Fog and Cloud Computing
Lab

Daniele Santoro *(dsantoro@fbk.eu)* - Expert Research Engineers
Silvio Cretti *(scretti@fbk.eu)* – *Senior* Research Engineers

RiSING (Robust and Secure Distributed Computing)
Fondazione Bruno Kessler (FBK)

Trento, May 20th 2022

Lab Resources

- Shared Etherpad: https://annuel2.framapad.org/p/6s5u416vo7-9t4b
- White Board: https://tinyurl.com/2p8j7yra
- Interaction:
  - Etherpad
    - *Exercises check, Share Troubleshooting, Questions and Logs*
  - Zoom Chat (for those remotely connected)
    - *Discuss with your colleagues during exercises or directly/privately with me*
  - Rise your Hand (also via Zoom)
    - *If you need my attention or want to speak, don't be shy !!!*
  - Course Forum: https://tinyurl.com/27vmd9pj
    - *Questions and answers could be useful to others, be collaborative*

Lab Resources

- Slides
  - Uploaded before any lesson in Moodle
- Repositories of exercises
  - https://gitlab.fbk.eu/dsantoro/fcc-lab-2022
- Lab Virtual Machine:
  - **Lab VM on Azure (reference for exercises)**
  - Vagrant and VirtualBox on your laptop (possible choice)
    - https://www.virtualbox.org/, https://www.vagrantup.com/ and https://gitlab.fbk.eu/dsantoro/fcc-lab-2022

Quick Recap & Today Lesson

- Recap of previous topics
  - Install a multi-node cluster
  - Overlay Network
  - Pod-to-Pod communication
- Today
  - K8s Services
  - External-World-to-Pod communication
  - Load Balancer
  - Namespaces
  - Labels & Selectors
  - Dashboard
  - ConfigMaps & Secrets [optional]

Namespaces

- If we have numerous users whom we would like to organize into teams/projects, we can partition the Kubernetes cluster into «sub-clusters» using Namespaces

- The names of the resources/objects created inside a Namespace are unique, but not across Namespaces

- Generally, Kubernetes creates two default namespaces:
  - **kube-system:** contains objects created by k8s system
  - **default:** contains objects which belong to any other user

- Using Resource Quotas, we can divide the cluster resources within Namespaces.

Exercise 28 – Namespaces

- **Time:** ~10 minutes
  - 4 minutes: *Try by yourself*
  - 6 minutes: *Check, Verify, Ask*

Description: Play with namespaces and explore “hidden” workload. Can you find out where is Kubernetes control plane running? What are those components: *Etcd, Scheduler, API server, Controller manager* and where they run?

- **Instructions:**
  https://gitlab.fbk.eu/dsantoro/fcc-lab-2022/-/tree/master/e28

Labels and Selectors

- Labels are key-value pairs that can be attached to any Kubernetes objects (e.g. Pods)
  - Labels are used to organize and group a subset of objects
  - Labels do not provide uniqueness to objects

- Examples: `app=webserver, app=database, env=dev, env=prod, env=qa`

- With Selectors, we can select a subset of objects.
  - **Equality-Based Selectors:** Filters based on label keys and values
    - Operators: `=, ==, !=` eg: `env==dev`
  - **Set-Based Selectors:** Filters based on a set of values
    - Operators: `in, noin, exists` eg: `env in (dev,qa)`

Exercise 29 – Labels and Selectors

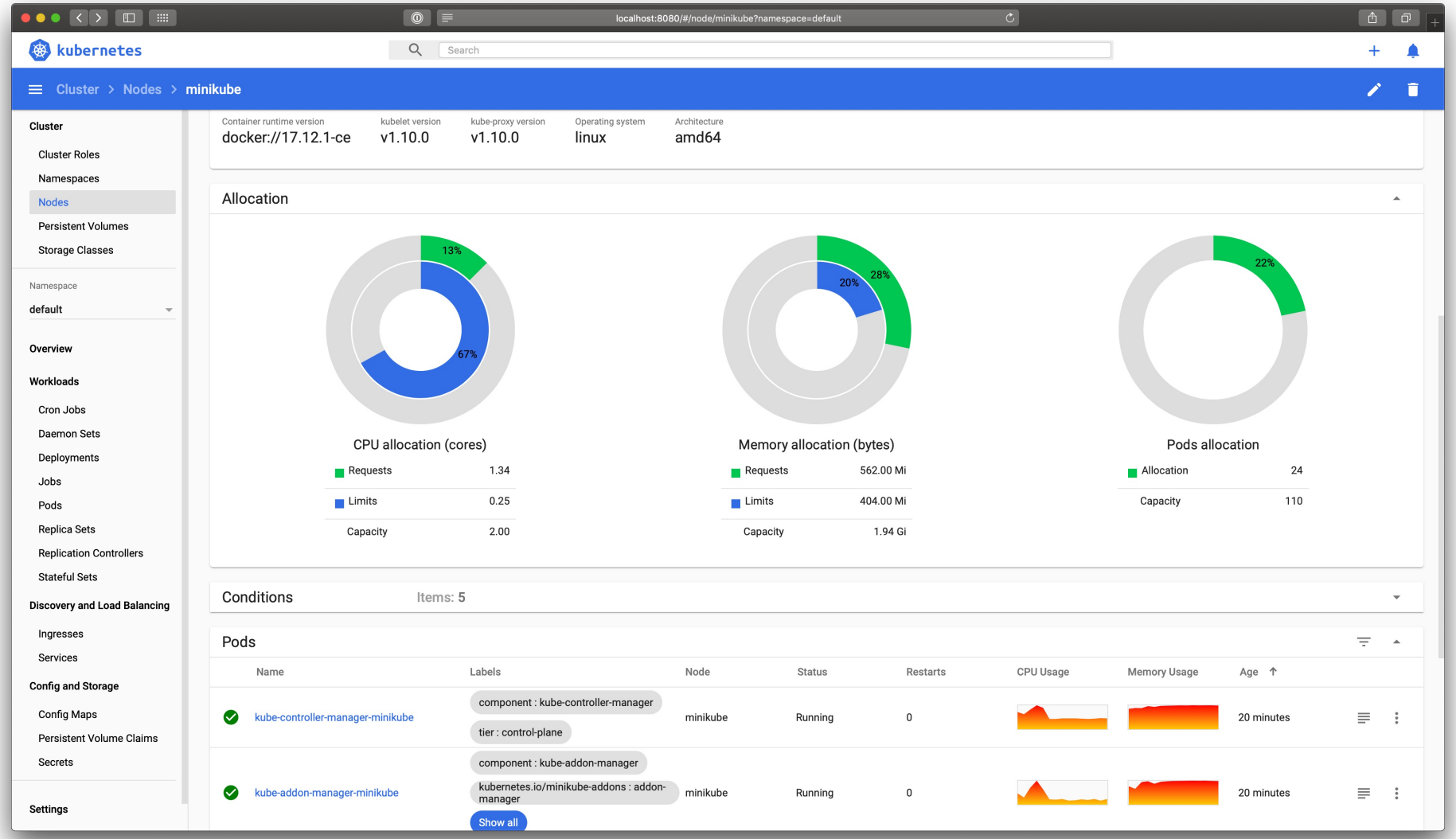
- **Time:** ~10 minutes
  - 4 minutes: *Try by yourself*
  - 6 minutes: *Check, Verify, Ask*

Description: Create five replicas of an example application. Mark the first three with a label `app=frontend` and `env=prod`, the last two with label `app=backend` and the last one with label `env=dev`. Then try to perform the following queries using selectors:

1) List all Pods with their labels
2) List Pods with label `app=backend`
3) List Pods with label `app!=backend`
4) List Pods with labels `env=prod` AND `env=dev`
5) List all Pods with labels `env=prod` OR `env=dev`

- **Instructions:**
  https://gitlab.fbk.eu/dsantoro/fcc-lab-2022/-/tree/master/e29

Kubernetes Dashboard



See the official repository for more information → https://github.com/kubernetes/dashboard

Exercise 30 – Install k8s Dashboard

- **Time:** ~10 minutes
  - 4 minutes: *Try by yourself*
  - 6 minutes: *Check, Verify, Ask*

Description: With the help of the official k8s Dashboard repository install the GUI on your cluster, access to it and perform some operations. For example:

1. Create from the terminal and check the Dashboard
2. Inspect the Pod logs from the dashboard UI
3. Deploy the example of exercise e17 using `lb-example.yaml` from the dashboard
4. Create some workload using the dashboard form

- **Instructions:**
  https://gitlab.fbk.eu/dsantoro/fcc-lab-2022/-/tree/master/e30

ConfigMaps & Secrets

- While deploying an application, we may need to pass runtime parameters like configuration details, passwords, etc.
- In such cases, we can use the **ConfigMap** API resource.
- Similarly, when we want to pass sensitive information, we can use the **Secret** API resource.
- Both ConfigMaps and Secrtes can be created and retrieved in various ways
  - Created from literal values, from files and from directory of files…
  - Used via `ENV_VARS`, Volumes…

Exercise 31 – ConfigMaps and Secrets

- **Time:** ~10 minutes
  - 4 minutes: *Try by yourself*
  - 6 minutes: *Check, Verify, Ask*

Description: Create ConfigMaps and Secrets in various ways, attach them to a Pod and try to retrieve them from inside the Pod.

- **Instructions:**
  https://gitlab.fbk.eu/dsantoro/fcc-lab-2022/-/tree/master/e31

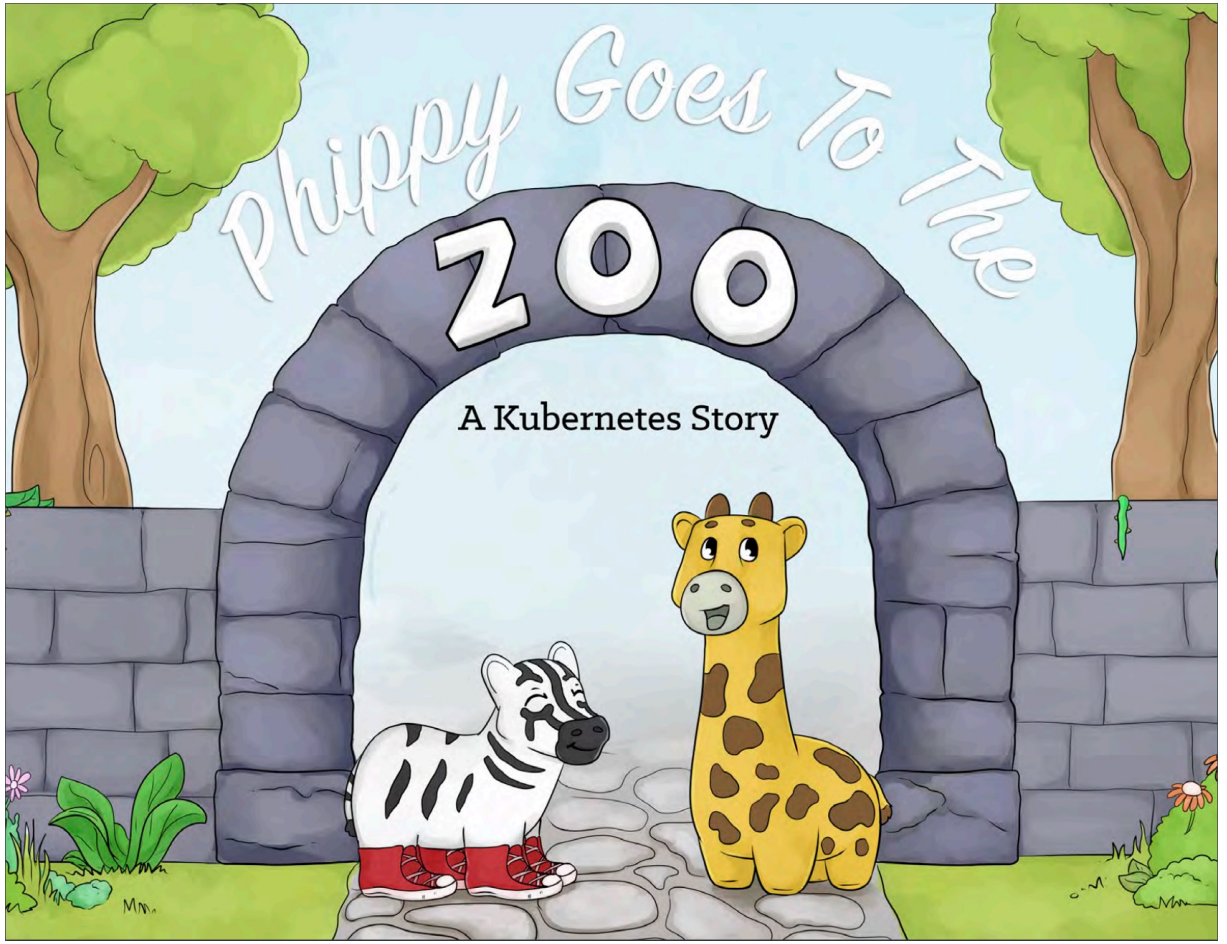
Other k8s Objects & Features

FONDAZIONE BRUNO KESSLER

- **Deployment Advanced Features:** Autoscaling, Proportional Scaling, Pausing and Resuming
- **ConfigMaps and Secrets:** Way to decouple configuration details from container image allowing to pass them as key-value pairs to k8s objects or system components. Also passing them as reference, controlling the usage and hiding the content. (*do you remember 3° rule of 12° Factor rules?*)
- **Ingress:** Collection of rules that allow inbound connections to reach the cluster Services.
- **Jobs:** Create one or more Pod to perform a given task. It makes sure the task is completed , then terminates the Pods. Cron Job is a Job on a time-based schedule
- **StatefulSets:** For application that require a unique identity, like name, net id, strict ordering, eg: mysql or etcd cluster (*it was called PetSet < 1.5*)
- **DaemonSets:** Special Pod that run on all nodes and is started/deleted automatically when node is added/removed
- **Quota Management:** Limit resource consumption per namespace: Compute, Storage, Object Count
- **CRD:** Create our own API objects and Controller that manages them
- **RBAC:** Authorization mechanism for managing permissions around Kubernetes resources
- **Kubernetes Federation:** Manage multiple Kubernetes clusters from a single control plane

… and many more…

Movie Time



Phippy Goes to the Zoo

https://www.youtube.com/watch?v=R9-SOzep73w