Kubernetes Workshop Series

JTC14
Kubernetes
Operators

Niklaus Hirt
DevOps Architect / Cloud Architect
nikh@ch.ibm.com

Welcome to the

Kubernetes Workshop Series

Housekeeping

Meeting is being recorded to be shared on Social Media

Meeting Mute All: Unmute to speak

Breaks: every 60mins (interrupt me if I forget ;-)

Questions:

- In Slack # (not in Webex!)
- Adressed at the end of the Module
- Additional questions: unmute to speak

We will monitor the Slack channel during the Labs
→ Feel free to answer other participants questions

Who am I?

Niklaus Hirt

Passionate about tech for over 35 years

- High-school in Berne
- Degree in Computer Science at EPFL
- ELCA
- CAST
- IBM

✉ nikh@ch.ibm.com

@nhirt

Agenda – Ansible Operators

Module 0: Prepare the Labs

Module 1: Ansible Operators

Module 2: Ansible Operators Hands-On

Videos, sources and documentation will be available here:

All Workshop Recordings
https://www.youtube.com/channel/UCIS0jmGOQrG2AKKPkTJYj9w/videos

https://github.com/niklaushirt/k8s_training_public

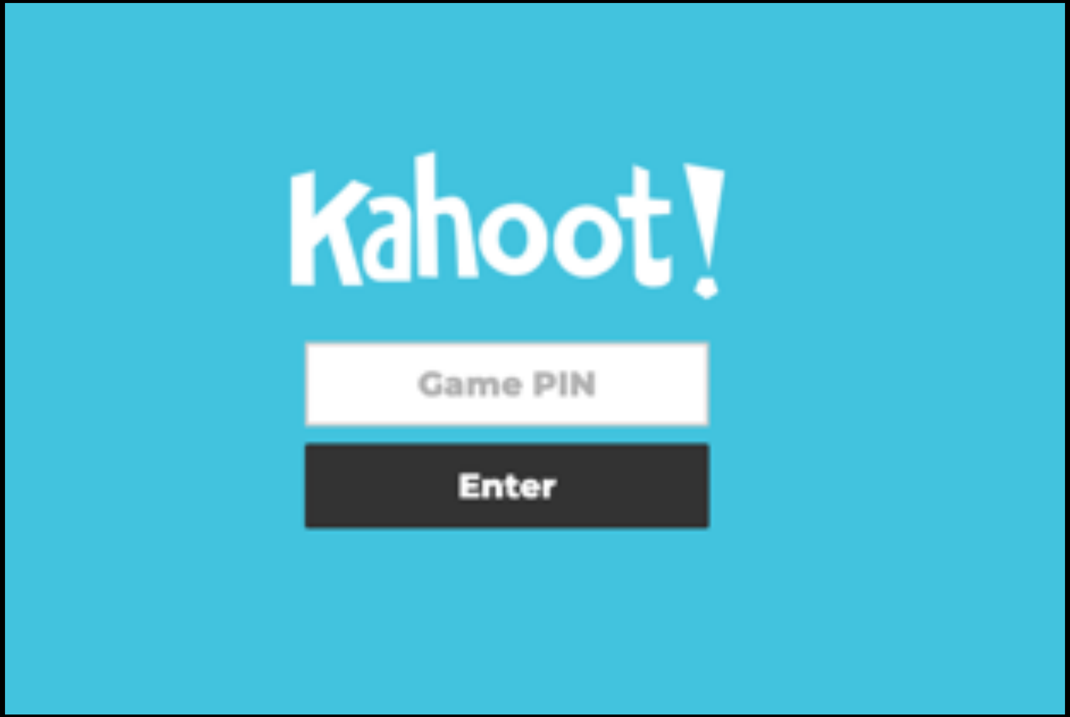
https://github.com/niklaushirt/training

Session Quiz & Feedback

We will collect some feedback.

Please make sure you can access https://kahoot.it/
 either on your PC of Phone.

You will get the Game PIN
later in the training.



Kubernetes Workshop Series

Prepare the Labs

00

Session Objectives

Attendees will run their own ***Personal Training Environment (PTE)*** in the VM.

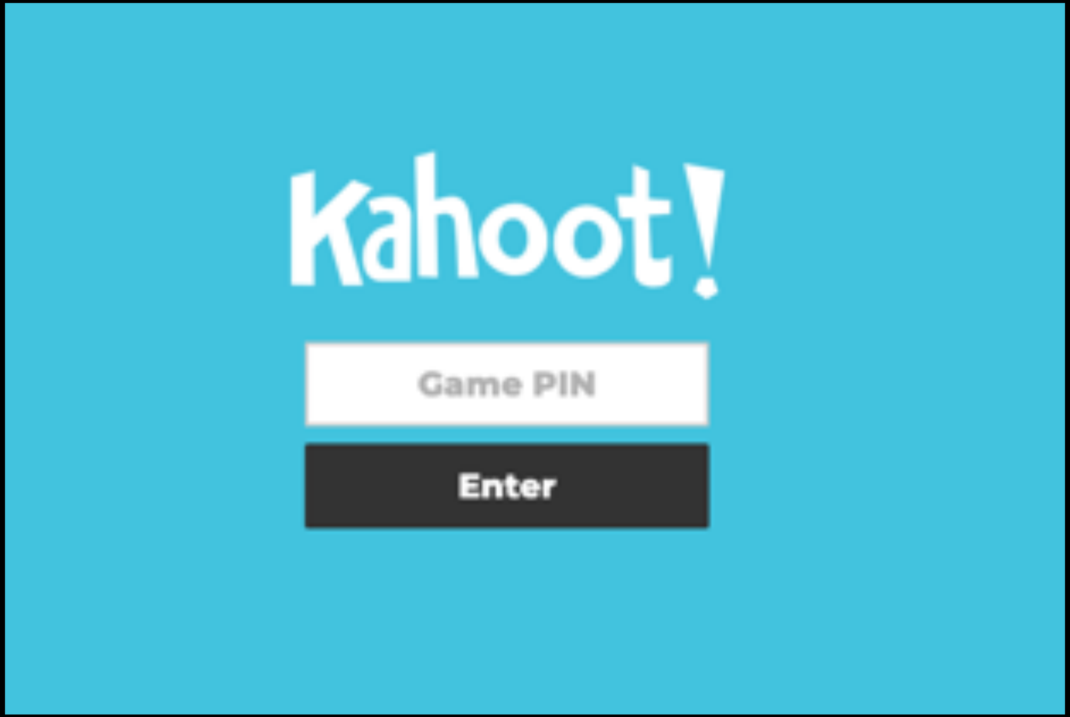
Following some lectures will be ***hands-on*** work that each participant can to complete.

Session Quiz & Feedback

We will collect some feedback and run a quiz or two.

Please make sure you can access https://kahoot.it/
either on your PC of Phone.

You will get the Game PIN
later in the training.



JTC90 Lab Setup

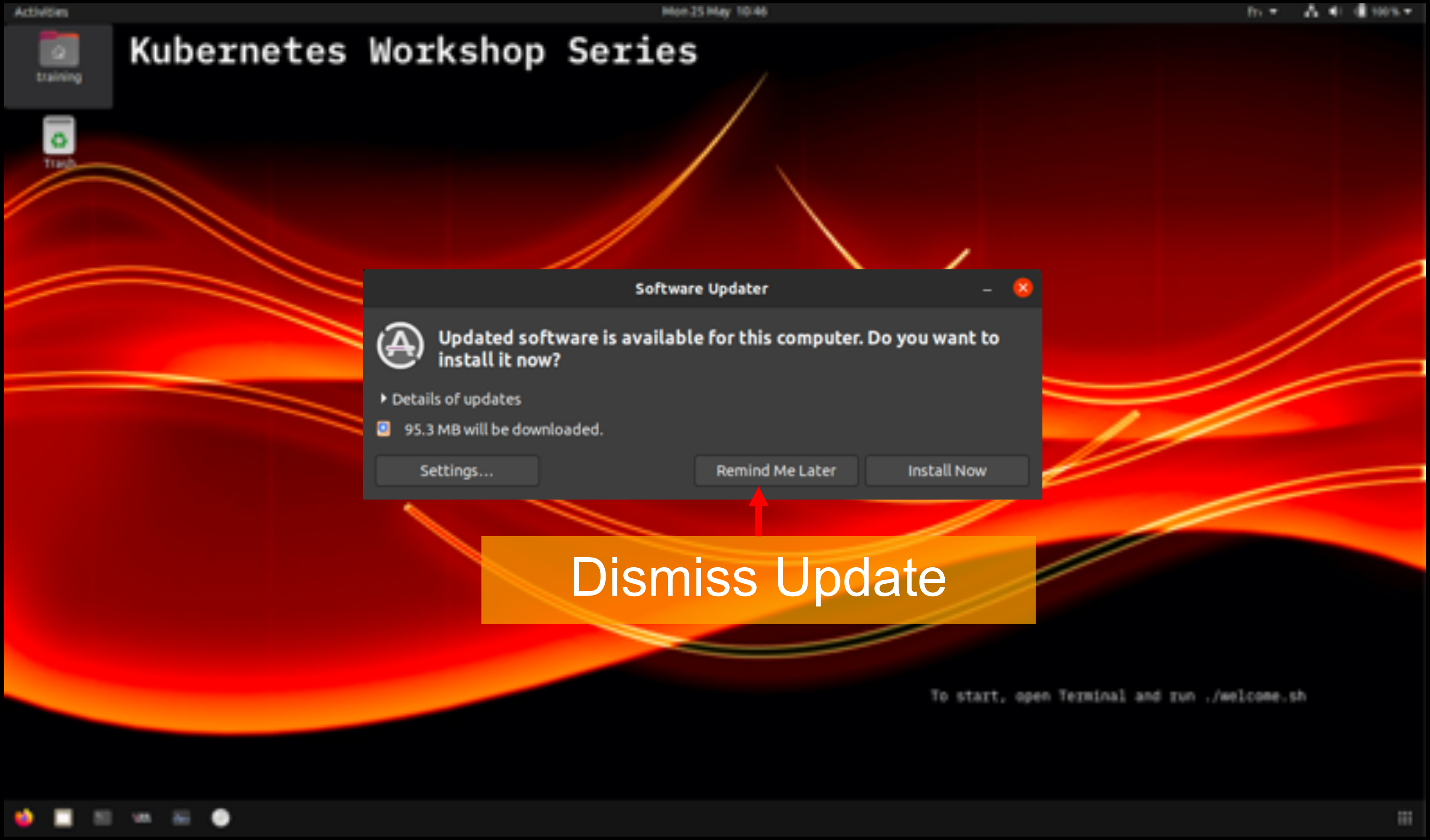
Task 1: Download Training VM

Task 2: Setup VMWare / VirtualBox

Task 3: Start Training VM

Task 4: Login / Check

Accessing your Personal Training Environment

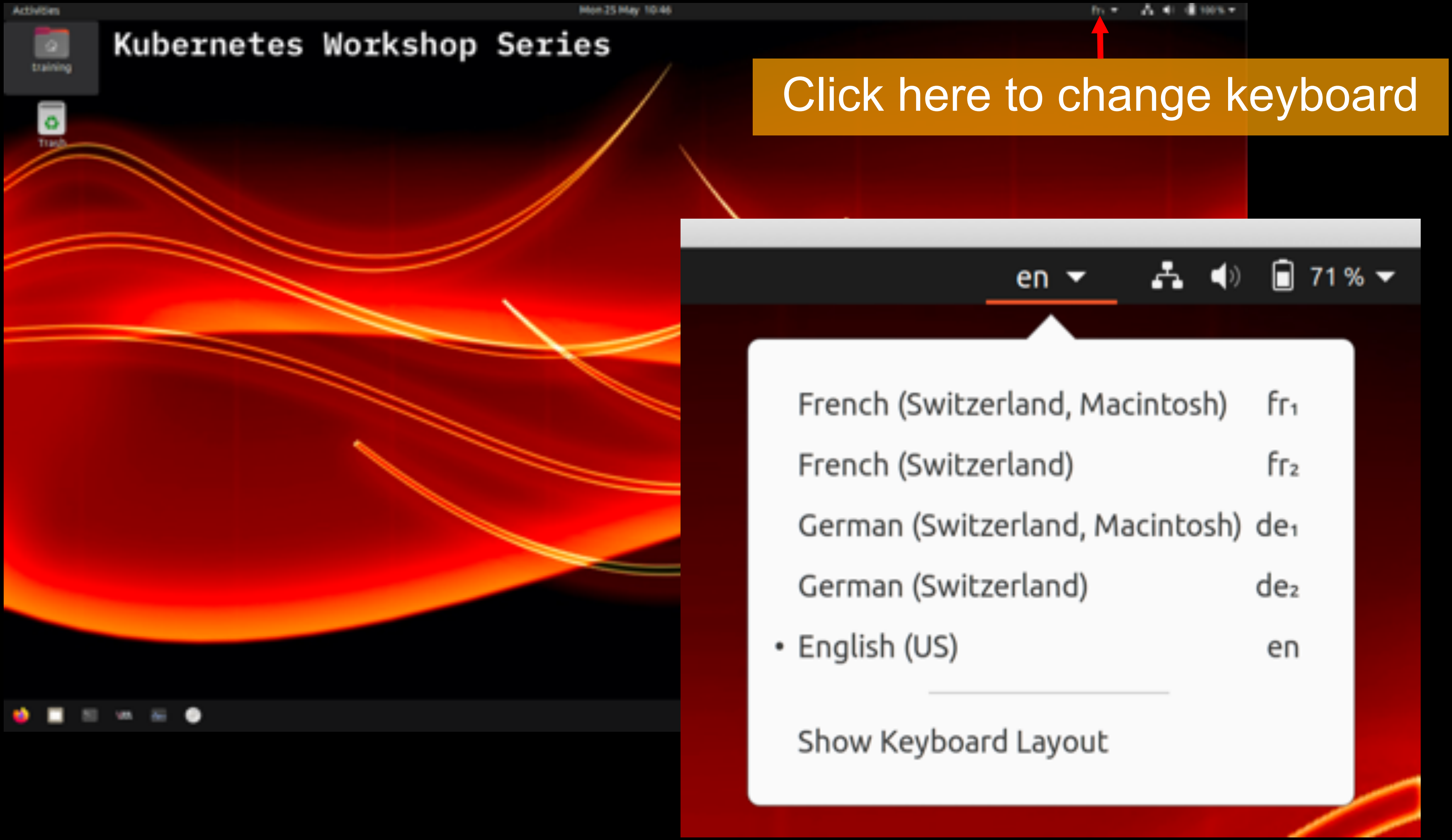


Updated software is available for this computer. Do you want to install it now?

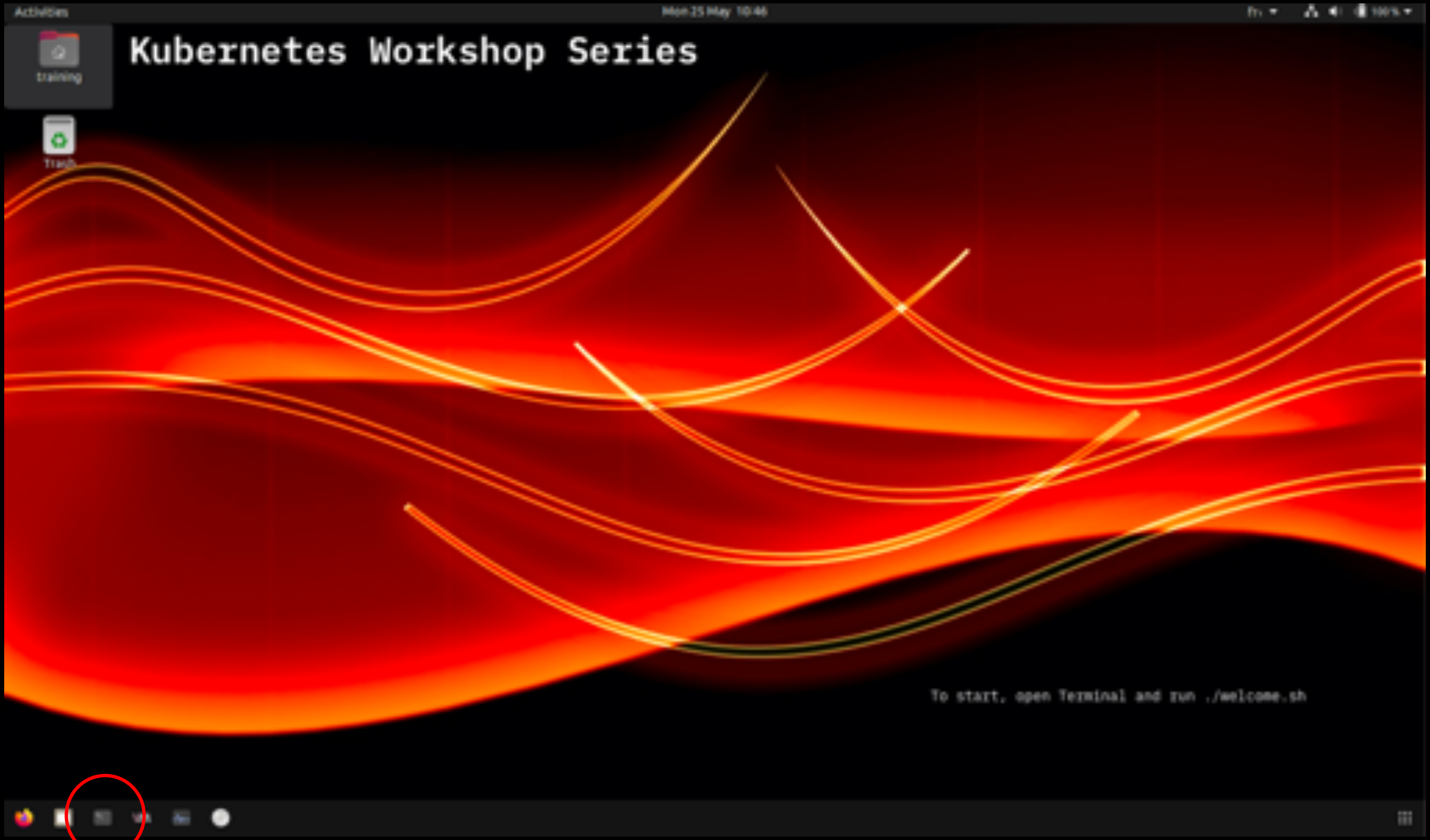
95.3 MB will be downloaded.

To start, open Terminal and run ./welcome.sh

Accessing your Personal Training Environment


Kubernetes Workshop Series
Click here to change keyboard
en
71 %
French (Switzerland, Macintosh) fr₁
French (Switzerland) fr₂
German (Switzerland, Macintosh) de₁
German (Switzerland) de₂
• English (US) en
Show Keyboard Layout

Accessing your Personal Training Environment

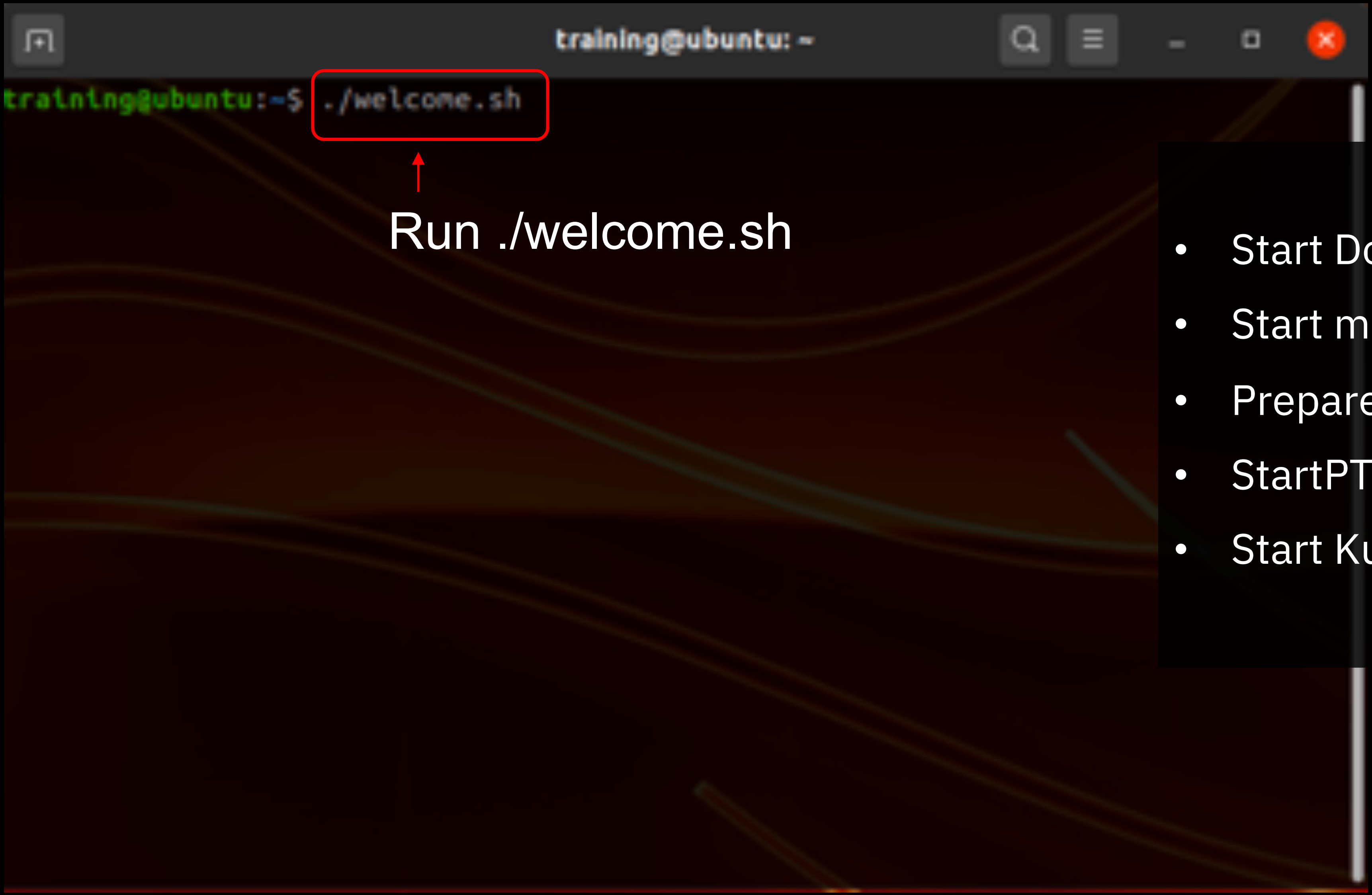


Kubernetes Workshop Series

To start, open Terminal and run ./welcome.sh

Start Terminal

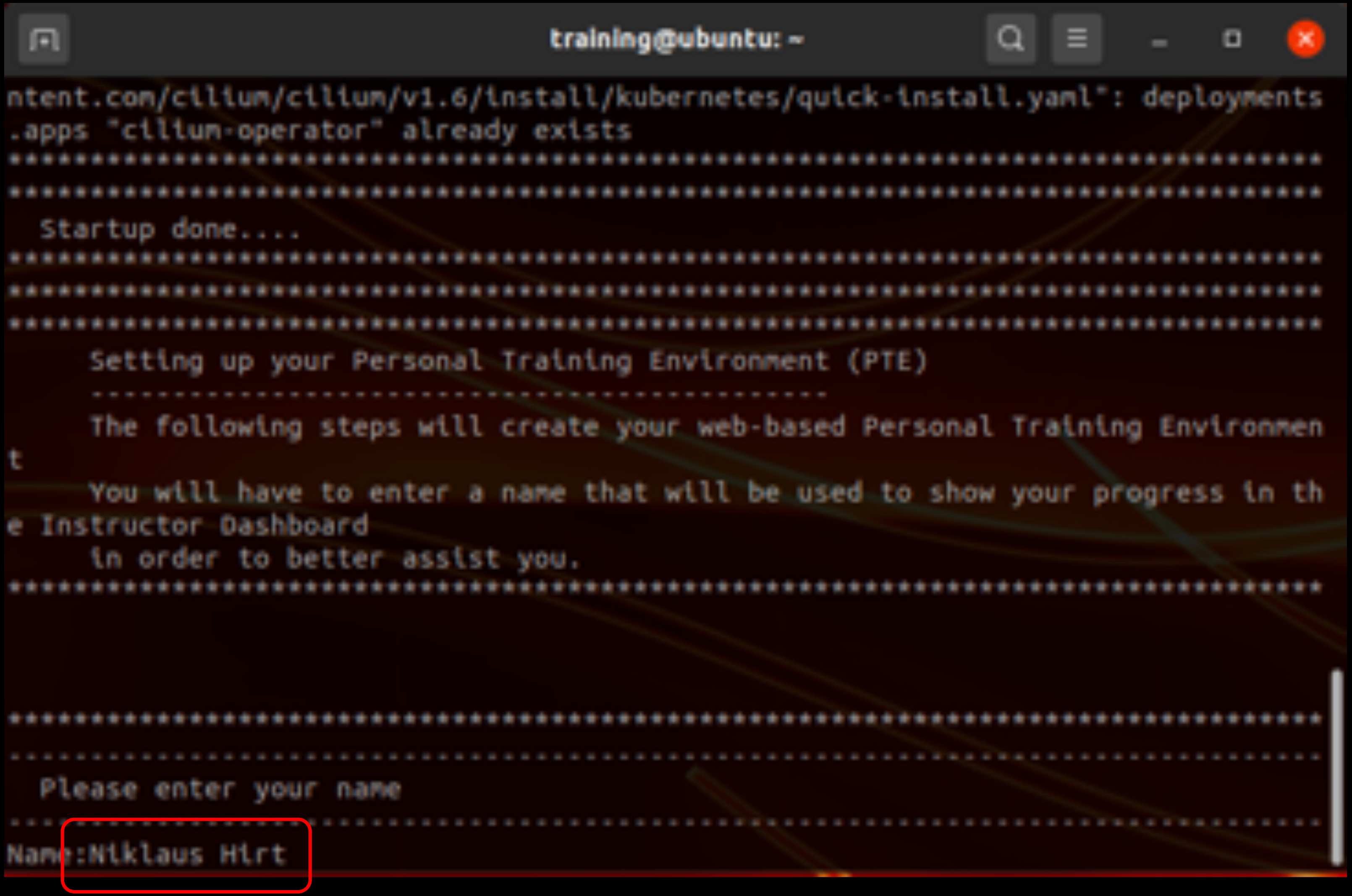
Accessing your Personal Training Environment



```
training@ubuntu:~$ ./welcome.sh
```

- Start Docker
- Start minikube
- Prepares networking
- StartPTE
- Start Kubernetes Dashboard

Accessing your Personal Training Environment



```
ntent.com/cilium/cilium/v1.6/install/kubernetes/quick-install.yaml": deployments
.apps "cilium-operator" already exists
******************************************************************************
******************************************************************************
  Startup done....
******************************************************************************
******************************************************************************
******************************************************************************
     Setting up your Personal Training Environment (PTE)
     --------------------------------------------
     The following steps will create your web-based Personal Training Environmen
t
     You will have to enter a name that will be used to show your progress in th
e Instructor Dashboard
     in order to better assist you.
******************************************************************************



******************************************************************************
------------------------------------------------------------------------------
  Please enter your name
------------------------------------------------------------------------------
Name:Niklaus Hirt
```

Enter your name

Name will be used to show your progress in the Instructor Dashboard in order to better assist you

Accessing your Personal Training Environment

Troubleshooting

- If the startup script doesn’t work you can run ./resetEnvironment.sh
  (this can take up to 30 minutes as it has to redownload all Docker images)

- If you lose your PTE Webpage just run `minikube service student-ui`

- Windows 10 problems can mostly be fixed by turning off Hyper-V by running (as admin)
  ```
  bcdedit /set hypervisorlaunchtype off
  ```
  and rebooting.
  This disables Hyper-V and allows Virtualbox to support nested virtualisation.

- You can turn it back on again with
  ```
  bcdedit /set hypervisorlaunchtype auto
  ```

Accessing your Personal Training Environment

Troubleshooting

I have added a standalone version to the Git repository for participants wishing to run the Labs directly on their PC.

This is **untested** and I cannot guarantee that all the Labs will be working 100%.

You must have the following setup on your PC:

- Minikube
- Docker
- Git

1. Clone the repository to your home directory
   ```
   git clone https://github.com/niklaushirt/training.git
   ```
2. Go to the installation directory
   ```
   cd ~/training/standalone
   ```
3. Run the preparation script
   ```
   ./welcome.sh
   ```

Accessing your Personal Training Environment

Troubleshooting

- Run ***k9s*** in the Terminal – wait for all the pods to be Running (blue – 1/1)

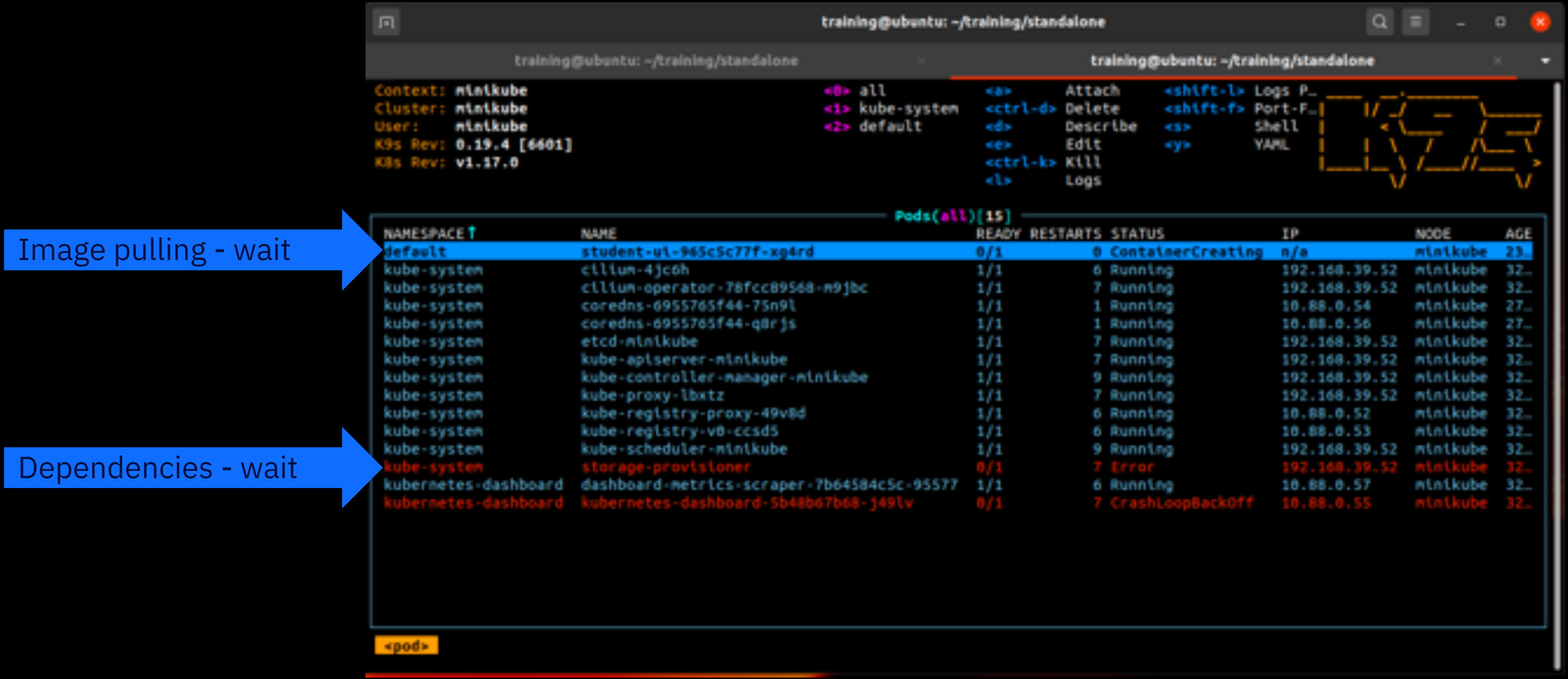


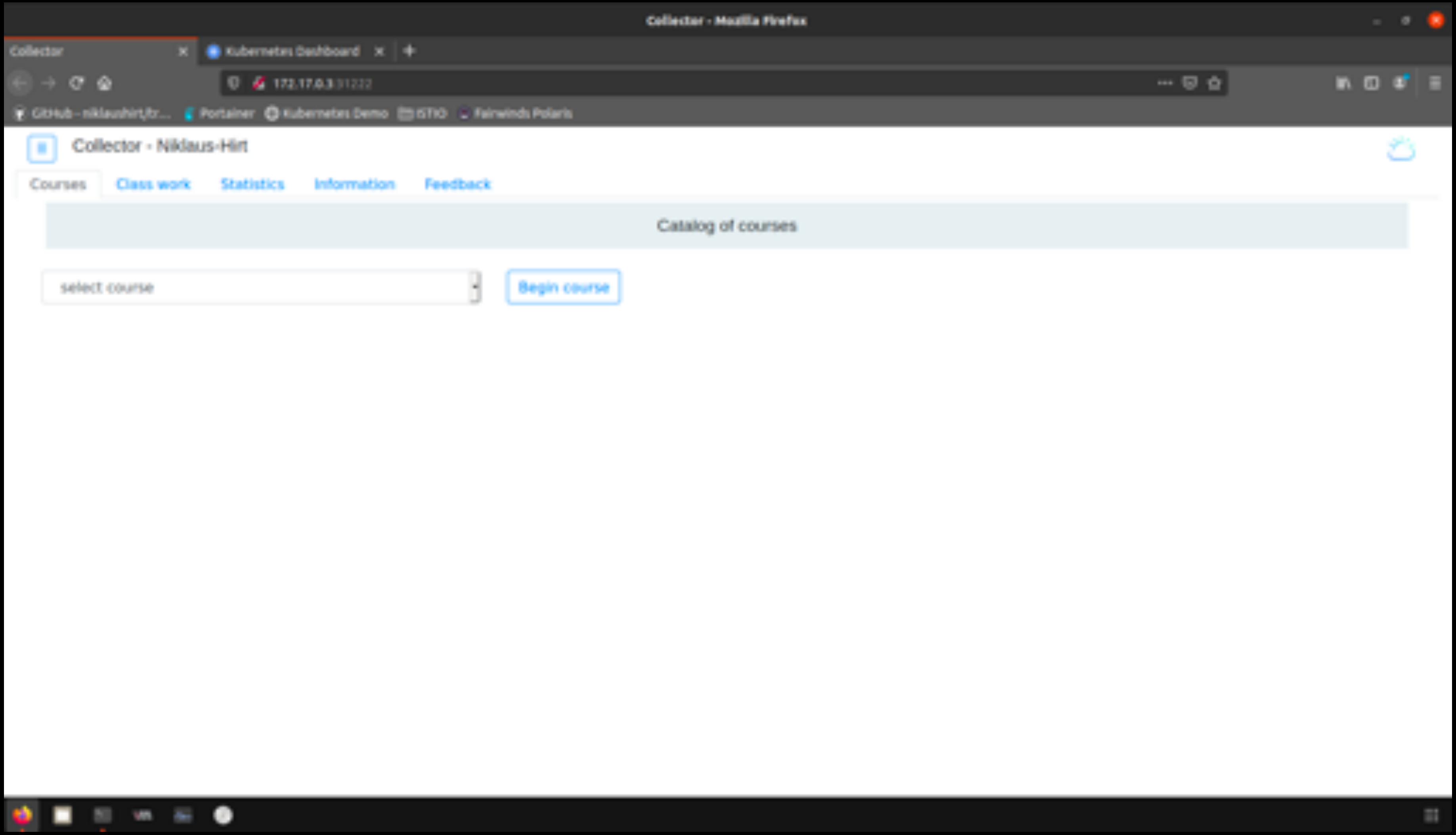
Image pulling - wait
Dependencies - wait
training@ubuntu: ~/training/standalone
training@ubuntu: ~/training/standalone
training@ubuntu: ~/training/standalone

```
Context: minikube          <0> all           <a>        Attach     <shift-l> Logs P…
Cluster: minikube          <1> kube-system   <ctrl-d>   Delete     <shift-f> Port-F…
User:    minikube          <2> default       <d>        Describe   <s>       Shell
K9s Rev: 0.19.4 [6601]                       <e>        Edit       <y>       YAML
K8s Rev: v1.17.0                             <ctrl-k>   Kill
                                             <l>        Logs

                                  Pods(all)[15]
NAMESPACE↑            NAME                                         READY RESTARTS STATUS             IP              NODE      AGE
default               student-ui-965c5c77f-xg4rd                   0/1          0 ContainerCreating  n/a             minikube  23…
kube-system           cilium-4jc6h                                 1/1          6 Running            192.168.39.52   minikube  32…
kube-system           cilium-operator-78fcc89568-m9jbc             1/1          7 Running            192.168.39.52   minikube  32…
kube-system           coredns-6955765f44-75n9l                     1/1          1 Running            10.88.0.54      minikube  27…
kube-system           coredns-6955765f44-q8rjs                     1/1          1 Running            10.88.0.56      minikube  27…
kube-system           etcd-minikube                                1/1          7 Running            192.168.39.52   minikube  32…
kube-system           kube-apiserver-minikube                      1/1          7 Running            192.168.39.52   minikube  32…
kube-system           kube-controller-manager-minikube             1/1          9 Running            192.168.39.52   minikube  32…
kube-system           kube-proxy-lbxtz                             1/1          7 Running            192.168.39.52   minikube  32…
kube-system           kube-registry-proxy-49v8d                    1/1          6 Running            10.88.0.52      minikube  32…
kube-system           kube-registry-v0-ccsd5                       1/1          6 Running            10.88.0.53      minikube  32…
kube-system           kube-scheduler-minikube                      1/1          9 Running            192.168.39.52   minikube  32…
kube-system           storage-provisioner                          0/1          7 Error              192.168.39.52   minikube  32…
kubernetes-dashboard  dashboard-metrics-scraper-7b64584c5c-95577   1/1          6 Running            10.88.0.57      minikube  32…
kubernetes-dashboard  kubernetes-dashboard-5b48b67b68-j49lv        0/1          7 CrashLoopBackOff   10.88.0.55      minikube  32…

<pod>
```

Accessing your Personal Training Environment


Collector - Mozilla Firefox

Collector - Niklaus-Hirt

Courses | Class work | Statistics | Information | Feedback

Catalog of courses

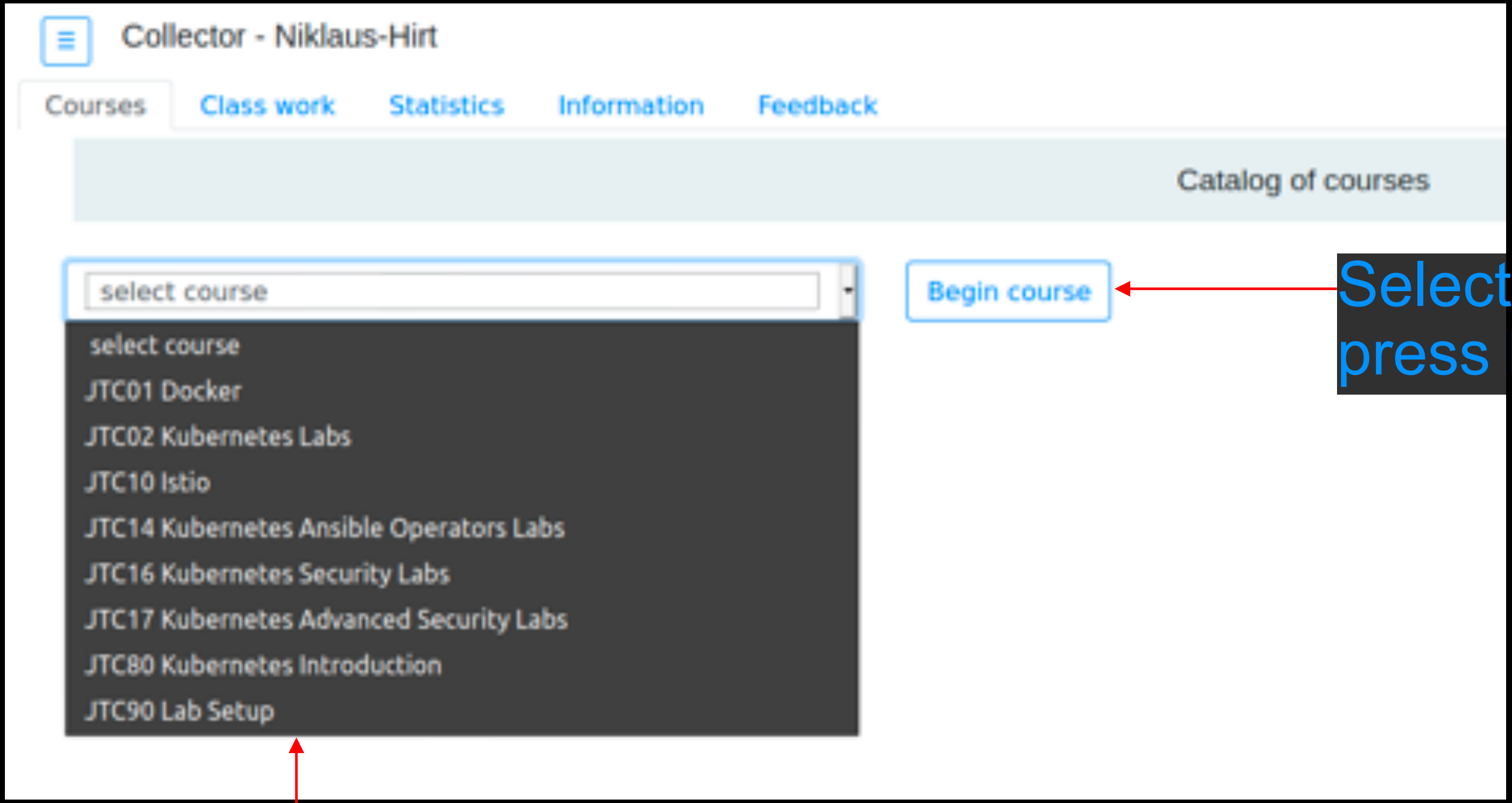
select course

Begin course

When completed, your PTE and Kubernetes Dashboard will open automatically

Accessing your Personal Training Environment

Name will be shown



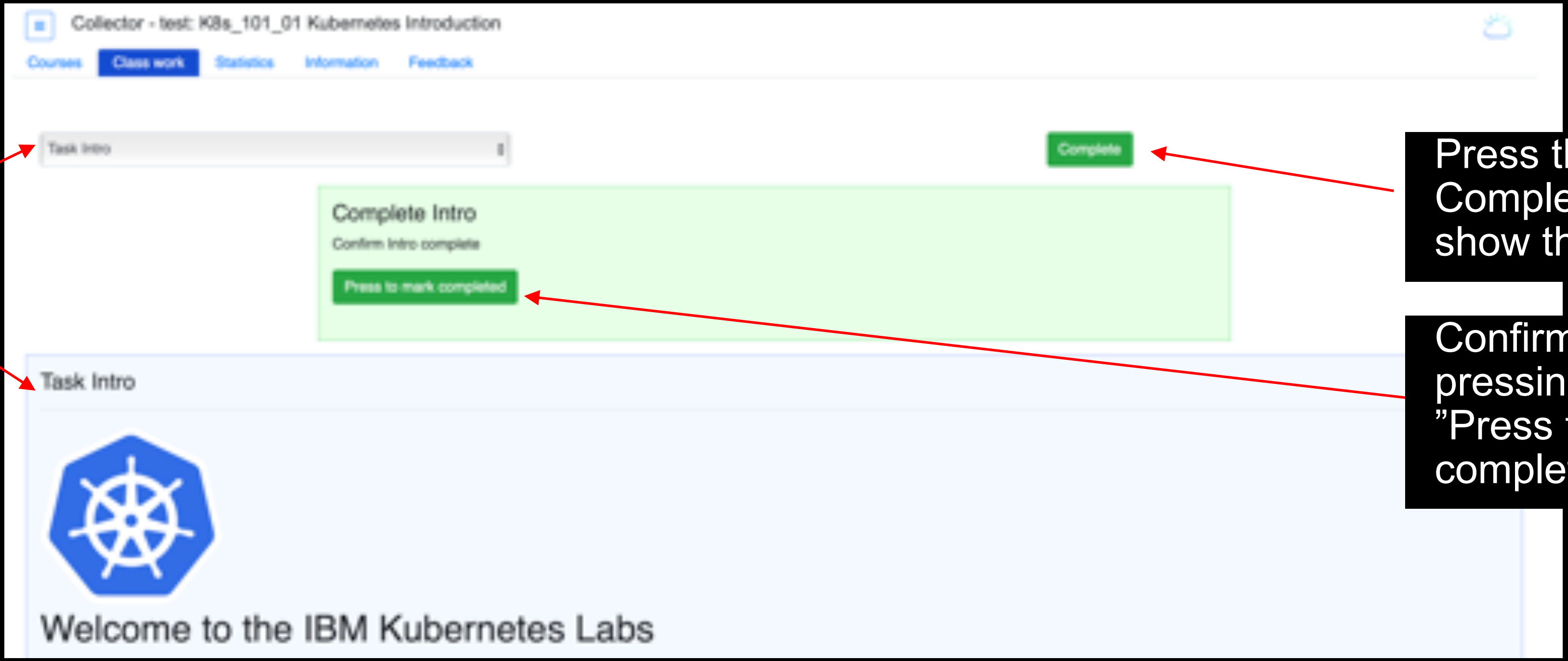
Collector - Niklaus-Hirt

select course
JTC01 Docker
JTC02 Kubernetes Labs
JTC10 Istio
JTC14 Kubernetes Ansible Operators Labs
JTC16 Kubernetes Security Labs
JTC17 Kubernetes Advanced Security Labs
JTC80 Kubernetes Introduction
JTC90 Lab Setup

Select course and
press button to begin

Current course catalog

Class Work



Complete Intro

Confirm Intro complete

Welcome to the IBM Kubernetes Labs

Select class work and the blue portion of the screen is shown

Press the green Complete button to show the green portion.

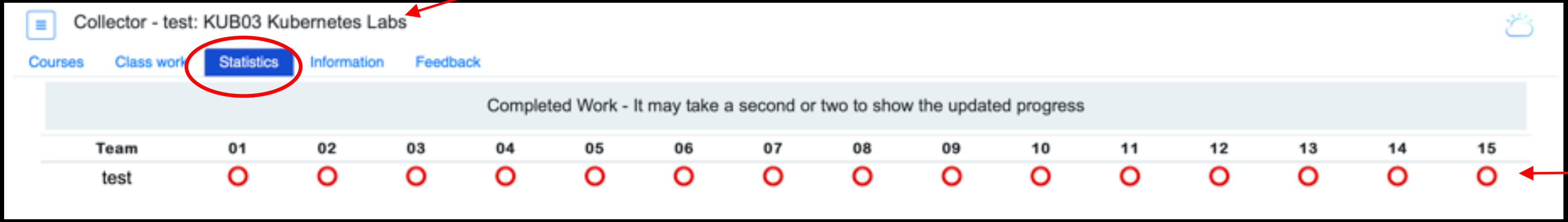
Confirm completion by pressing the green ”Press to mark completed” button.

!

The Complete Button might not show instantly depending on the course settings

Following your progress

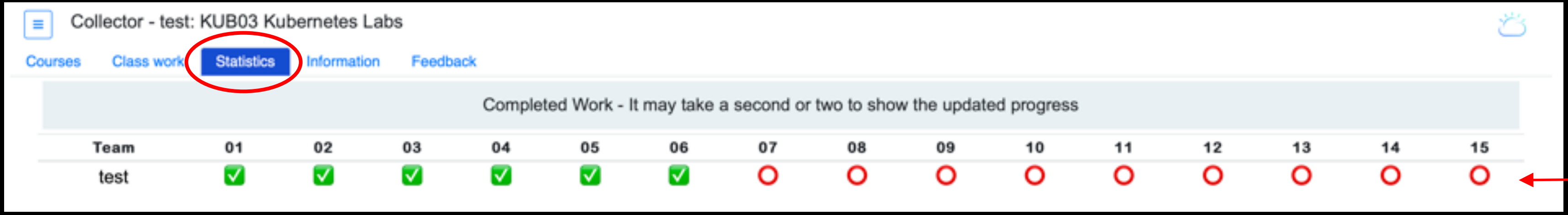
Course title



Collector - test: KUB03 Kubernetes Labs

Completed Work - It may take a second or two to show the updated progress

Team	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
test	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○



Collector - test: KUB03 Kubernetes Labs

Completed Work - It may take a second or two to show the updated progress

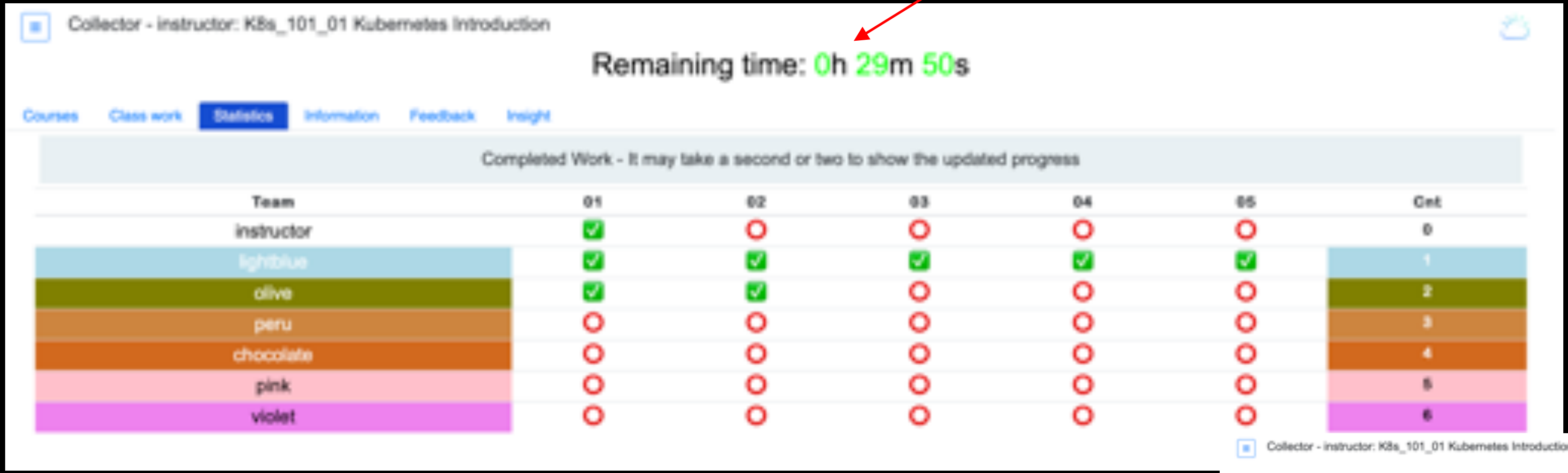
Team	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
test	✓	✓	✓	✓	✓	✓	○	○	○	○	○	○	○	○	○

The number of items tracked will change based on the current course selected.

Green checkmark - item is completed
Red circle - item is waiting to be completed

Instructor Dashboard

Remaining Time for the Lab

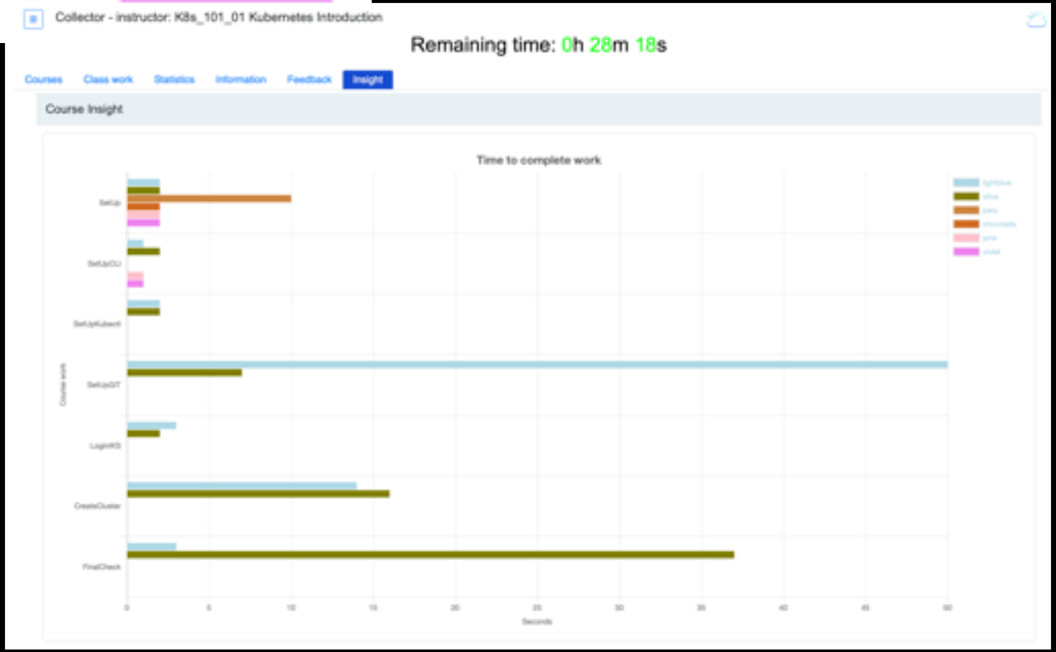


Collector - instructor: K8s_101_01 Kubernetes Introduction

Remaining time: 0h 29m 50s

Completed Work - It may take a second or two to show the updated progress

Team	01	02	03	04	05	Cnt
instructor	✓	○	○	○	○	0
lightblue	✓	✓	✓	✓	✓	1
olive	✓	✓	○	○	○	2
peru	○	○	○	○	○	3
chocolate	○	○	○	○	○	4
pink	○	○	○	○	○	5
violet	○	○	○	○	○	6



Collector - instructor: K8s_101_01 Kubernetes Introduction

Remaining time: 0h 28m 18s

QUESTIONS?

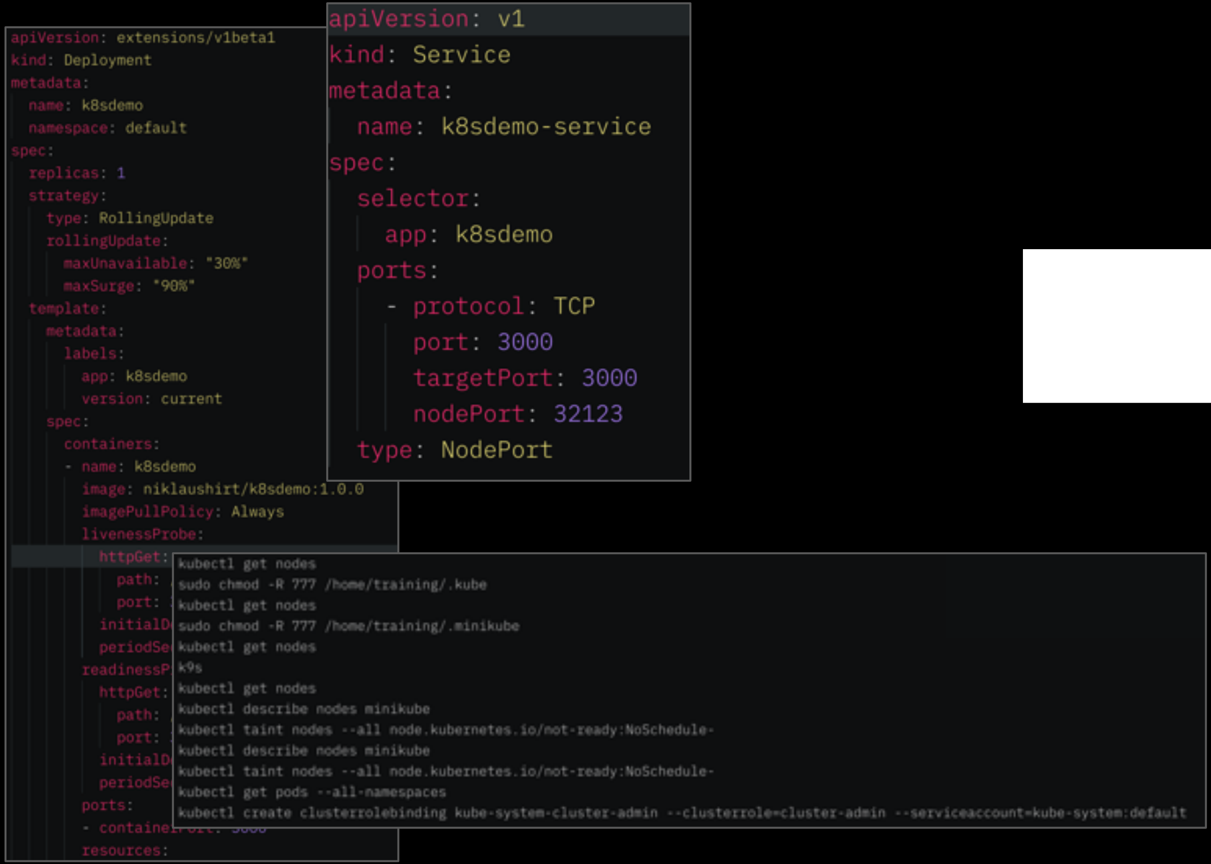
Kubernetes Workshop Series

Kubernetes Operators

01

Kubernetes Operators

From this…

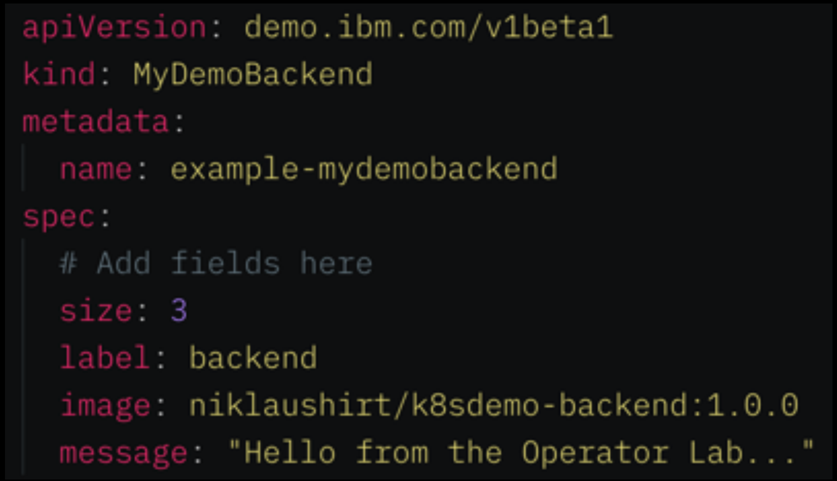


```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: k8sdemo
  namespace: default
spec:
  replicas: 1
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxUnavailable: "30%"
      maxSurge: "90%"
  template:
    metadata:
      labels:
        app: k8sdemo
        version: current
    spec:
      containers:
      - name: k8sdemo
        image: niklaushirt/k8sdemo:1.0.0
        imagePullPolicy: Always
        livenessProbe:
          httpGet:
            path:
            port:
          initialD
          periodSe
        readinessP
          httpGet:
            path:
            port:
          initialD
          periodSe
        ports:
        - containe
        resources:
```

```
apiVersion: v1
kind: Service
metadata:
  name: k8sdemo-service
spec:
  selector:
    app: k8sdemo
  ports:
    - protocol: TCP
      port: 3000
      targetPort: 3000
      nodePort: 32123
  type: NodePort
```

```
kubectl get nodes
sudo chmod -R 777 /home/training/.kube
kubectl get nodes
sudo chmod -R 777 /home/training/.minikube
kubectl get nodes
k9s
kubectl get nodes
kubectl describe nodes minikube
kubectl taint nodes --all node.kubernetes.io/not-ready:NoSchedule-
kubectl describe nodes minikube
kubectl taint nodes --all node.kubernetes.io/not-ready:NoSchedule-
kubectl get pods --all-namespaces
kubectl create clusterrolebinding kube-system-cluster-admin --clusterrole=cluster-admin --serviceaccount=kube-system:default
```

… to this



```
apiVersion: demo.ibm.com/v1beta1
kind: MyDemoBackend
metadata:
  name: example-mydemobackend
spec:
  # Add fields here
  size: 3
  label: backend
  image: niklaushirt/k8sdemo-backend:1.0.0
  message: "Hello from the Operator Lab..."
```

Kubernetes Operators

The Operator Framework is an **open source toolkit** to **manage Kubernetes native applications**, called Operators, in an effective, automated, and scalable way.

https://operatorhub.io/

Kubernetes Operators

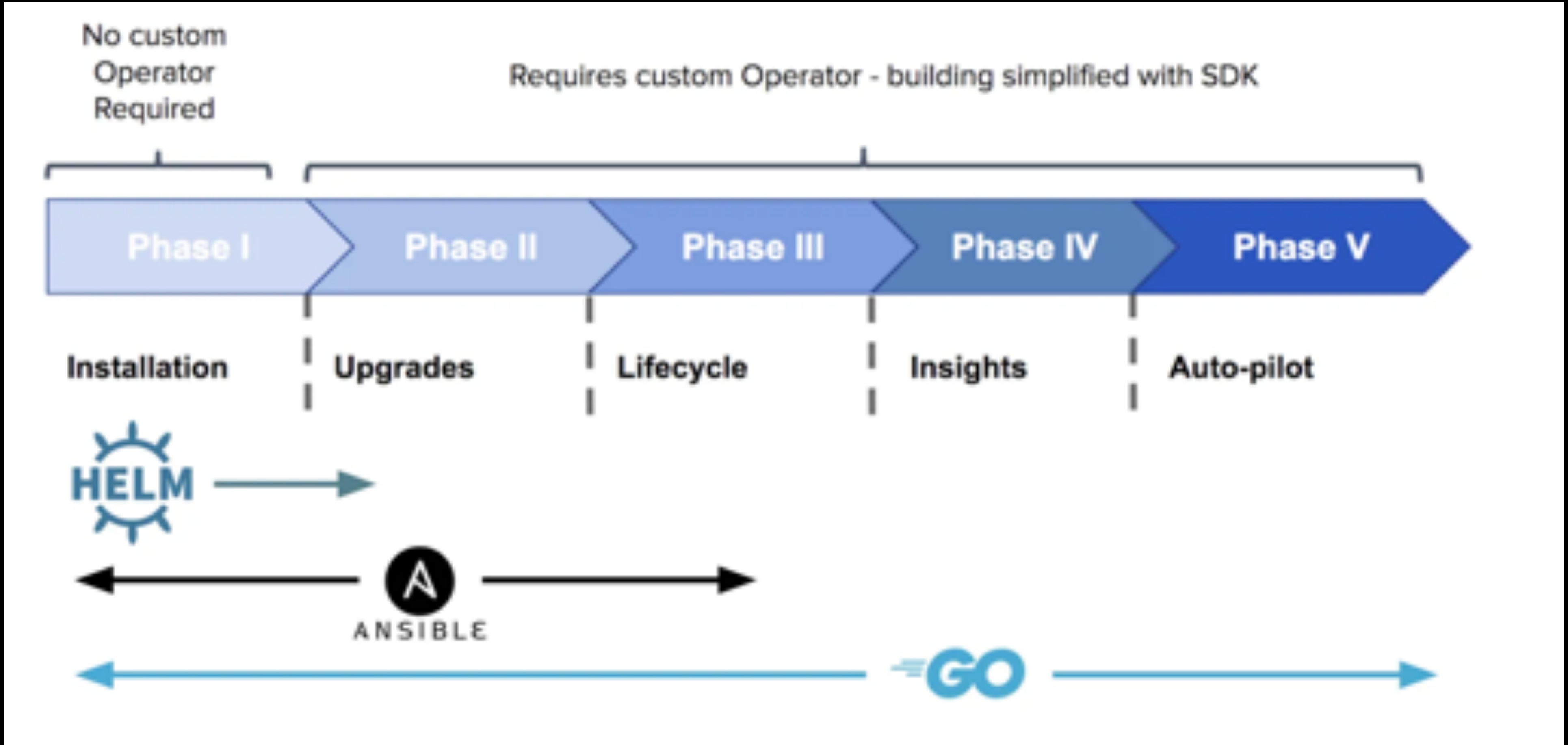
Operators are a **design pattern** made public in a 2016 CoreOS blog post.

The goal of an Operator is to put **operational knowledge into software**.

Previously this knowledge only resided in the minds of administrators, various combinations of shell scripts or automation software like Ansible. It was outside of your Kubernetes cluster and hard to integrate.

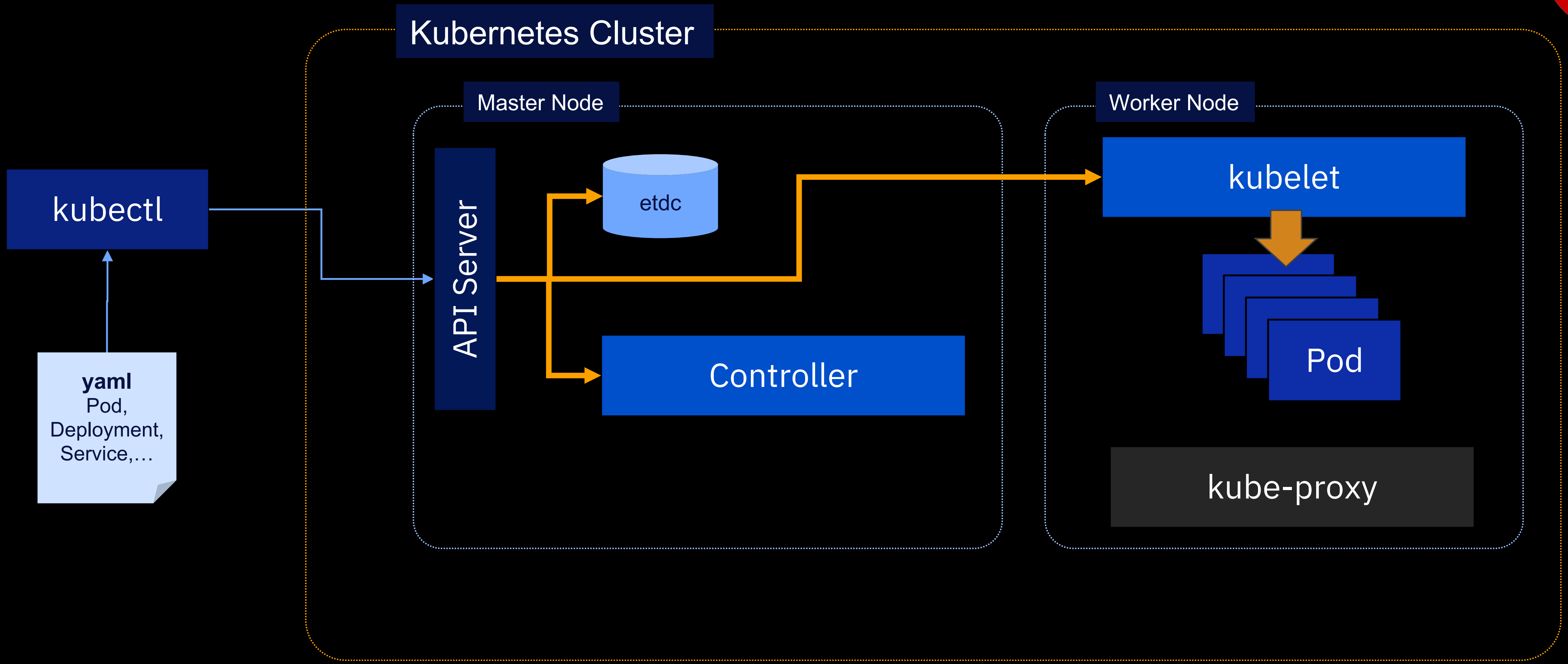
Operators implement and automate common **Day-1** (installation, configuration, etc) and **Day-2** (re-configuration, update, backup, failover, restore, etc.) **activities** in a piece of software running inside your Kubernetes cluster, by integrating natively with Kubernetes concepts and APIs.

Kubernetes Operators



Operator Maturity Model from the Operator SDK

Kubernetes Operators



Kubernetes Operators

Operators extend Kubernetes by allowing you to define a **Custom Controller** to watch your application and perform custom tasks based on its state (a perfect fit to automate maintenance of the stateful application we described above).

The application you want to watch is defined in Kubernetes as a new object: a **Custom Resource** (CR) that has its own yaml spec and object type (in K8s, a kind) that is understood by the API server.

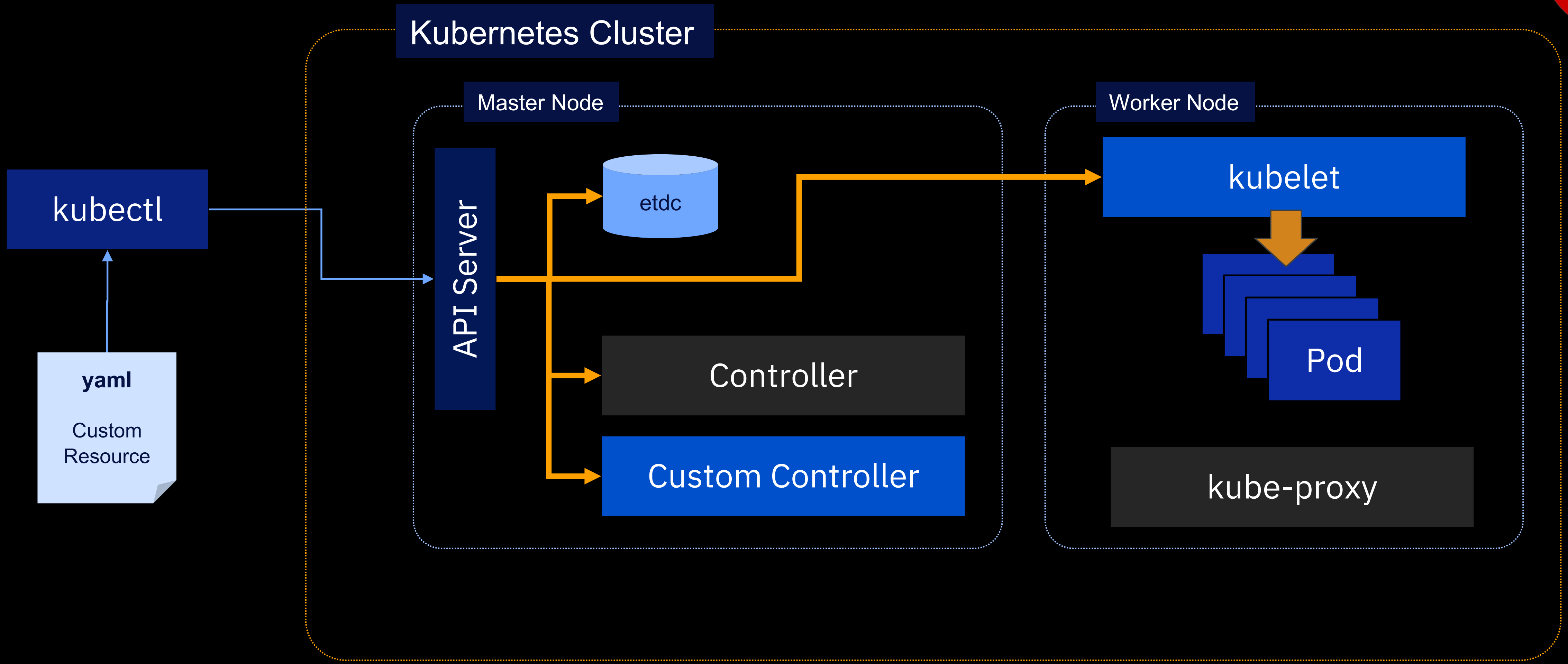
That way, you can define any specific criteria in the custom spec to watch out for, and reconcile the instance when it doesn’t match the spec.
The way an operator’s controller reconciles against a spec is very similar to native Kubernetes’ controllers, though it is using mostly custom components.

Kubernetes Operators

- A **Custom Resource Definition** (CRD) spec that defines the format of the Custom Resource
- A **Custom Controller** to watch our application

  Custom code within the new controller that dictates how to reconcile our CR against the spec
- An **Operator** to manage the Custom Controller
- A **Custom Resource** (CR) spec that defines the application we want to watch
- A **deployment** for the Operator and Custom Resource

Kubernetes Operators



Kubernetes Operators – Create Operator

Enables developers to build Operators based on their expertise without requiring knowledge of Kubernetes API complexities.

```
# Create operator project
operator-sdk new ansible-operator-frontend
  --type=ansible
  --api-version=ansiblenlab.ibm.com/v1beta1
  --kind=MyAnsibleLabDemo

# Create operator controller
gedit   ansible-operator-frontend/roles/myansiblelabdemo/tasks/main.yml

# Build operator project
operator-sdk build localhost:5000/ansible-operator-frontend:ansible
```

Kubernetes Operators – Create the Lab Operator Project

```
operator-sdk new ansible-operator-frontend
  --type=ansible
  --api-version=ansiblenlab.ibm.com/v1beta1
  --kind=MyAnsibleLabDemo
```

ansiblenlab_v1beta1_myansiblelabdemo_cr

```
apiVersion: ansiblenlab.ibm.com/v1beta1
kind: MyAnsibleLabDemo
metadata:
  name: example-MyAnsibleLabDemo
spec:
  # Add fields here
  size: 3
```

Kubernetes Operators – Create the Lab Operator Project

```
operator-sdk new ansible-operator-frontend
  --type=ansible
  --api-version=ansiblenlab.ibm.com/v1beta1
  --kind=MyAnsibleLabDemo
```

ansiblenlab_v1beta1_myansiblelabdemo_cr

```
apiVersion: ansiblenlab.ibm.com/v1beta1
kind: MyAnsibleLabDemo
metadata:
  name: example-MyAnsibleLabDemo
spec:
  # Add fields here
  size: 3
  demo:
    image: niklaushirt/k8sdemo:1.0.0
```

Kubernetes Operators – Define Controller

ansible-operator-frontend/roles/myansiblelabdemo/tasks/main.yml

Ansible

Kubernetes

```
- name: Create the k8sdemo service
  k8s:
    definition:
      apiVersion: v1
      kind: Service
      metadata:
        name: k8sdemo-service
        namespace: default
      spec:
        selector:
          app: k8sdemo
        ports:
          - protocol: TCP
            port: 3000
            targetPort: 3000
            nodePort: 32123
        type: NodePort
```

Ansible code that creates a Kubernetes Object

Kubernetes Operators – Define Controller

ansible-operator-frontend/roles/myansiblelabdemo/tasks/main.yml

Ansible

Kubernetes

```
- name: Create the k8sdemo deployment
  k8s:
    definition:
      apiVersion: apps/v1
      kind: Deployment
      metadata:
        name: k8sdemo
        namespace: default
…
        replicas: 1
        template:
…
          spec:
            containers:
            - name: k8sdemo
              image: "{{demo.image}}"
              imagePullPolicy: IfNotPresent
…
```

```
apiVersion: ansiblenlab.ibm.com/v1beta1
kind: MyAnsibleLabDemo
metadata:
  name: example-MyAnsibleLabDemo
spec:
  # Add fields here
  size: 3
  demo:
    image: niklaushirt/k8sdemo:1.0.0
```

Ansible code that creates a Kubernetes Object

Kubernetes Operators – Build the Operator

```
operator-sdk build localhost:5000/ansible-operator-frontend:ansible


docker push localhost:5000/ansible-operator-frontend:ansible
```

Kubernetes Operators – Deploy the Operator

Create the Custom Resource Definition

```
kubectl create -f
   ~/ansible-operator/ansible-operator-frontend/deploy/crds/ansiblenlab_v1beta1_myansiblelabdemo_crd.yaml
```

Create the ServiceAccount

```
kubectl create -f ~/ansible-operator/ansible-operator-frontend/deploy/service_account.yaml kubectl create -f
~/ansible-operator/ansible-operator-frontend/deploy/role.yaml kubectl create -f ~/ansible-operator/ansible-
operator-frontend/deploy/role_binding.yaml
```

Create the Operator

```
kubectl create -f ~/ansible-operator/ansible-operator-frontend/deploy/operator.yaml
```

```
kubectl get pods

> NAME                                   READY   STATUS    RESTARTS   AGE
> ansible-operator-frontend-fd78bcf5-zxgws   1/1     Running   0          43m
```

Kubernetes Operators – Deploy the Custom Resource

```
kubectl create
 -f ~/ansible-operator/ansible-operator-frontend/deploy/crds/ansiblenlab_v1beta1_myansiblelabdemo_cr.yaml


kubectl get pods

> NAME                                        READY   STATUS    RESTARTS   AGE
> ansible-operator-frontend-fd78bcf5-zxgws     2/2     Running   0          3m11s
> k8sdemo-7fc8554dff-2krkz                     1/1     Running   0          45s


kubectl logs -c operator ansible-operator-frontend-fd78bcf5-zxgws

> … msg":"Watching resource","Options.Group":"ansiblenlab.ibm.com",
      "Options.Version":"v1beta1","Options.Kind":"MyAnsibleLabDemo"}
> … Starting EventSource","controller":"myansiblelabdemo-controller",
     "source":"kind source: ansiblenlab.ibm.com/v1beta1, Kind=MyAnsibleLabDemo"}
> … "msg":"Starting Controller","controller":"myansiblelabdemo-controller"}
> … "logging_event_handler","msg":"[playbook task]","name":"example-myansiblelabdemo",
      "namespace":"default","gvk":"ansiblenlab.ibm.com/v1beta1, Kind=MyAnsibleLabDemo",...
```

TODO

ANSIBLE OPERATOR

Kubernetes Operators – Create Operator

OPERATOR SDK

Enables developers to build Operators based on their expertise without requiring knowledge of Kubernetes API complexities.

```
# Create operator project
operator-sdk new lab-operator

# Create operator API / CRD
operator-sdk add api --api-version=lab.ibm.com/v1beta1 --kind=MyLabDemo

# Create operator controller
operator-sdk add controller --api-version=lab.ibm.com/v1beta1 --kind=MyLabDemo

# Build operator project
operator-sdk build localhost:5000/lab-operator:v0.0.1
```

Kubernetes Operators – Define API

./pkg/apis/lab/v1beta1/mylabdemo_types.go

```go
type MyLabDemoSpec struct {
  // Image is the Docker image to run for the daemon
  Image string `json:"image"`
 }
```

custom_resource.yaml

```yaml
apiVersion: lab.ibm.com/v1beta1
kind: MyLabFrontend
metadata:
name: labdemo
spec:
  image: niklaushirt/k8sdemo:1.0.0
```

Kubernetes Operators – Define Controller

./pkg/controller/mylabdemo/ mylabdemo_controller.go

```
// Watch for changes to secondary resource Deployment

err = c.Watch(&source.Kind{Type: &appsv1.Deployment{}}, &handler.EnqueueRequestForOwner{
    IsController: true,
    OwnerType: &labv1beta1.MyLabFrontend{},
})
```

Kubernetes Operators – Define Controller

./pkg/controller/mylabdemo/ mylabdemo_controller.go

```
func newMyDemoDeployment(cr *demov1beta1.MyLabDemo) *appsv1.Deployment {
  return &appsv1.Deployment{

   TypeMeta: metav1.TypeMeta{
      Kind: "Deployment",
      APIVersion: "apps/v1",
   },

   ObjectMeta: metav1.ObjectMeta{
      Name: cr.Name + "-deployment",
      Namespace: cr.Namespace,
   },

   Spec: appsv1.DeploymentSpec{
      Selector: &metav1.LabelSelector{
       MatchLabels: map[string]string{"deployment": cr.Name + "-deployment"},
      },
   ….
```

Find the API specifications here: https://godoc.org/k8s.io/api/apps/v1#Deployment

Kubernetes Operators – Define Controller

./pkg/controller/mylabdemo/ mylabdemo_controller.go

```
foundDeployment := &appsv1.Deployment{}
 err = r.client.Get(context.TODO(), types.NamespacedName{Name: deployment.Name,
                                 Namespace: deployment.Namespace}, foundDeployment)

 if err != nil && errors.IsNotFound(err) {

     //If Deplyoment not found → Create a new one
     err = r.client.Create(context.TODO(), deployment)

     // Deployment created successfully - don't requeue
     return reconcile.Result{}, nil
 }
```

Kubernetes Operators – Define Controller

./pkg/controller/mylabdemo/ mylabdemo_controller.go

```
// Check if the Image in CR has changed and update accordingly

  // Image name as specified in the CR
  image:=instance.Spec.Image

  // Image name as specified in the CR
  if foundDeployment.Spec.Template.Spec.Containers[0].Image != image {
      // Update in the existing Deployment definition
      foundDeployment.Spec.Template.Spec.Containers[0].Image = image
       // Update the existing Deployment
       r.client.Update(context.TODO(), foundDeployment)
  }
```

Kubernetes

Assemble vs. Operate

OPERATOR
SDK

TODO

OPERATOR CATALOG

QUESTIONS?

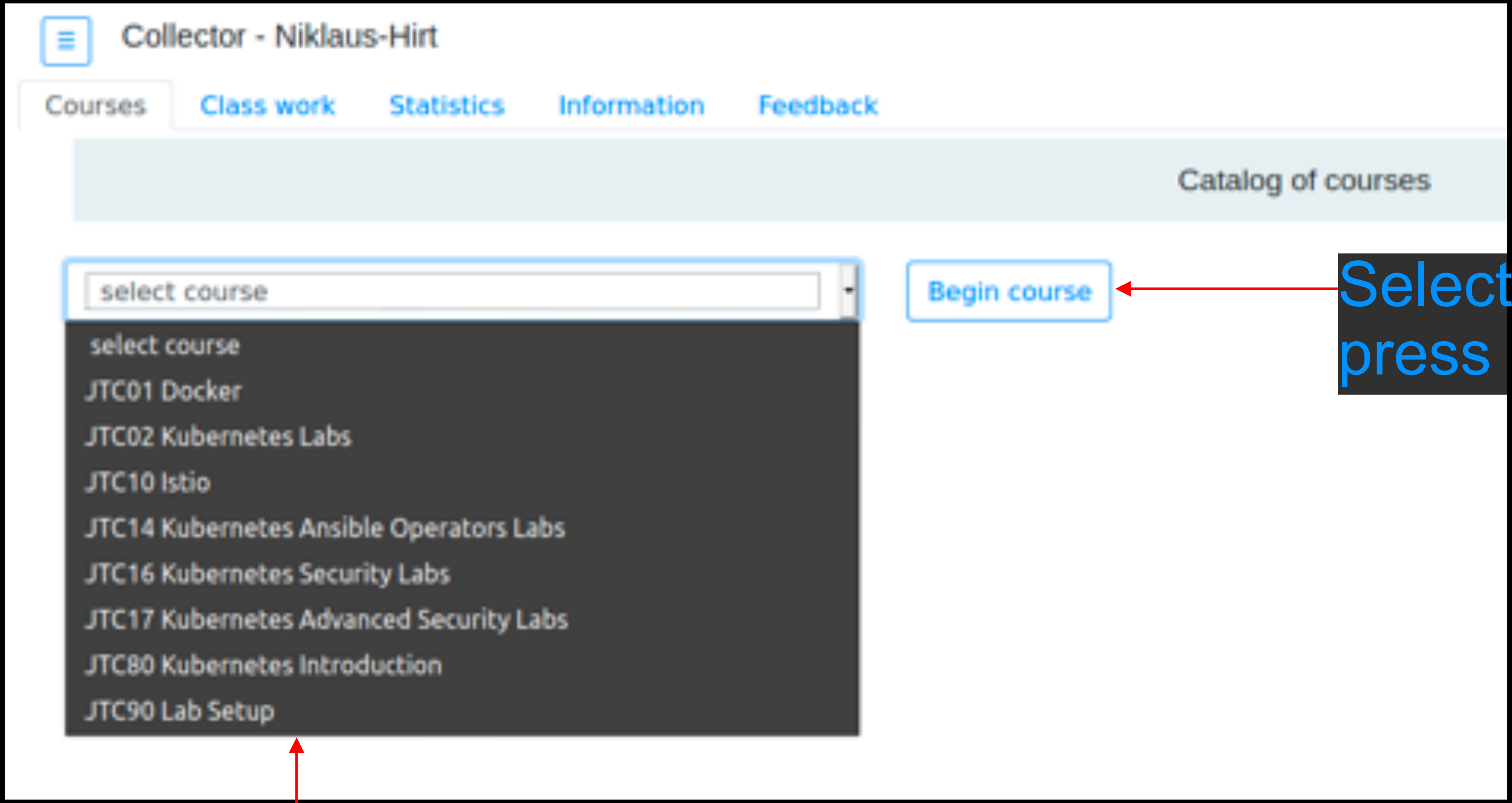
Kubernetes Workshop Series

Operators - *Hands-On*

02

Starting Course JTC14 Ansible Operators

Name will be shown



Collector - Niklaus-Hirt

select course
JTC01 Docker
JTC02 Kubernetes Labs
JTC10 Istio
JTC14 Kubernetes Ansible Operators Labs
JTC16 Kubernetes Security Labs
JTC17 Kubernetes Advanced Security Labs
JTC80 Kubernetes Introduction
JTC90 Lab Setup

Select copurse and press button to begin

Current course catalog

JTC14 Labs

Lab 1 : Introduction

Lab 2 : Ansible Operators

- Creating the Operator Project
- Creating the Operator API
- Creating the Operator Controller
- Build and deploy the Operator
- Create and deploy the Custom Resource
- Update the Custom Resource

READY
SET
GO!!!!

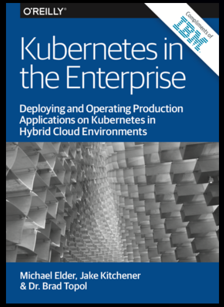
Duration: 90 mins

QUESTIONS?

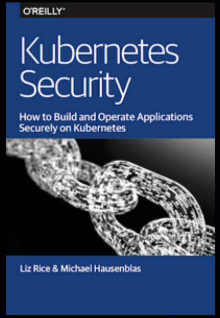
Kubernetes – Some Reading Tips

FREE!
The Cloud Adoption Playbook
Proven strategies for transforming your organization with the cloud
WILEY

The de facto guide to improving your enterprise with the cloud, created by distinguished members of our Solution Engineering team
http://ibm.biz/playbook


O'REILLY
Kubernetes in the Enterprise
Deploying and Operating Production Applications on Kubernetes in Hybrid Cloud Environments
Michael Elder, Jake Kitchener & Dr. Brad Topol

Deploying and Operating Production Applications on Kubernetes in Hybrid Cloud Environments
https://ibm.co/2LQketN (excerpt)


O'REILLY
Kubernetes Security
How to Build and Operate Applications Securely on Kubernetes
Liz Rice & Michael Hausenblas

https://kubernetes-security.info/

Videos, sources and documentation will be available here:

All Workshop Recordings
https://www.youtube.com/channel/UCIS0jmGOQrG2AKKPkTJYj9w/videos

https://github.com/niklaushirt/k8s_training_public

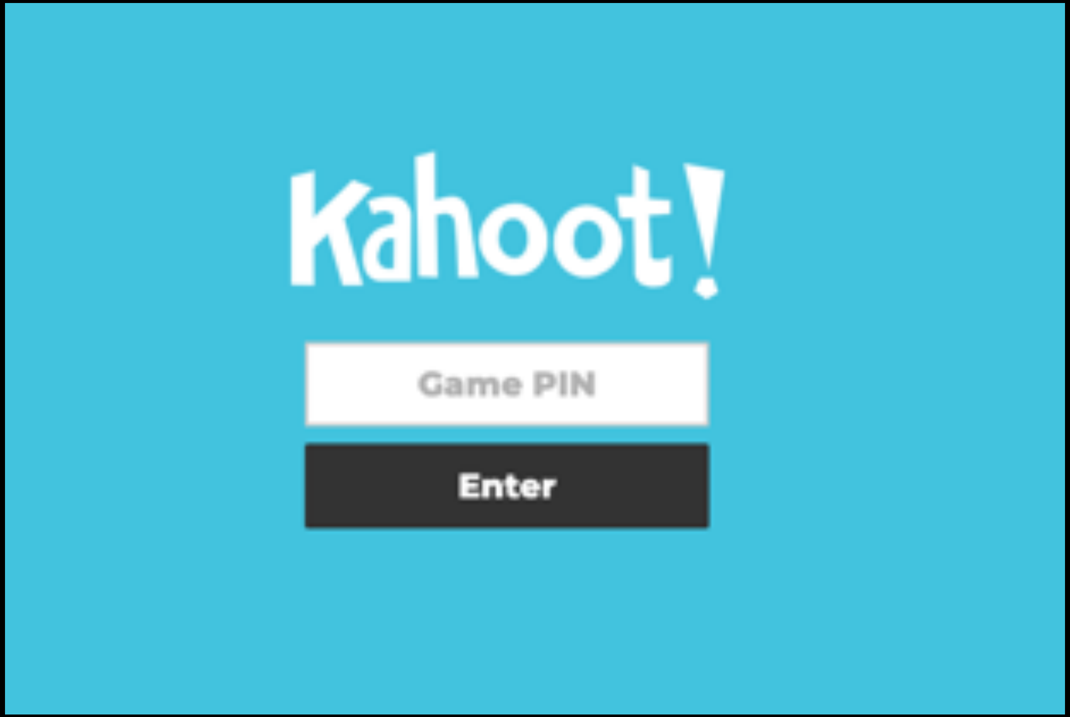
https://github.com/niklaushirt/training

Before you go...

We will collect some feedback.

Please make sure you can access https://kahoot.it/
 either on your PC of Phone.

You will get the Game PIN
later in the training.



See you next week!

- **Same place**
- **Same time**

Kubernetes Workshop Series
ISTIO

Please keep in mind that you'll need 16GB of RAM for the Labs or create a cloud instance:
https://github.com/niklaushirt/training/blob/master/standalone/istio/install-istio-standalone.md

Niklaus Hirt

✉ nikh@ch.ibm.com

@nhirt

THANK YOU!!!!