

# Kubeflow Deployment & Development

葉信和 / Hsin-Ho Yeh  
Software Engineer / CEO @ 信誠金融科技  
[hsinho.yeh@footprint-ai.com](mailto:hsinho.yeh@footprint-ai.com)

# Download Slides

<https://reurl.cc/7DV8k5>



# About me

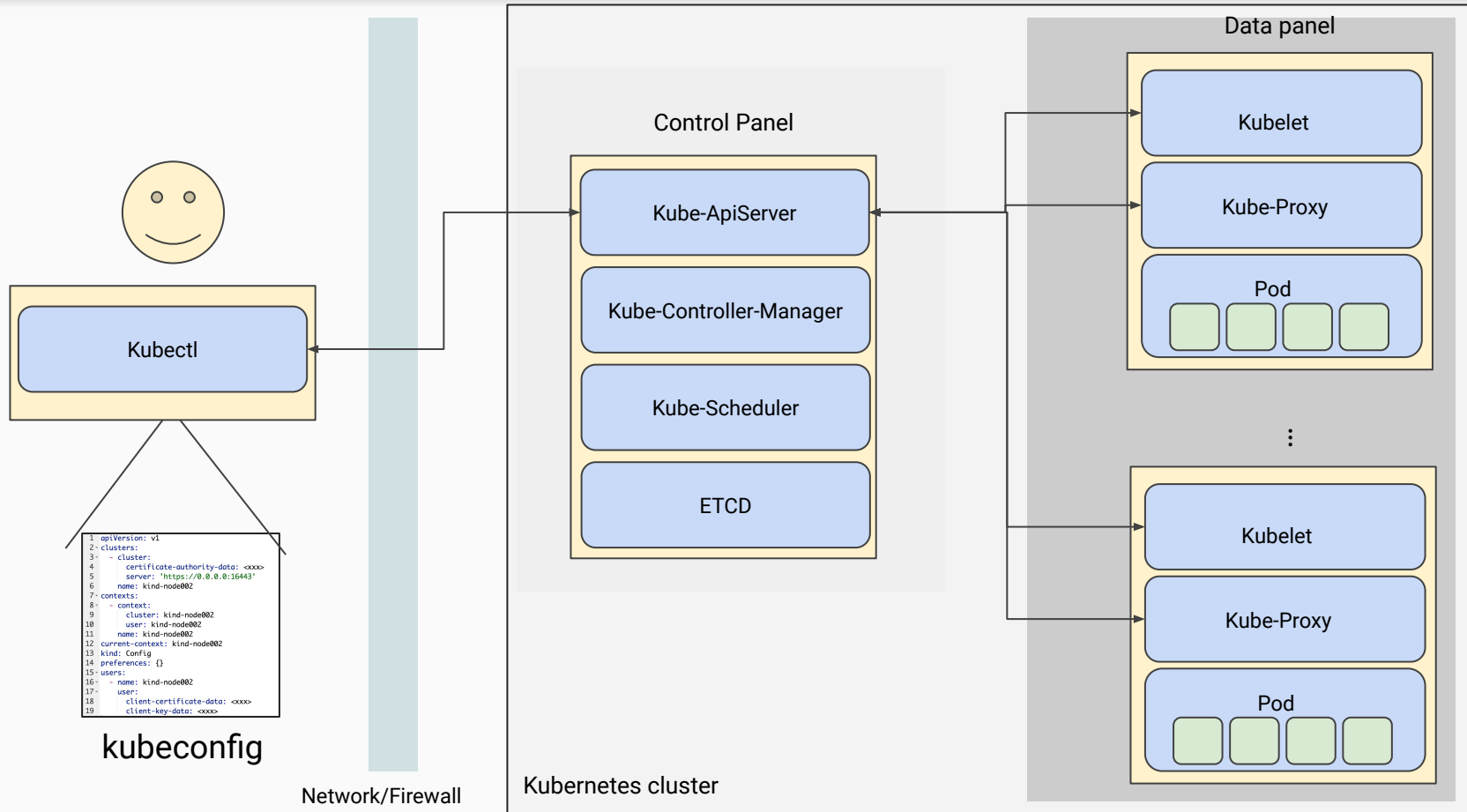
- 2020 - Present at 信誠金融科技
  - Shrimping: A data-sharing platform
    - <https://get-shrimping.footprint-ai.com>
  - Tintin: a machine learning platform for everyone
    - <https://get-tintin.footprint-ai.com>
- 2016 - 2020 at IglooInsure (16M+ in series A+ 2020)
  - Provide digital insurance for e-economic world
  - Funded in KUL, Headquartered in Singapore
  - First employee/ Engineering Lead / Regional Head/ Chief Engineer
- 2013 - 2016 at Studio Engineering @ hTC
  - Principal Engineer on Cloud Infrastructure Team
- 2009 - 2012 at IIS @ Academia Sinica
  - Computer vision, pattern recognition, and data mining
- CS@CCU, CS@NCKU alumni



# Agenda

- Kubernetes & Kubeflow Deployment Tool
- Multikf: a simplified version to deploying kubeflow(s) on a host machine
- Development hints
- Q&A

# What is Kubectl? (1/2)



## What is Kubectl? (2/2)

// when you key in this command on console

```
kubectl get pods
```

// it will be translated into

```
kubectl --kubeconfig=~/.kube/config get pods
```

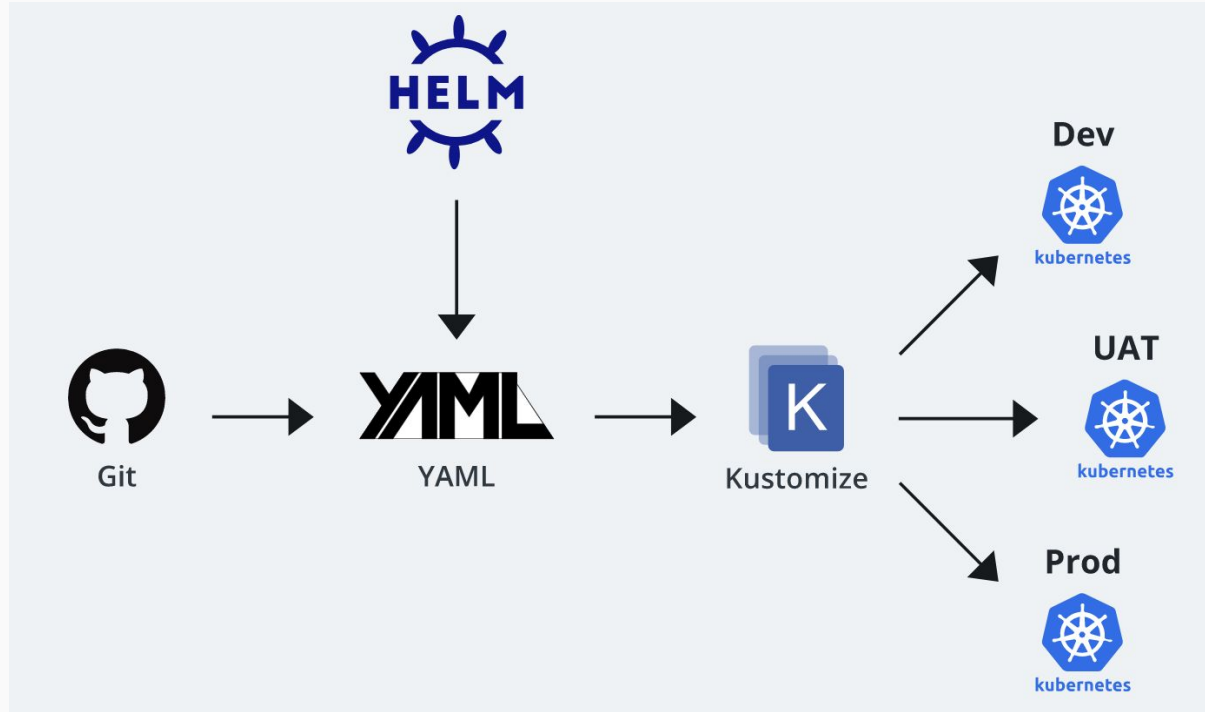
// so it is totally okay to talk to any kube-control panel by specifying kube config path, only if you know how to talk with them

```
1  apiVersion: v1
2  clusters:
3  - cluster:
4      certificate-authority-data: <xxx>
5      server: 'https://0.0.0.0:16443'
6      name: kind-node002
7  contexts:
8  - context:
9      cluster: kind-node002
10     user: kind-node002
11     name: kind-node002
12  current-context: kind-node002
13  kind: Config
14  preferences: {}
15  users:
16  - name: kind-node002
17    user:
18      client-certificate-data: <xxx>
19      client-key-data: <xxx>
```

# Declarative Management with Kubectl

```
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: static-html-deployment
5    namespace: demo1
6    labels:
7      app: http-service
8      version: v1
9  spec:
10   replicas: 1
11   selector:
12     matchLabels:
13       app: http-service
14       version: v1
15   template:
16     metadata:
17       labels:
18         app: http-service
19         version: v1
20     spec:
21       containers:
22       - name: main
23         image: footprintai/k8sworkshop:static-html-demo
24         imagePullPolicy: IfNotPresent
25       ports:
26       - containerPort: 80
```

## Kustomize: create a overwrite layer on existing resources (1/2)





An environment configuration =  
**base** configuration + environment **specific** configuration

base: **kustomization** + **resources**

### kustomization.yaml

```
commonLabels:  
  app: myWord  
resources:  
- deployment.yaml  
- service.yaml  
configMapGenerator:  
- name: wordpress-map  
  files:  
  - env.startup.txt
```

### deployment.yaml

```
apiVersion: v1  
kind: Deployment  
metadata:  
  name: wordpress  
  labels:  
    app: wordpress  
spec:  
  replicas: 1  
  selector:  
    matchLabels:  
      app: wordpress  
  template: ...
```

### service.yaml

```
apiVersion: v1  
kind: Service  
metadata:  
  name: wordpress  
spec:  
  ports:  
    - port: 389  
  selector:  
    app: wordpress
```

# Kubeflow Manifests

```
// during kubeflow installation:
```

```
https://github.com/FootprintAI/kubeflow-workshop/blob/main/install/kubeflow.v14.sh
```

```
wget https://github.com/kubeflow/manifests/archive/refs/tags/v1.4.0.tar.gz && \
```

```
tar -xzf v1.4.0.tar.gz && \
```

```
cd manifests-1.4.0
```

```
while ! kustomize build example | kubectl apply -f -; do echo "Retrying to apply resources"; sleep 10; done
```

```
// and also you could run :
```

```
kustomize build example > manifest-v1.4.yaml // which generate CRDs that kubeflow is needed for installation
```

```
// https://github.com/FootprintAI/multikf/blob/main/kfmanifests/kubeflow-manifest-v1.4.1.yaml
```

# Few drawbacks of the current installation process

- Requires knowledge of command line
  - Customize version is tied into v3.2 [1]
  - Heavily rely on kubectl for troubleshooting.
- Hard to Resource customization
  - Default requirements for system itself costs 60 Gbytes Disk, 16G Memory, and at least 4 Cores CPU.
  - Resources can be further reduced for different purposes, like personal usage vs production environment with thousands RPS.
- Resource sharing from powerful PC
  - Thin-Client vs Thick-Client
  - How to utilize resources from powerful PC?



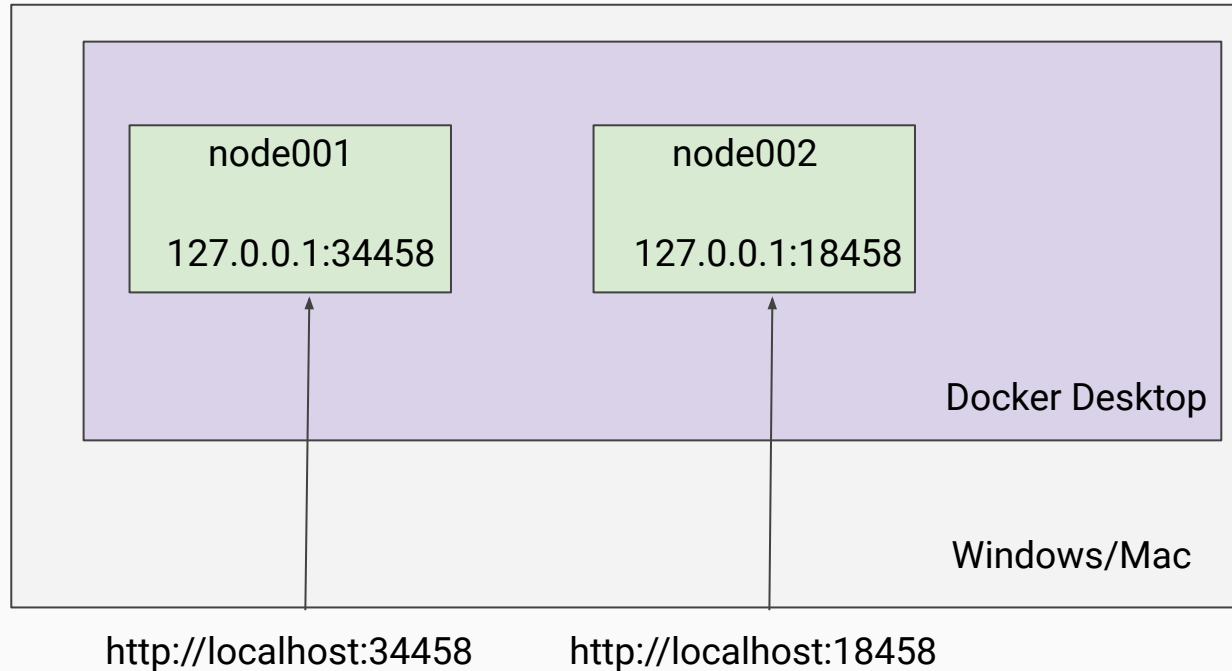
[1] <https://github.com/kubeflow/manifests#prerequisites>  
<https://www.javatpoint.com/thin-client-vs-thick-client>

multikf(馬蒂庫夫)

Our open-source project for one-clicked running multiple Kubeflow instances within the single host machine.

# multikf: One-click Installation

- Multikf: <https://github.com/footprintai/multikf>



## multikf: One-click Installation

```
// Prerequisite: install dockerd (windows)
https://github.com/FootprintAI/kubeflow-workshop/blob/main/install/windows/dockerd.bat.md

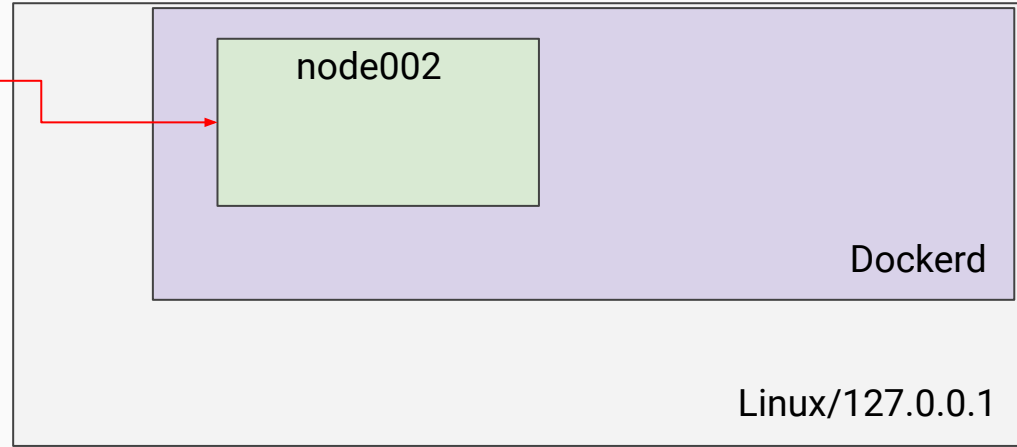
// for windows user
// install multikf
wget https://github.com/FootprintAI/multikf/raw/main/build/multikf.windows.exe
chmod +x multikf.windows.exe

// add an instances
./multikf.windows.exe add node002

// connect kubeflow
./multikf.windows.exe connect kubeflow node002
>> now you can open http://localhost:38541
```

## Multikf: demo (1/3)

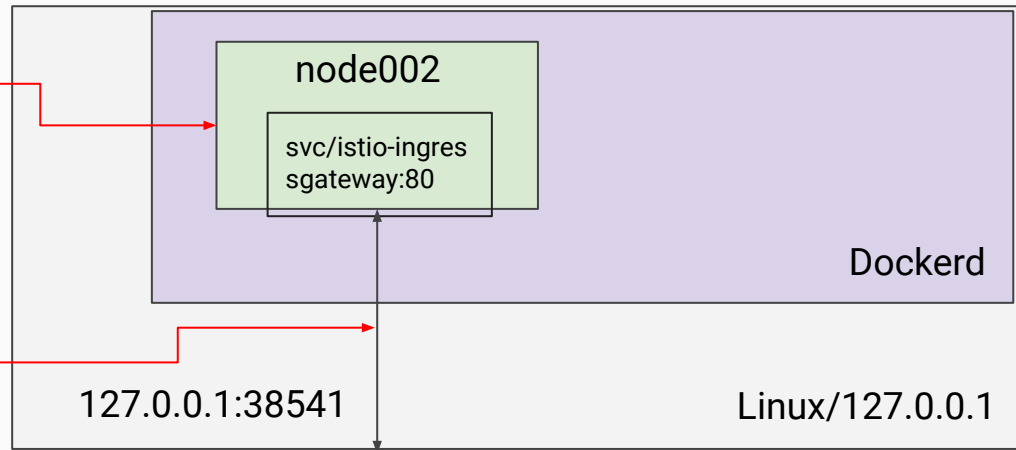
1. multikf.linux add node002



## Multikf: demo (2/3)

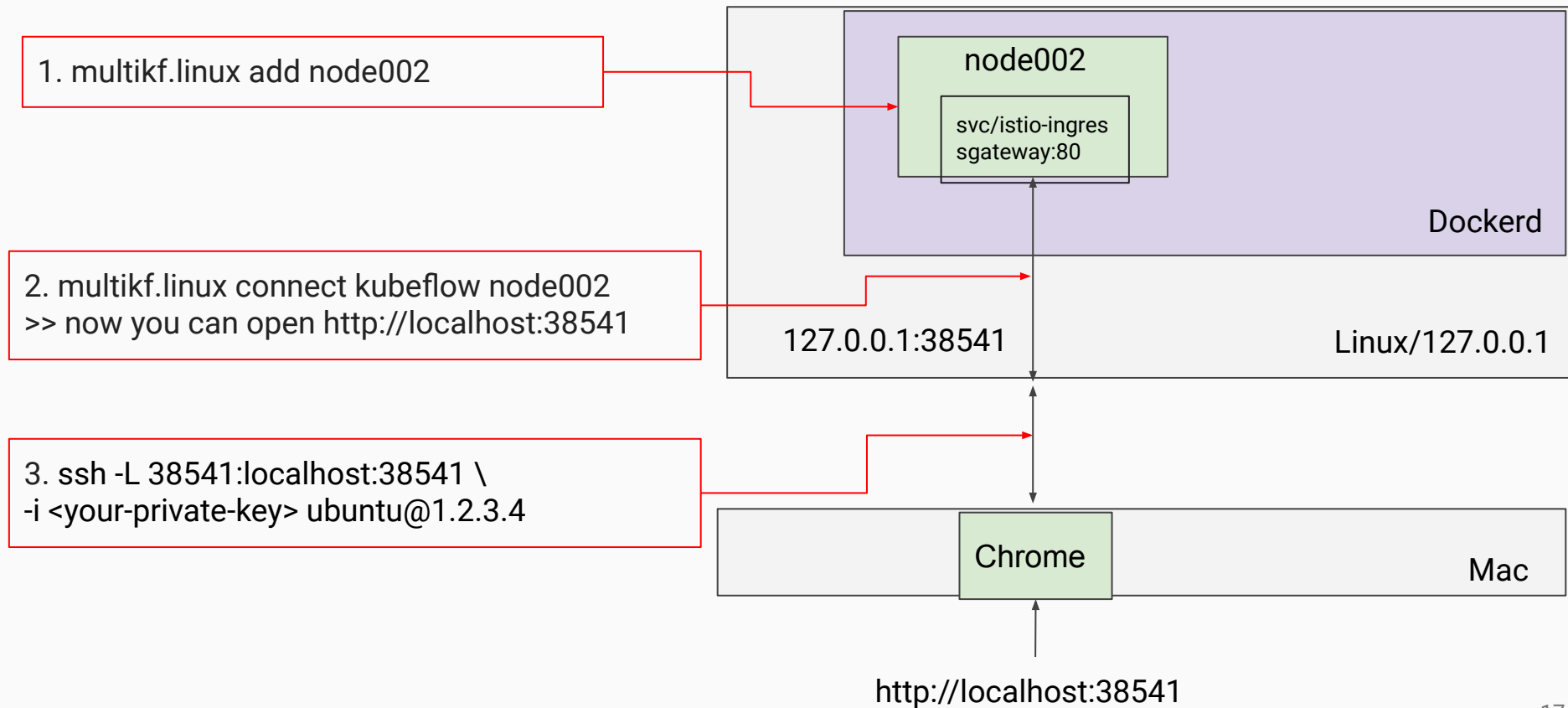
1. multikf.linux add node002

2. multikf.linux connect kubeflow node002  
>> now you can open <http://localhost:38541>



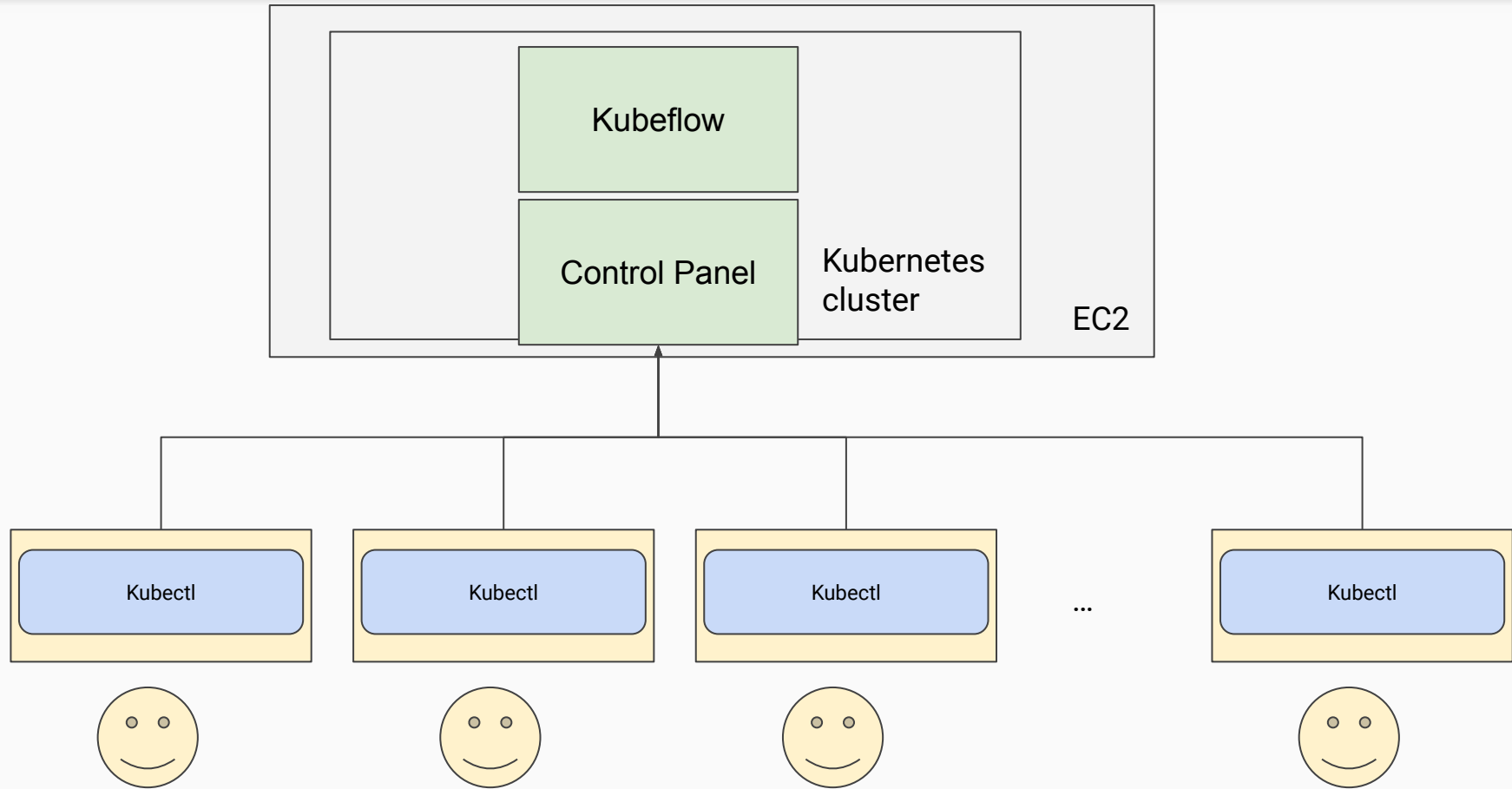


## Multikf: demo (3/3)



- Minimize knowledge on installation and debugging
  - All the lifting is picked up by the tooling.
  - Saved more times and easily for troubleshooting.
- Customizable resources used
  - Allows customization on resources used per environment per project with kustomize.
- Thin-client model and leave the loads on power PC

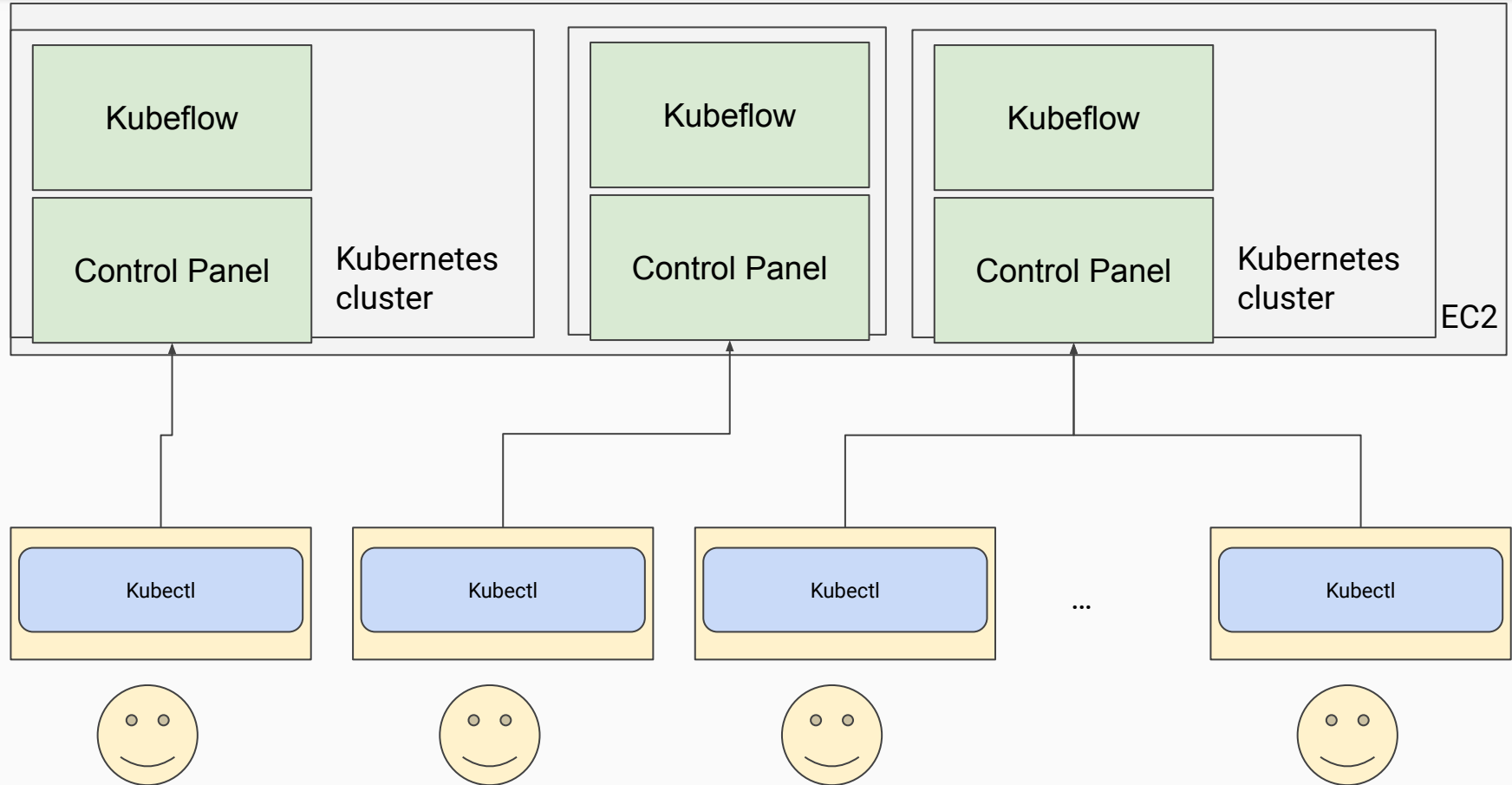
## Development Hints (1/4)



### Centralized Development Model:

- Cons:
  - Longer debugging cycle
    - containerized -> transferred to registry -> downloaded and deploy to k8s cluster
  - Broken upstream
    - Someone updated a new version on dev environment and just got crashed.
  - Confliction between users
    - User A deleted dummy resources that he/she think it might not use anymore.
    - User B relies on that dummy resources for debugging...

## Development Hints (3/4)

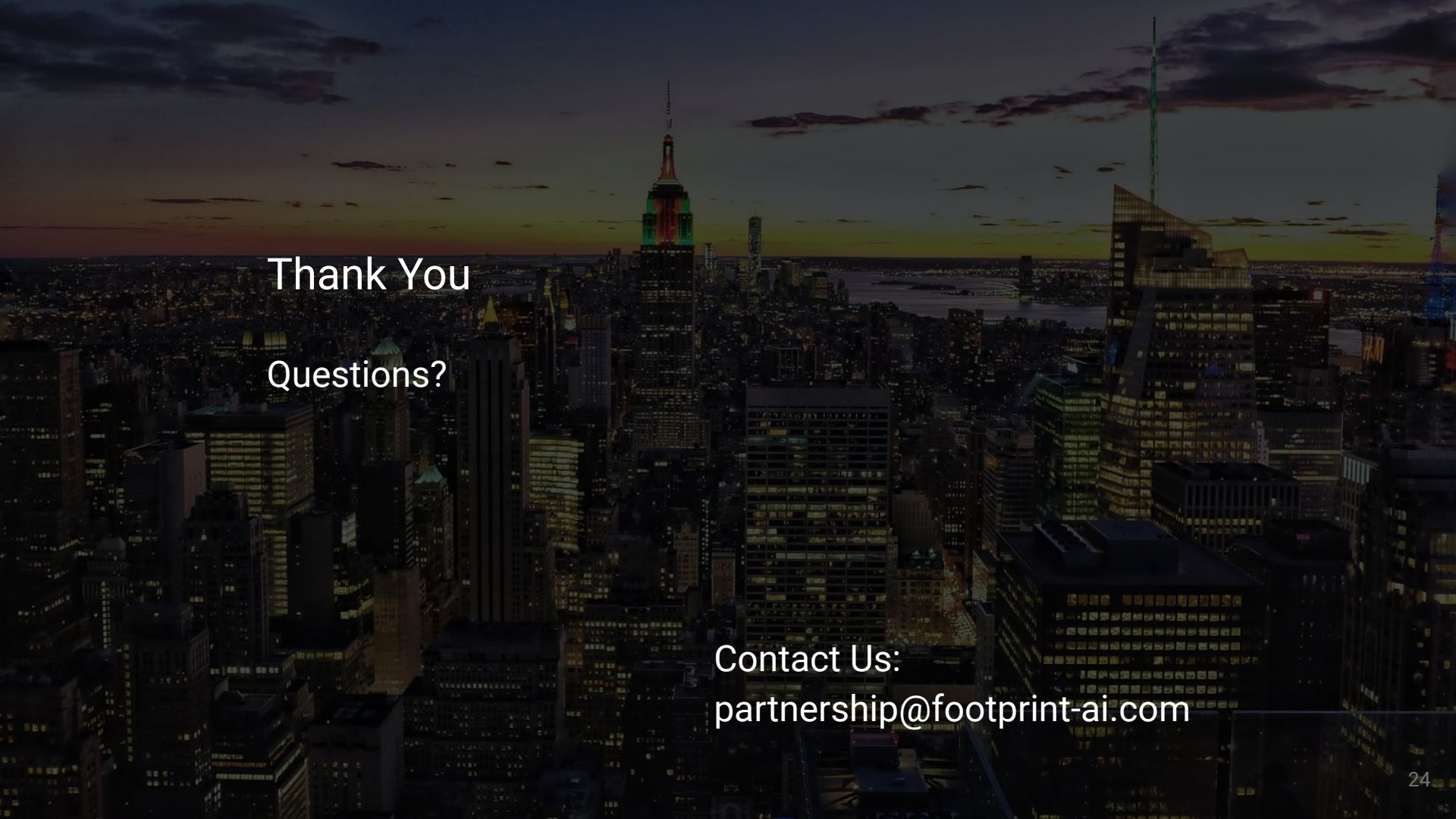


## De-Centralized Development Model:

- One click deployment with multikf.
- Customized configuration with kustomize.
- Cons:
  - Longer debugging cycle
    - Require port-forward from your machine to the destination service
  - ~~Broken upstream~~
  - ~~Conflict between users~~

```
from [::1]:17443 -> 443
from 127.0.0.1:5432 -> 5432
from [::1]:5432 -> 5432
16:43:02 portforward.pool: create a new conn
16:43:02 new forwarder conn, req:grandturks/minio-service/[19001:9001 9000:9000]
16:43:03 portforward: haven't start yet, forwarding now
from 127.0.0.1:19001 -> 9001
from [::1]:19001 -> 9001
from 127.0.0.1:9000 -> 9000
from [::1]:9000 -> 9000
16:43:03 portforward.pool: create a new conn
16:43:03 new forwarder conn, req:grandturks/influxdb-service/[8086:8086]
16:43:03 portforward: haven't start yet, forwarding now
from 127.0.0.1:8086 -> 8086
from [::1]:8086 -> 8086
16:43:04 portforward.pool: create a new conn
16:43:04 new forwarder conn, req:grandturks/authentication-service/[50090:50090]
16:43:04 portforward: haven't start yet, forwarding now
from 127.0.0.1:50090 -> 50090
from [::1]:50090 -> 50090
16:43:05 portforward.pool: create a new conn
16:43:05 new forwarder conn, req:grandturks/authz-service/[50100:50100]
16:43:05 portforward: haven't start yet, forwarding now
from 127.0.0.1:50100 -> 50100
from [::1]:50100 -> 50100
16:43:05 portforward.pool: create a new conn
16:43:05 new forwarder conn, req:grandturks/taskworker-service/[50120:50120]
16:43:06 portforward: haven't start yet, forwarding now
from 127.0.0.1:50120 -> 50120
from [::1]:50120 -> 50120
16:43:06 portforward.pool: create a new conn
16:43:06 new forwarder conn, req:kubeflow/ml-pipeline/[8888:8888 8887:8887]
16:43:07 portforward: haven't start yet, forwarding now
from 127.0.0.1:8888 -> 8888
from [::1]:8888 -> 8888
from 127.0.0.1:8887 -> 8887
from [::1]:8887 -> 8887
```


- Introduced deployment tools including kubectl and kustomize
- multikf provides several benefits for using and deploying kubeflow with easily, simplified, and time-saving process.
- Some features are in the roadmap:
  - Client-server architecture: grants you ability to remote deploy clusters
  - Web-service: provides SaaS solution to customers.
  -

An aerial photograph of the New York City skyline at dusk. The sky is a mix of dark purple, blue, and orange. The city is densely packed with skyscrapers, many of which are illuminated with their interior lights. The Empire State Building is prominent in the center, with its top lit in red and green. The Hudson River is visible on the right side of the image.

Thank You  
Questions?

Contact Us:  
[partnership@footprint-ai.com](mailto:partnership@footprint-ai.com)





***“The Best Engineers  
Are Lazy”***

-Ancient Engineering Proverb

- Slides:
  - <https://github.com/FootprintAI/talks/tree/main/slides>
- Multikf
  - <https://github.com/FootprintAI/multikf>
- Kubeflow Workshop
  - <https://github.com/footprintai/kubeflow-workshop>