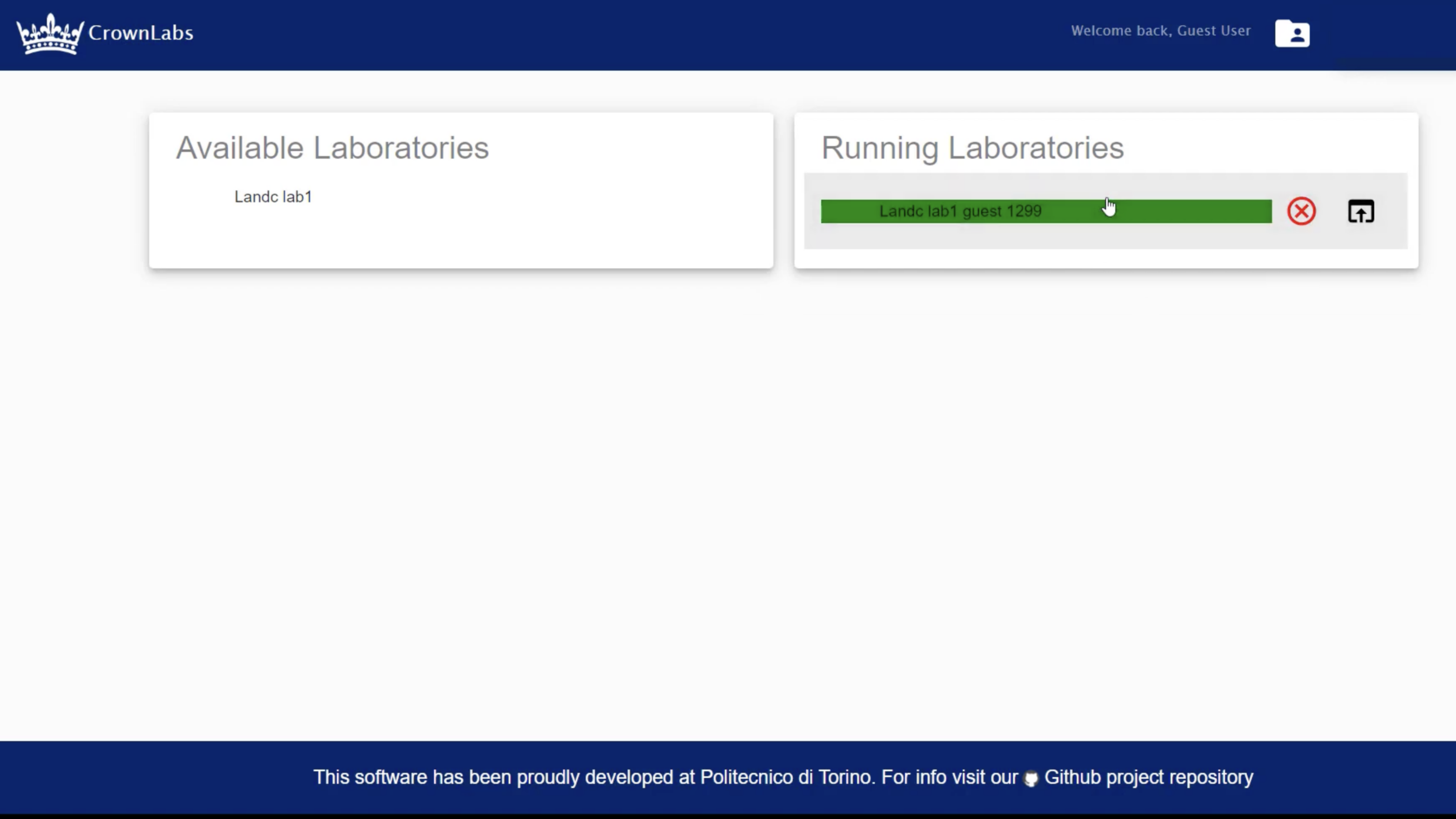
- Chain of Owner References

The CrownLabs Architecture



The Dashboard is a Graphical kubectl

- Historically interacted with k8s through a generic JavaScript Client
- Now leverages a (mostly) generic GraphQL adapter
- Lists and modifies custom resources:
  - Show available templates → watch templates
  - Show running environments → watch instances
  - Start a new environment → create instance
  - Stop an environment → delete instance



CrownLabs

Welcome back, Guest User

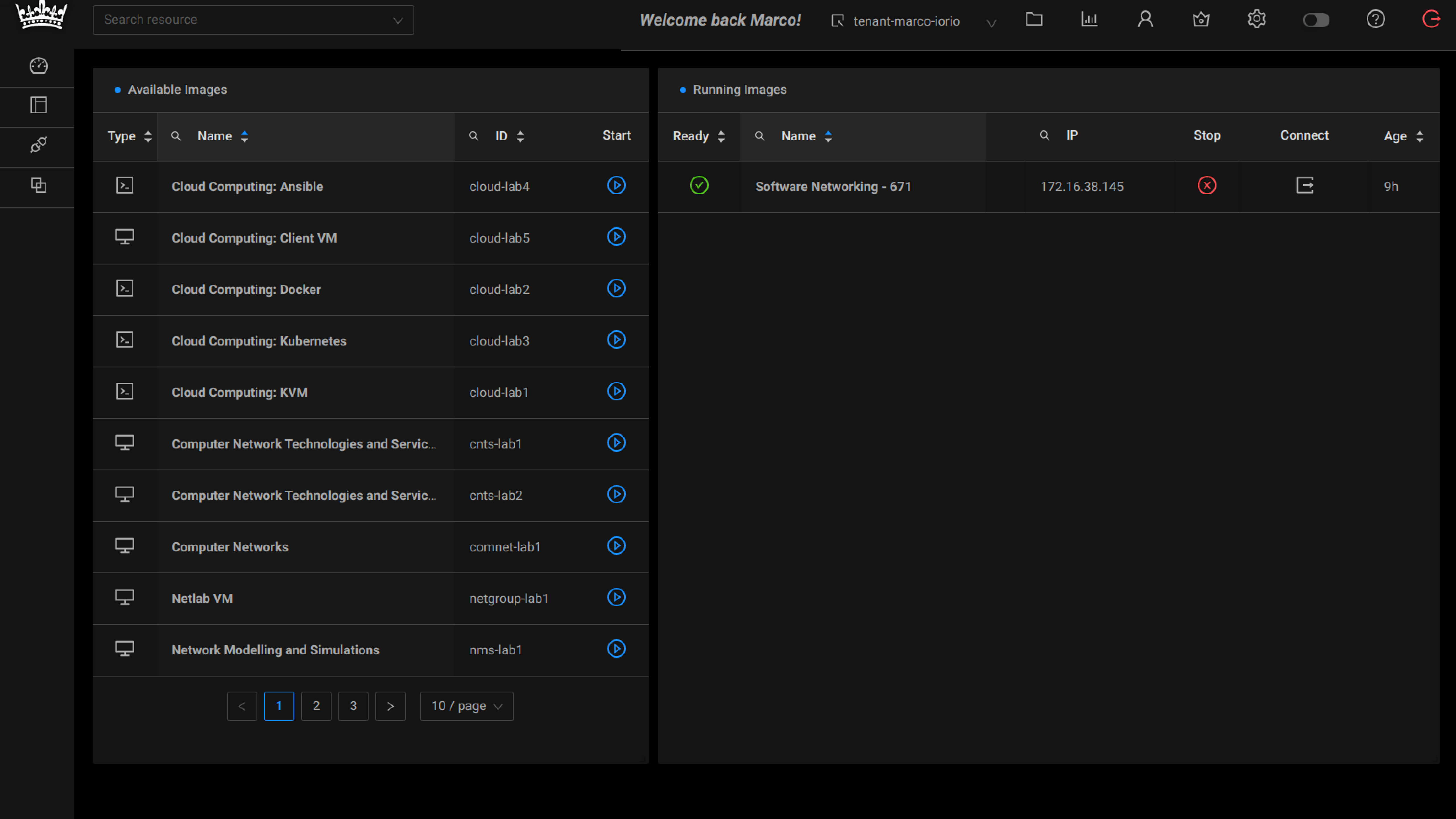
Available Laboratories

Landc lab1

Running Laboratories

Landc lab1 guest 1299

This software has been proudly developed at Politecnico di Torino. For info visit our Github project repository



Search resource

Welcome back Marco!

tenant-marco-iorio

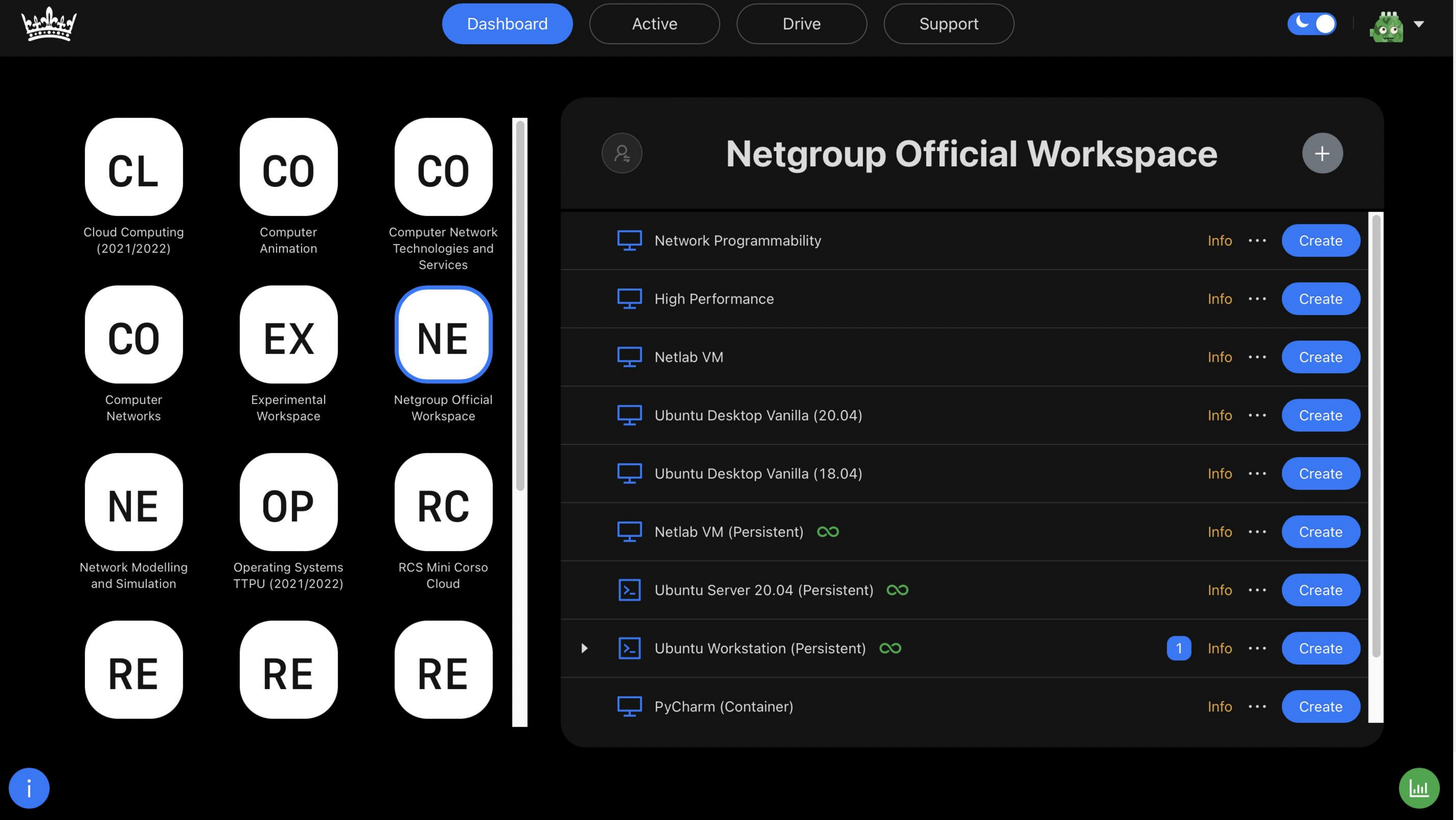
Available Images

Type	Name	ID	Start
>_	Cloud Computing: Ansible	cloud-lab4	▷
🖥	Cloud Computing: Client VM	cloud-lab5	▷
>_	Cloud Computing: Docker	cloud-lab2	▷
>_	Cloud Computing: Kubernetes	cloud-lab3	▷
>_	Cloud Computing: KVM	cloud-lab1	▷
🖥	Computer Network Technologies and Servic...	cnts-lab1	▷
🖥	Computer Network Technologies and Servic...	cnts-lab2	▷
🖥	Computer Networks	comnet-lab1	▷
🖥	Netlab VM	netgroup-lab1	▷
🖥	Network Modelling and Simulations	nms-lab1	▷

< 1 2 3 > 10 / page

Running Images

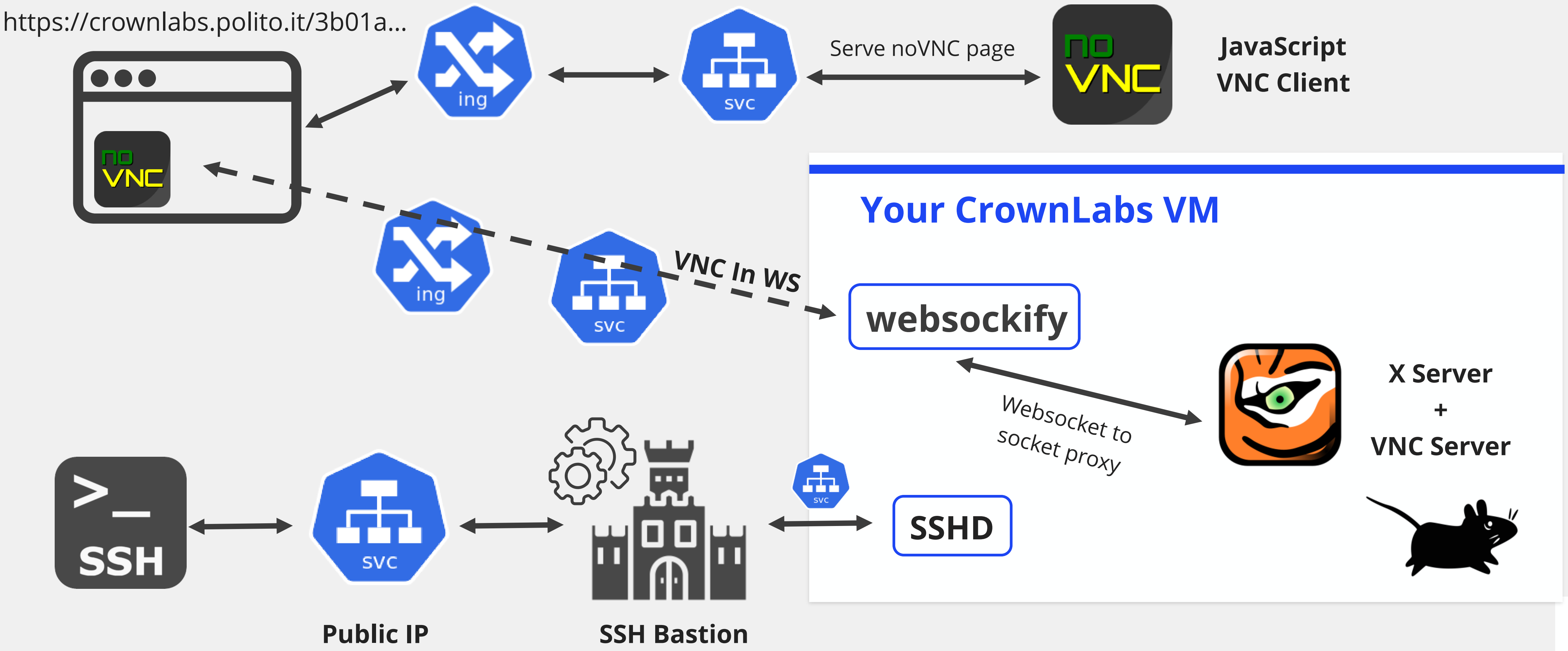
Ready	Name	IP	Stop	Connect	Age
✓	Software Networking - 671	172.16.38.145	⊗	⇥	9h


Dashboard
Active
Drive
Support
CL
Cloud Computing (2021/2022)
CO
Computer Animation
CO
Computer Network Technologies and Services
CO
Computer Networks
EX
Experimental Workspace
NE
Netgroup Official Workspace
NE
Network Modelling and Simulation
OP
Operating Systems TTPU (2021/2022)
RC
RCS Mini Corso Cloud
RE
RE
RE

Netgroup Official Workspace

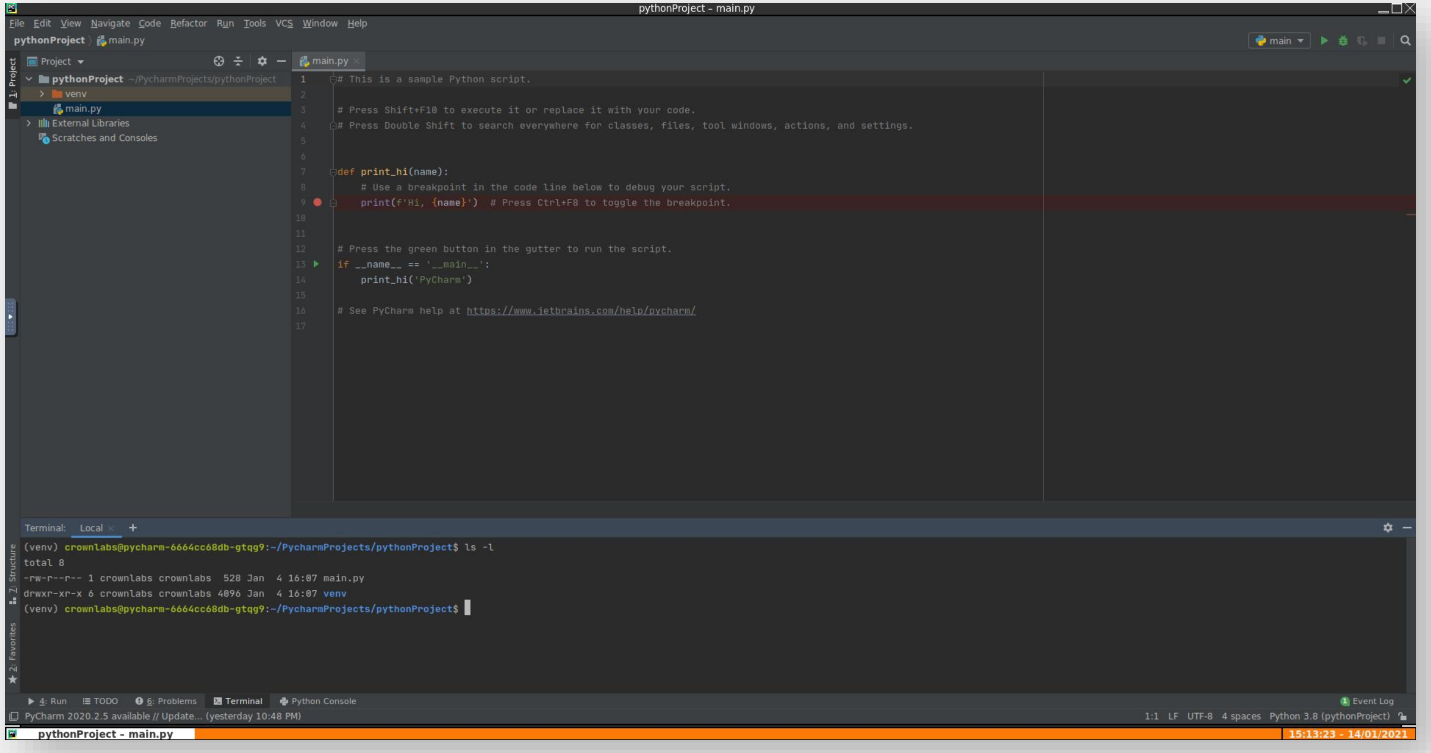
Network Programmability	Info	···	Create
High Performance	Info	···	Create
Netlab VM	Info	···	Create
Ubuntu Desktop Vanilla (20.04)	Info	···	Create
Ubuntu Desktop Vanilla (18.04)	Info	···	Create
Netlab VM (Persistent) ∞	Info	···	Create
Ubuntu Server 20.04 (Persistent) ∞	Info	···	Create
▸ Ubuntu Workstation (Persistent) ∞ 1	Info	···	Create
PyCharm (Container)	Info	···	Create

Accessing the Remote Environments



From VMs to Containers

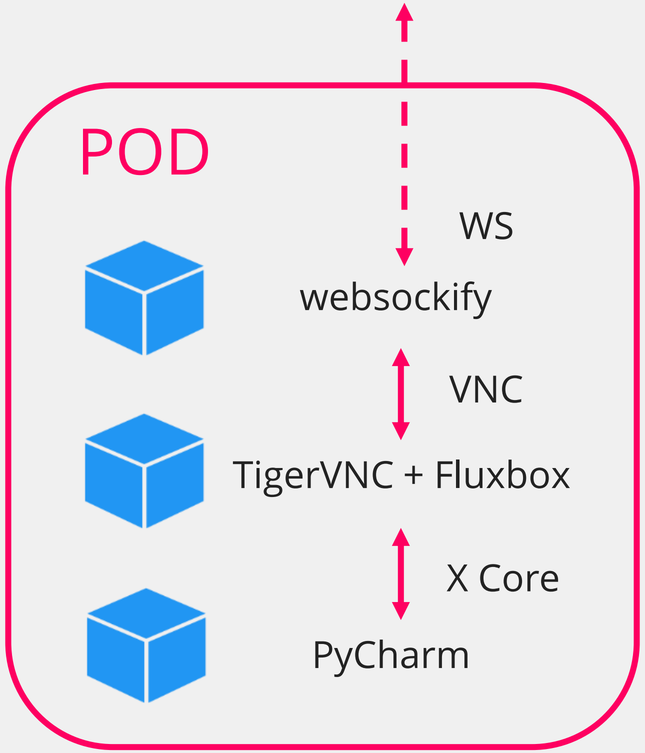
Graphical Containers actually work!



```python
# This is a sample Python script.

# Press Shift+F10 to execute it or replace it with your code.
# Press Double Shift to search everywhere for classes, files, tool windows, actions, and settings.


def print_hi(name):
    # Use a breakpoint in the code line below to debug your script.
    print(f'Hi, {name}')  # Press Ctrl+F8 to toggle the breakpoint.


# Press the green button in the gutter to run the script.
if __name__ == '__main__':
    print_hi('PyCharm')

# See PyCharm help at https://www.jetbrains.com/help/pycharm/

```

```
(venv) crownlabs@pycharm-6664cc68db-gtqg9:~/PycharmProjects/pythonProject$ ls -l
total 8
-rw-r--r-- 1 crownlabs crownlabs  528 Jan  4 16:07 main.py
drwxr-xr-x 6 crownlabs crownlabs 4096 Jan  4 16:07 venv
(venv) crownlabs@pycharm-6664cc68db-gtqg9:~/PycharmProjects/pythonProject$
```

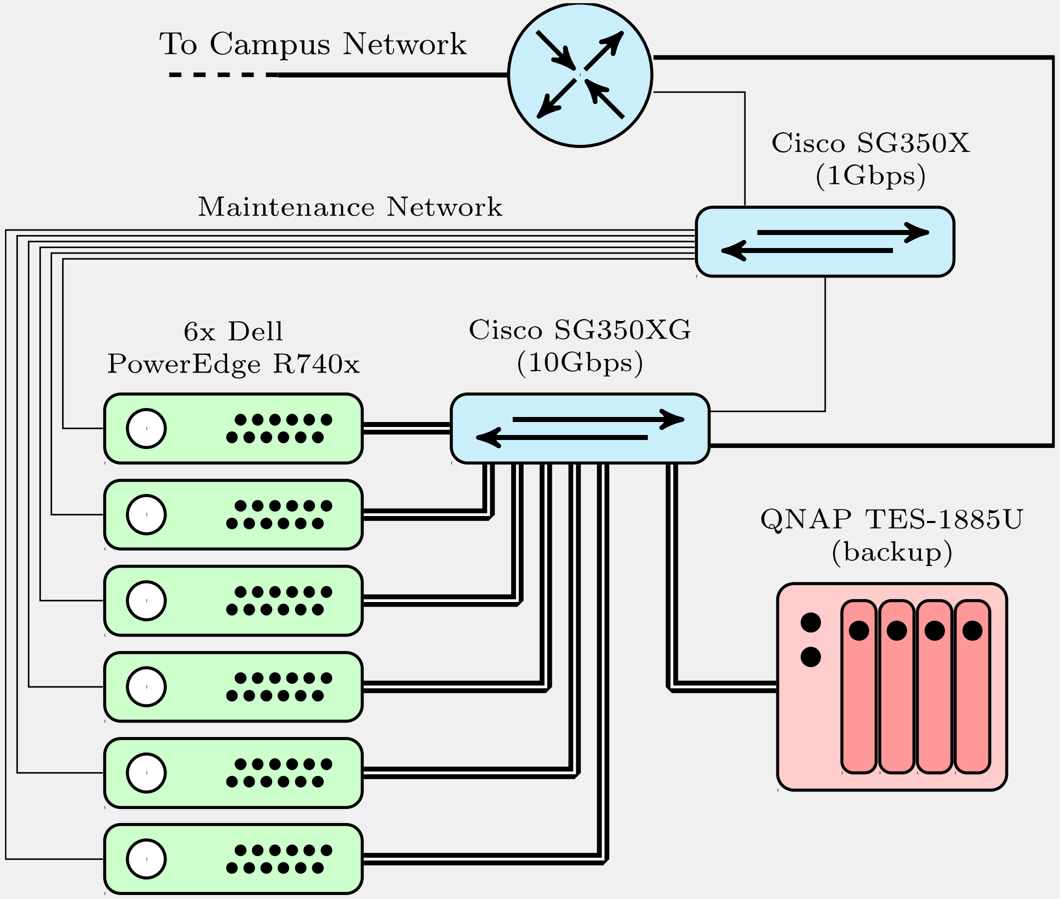
PyCharm 2020.2.5 available // Update... (yesterday 10:48 PM)



The CrownLabs horse powers

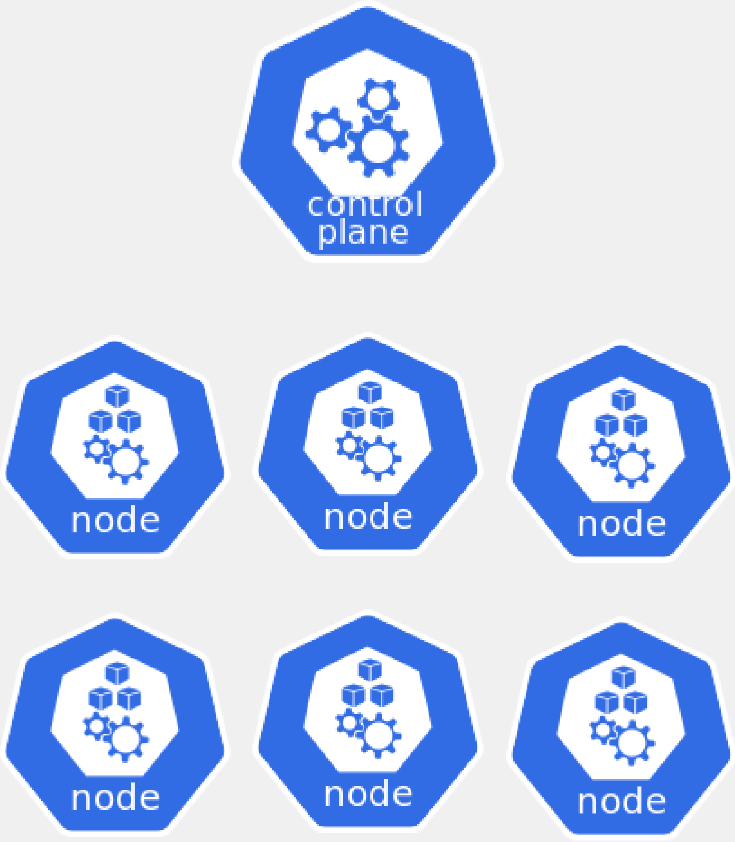
A quick look at the infrastructure

The Infrastructural View



- 336 logical cores
- 2.0TB RAM
- ≅25TB SSD
- ≅10TB HDD
- 2x 10G NICs (link aggregation)

The Logical View



Control plane:

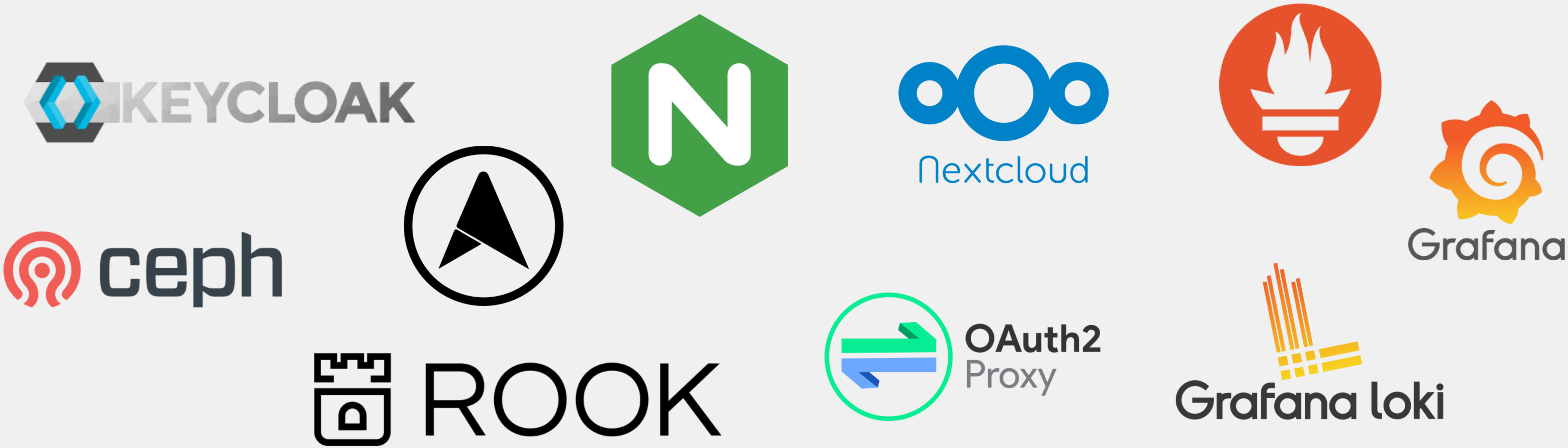
- Hosted by a VM, to simplify migration
- Not yet HA, due to initially limited resources

Networking:

- CNI: Project Calico
- No overlay
- Supports advanced network policies

What powers CrownLabs?

A journey among the main components

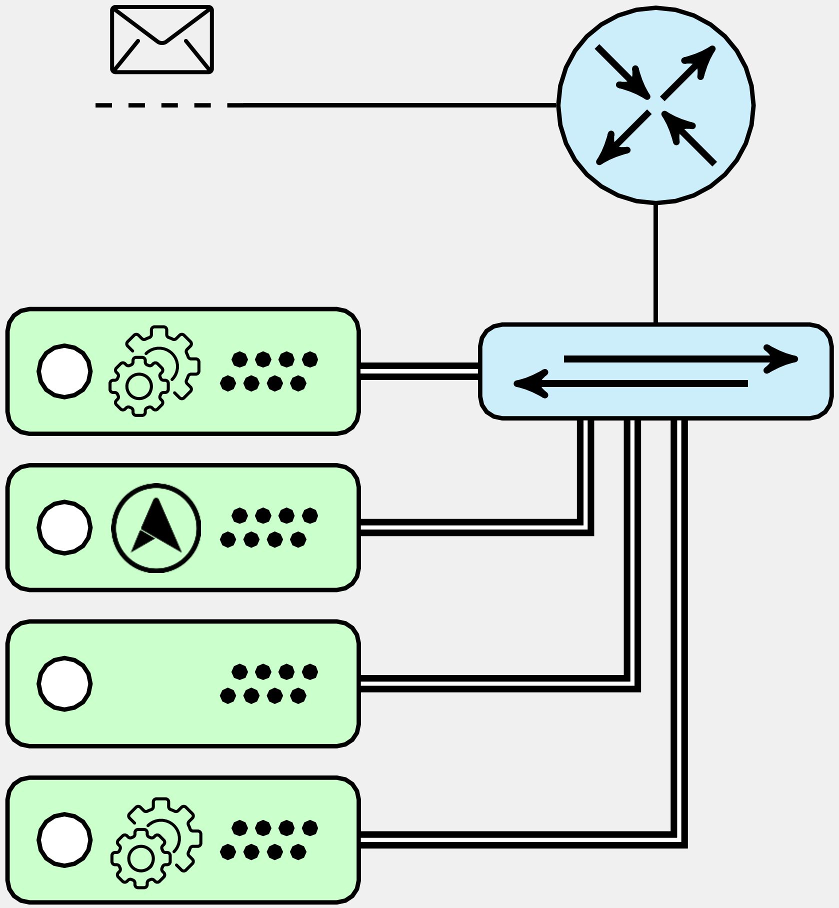


First

How to access the services?

https://crownlabs.polito.it

The journey of a request (1)



Access + Resiliency (MetalLB):

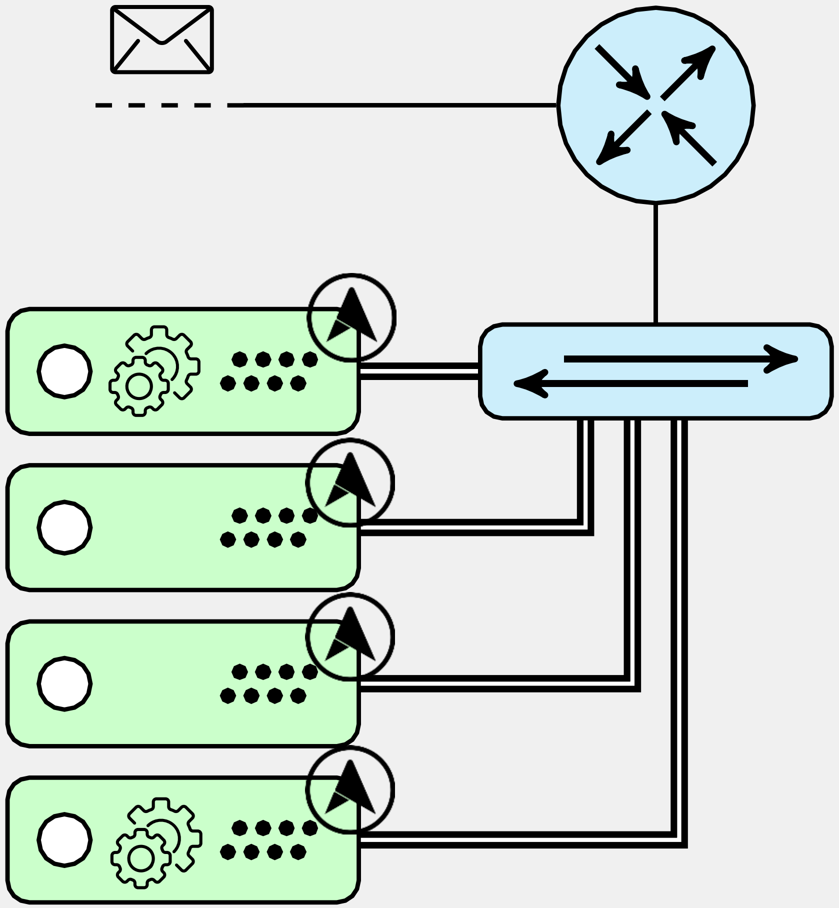
- Takes care of the reachability of the “public” IPs
- Operates at L2, with gratuitous ARPs
- Does NOT perform load balancing

☹

Load Balancing (Kube Proxy):

- Redistribute the traffic to one of the backends

The journey of a request (1)



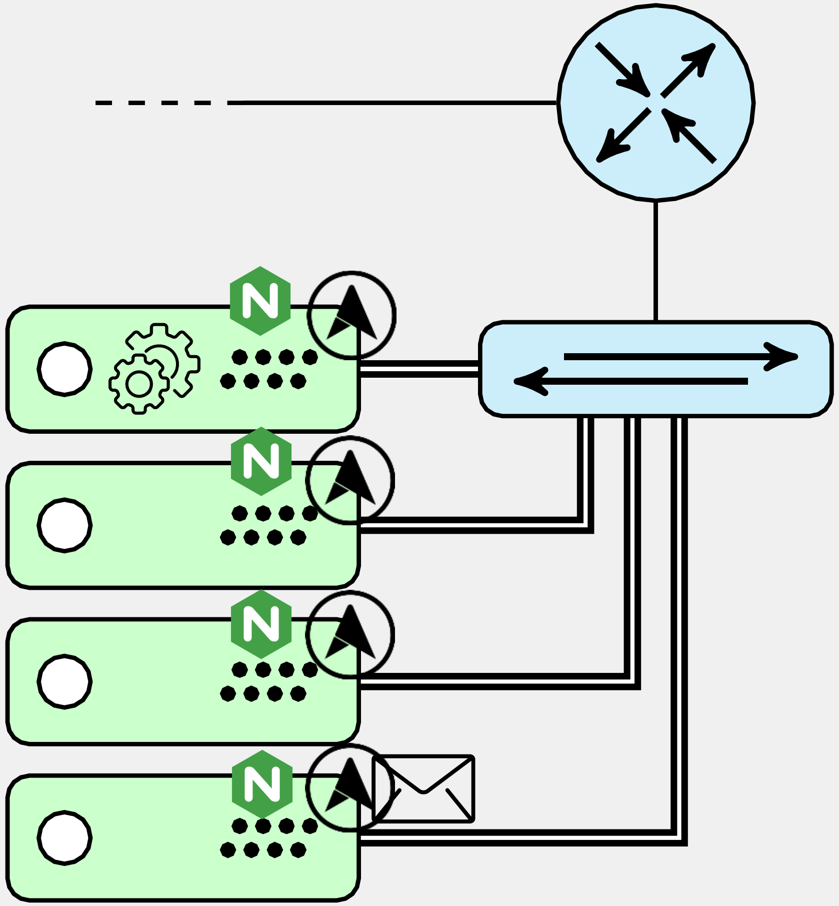
Access + Resiliency + LB (MetalLB):

- Takes care of the reachability of the "public" IPs
- Operates at L3, announcing IP addr. through BGP
- The router DOES perform load balancing (ECMP)

2nd Load Balancing Step (Kube Proxy):

- Optional, based on service configuration
- Redistribute the traffic to one of the backends

The journey of a request (2)



HTTPs proxy (NGINX Ingress Controller):

- Exposed through a “load-balanced” virtual IP
- Terminates all HTTP/HTTPs connections
- Selects the backend service depending on the host name and the requested path
- Multiple replicas, for HA (DaemonSet)

Ingresses and companion components

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: crownlabs-website
  namespace: crownlabs-website
  annotations:
    cert-manager.io/cluster-issuer: letsencrypt-production
spec:
  rules:
  - host: crownlabs.polito.it
    http:
      paths:
      - path: /
        backend:
          service:
            ...
  tls:
  - hosts:
    - crownlabs.polito.it
    secretName: crownlabs-website-certificate
```

external-dns

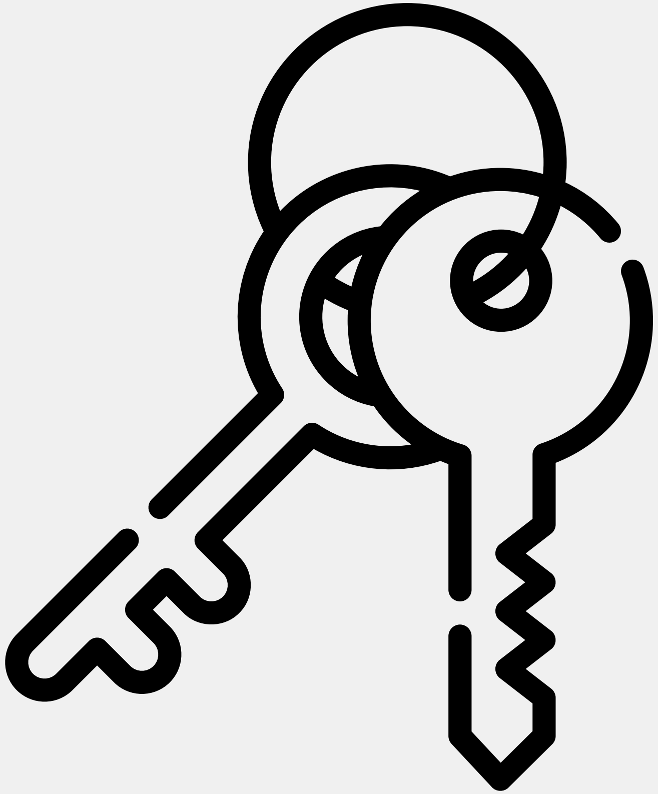
- Automatic configuration of DNS records
- Interacts with the bind9 server of the netgroup

cert-manager

- Issuance and renewal of valid TLS certificates
- Configured to leverage Let’s Encrypt as backend

Second

We need to authenticate

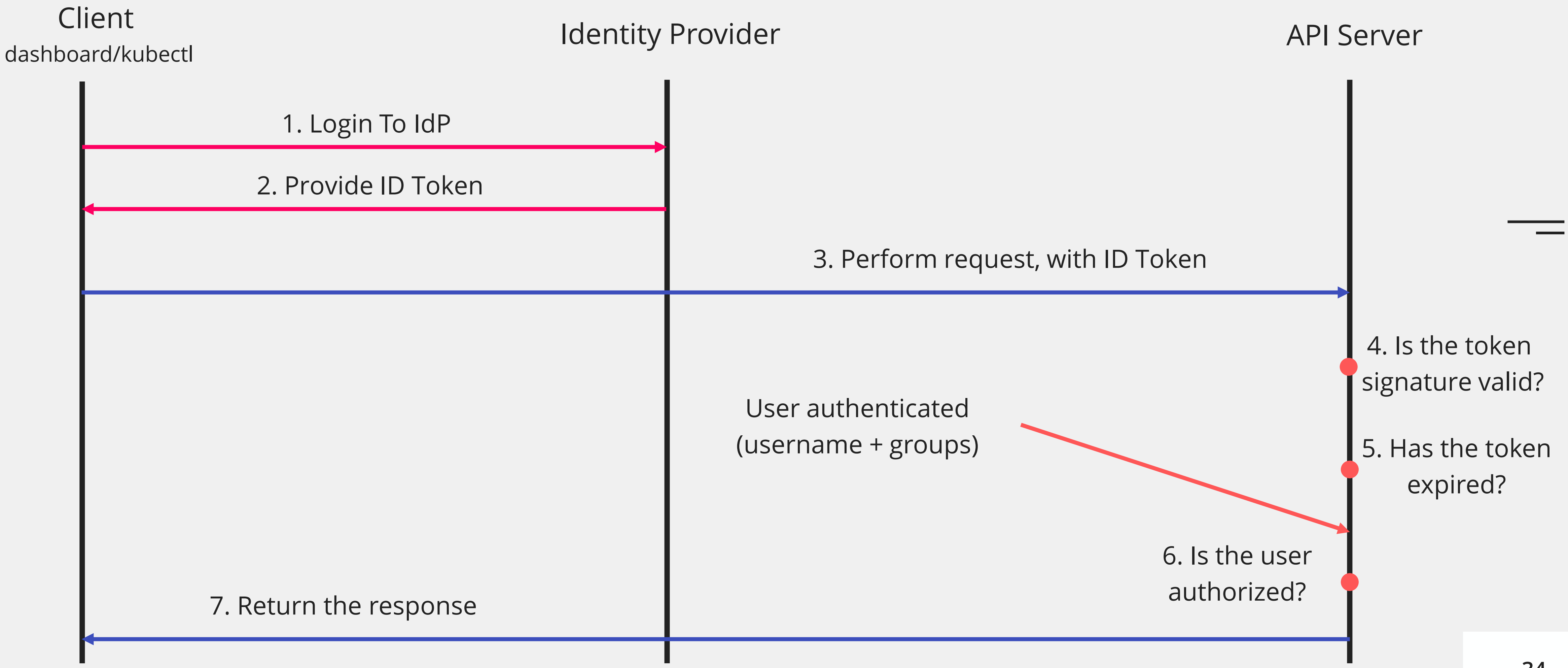


Authentication and SSO

- One identity to access all services (dashboard, k8s, monitoring, ...)
- Different authorization policies based on users and groups

KEYCLOAK + OAuth2 Proxy

Authentication Workflow?


Client
dashboard/kubectl
Identity Provider
API Server
1. Login To IdP
2. Provide ID Token
3. Perform request, with ID Token
4. Is the token signature valid?
User authenticated (username + groups)
5. Has the token expired?
6. Is the user authorized?
7. Return the response

Why keycloak?

- An identity and management solution with advanced features
- Exposes a Standard OpenID Connect (OIDC) interface
- More control and less complexity wrt the campus OIDC system

- High Availability configuration:
  - Three replicas of the keycloak server
  - Three replicas of the database

Deploying complex apps

- Do not expect “kubectl apply –f keycloak.yaml” to be enough
- Many pre-requisites (e.g. storage, ingress-controller, cert-manager, …)
- HA isn’t free: you need state synchronization (e.g. databases)
  - postgres-operator: create and configure PostgreSQL clusters
  - One pod per replica, synchronization managed by PostgreSQL
- Even with everything in place, multiple aspects to tune:
  - Better to leverage Helm charts
  - Even better with declarative GitOps approaches

Third

Store data, keep it safe

Disk Partitioning

- Do not underestimate the importance of the partitioning scheme
- Better to isolate the important directories:
  - /var/lib/docker: docker images + ephemeral storage (overlayfs)
  - /var/lib/kubelet: ephemeral storage (emptyDir)
- Limit the amount of ephemeral storage per pod with resource quotas
- Slow disks and I/O intensive workloads are the recipe for a disaster

Storage Provisioning

- Applications and users want persistent storage to save data
- Kubernetes leverages the PersistentVolumeClaim abstraction
- Needs to survive disk/node failures, without losing data

ROOK

Automates the deployment, configuration
and upgrade of storage providers

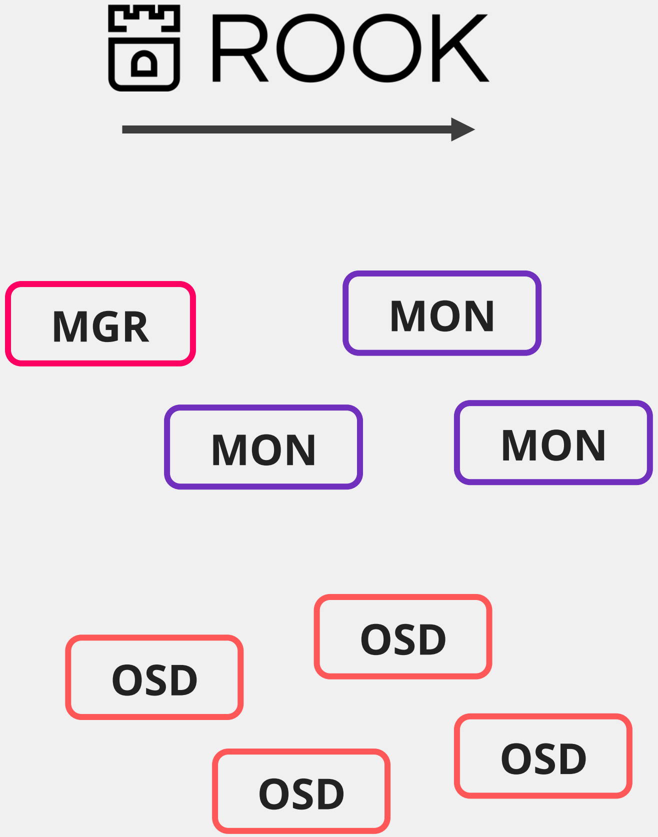
+

ceph

Storage provider supporting block,
file-system and object storage (i.e. S3)

Ceph Cluster

```
apiVersion: ceph.rook.io/v1
kind: CephCluster
metadata: ...
spec:
  cephVersion:
    image: ceph/ceph:v15.2.5
  dataDirHostPath: /var/lib/rook
  ...
  mon:
    count: 3
  storage:
    nodes:
    - devices:
      - name: sdd7
      name: worker-1
    - devices:
      - name: sdd7
      name: worker-2
    - ...
    useAllDevices: false
```



ROOK

MGR

MON

MON

MON

OSD

OSD

OSD

OSD

OSD

```
cluster:
  id:     5e5755fb-3294-442b-8576-8733460cdcfb
  health: HEALTH_OK

services:
  mon: 3 daemons, quorum u,x,y (age 10w)
  mgr: a(active, since 10w)
  osd: 12 osds: 12 up (since 3d), 12 in (since 10w)

data:
  pools:   10 pools, 265 pgs
  objects: 181.78k objects, 701 GiB
  usage:   2.0 TiB used, 17 TiB / 19 TiB avail
  pgs:     265 active+clean
```

Personal Storage

- A place where users can persist their files even if VMs are deleted
- Accessible from the VMs, through davfs2 (with some limitations)
- Feature-rich graphical interface to access the files

Nextcloud

- High Availability configuration:
  - Three replicas of the nextcloud server
  - Three replicas of the database
  - One Redis (in-memory cache) instance

Private Docker Registry

- A place to store and distribute Docker images
- Reduce start-up time of heavyweight VMs/containers
- Reduce impact of DockerHub rate limiting policies

HARBOR

- Microservice-based architecture
- Supports many advanced functionalities
- High Availability configuration

Fourth

A monitored cluster is a healthy cluster

Cluster Monitoring (1)

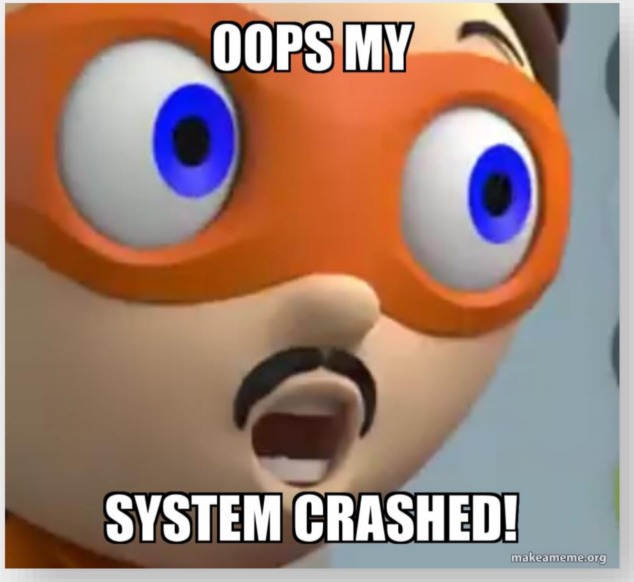
- Fairly standard monitoring stack:
  - Node Exporters + Prometheus/Thanos: metrics collection and storage
  - Promtail + Loki: logs collection and storage
  - Grafana: visualization platform to graphically display the metrics
  - Alertmanager: alerts transmission when something goes wrong
- Lessons learned:
  - Push notifications are fundamental to react (quickly)
  - Cluster monitoring is useless in catastrophic scenarios → freshping

Cluster Monitoring (2)

… and most of all …

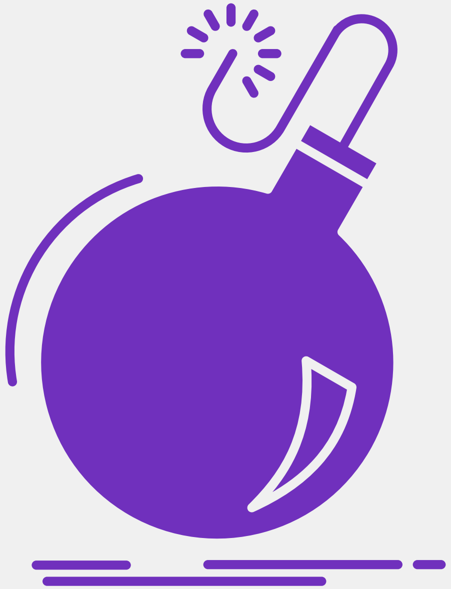
If your users complain about a problem before you get an alert,

then your monitoring sucks!



Fifth

Be prepared when things go south



Disaster Recovery

VELERO

Schedule backups…
and pray you’ll never
need to restore them

CrownLabs in a few months

More exciting features to come

What’s Next?

- More environments supported (Cloud VMs, standalone applications, …)
- Automation of sandbox namespaces (i.e., k8s playgrounds) setup
- Advanced resource accounting for high-performance environments
- Computer science (first year’s course) exams & integration with Moodle
- Federation of additional clusters through liqo to increase resources

"That's all Folks!"

The CrownLabs Contributors

Want to know more?

[1]: https://github.com/netgroup-polito/CrownLabs/

[2]: Marco Iorio, Alex Palesandro and Fulvio Risso, “CrownLabs — A Collaborative Environment to Deliver Remote Computing Laboratories,” in IEEE Access, vol. 8, pp. 126428-126442, 2020. Available at https://ieeexplore.ieee.org/document/9136697

THANK YOU

BİYAN
SHUKRIA
TINGKI
DANKSCHEEN
YUSPAGARÀTAM
CHALTU
YAQHANYELAY
NUHUN
SPASSIBO
SNACHALHUYA
TASHAKKUR ATU
WABEEJA
MAITEKA
HUI
GRACIAS
DHANYABAAD
SUKSAMA
GUI
ANIHA
EKHMET
UNALCHÉESH
HATUR
MEHRBANI
SPASIBO
MERSI
ATTO
DENKAUJA
EKOJU
NENACHALHYA
SIKOMO
ARIGATO
SANCO
KOMAPSUMNIDA
MAAKE
GRAZIE
MAKETAI
SHUKURIA
MERASTAWHY
GAEJTHO
PALDIES
TAVTAPUCH
LAH
GOZAIMASHITA
MEDAWAGSE
EFCHARISTO
AGUYJE
BAIIKA
FAKAAUE
JUSPAXAR
MINMONCHAR
BOLZÏN
MERCI