



## INTERNATIONAL SUMMER SCHOOL ON INDUSTRIAL AGENTS 2024

STANDARDIZATION OF I4.0 SYSTEMS

# Microservice-based Industrial Agents: Managing manufacturing applications through the Edge-Fog-Cloud continuum

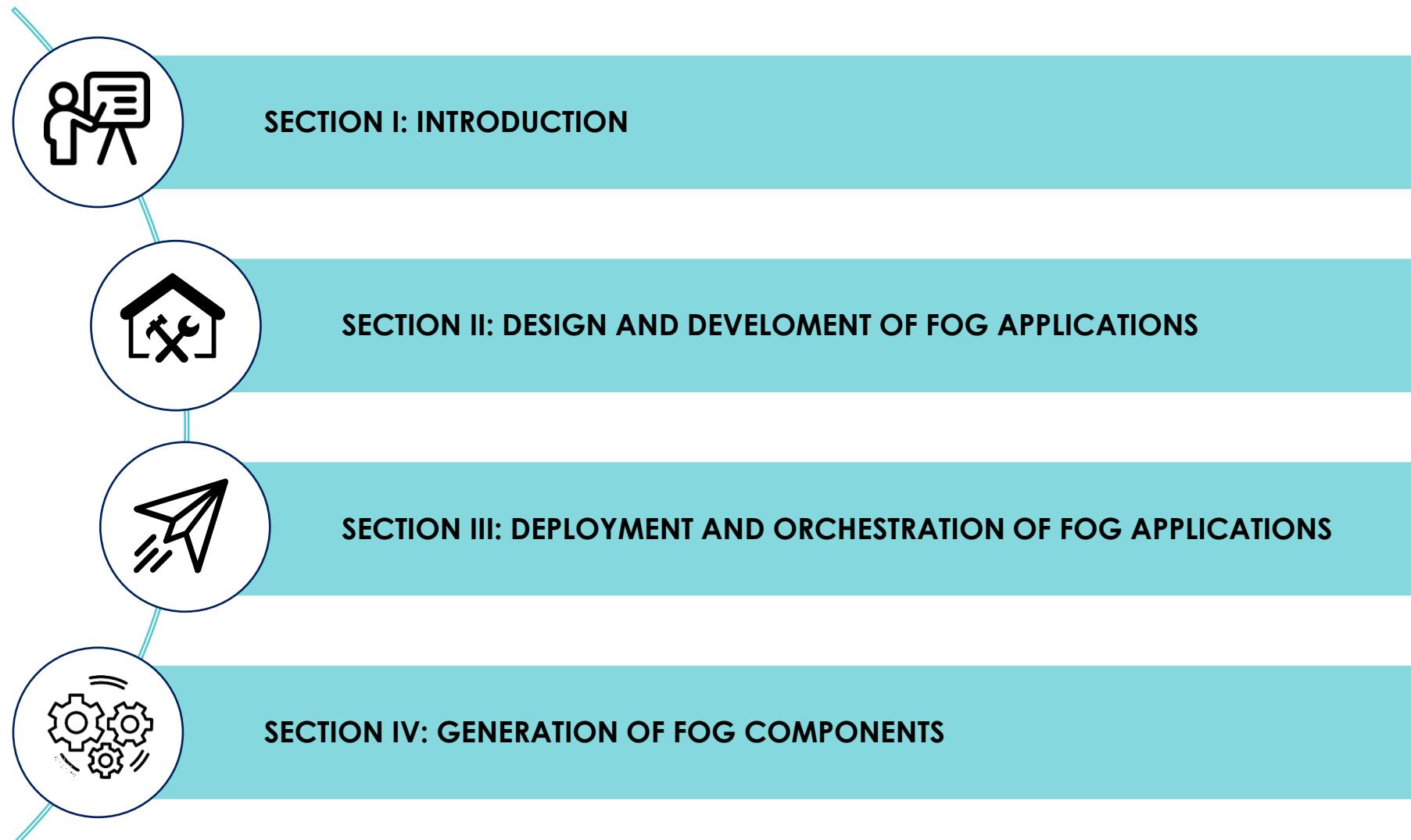
Aintzane Armentia, Oskar Casquero, Julen Cuadra, Ekaitz Hurtado

Bilbao, 28<sup>th</sup> June 2024

This work is licensed  
under [CC BY 4.0](#).



# TABLE OF CONTENTS



# INTRODUCTION

## TABLE OF CONTENTS

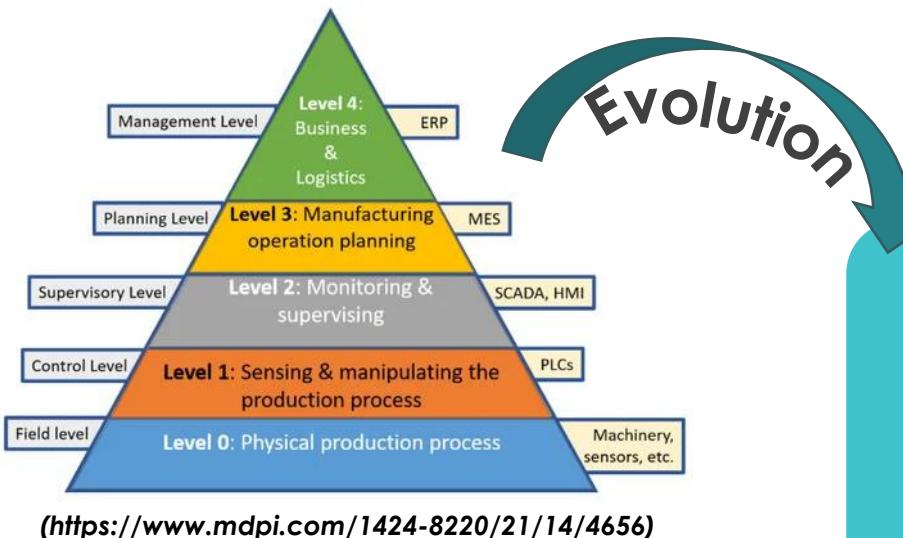
### SECTION I



- Motivation
- Technologies for Digital Transformation
  - Virtualization technologies
  - Orchestration technologies
- Getting Started
  - Kubernetes cluster
  - Kubernetes dashboard

# INTRODUCTION

## MOTIVATION



### Cloud

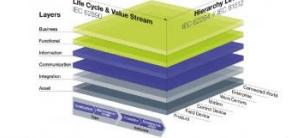
- Execution of business logic.
- Peaks of processing demand.



### Fog

- Improvement of the production process.
- Balance between computation, network latency, and data security.

RAMI4.0



### Plant and Edge

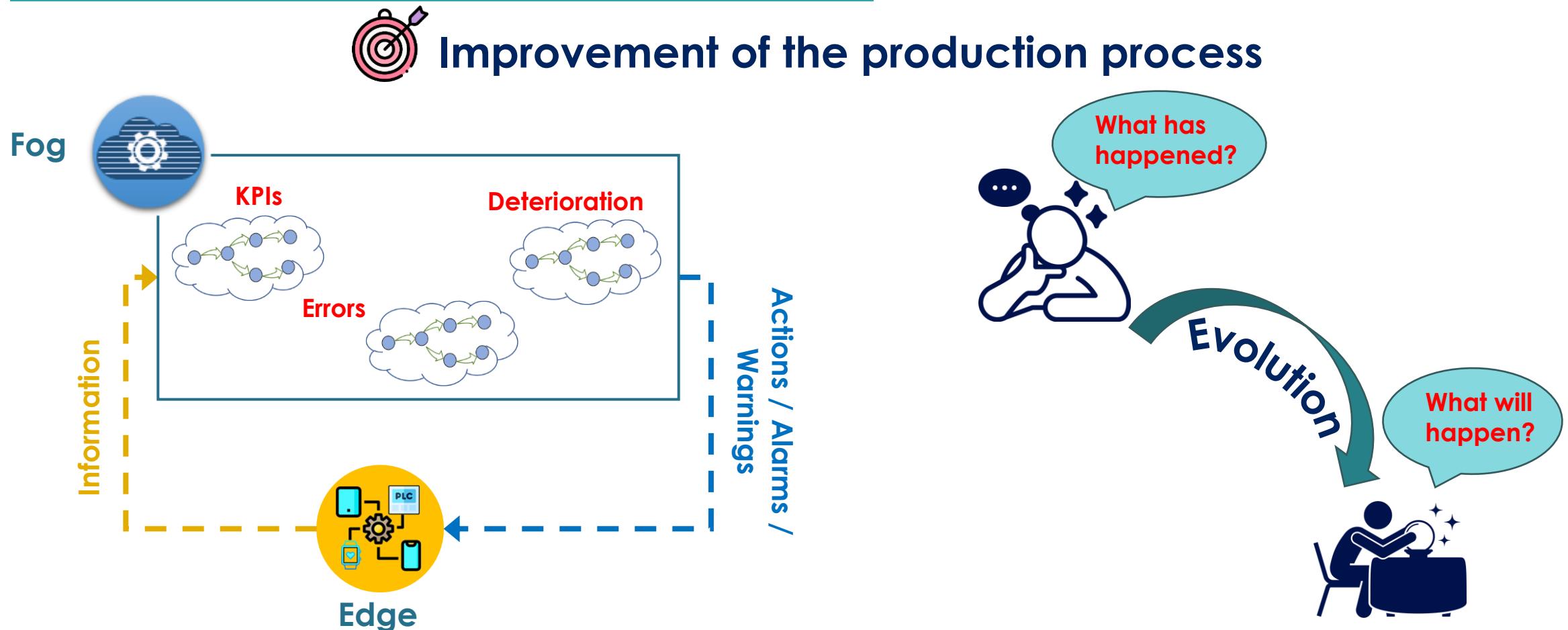
- Control of physical assets.
- Thin data processing.

EDGE/FOG/CLOUD CONTINUUM

# INTRODUCTION

## MOTIVATION

### DATA ANALYTICS APPLICATIONS



# INTRODUCTION

## MOTIVATION

### DATA ANALYTICS APPLICATIONS



#### Improvement of the production process

Cloud



Trend



Fog



IEEE Std 1934-2018  
IIC, nov. 2022, «The  
Industrial Internet  
Reference Architecture»



Design



Microservices



Containers



docker



Application

Development

Deployment

Operation



Orchestrators



kubernetes

# INTRODUCTION

## TABLE OF CONTENTS

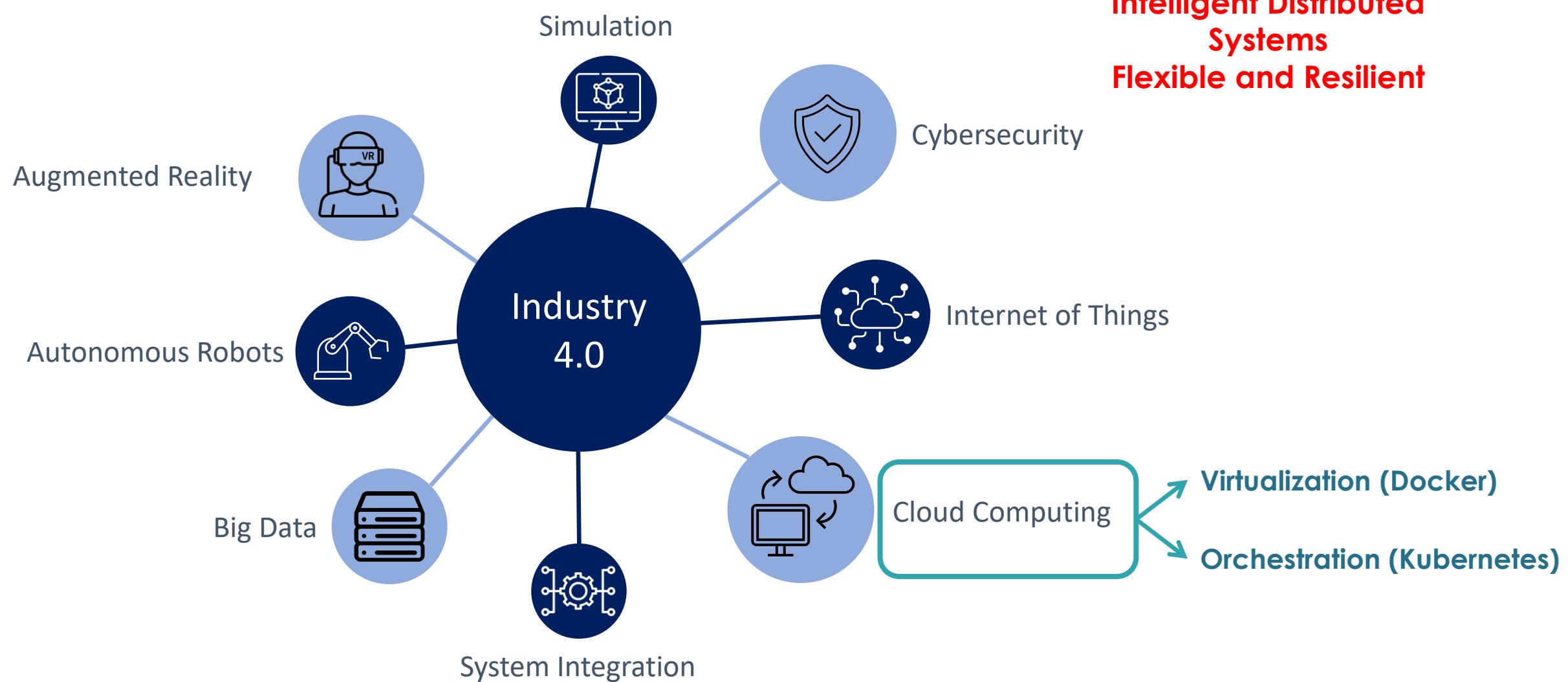
### SECTION I



- Motivation
- Technologies for Digital Transformation
  - Virtualization technologies
  - Orchestration technologies
- Getting Started
  - Kubernetes cluster
  - Kubernetes dashboard

# INTRODUCTION

## TECHNOLOGIES FOR DIGITAL TRANSFORMATION



# INTRODUCTION

## TECHNOLOGIES FOR DIGITAL TRANSFORMATION

### VIRTUALIZATION TECHNOLOGIES



Why containers?

- 1 Start the Virtual Machine
- 2 Open the “issia/Section1” directory
- 3 Open the “my\_program.py” file
- 4 Open a Terminator terminal



```
import psutil
import time

def cpu_ram():
    while True:
        cpu = psutil.cpu_percent(interval=5)
        ram = psutil.virtual_memory().percent
        print("CPU: % " + str(cpu) + "\tRAM: % " + str(ram))

if __name__ == "__main__":
    cpu_ram()
```

# INTRODUCTION

## TECHNOLOGIES FOR DIGITAL TRANSFORMATION

### VIRTUALIZATION TECHNOLOGIES



#### Why containers?

#### 5 Execute "my\_program.py"

**python3 my\_program.py**

```
issia@cluster:~/SectionI$ python3 my_program.py
Traceback (most recent call last):
  File "my_program.py", line 1, in <module>
    import psutil
ModuleNotFoundError: No module named 'psutil'
```

#### 6 Install "psutil" library

**pip install psutil**

```
issia@cluster:~/SectionI$ pip install psutil
bash: /usr/bin/pip: No such file or directory
issia@cluster:~/SectionI$
```

#### 7 Install pip package

**sudo apt install python3-pip**

```
issia@cluster:~/SectionI$ sudo apt install python3-pip
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  gir1.2-keybinder-3.0 libkeybinder-3.0-0 python3-configobj
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  python3-pip
0 upgraded, 1 newly installed, 0 to remove and 5 not upgraded.
Need to get 231 kB of archives.
After this operation, 1.050 kB of additional disk space will be used.
Get:1 http://es.archive.ubuntu.com/ubuntu focal-updates/universe amd64 python3-pip all 20.0.2-5ubuntu1.10 [231 kB]
Fetched 231 kB in 0s (481 kB/s)
Selecting previously unselected package python3-pip.
(Reading database ... 165108 files and directories currently installed.)
Preparing to unpack .../python3-pip_20.0.2-5ubuntu1.10_all.deb ...
Unpacking python3-pip (20.0.2-5ubuntu1.10) ...
Setting up python3-pip (20.0.2-5ubuntu1.10) ...
Processing triggers for man-db (2.9.1-1) ...
```

# INTRODUCTION

## TECHNOLOGIES FOR DIGITAL TRANSFORMATION

### VIRTUALIZATION TECHNOLOGIES



Why containers?

- 8 Install “psutil” library

```
pip install psutil
```

```
issia@cluster:~$ pip install psutil
Collecting psutil
  Using cached psutil-5.9.8-cp36-abi3-many
  4.manylinux2014_x86_64.whl (288 kB)
Installing collected packages: psutil
Successfully installed psutil-5.9.8
```

- 9 Execute “my\_program.py”

```
python3 my_program.py
```

```
issia@cluster:~$ python3 my_program.py
CPU: %1.2      RAM: %79.2
CPU: %1.4      RAM: %79.2
CPU: %1.3      RAM: %79.2
CPU: %1.6      RAM: %79.3
CPU: %2.2      RAM: %79.3
CPU: %1.4      RAM: %79.3
CPU: %1.4      RAM: %79.3
CPU: %1.3      RAM: %79.3
CPU: %1.1      RAM: %79.3
CPU: %1.7      RAM: %79.3
```



Ctrl+C to exit the program

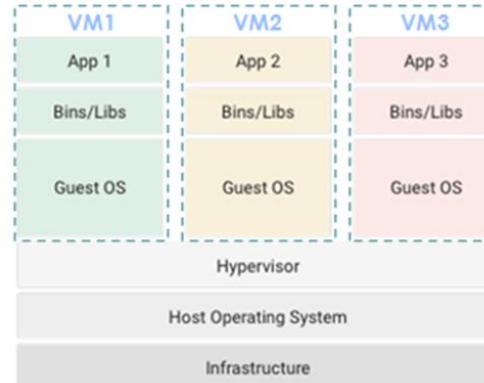
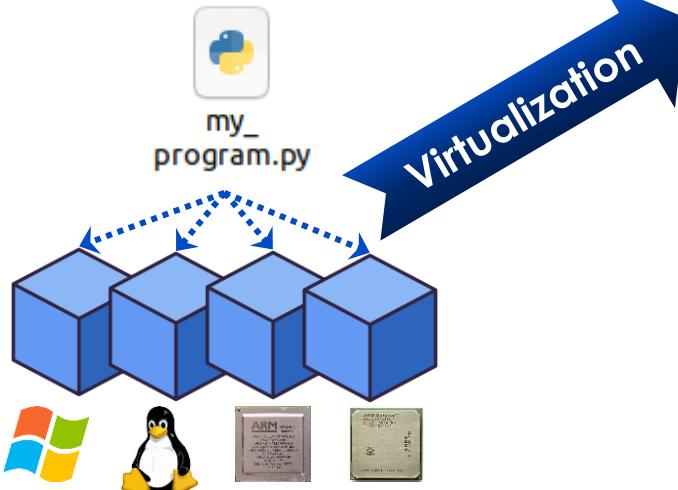
# INTRODUCTION

## TECHNOLOGIES FOR DIGITAL TRANSFORMATION

### VIRTUALIZATION TECHNOLOGIES

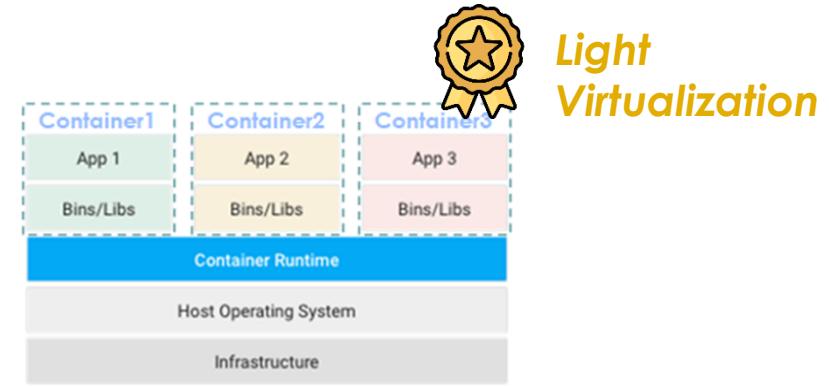


Why containers?



### Virtual Machines

- Virtualize an entire machine down to the HW layers.
- Each VM includes a complete copy of an OS, the application, the necessary binaries and libraries.



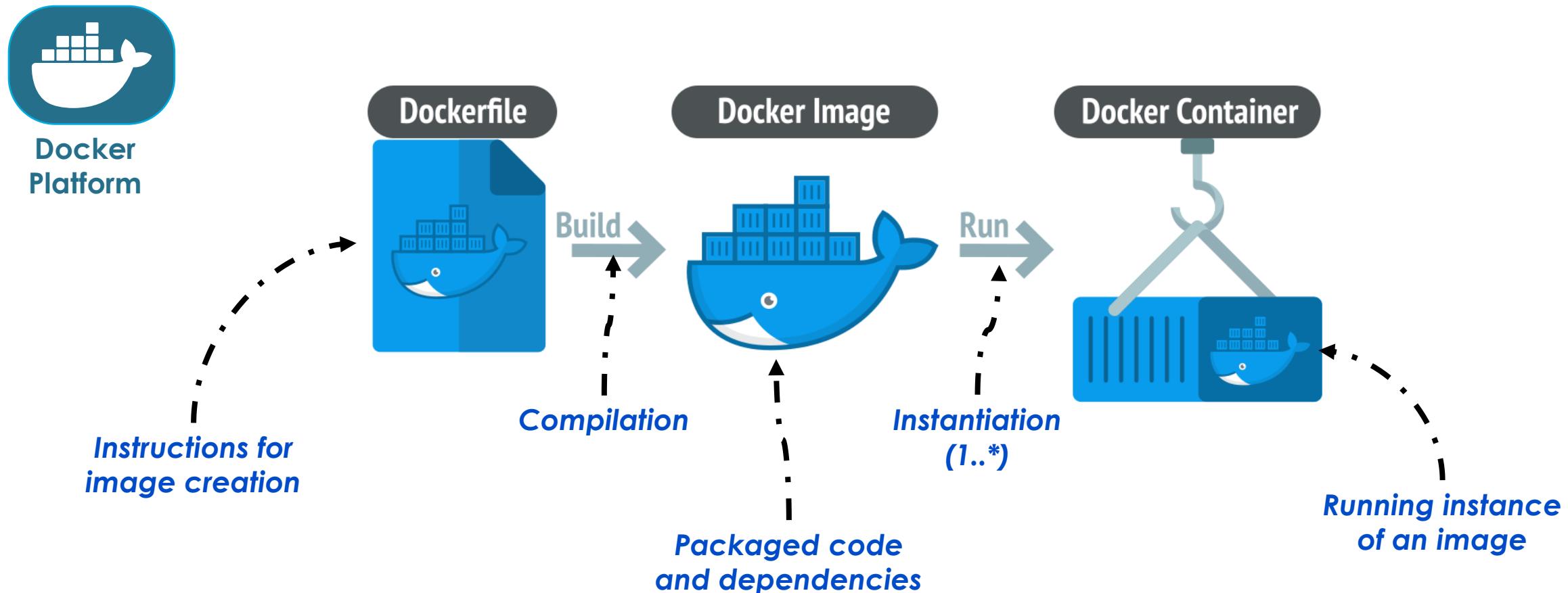
### Containers

- Virtualize the software layers above the OS level.
- Each container includes the application, the necessary binaries and libraries.
- Several containers can share the OS's kernel, running as isolated processes in user space.

# INTRODUCTION

## TECHNOLOGIES FOR DIGITAL TRANSFORMATION

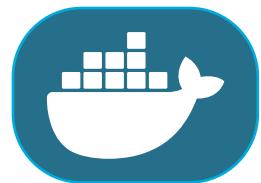
### VIRTUALIZATION TECHNOLOGIES



# INTRODUCTION

## TECHNOLOGIES FOR DIGITAL TRANSFORMATION

### VIRTUALIZATION TECHNOLOGIES



Let's run a  
Docker image

#### Program Dependencies

my\_program.py

psutil

python

Operating system

Operating System

“pip” package

“psutil” library

Program Code

Application Startup

#### Dockerfile

FROM ubuntu:20.04

RUN apt update && apt install -y python3-pip

RUN pip3 install psutil

COPY my\_program.py /

CMD ["python3", "-u", "my\_program.py"]

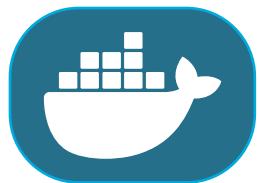


The Dockerfile is already  
available in the VM

# INTRODUCTION

## TECHNOLOGIES FOR DIGITAL TRANSFORMATION

### VIRTUALIZATION TECHNOLOGIES



Let's run a Docker image

#### 1 Uninstall “psutil” library

**pip uninstall psutil**

```
issia@cluster:~$ pip uninstall psutil
Found existing installation: psutil 5.9.8
Uninstalling psutil-5.9.8:
Would remove:
/home/issia/.local/lib/python3.8/site-packages/psutil-5.9.8.dist-info/*
/home/issia/.local/lib/python3.8/site-packages/psutil/*
Proceed (y/n)? y
Successfully uninstalled psutil-5.9.8
```

#### 2 Build a docker image

**sudo docker build -t my-app .**

Context directory

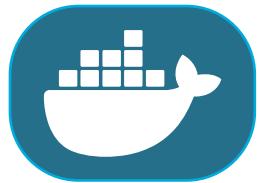
Image name

```
issia@cluster:~$ sudo docker build -t my-app .
[+] Building 26.3s (10/10) FINISHED                                            docker:default
=> [internal] load build definition from Dockerfile                         0.0s
=> => transferring dockerfile: 187B                                         0.0s
=> [internal] load metadata for docker.io/library/ubuntu:20.04             1.9s
=> [auth] library/ubuntu:pull token for registry-1.docker.io               0.0s
=> [internal] load .dockerignore                                           0.0s
=> => transferring context: 2B                                           0.0s
=> [1/4] FROM docker.io/library/ubuntu:20.04@sha256:0b897358ff6624825fb   2.0s
=> => resolve docker.io/library/ubuntu:20.04@sha256:0b897358ff6624825fb   0.0s
=> => sha256:d86db849e59626d94f768c679aba441163c996caf7a342 424B / 424B  0.0s
=> => sha256:f5250218d28ad6612bf653eced407165dd6475a4d 2.30kB / 2.30kB  0.0s
=> => sha256:9ea8908f47652b59b8055316d9c0e16b365e2b5c 27.51MB / 27.51MB 1.2s
=> => sha256:0b897358ff6624825fb50d20ffb605ab0eaea77ced 1.13kB / 1.13kB  0.0s
=> => extracting sha256:9ea8908f47652b59b8055316d9c0e16b365e2b5cee15d3e  0.7s
=> [internal] load build context                                           0.0s
=> => transferring context: 295B                                         0.0s
=> [2/4] RUN apt update && apt install -y python3-pip                  19.6s
=> [3/4] RUN pip3 install psutil                                         1.8s
=> [4/4] COPY my_program.py /                                           0.0s
=> => exporting to image                                                 0.9s
=> => exporting layers                                                 0.9s
=> => writing image sha256:db6dfdc7aa8e32a6982d0c708b58ee2d01543a31618  0.0s
=> => naming to docker.io/library/my-app                                0.0s
```

# INTRODUCTION

## TECHNOLOGIES FOR DIGITAL TRANSFORMATION

### VIRTUALIZATION TECHNOLOGIES



Let's run a  
Docker image

- 3 Check the image

```
sudo docker image ls
```

```
issia@cluster:~$ sudo docker image ls
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
my-app          latest   db6dfdca7aa8  5 minutes ago  448MB
hello-world     latest   d2c94e258dcb  13 months ago  13.3kB
```

- 4 Run a container

```
sudo docker run --name=inst1-myapp my-app
```

```
issia@cluster:~$ sudo docker run --name=inst1-myapp my-app
CPU: %2.3      RAM: %75.5
CPU: %2.3      RAM: %75.5
CPU: %2.2      RAM: %75.5
CPU: %2.9      RAM: %75.2
```

- 5 Store in DockerHub → Create an alias for the image

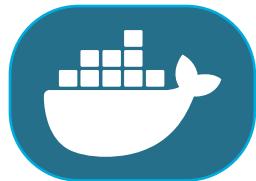
```
sudo docker tag my-app aarmentia/my-app
```

```
issia@cluster:~$ sudo docker tag my-app aarmentia/my-app
issia@cluster:~$ sudo docker image ls
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
aarmentia/my-app  latest   db6dfdca7aa8  11 minutes ago  448MB
my-app          latest   db6dfdca7aa8  11 minutes ago  448MB
hello-world     latest   d2c94e258dcb  13 months ago  13.3kB
```

# INTRODUCTION

## TECHNOLOGIES FOR DIGITAL TRANSFORMATION

### VIRTUALIZATION TECHNOLOGIES



Let's run a Docker image

- 6 Store in DockerHub → Log in
- `sudo docker login`

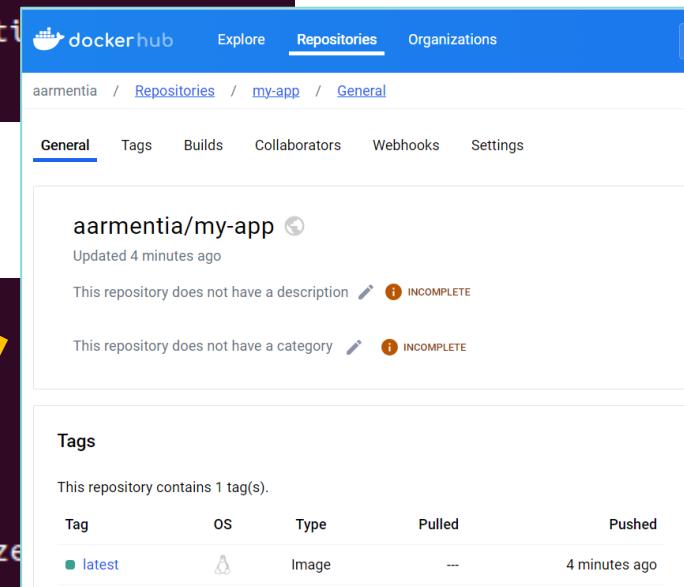
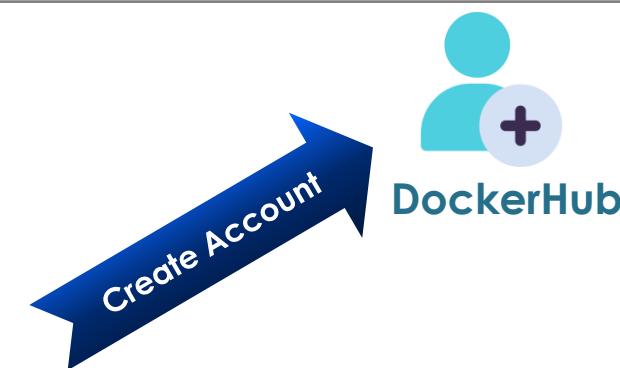
```
issia@cluster:~$ sudo docker login -u aarmentia
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

- 7 Store in DockerHub → Push the image

`sudo docker push aarmentia/my-app`

```
issia@cluster:~$ sudo docker push aarmentia/my-app
Using default tag: latest
The push refers to repository [docker.io/aarmentia/my-app]
5db5751d81c3: Pushed
5cfbb9cc2801: Pushed
d6b866735af5: Pushed
3ec3ded77c0c: Mounted from library/ubuntu
latest: digest: sha256:c9b78ebc6a225aa91768a761d813c42f38f584f4feeaab0d8b8cd75869b8e68c size: 1159
```



# INTRODUCTION

## TECHNOLOGIES FOR DIGITAL TRANSFORMATION

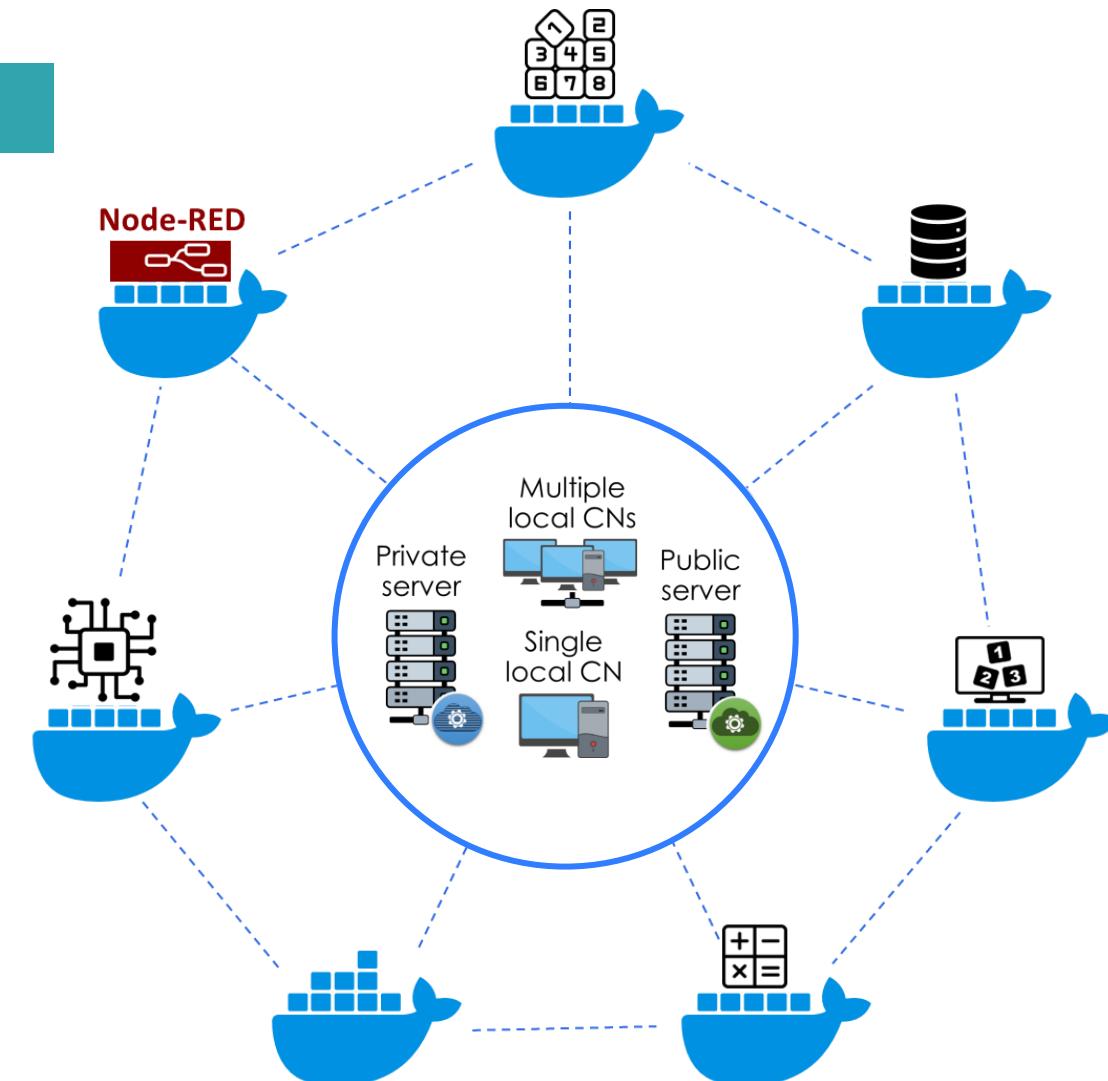
### ORCHESTRATION TECHNOLOGIES



#### Why orchestrators?

- Easy integration with container technologies.
- Ensure that the containers will run correctly on all CNs.
- Management of containers lifecycle.
- Self-management of workloads.

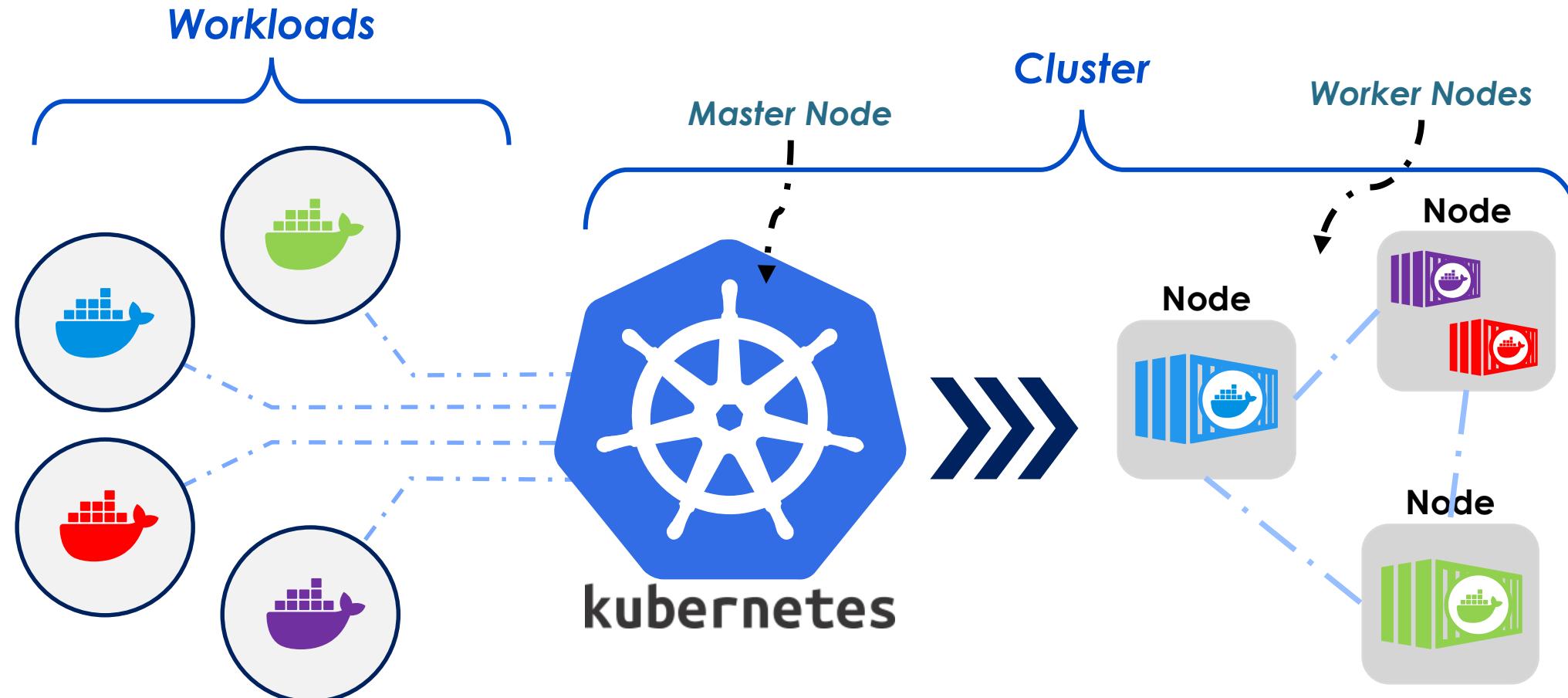
\* CN: Computing Node



# INTRODUCTION

## TECHNOLOGIES FOR DIGITAL TRANSFORMATION

### ORCHESTRATION TECHNOLOGIES

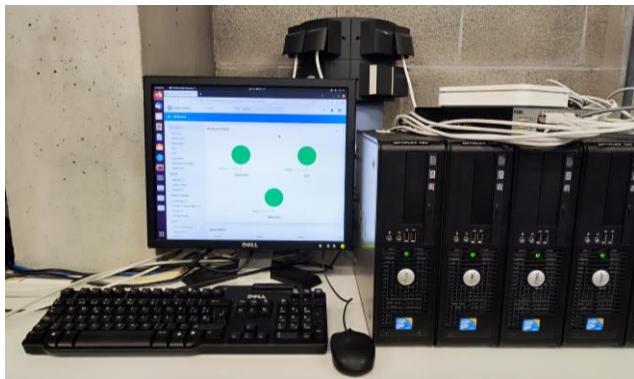


# INTRODUCTION

## TECHNOLOGIES FOR DIGITAL TRANSFORMATION

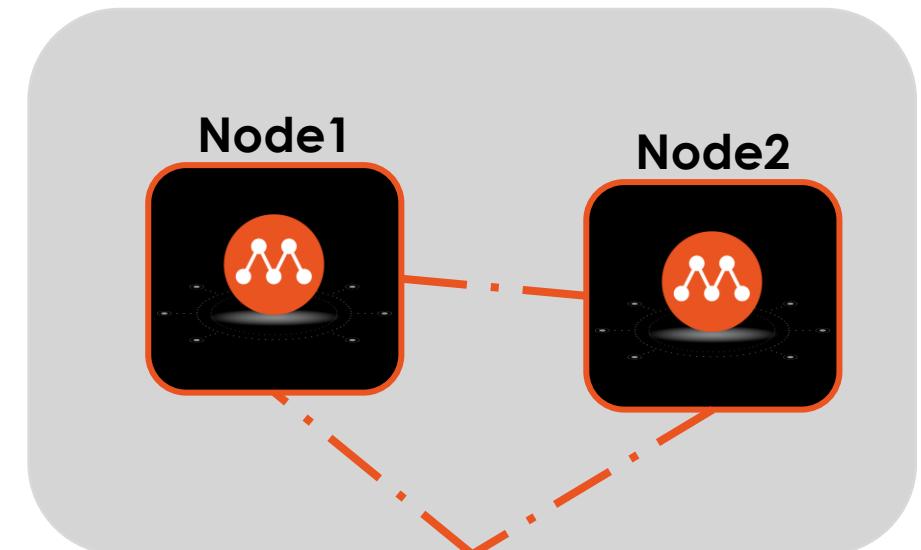
### ORCHESTRATION TECHNOLOGIES

#### Physical Cluster



#### Virtual Cluster

##### Virtual Machine



# INTRODUCTION

## TABLE OF CONTENTS

### SECTION I



- Motivation
- Technologies for Digital Transformation
  - Virtualization technologies
  - Orchestration technologies
- **Getting Started**
  - **Kubernetes cluster**
  - **Kubernetes dashboard**

# INTRODUCTION

## GETTING STARTED

### KUBERNETES CLUSTER



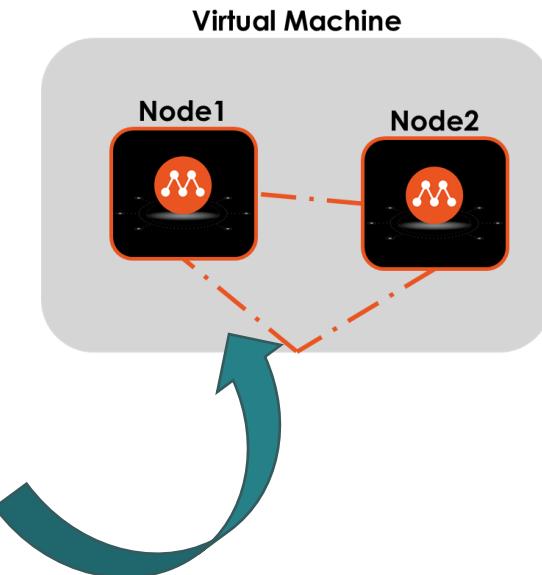
K3S

**Lightweight Kubernetes**

- 1 IP address related to the “ens33” interface

**ifconfig**

```
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.237.130 netmask 255.255.255.0 broadcast 192.168.237.255
      inet6 fe80::c6fe:427c:894b:675 prefixlen 64 scopeid 0x20<link>
        ether 00:0c:29:3b:05:83 txqueuelen 1000 (Ethernet)
          RX packets 566301 bytes 819704043 (819.7 MB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 130498 bytes 9418568 (9.4 MB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```



# INTRODUCTION

## GETTING STARTED

### KUBERNETES CLUSTER

- 2 Delete the current cluster ("Scripts" directory)

```
./delete_cluster.sh
```

```
issia@cluster:~$ ./delete_cluster.sh
deployment.apps "nodered" deleted
node "node1" deleted
node "node2" deleted
node "cluster" deleted
+ id -u
+ [ 0 -eq 0 ]
+ remove_uninstall
+ rm -f /usr/local/bin/k3s-uninstall.sh
Enter number of nodes: 2
Deleting node1...
Deleting node2...
```

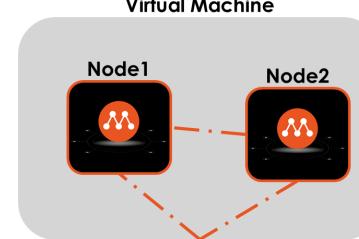
- 3 Create a new cluster ("Scripts" directory)

```
./create_cluster_v2_k3s.sh
```

```
issia@cluster:~$ ./create_cluster_v2_k3s.sh
##### INSTALLING K3S MASTER ON HOST #####
[INFO] Finding release for channel stable
[INFO] Using v1.29.5+k3s1 as release
[INFO] Downloading hash https://github.com/k3s-io/k3s/releases/download/v1.29.5+k3s1/sha256sum-amd64.txt
[INFO] Downloading binary https://github.com/k3s-io/k3s/releases/download/v1.29.5+k3s1/k3s
[INFO] Verifying binary download
[INFO] Installing k3s to /usr/local/bin/k3s
[INFO] Skipping installation of SELinux RPM
[INFO] Creating /usr/local/bin/kubectl symlink to k3s
[INFO] Creating /usr/local/bin/crictl symlink to k3s
[INFO] Skipping /usr/local/bin/ctr symlink to k3s, command exists in PATH at /usr/bin/ctr
[INFO] Creating killall script /usr/local/bin/k3s-killall.sh
[INFO] Creating uninstall script /usr/local/bin/k3s-uninstall.sh
[INFO] env: Creating environment file /etc/systemd/system/k3s.service.env
[INFO] systemd: Creating service file /etc/systemd/system/k3s.service
[INFO] systemd: Enabling k3s unit
Created symlink /etc/systemd/system/multi-user.target.wants/k3s.service → /etc/systemd/system/k3s.service.
[INFO] systemd: Starting k3s

##### CREATING NODES #####
Enter number of nodes: 2
##### CREATING node1 #####
Launched: node1

##### INSTALLING K3S AGENT ON node1 #####
'curl -sfL https://get.k3s.io | K3S_URL=https://192.168.237.130:6443 K3S_TOKEN=K10a04cc91776e0d63631e6e5a47'
```



# INTRODUCTION

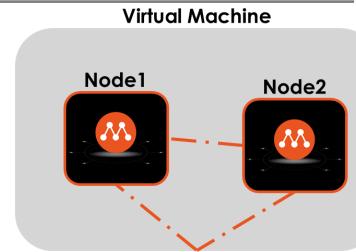
## GETTING STARTED

### KUBERNETES CLUSTER

- 3 Confirm → multipass nodes

`multipass list`

Name	State	IPv4	Image
node1	Running	10.128.11.240 10.42.1.0 10.42.1.1	Ubuntu 20.04 LTS
node2	Running	10.128.11.17 10.42.2.0 10.42.2.1	Ubuntu 20.04 LTS



- 4 Confirm → K3s cluster

`kubectl get nodes -o wide`

NAME	STATUS	ROLES	AGE	VERSION	INTERNAL-IP	EXTERNAL-IP	OS-IMAGE	KERNEL-VERSION	CONTAINER-RUNTIME
node2	Ready	<none>	41m	v1.29.5+k3s1	10.128.11.17	<none>	Ubuntu 20.04.6 LTS	5.4.0-182-generic	containerd://1.7.15-k3s1
cluster	Ready	control-plane,master	43m	v1.29.5+k3s1	192.168.237.130	<none>	Ubuntu 20.04.6 LTS	5.15.0-107-generic	containerd://1.7.15-k3s1
node1	Ready	<none>	42m	v1.29.5+k3s1	10.128.11.240	<none>	Ubuntu 20.04.6 LTS	5.4.0-182-generic	containerd://1.7.15-k3s1

# INTRODUCTION

## GETTING STARTED

### KUBERNETES DASHBOARD

- 1 Connect to the dashboard from your computer

[https://<IP\\_ADDRESS>:30443](https://<IP_ADDRESS>:30443)



#### Warning: Potential Security Risk Ahead

Firefox detected a potential security threat and did not continue to **192.168.237.130**. If you visit this site, attackers could try to steal information like your passwords, emails, or credit card details.

[Learn more...](#)

[Go Back \(Recommended\)](#)

[Advanced...](#)

192.168.237.130:30443 uses an invalid security certificate.

The certificate is not trusted because it is self-signed.

Error code: [MOZILLA\\_PKIX\\_ERROR\\_SELF\\_SIGNED\\_CERT](#)

[View Certificate](#)

[Go Back \(Recommended\)](#)

[Accept the Risk and Continue](#)

ens33: flags=4163<UP,BROADCAST  
inet 192.168.237.130

# INTRODUCTION

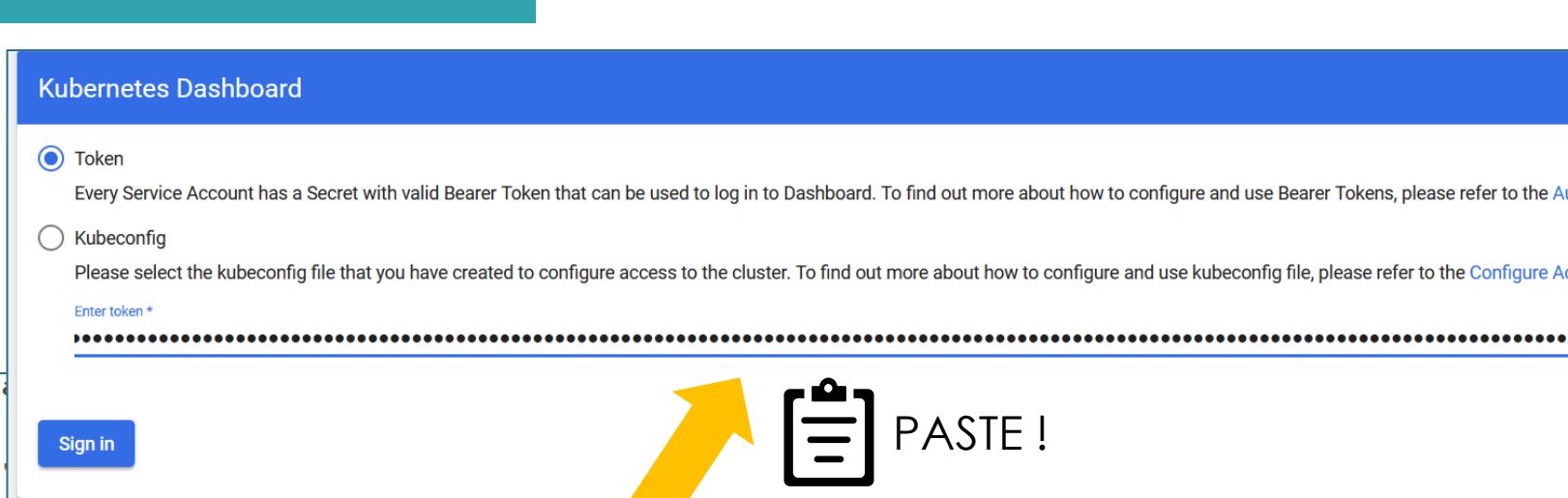
## GETTING STARTED

### KUBERNETES CLUSTER

#### 2 Insert the token



```
1 apiVersion: authentication/v1
2 kind: TokenRequest
3 metadata:
4   creationTimestamp:
5   name: admin-user
6   namespace: kubernetes-dashboard
7 spec:
8   audiences:
9     - https://kubernetes.default.svc.cluster.local
10    - k3s
11   boundObjectRef: null
12   expirationSeconds: 2592000
13 status:
14   expirationTimestamp: "2024-07-10T12:22:31Z"
15   token:
eyJhbGciOiJSUzI1NiIsImtpZCI6IkpNZ2Z4QjMzc2J6YURvZEVjdXZlX8ouZHqV7snsPw54BEci1rT4Lk6T6i3QLSnQI2oNrZ1q024VYWp7xDfVM0333FOiytOW6MjCHVnkWfHYTS30kKF_hfqKJFFnqfw1UNMtCu-1...
```



# INTRODUCTION

## GETTING STARTED

### KUBERNETES CLUSTER

The screenshot shows the Kubernetes Dashboard interface. The left sidebar contains a navigation menu with the following items:

- Workloads
  - Cron Jobs
  - Daemon Sets
  - Deployments
  - Jobs
  - Pods** (highlighted with a yellow box)
  - Replica Sets
  - Replication Controllers
  - Stateful Sets
- Service
  - Ingresses
  - Ingress Classes
  - Services
- Config and Storage
  - Config Maps
  - Persistent Volume Claims
  - Secrets
  - Storage Classes
- Cluster
  - Cluster Role Bindings
  - Cluster Roles
  - Events
  - Namespaces
  - Nodes
  - Persistent Volumes
  - Role Bindings
  - Roles
  - Service Accounts
  - Custom Resource Definitions

The main content area displays three large green circles representing the status of different workload types:

- Deployments:** Running: 6
- Pods:** Running: 6
- Replica Sets:** Running: 12

Below these summary metrics, there are two tables:

#### Deployments

Name	Images	Labels	Pods	Created
numberdisplay-mycustomapp2	ekhurtado/gcis-issia-24:number-display	applicationName: mycustomapp2 microservice.name: numberdisplay resource.controller: microservice-controller Show all	1 / 1	a day ago
aintzanecustom-mycustomapp2	aarmentia/issia2024:custom-app	applicationName: mycustomapp2 microservice.name: aintzanecustom resource.controller: microservice-controller Show all	1 / 1	a day ago
numbergeneratorbutton-mycustomapp2	ekhurtado/gcis-issia-24:number-generator	applicationName: mycustomapp2 microservice.name: numbergeneratorbutton resource.controller: microservice-controller Show all	1 / 1	a day ago
microservice-controller	ekhurtado/gcis-issia-24:microservice-controller	io.kompose.service: microservice-controller	1 / 1	a day ago
application-controller	ekhurtado/gcis-issia-24:application-controller	io.kompose.service: application-controller	1 / 1	a day ago
nodered	ekhurtado/gcis-fog:nodered	io.kompose.service: nodered	1 / 1	2 days ago

#### Pods

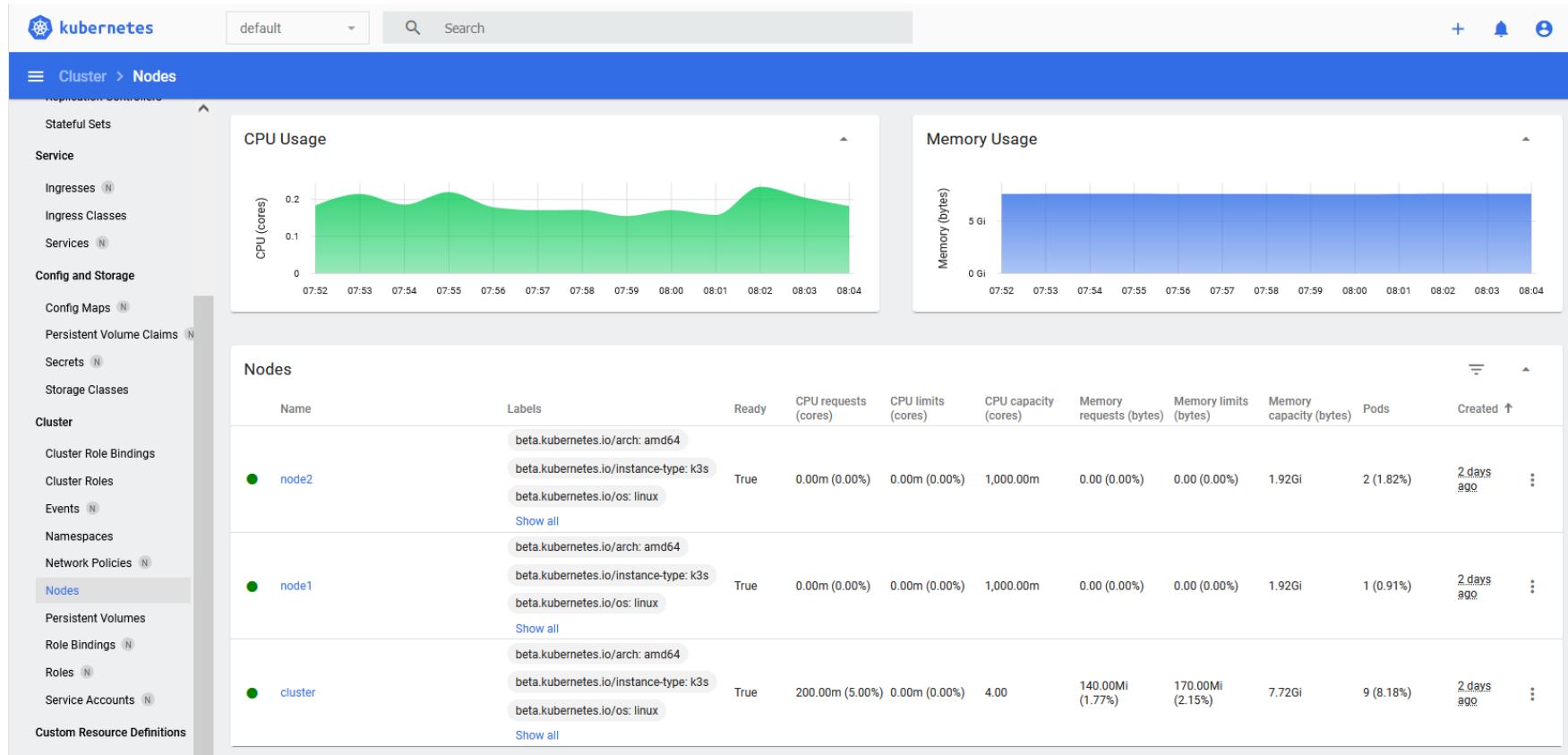
Name	Images	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created
numberdisplay-mycustomapp2-76f9cd6fcbb-qv2f	ekhurtado/gcis-issia-24:number-display	pod-template-hash: 76f9cd6fcbb resource.name: numberdisplay-mycustomapp2	node1	Running	0	<div style="width: 200px; height: 10px; background-color: #2e7131;"></div>	61.51Mi	a day ago
aintzanecustom-mycustomapp2-67c94ff76-6qlts	aarmentia/issia2024:custom-app	pod-template-hash: 67c94ff76 resource.name: aintzanecustom-mycustomapp2	node1	Running	0	<div style="width: 100px; height: 10px; background-color: #2e7131;"></div>	23.32Mi	a day ago

# INTRODUCTION

## GETTING STARTED

### KUBERNETES CLUSTER

#### Nodes Section

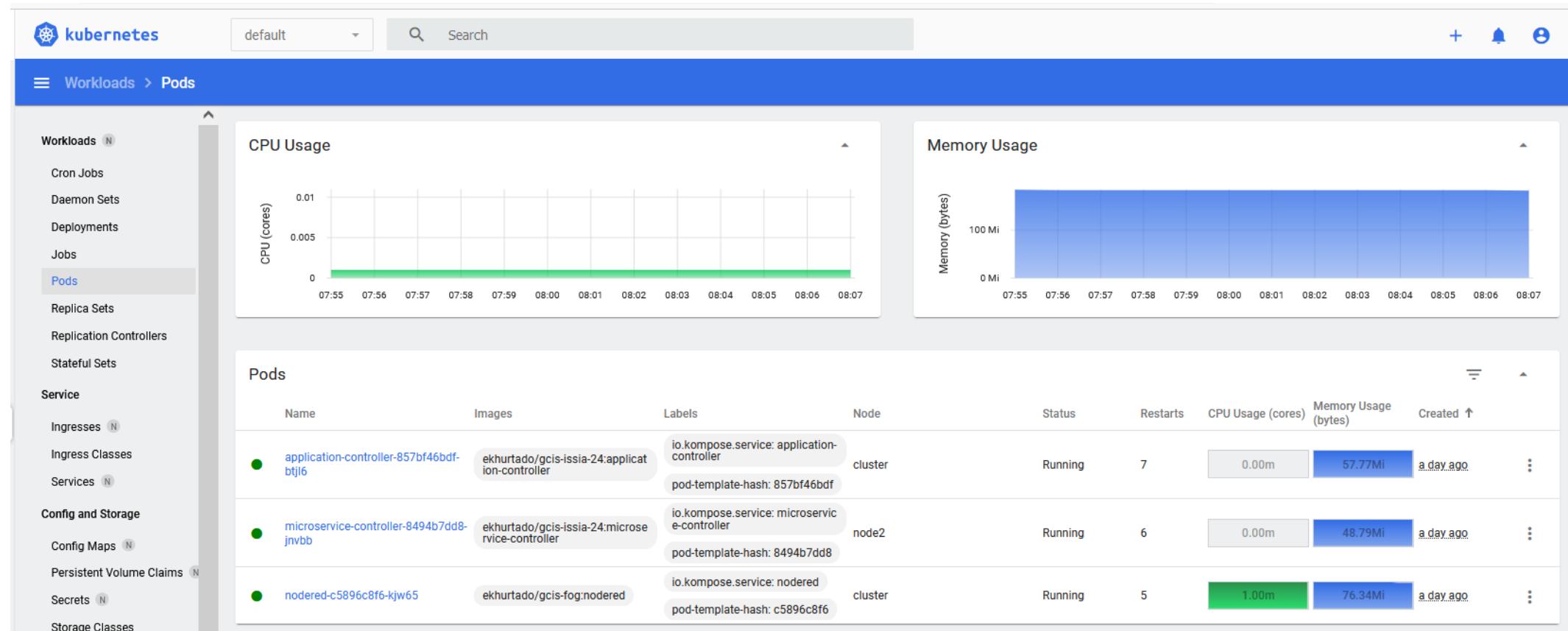


# INTRODUCTION

## GETTING STARTED

### KUBERNETES CLUSTER

#### Pods Section



## TABLE OF CONTENTS

## SECTION II



- **Introduction**
  - Microservices architecture
  - Smart manufacturing domain
  - Applications design overview
- Node-RED based Tool
- Fog Applications Design
  - Fog Components
  - Application with a Node-RED's component
  - Application with Fog Components

# DESIGN AND DEVELOPMENT OF FOG APPLICATIONS

## INTRODUCTION

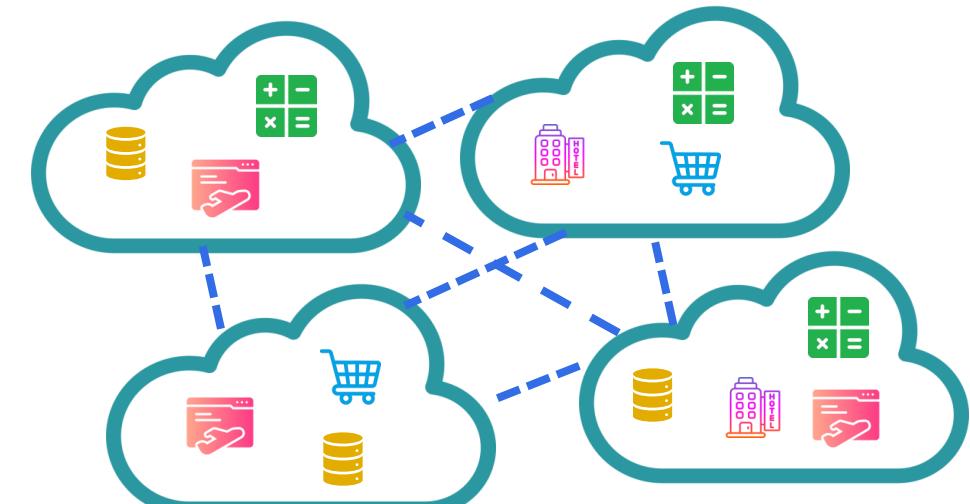
### MICROSERVICES ARCHITECTURE



**Monolithic Applications**



**Service Oriented Architectures**



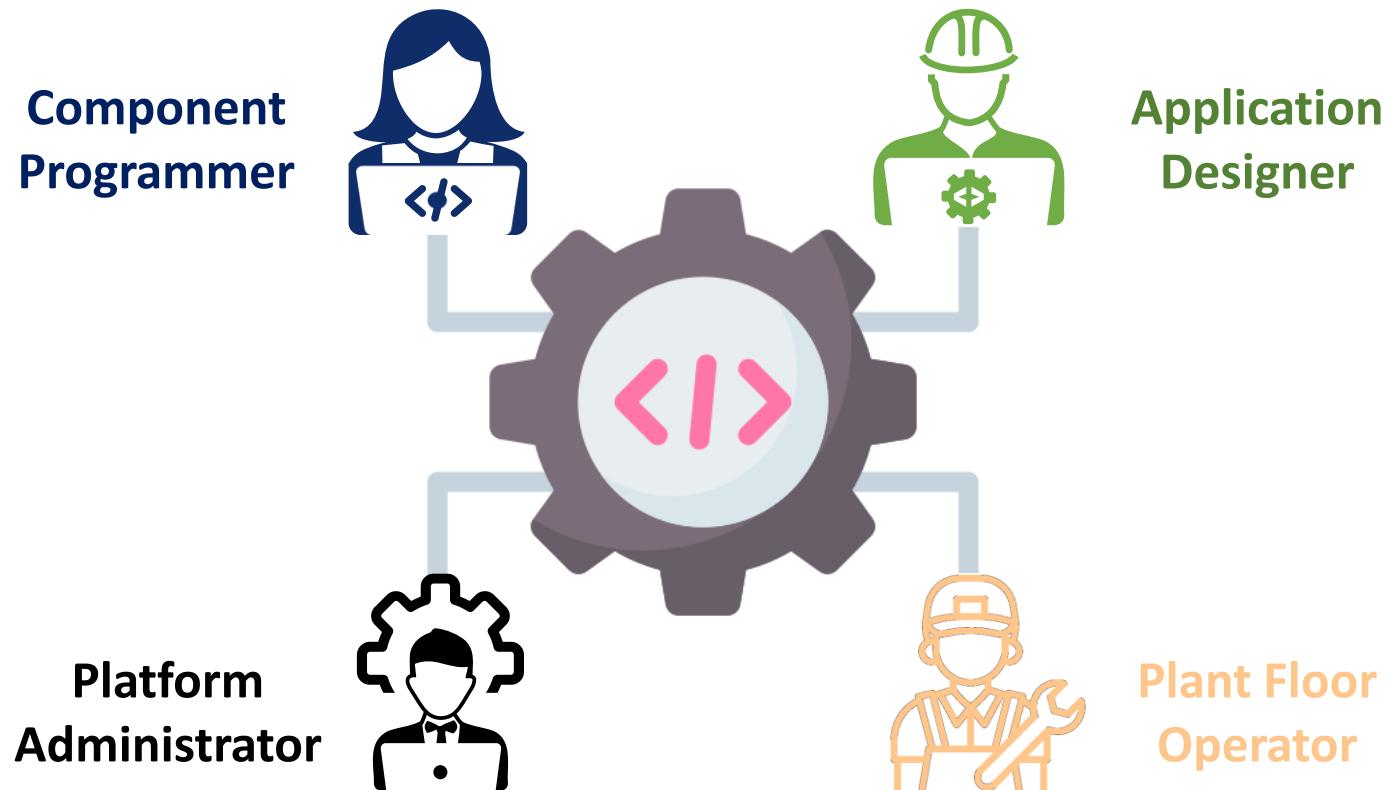
**Microservice Architecture**

# DESIGN AND DEVELOPMENT OF FOG APPLICATIONS

## INTRODUCTION

### SMART MANUFACTURING DOMAIN

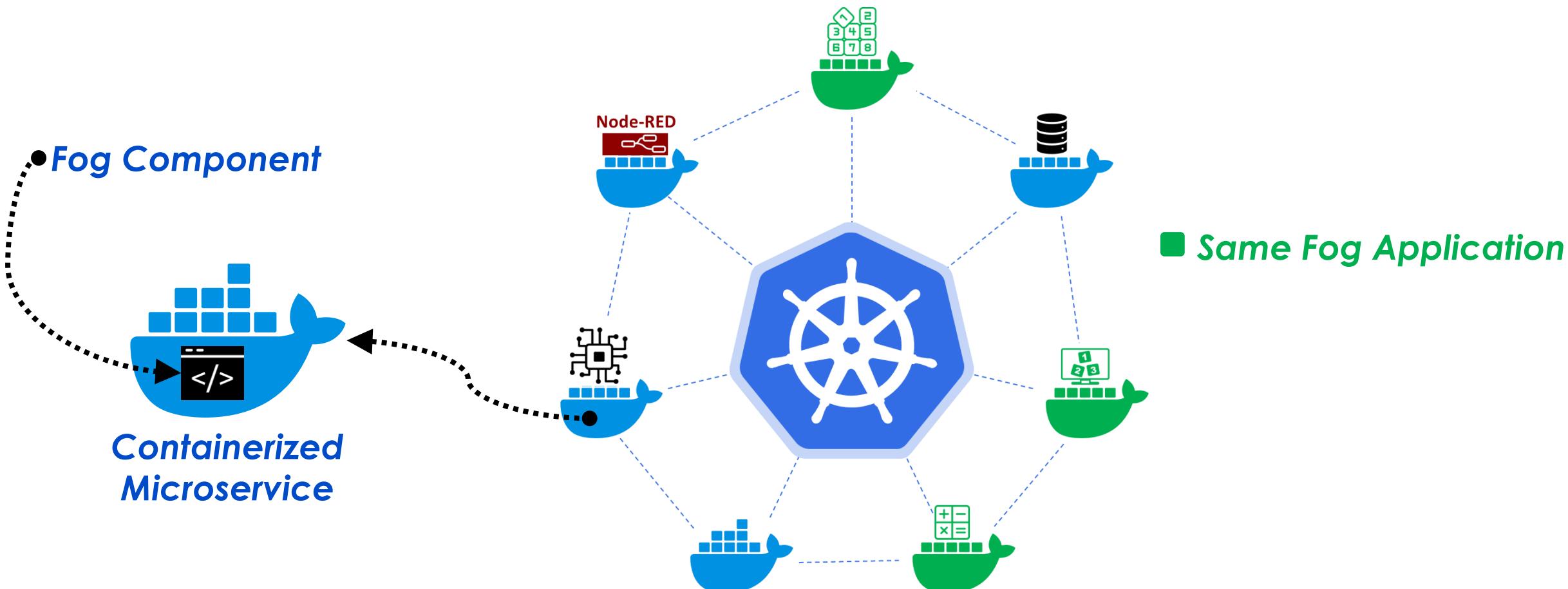
Separation of Concerns  
Abstraction  
Collaboration



# DESIGN AND DEVELOPMENT OF FOG APPLICATIONS

## INTRODUCTION

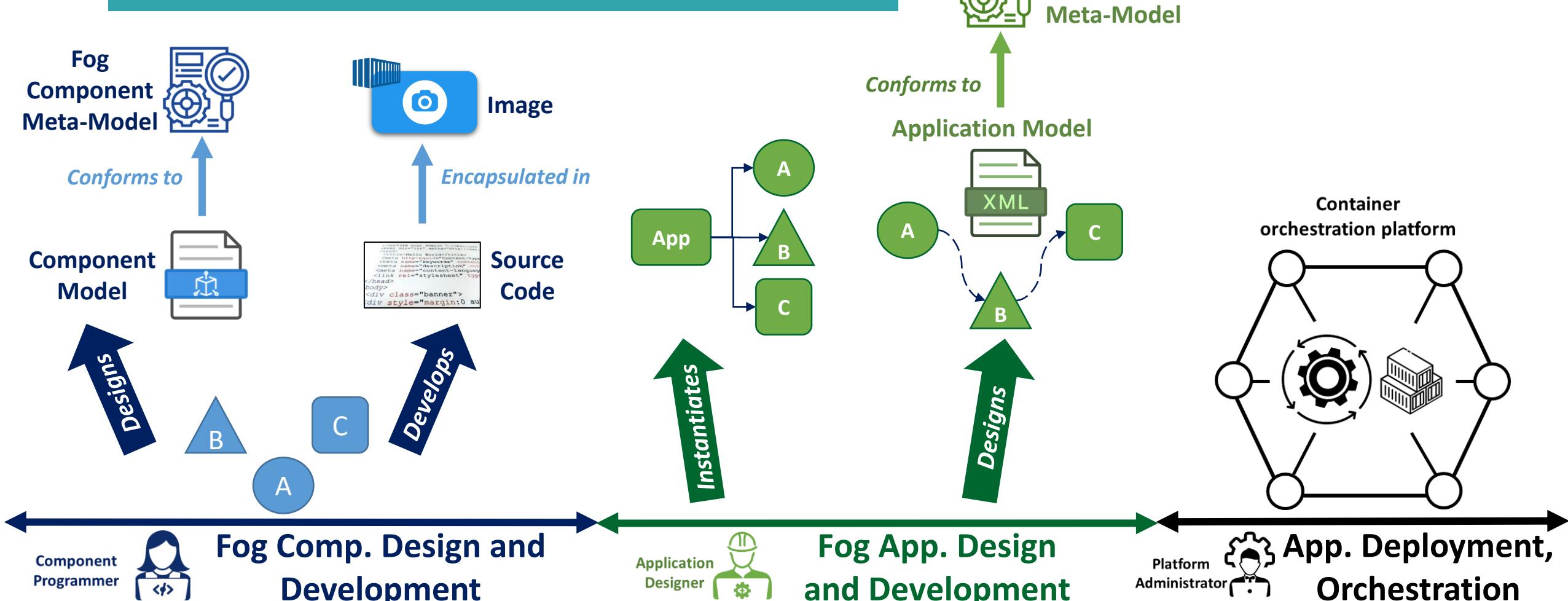
### APPLICATIONS DESIGN OVERVIEW



# DESIGN AND DEVELOPMENT OF FOG APPLICATIONS

## INTRODUCTION

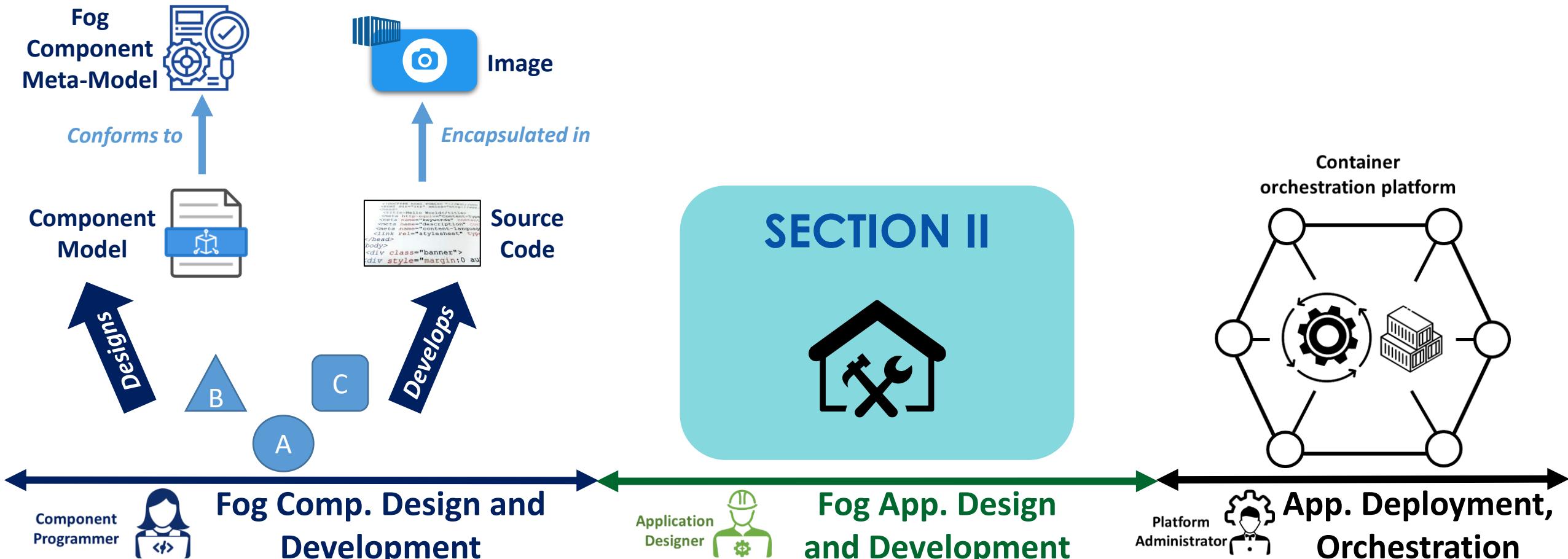
### APPLICATIONS DESIGN OVERVIEW



# DESIGN AND DEVELOPMENT OF FOG APPLICATIONS

## INTRODUCTION

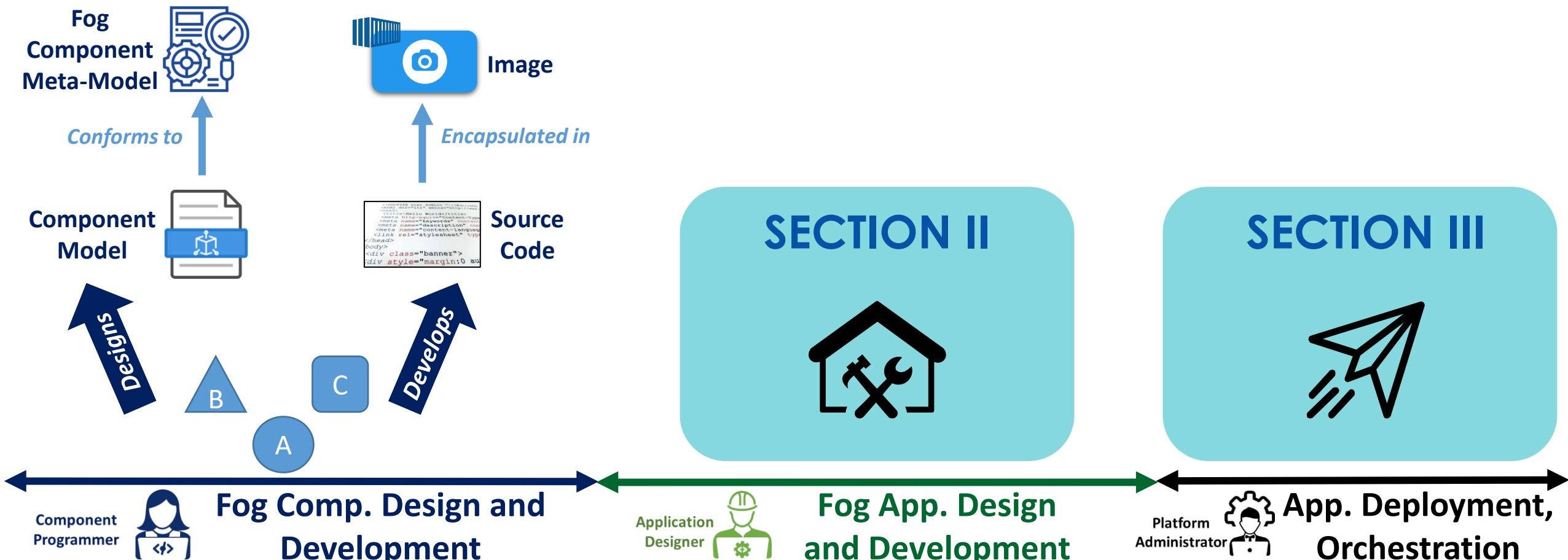
### APPLICATIONS DESIGN OVERVIEW



# DESIGN AND DEVELOPMENT OF FOG APPLICATIONS

## INTRODUCTION

### APPLICATIONS DESIGN OVERVIEW



## INTRODUCTION

### APPLICATIONS DESIGN OVERVIEW



## TABLE OF CONTENTS

### SECTION II



- Introduction
  - Microservices architecture
  - Smart manufacturing domain
  - Applications design overview
- Node-RED based Tool
- Fog Applications Design
  - Kubernetes cluster

# DESIGN AND DEVELOPMENT OF FOG APPLICATIONS

## NODE-RED BASED TOOL

### Fog Application Model (xml)

```
<Application name="DataProcessing">
    <Microservice name="eXist" service="GetAssemblyStationData"
        imgBase="gcr.io/upv-ehu/gcis:exist-services"
        customization="{'custom_stationID': 'AS1'}">
        <outPort name="eXistOPort" protocol="HTTP" dataType="TDBStation"/>
    </Microservice>
    <Microservice name="AssemblyStation" service="ProcessingOEE"
        imgBase="gcr.io/upv-ehu/gcis:processing-services">
        <inPort name="AssemblyStationIPort" protocol="HTTP"
            dataType="TDBStation" number="6000"/>
        <outPort name="AssemblyStationOPort" protocol="HTTP" dataType="TStationOEE"/>
    </Microservice>
    <Microservice name="OEEEEvents" service="NotifyOperator"
        imgBase="gcr.io/upv-ehu/gcis:oee-events"
        customization="{'custom_threshold': 70}">
        <inPort name="OEEEEventsIPort" protocol="HTTP"
            dataType="TStationOEE" number="7000"/>
        <outPort name="OEEEEventsOPort" protocol="HTTP" dataType="TDBOEE"/>
    </Microservice>
    <Microservice name="Influx" service="StoreOEEData"
        imgBase="gcr.io/upv-ehu/gcis:influx-services">
        <inPort name="InfluxIPort" protocol="HTTP"
            dataType="TDBOEE" number="8000"/>
    </Microservice>
    <channel from="eXistOPort" to="AssemblyStationIPort"/>
    <channel from="AssemblyStationOPort" to="OEEEEventsIPort"/>
    <channel from="OEEEEventsOPort" to="InfluxIPort"/>
</Application>
```

### Fog Component Model (xml)

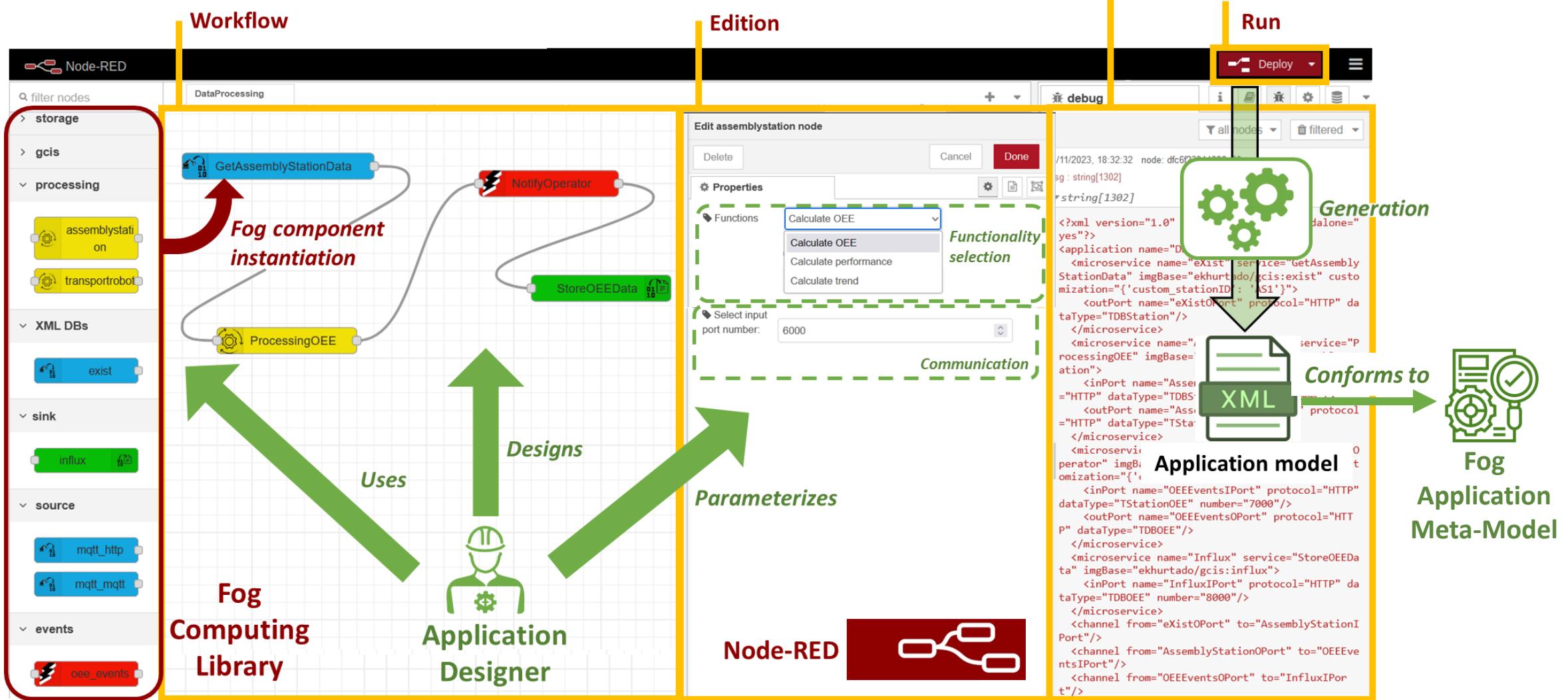
```
<Component name="AssemblyStation" category="processing" description="This component
calculates several data related to the operation of an assembly station" imgBase=
"processing_assembly_base_image">
    <functionality id="ProcessingOEE" name="Calculate OEE" description="It calculates
the OEE related to the station" customization="custom_threshold">
        <inputs name="DBdata" dataType="TDBStation" protocol="HTTP"/>
        <outputs name="StationOEE" dataType="TStationOEE" protocol="HTTP"/>
    </functionality>
    <functionality id="ProcessingPerformance" name="Calculate performance" description=
"It calculates the performance of the station">
        <inputs name="DBdata" dataType="TDBPerformance" protocol="HTTP"/>
        <outputs name="StationPerformance" dataType="TStationPerformance" protocol=
"HTTP"/>
    </functionality>
    <functionality id="ProcessingTrend" name="Calculate Trend" description="It
calculates the trend of the station">
        <inputs name="DBdata" dataType="TDBTrend" protocol="HTTP"/>
        <outputs name="StationTrend" dataType="TStationTrend" protocol="HTTP"/>
    </functionality>
</Component>
```

### Instantiated Fog Components

### Fog Application workflow

# DESIGN AND DEVELOPMENT OF FOG APPLICATIONS

## NODE-RED BASED TOOL



## TABLE OF CONTENTS

### SECTION II

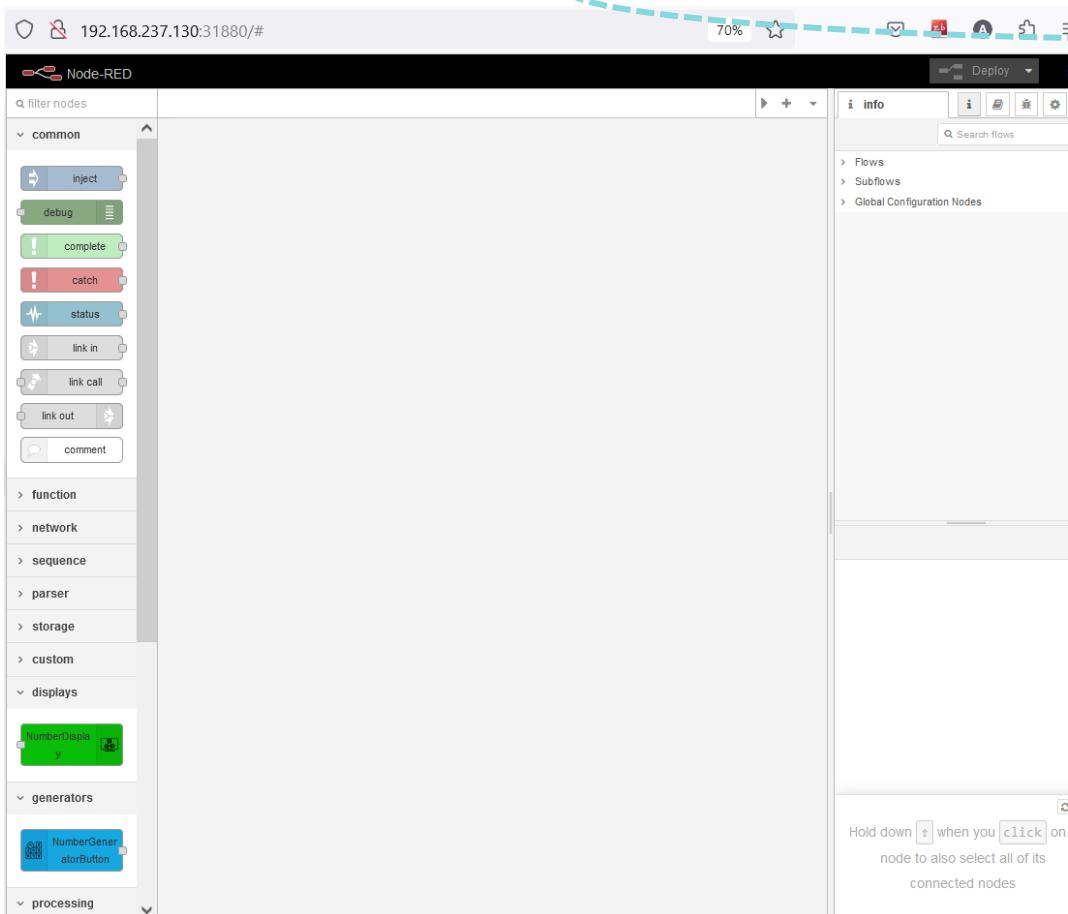


- Introduction
  - Microservices architecture
  - Smart manufacturing domain
  - Applications design overview
- Node-RED based Tool
- **Fog Applications Design**
  - Fog Components
  - Application with a Node-RED's component
  - Application with Fog Components

# DESIGN AND DEVELOPMENT OF FOG APPLICATIONS

## FOG APPLICATIONS DESIGN

- 1 Connect to Node-RED from your computer  
[https://<IP\\_ADDRESS>:31880](https://<IP_ADDRESS>:31880)



```
ens33: flags=4163<UP,BROADCAST  
      inet 192.168.237.130
```

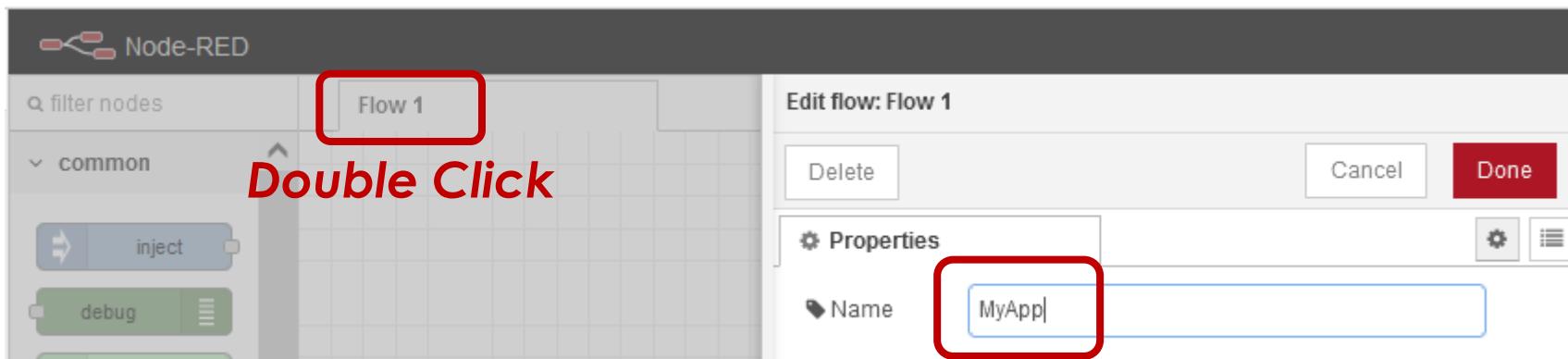
# DESIGN AND DEVELOPMENT OF FOG APPLICATIONS

## FOG APPLICATIONS DESIGN

- 2 Create a Flow



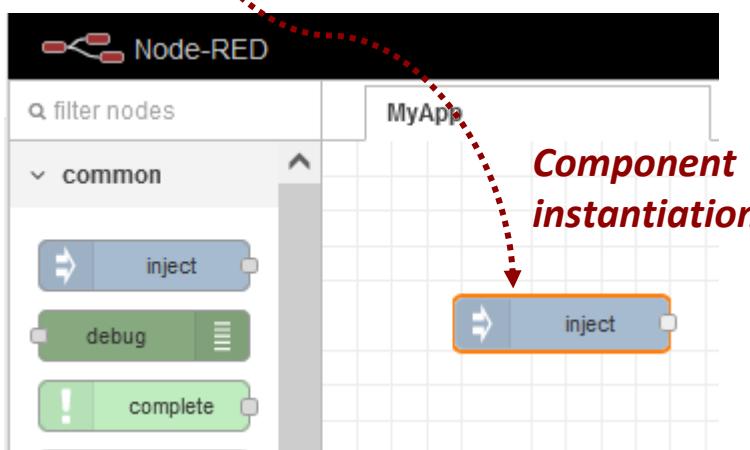
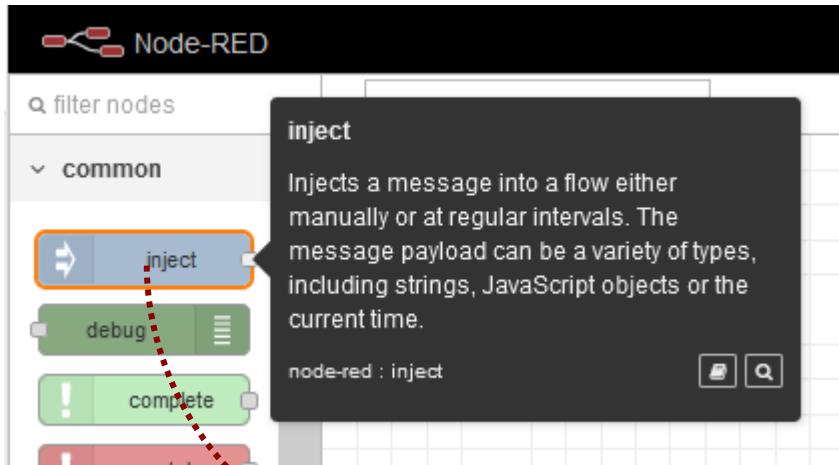
- 3 Name it



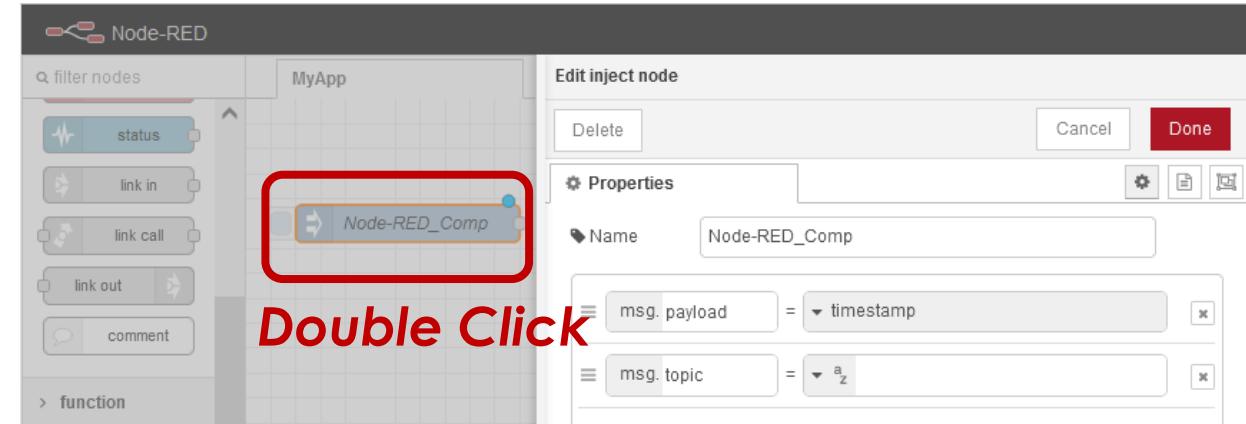
# DESIGN AND DEVELOPMENT OF FOG APPLICATIONS

## FOG APPLICATIONS DESIGN

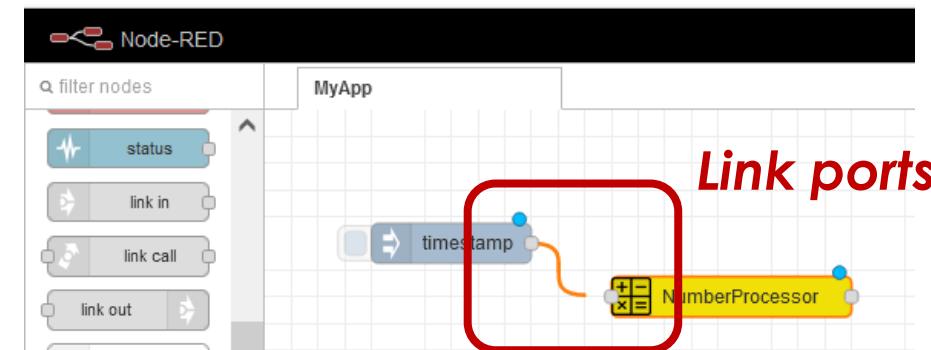
- 4 Instantiate a component: select and drag



- 5 Component information and configuration



- 6 Connect components



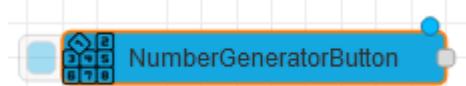
# DESIGN AND DEVELOPMENT OF FOG APPLICATIONS

## FOG APPLICATIONS DESIGN

### FOG COMPONENTS



Number Generator



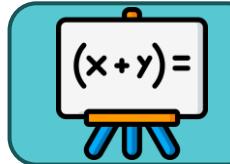
Edit NumberGeneratorButton node

Delete Cancel Done

Properties

Functionalities

- Select generation type
- Natural numbers
- Integer numbers
- Decimal numbers



Processor



NumberProcessor

Edit NumberProcessor node

Delete Cancel Done

Properties

Functionalities

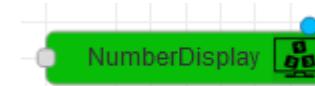
- Select input port number
- Increase value
- Decrease value
- Multiply value

Select step size

Select multiplier



Display



NumberDisplay

Edit NumberDisplay node

Delete Cancel Done

Properties

Functionalities

- Select input port number
- Display the value in the console
- Save (.txt)
- Save (.csv)

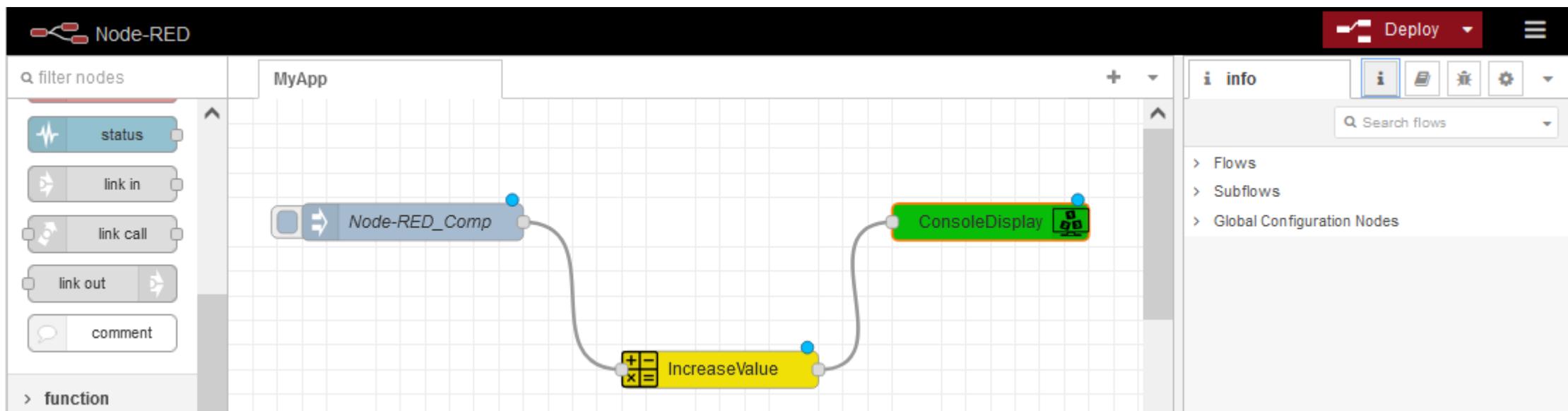
Add the name of the file

File path and name

### FOG APPLICATIONS DESIGN

#### APP. WITH A NODE-RED'S COMPONENT

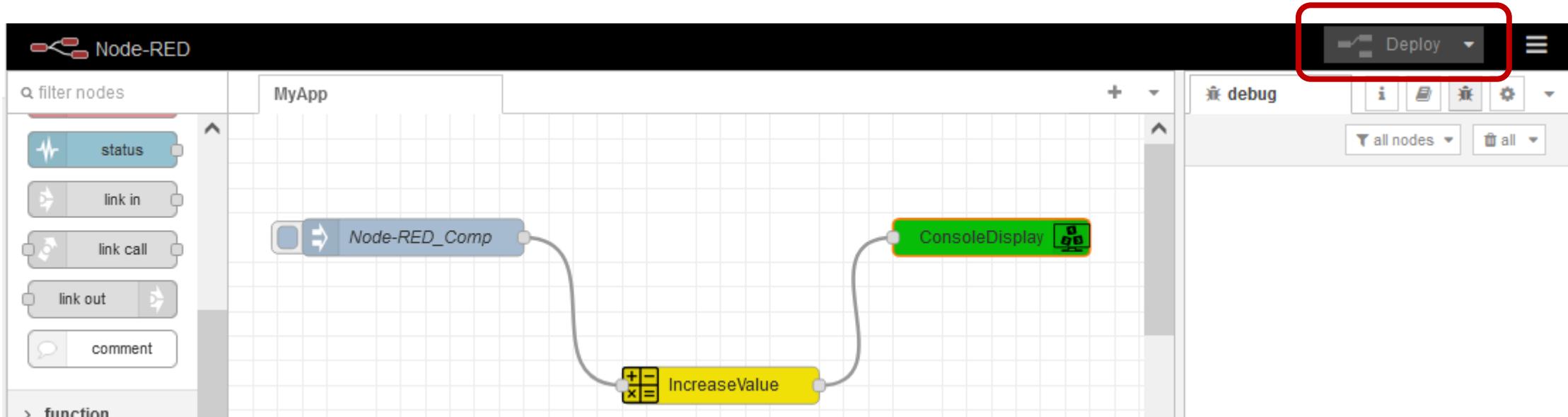
- 1 Create the application + instantiate and connect components



### FOG APPLICATIONS DESIGN

#### APP. WITH A NODE-RED'S COMPONENT

2 Deploy (debug window)

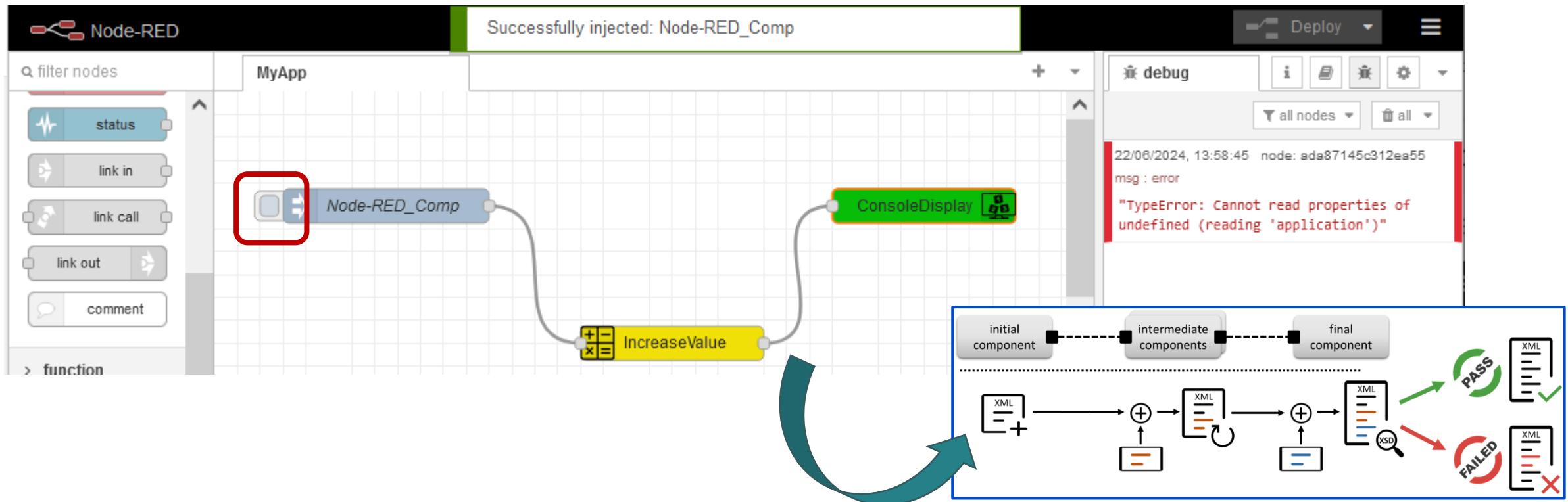


# DESIGN AND DEVELOPMENT OF FOG APPLICATIONS

## FOG APPLICATIONS DESIGN

### APP. WITH A NODE-RED'S COMPONENT

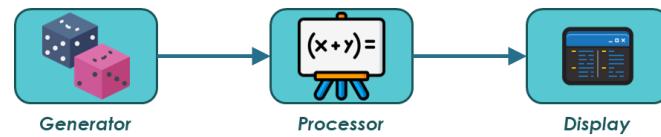
- 3 Generate the Fog Application model → Click the starting component's button → Error!!



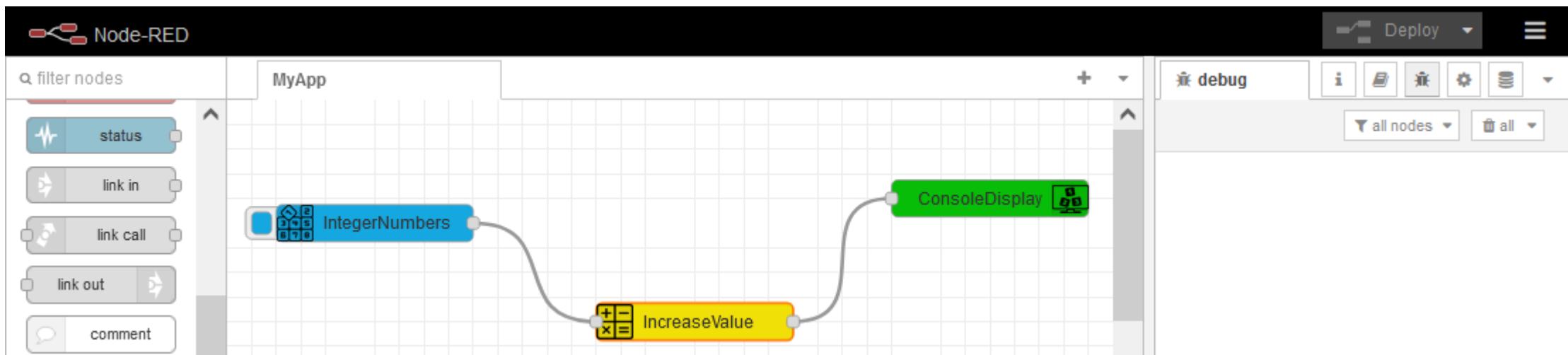
# DESIGN AND DEVELOPMENT OF FOG APPLICATIONS

## FOG APPLICATIONS DESIGN

### APP. WITH FOG COMPONENTS



- 1 Instantiate and connect components + Deploy

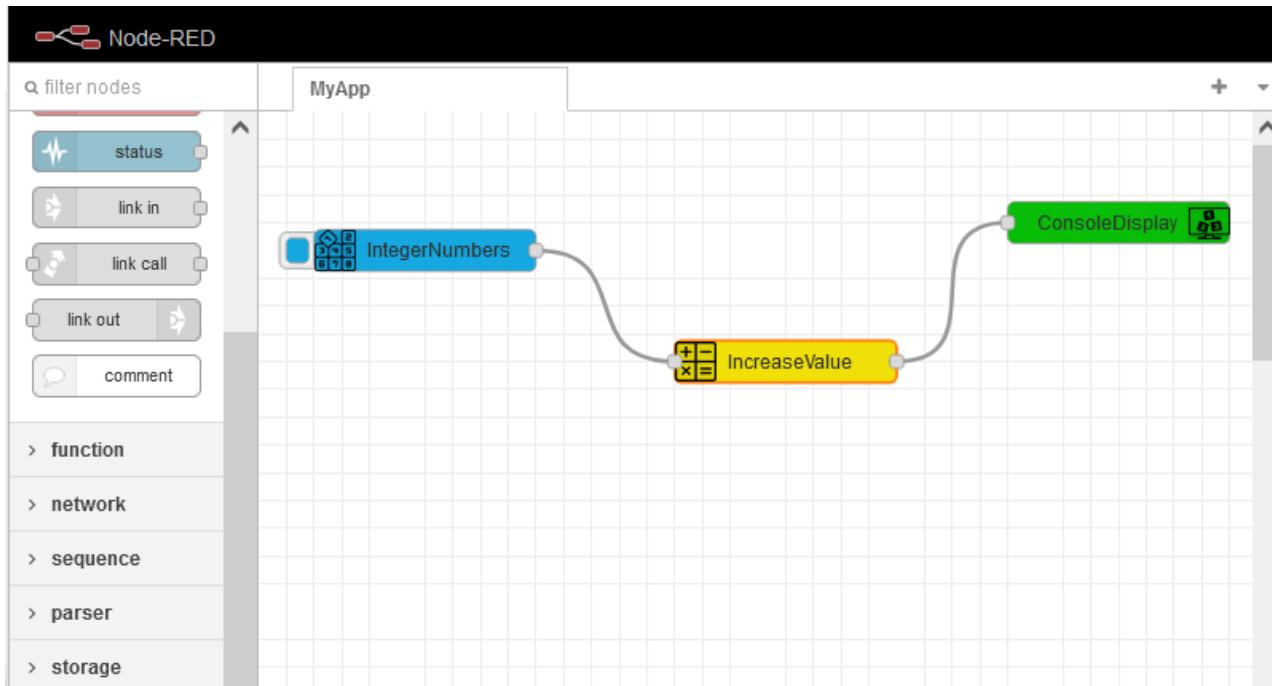


# DESIGN AND DEVELOPMENT OF FOG APPLICATIONS

## FOG APPLICATIONS DESIGN

### APP. WITH FOG COMPONENTS

- 2 Click the starting component's button



### Application Model

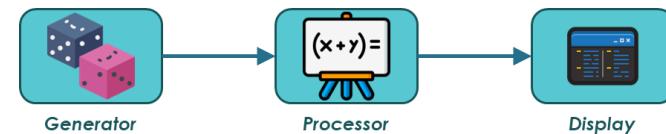


22/06/2024, 14:06:41 node: 9ff2217568b5857c  
msg : string[1093]  
▼ string[1093]  
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>  
<application name="MyApp">  
  <microservice name="NumberGeneratorButton" service="IntegerNumbers" imgBase="ekhurtado/gcis-issia-24:number-generator" customization="{'custom\_type': 'random', 'custom\_initialvalue': 0}">  
    <outPort name="NumberGeneratorButtonOPort" protocol="HTTP" dataType="TNumber"/>  
  </microservice>  
  <microservice name="NumberProcessor-1" service="IncreaseValue" imgBase="ekhurtado/gcis-issia-24:number-processor" customization="{'custom\_step': 2}">  
    <inPort name="NumberProcessor-1IPort" protocol="HTTP" dataType="TNumber" number="7000"/>  
    <outPort name="NumberProcessor-1OPort" protocol="HTTP" dataType="TNumber"/>  
  </microservice>  
  <microservice name="NumberDisplay" service="ConsoleDisplay" imgBase="ekhurtado/gcis-issia-24:number-display">  
    <inPort name="NumberDisplayIPort" protocol="HTTP" dataType="TNumber" number="7000"/>  
  </microservice>  
  <channel from="NumberGeneratorButtonOPort" to="NumberProcessor-1IPort"/>  
  <channel from="NumberProcessor-1OPort" to="NumberDisplayIPort"/>  
</application>

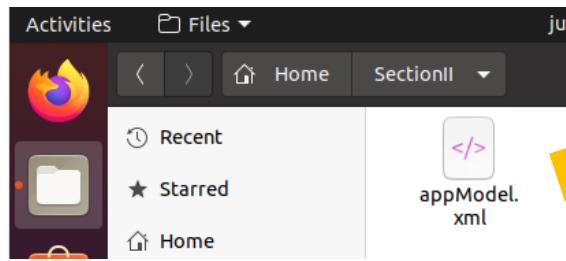
# DESIGN AND DEVELOPMENT OF FOG APPLICATIONS

## FOG APPLICATIONS DESIGN

### APP. WITH FOG COMPONENTS



- 3 Create the application model XML file in the Section II directory (appModel.xml)



```
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <application name="MyApp">
3   <microservice name="NumberGeneratorButton" service="IntegerNumbers"
 imgBase="ekhurtado/gcis-issia-24:number-generator"
customization="{'custom_type': 'random', 'custom_initialvalue': 0}">
4     <outPort name="NumberGeneratorButtonOPort" protocol="HTTP"
dataType="TNumber"/>
5   </microservice>
6   <microservice name="NumberProcessor-1" service="IncreaseValue"
 imgBase="ekhurtado/gcis-issia-24:number-processor"
customization="{'custom_step': 2}">
7     <inPort name="NumberProcessor-1IPort" protocol="HTTP" dataType="TNumber"
number="7000"/>
8     <outPort name="NumberProcessor-10Port" protocol="HTTP" dataType="TNumber"/>
9   </microservice>
10  <microservice name="NumberDisplay" service="ConsoleDisplay"
 imgBase="ekhurtado/gcis-issia-24:number-display">
11    <inPort name="NumberDisplayIPort" protocol="HTTP" dataType="TNumber"
number="7000"/>
12  </microservice>
13  <channel from="NumberGeneratorButtonOPort" to="NumberProcessor-1IPort"/>
14  <channel from="NumberProcessor-10Port" to="NumberDisplayIPort"/>
15 </application>
```

## TABLE OF CONTENTS

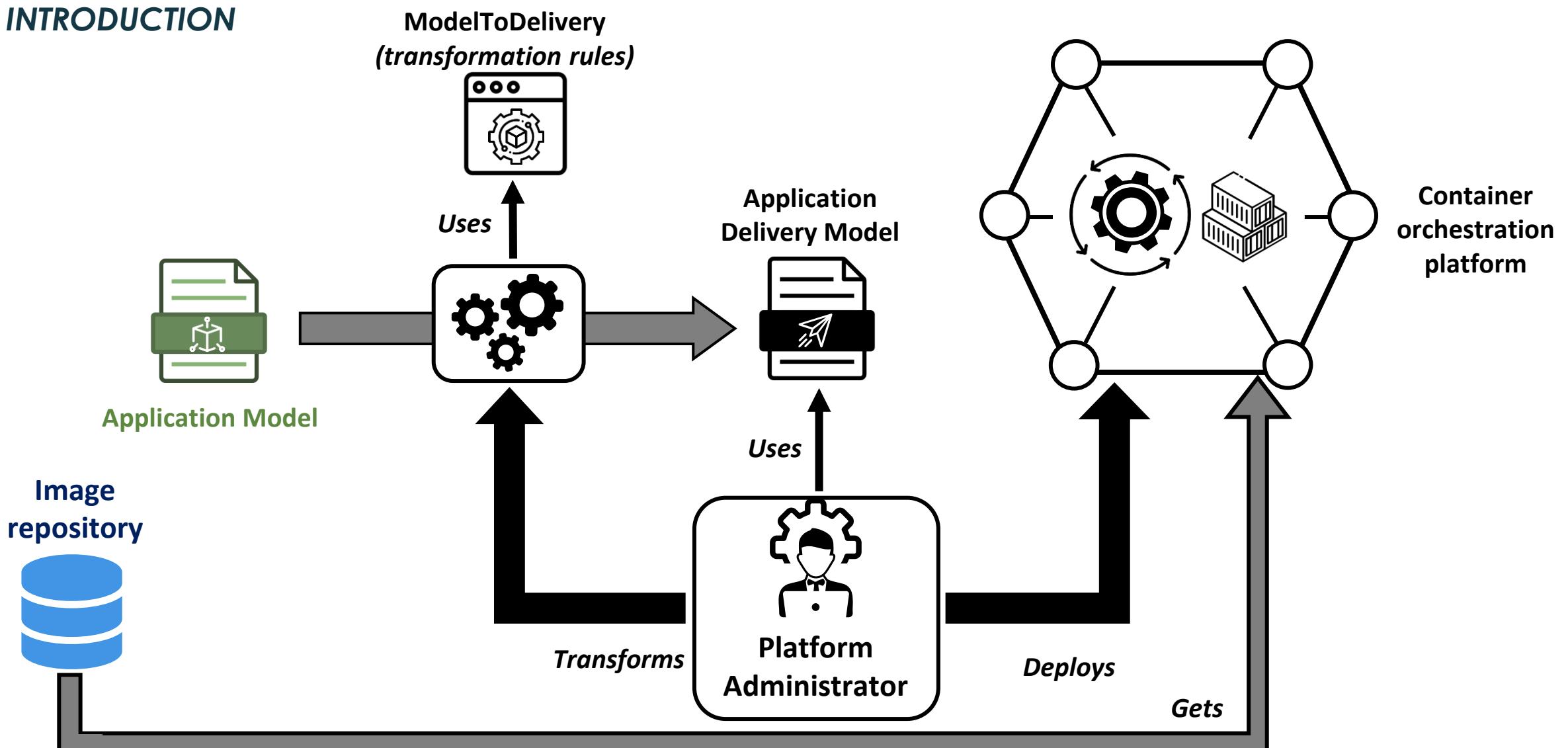
### SECTION III



- Introduction
- Kubernetes Extension Mechanisms
- Application Deployment
- Application Orchestration
  - Node failure
- More complex application

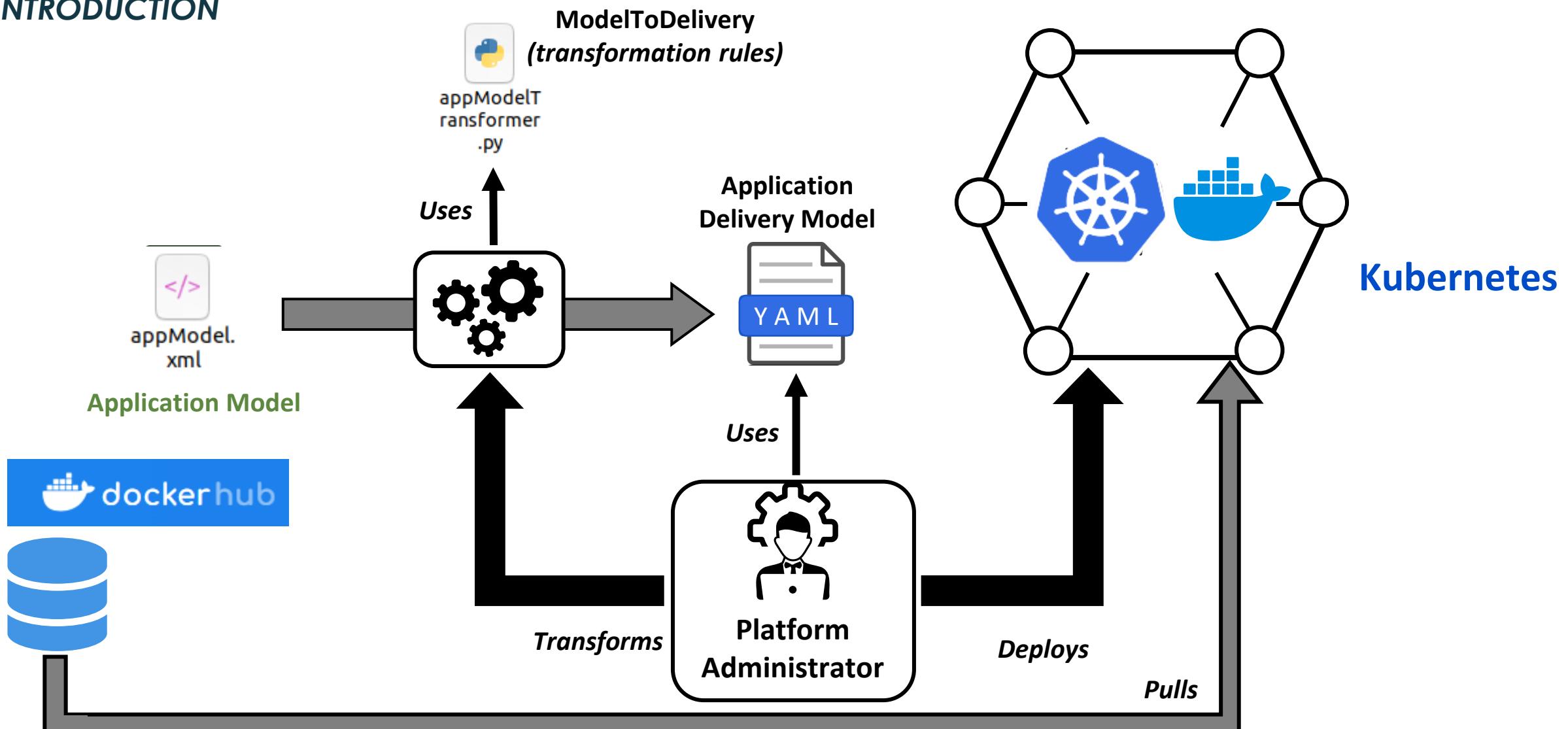
# DEPLOYMENT AND ORCHESTRATION OF FOG APPLICATIONS

## INTRODUCTION



# DEPLOYMENT AND ORCHESTRATION OF FOG APPLICATIONS

## INTRODUCTION

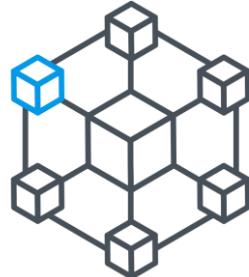


# DEPLOYMENT AND ORCHESTRATION OF FOG APPLICATIONS

INTERNATIONAL SUMMER SCHOOL ON  
INDUSTRIAL AGENTS 2024

STANDARDIZATION OF I4.0 SYSTEMS

## INTRODUCTION



### *Application Deployment*

- In a single CN environment
  - Same result for each deployment request.
- In a multiple CNs environment
  - On which CN should it be deployed for each request?
  - Does the deployment request have any criteria?
  - How to balance the workload across all CNs?



### *Application Concept*

- Component-level orchestration.
  - Pods for containers creation.
  - Deployments for containers deployment in the cluster.
  - Services for microservices communication.
- For Kubernetes, microservices (instantiated Fog Components) are simply deployments.
- Kubernetes does not know the relationships between microservices.
- All deployments are considered the same.

## TABLE OF CONTENTS

### SECTION III



- Introduction
- **Kubernetes Extension Mechanisms**
- Application Deployment
- Application Orchestration
  - Node failure
- More complex application

# DEPLOYMENT AND ORCHESTRATION OF FOG APPLICATIONS

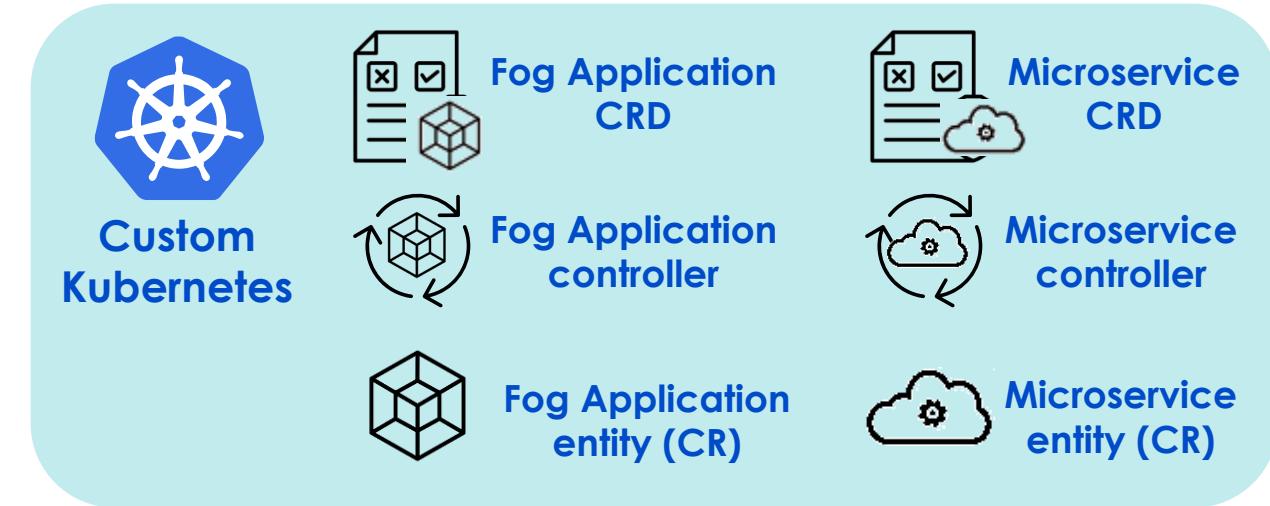
INTERNATIONAL SUMMER SCHOOL ON  
INDUSTRIAL AGENTS 2024

STANDARDIZATION OF I4.0 SYSTEMS

## KUBERNETES EXTENSION MECHANISMS



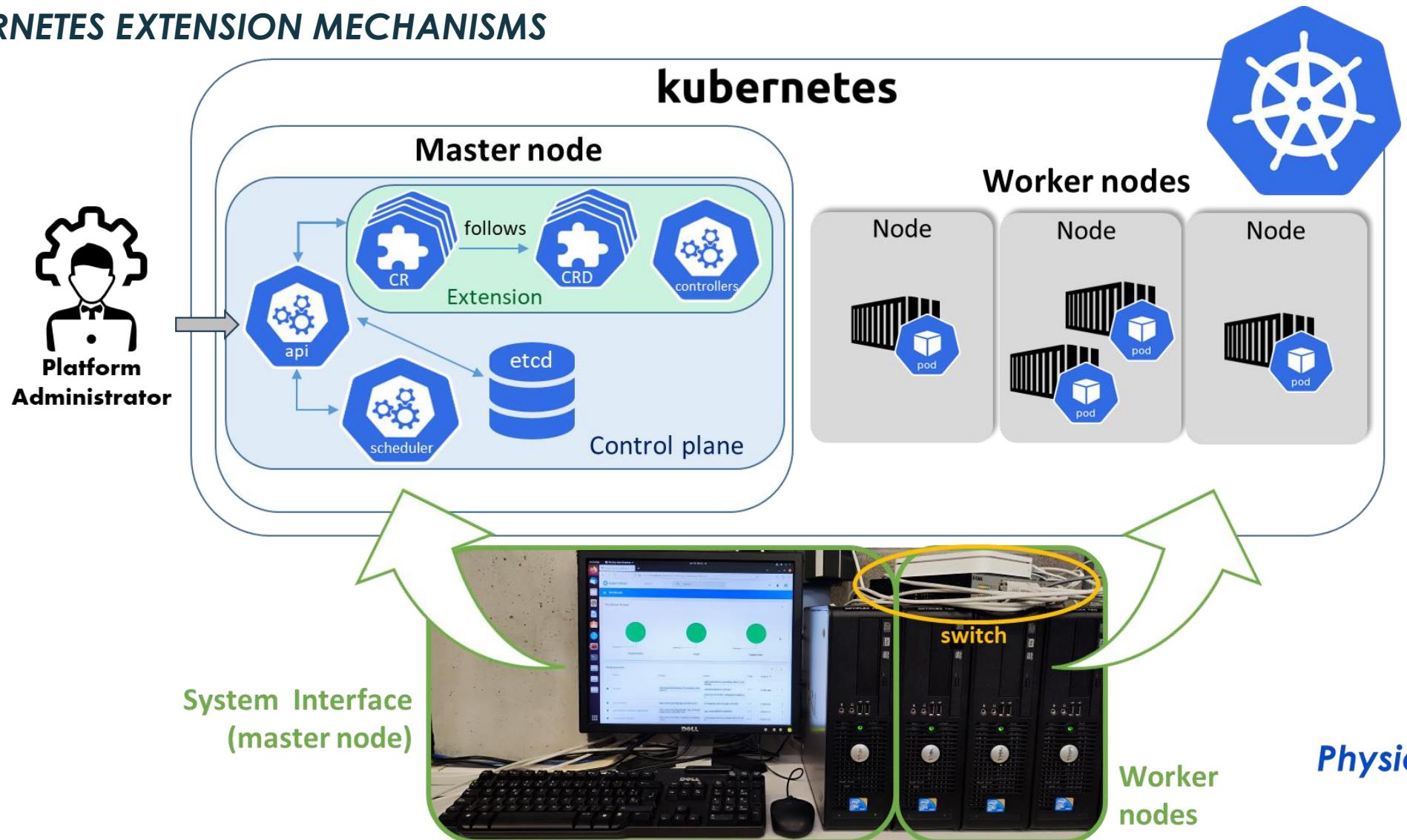
EXTENSION  
»»» »»» »»»



- How to integrate the fog application concept?
  - Create Custom Resources (CR) to add the application concept as first level entity, and to add the microservice concept as second level entity.
  - Create Controllers to manage CRs.
  - Define Custom Resource Definitions (CRDs) as the meta-models that CRs must follow.

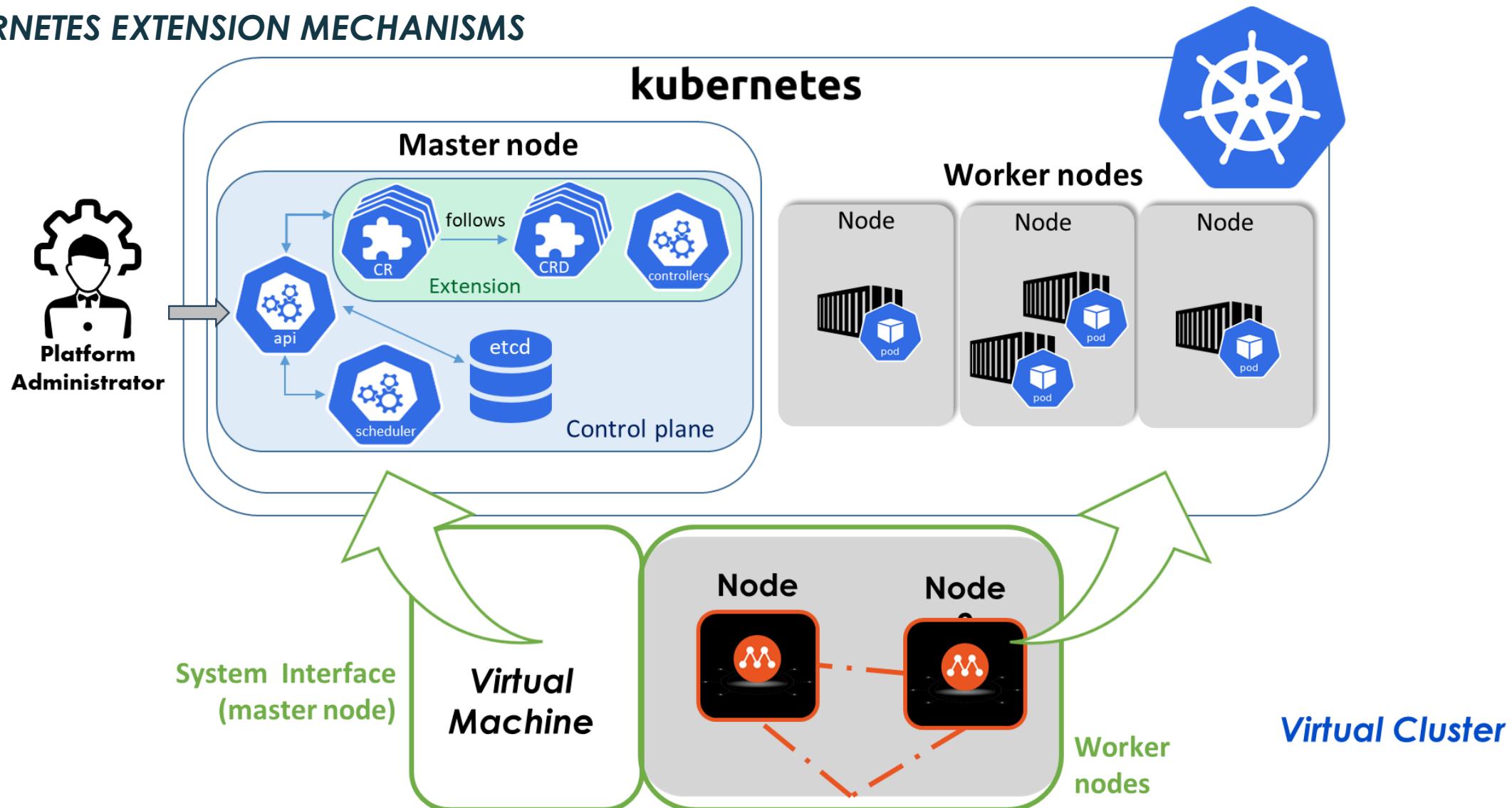
# DEPLOYMENT AND ORCHESTRATION OF FOG APPLICATIONS

## KUBERNETES EXTENSION MECHANISMS



# DEPLOYMENT AND ORCHESTRATION OF FOG APPLICATIONS

## KUBERNETES EXTENSION MECHANISMS



## TABLE OF CONTENTS

### SECTION III



- Introduction
- Kubernetes Extension Mechanisms
- Application Deployment
- Application Orchestration
  - Node failure
- More complex application

# DEPLOYMENT AND ORCHESTRATION OF FOG APPLICATIONS

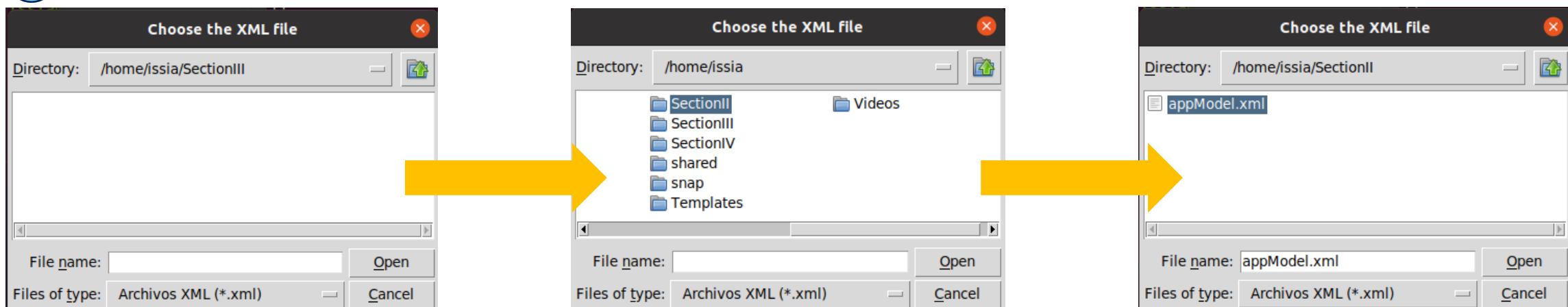
## APPLICATION DEPLOYMENT

- 1 Execute the “appModelTransformer.py” program, located in the SectionIII directory

```
python3 appModelTransformer.py
```

```
issia@cluster:~$ cd SectionIII
issia@cluster:~/SectionIII$ ls
appModelTransformer.py
issia@cluster:~/SectionIII$ python3 appModelTransformer.py
To enter the XML application model, select one of the following options:
    -> 1: Enter the application model as a file.
    -> 2: Enter the application model as text.
    -> 3: Exit the program.
Enter the option number: 1
```

- 2 Select the option 1, and choose the application model file

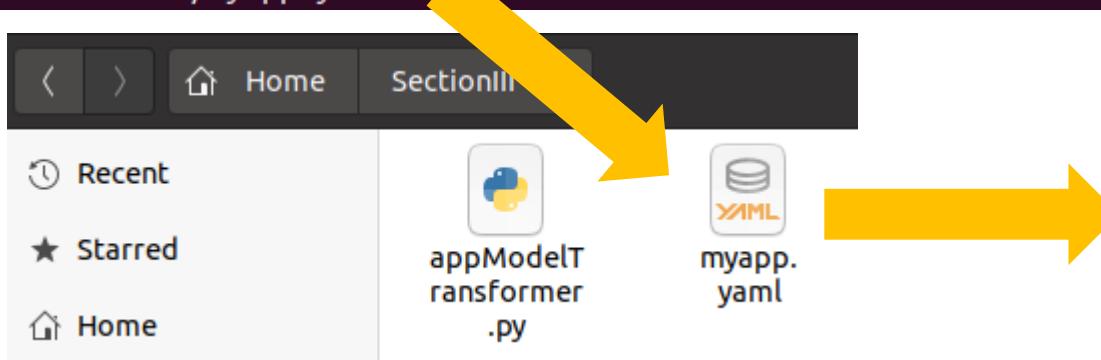


# DEPLOYMENT AND ORCHESTRATION OF FOG APPLICATIONS

## APPLICATION DEPLOYMENT

- 3 The Application Delivery Model has been created in the SectionIII directory

```
issia@cluster:~$ cd SectionIII
issia@cluster:~/SectionIII$ ls
appModelTransformer.py
issia@cluster:~/SectionIII$ python3 appModelTransformer.py
To enter the XML application model, select one of the following options:
    -> 1: Enter the application model as a file.
    -> 2: Enter the application model as text.
    -> 3: Exit the program.
Enter the option number: 1
1
Before starting the process, the entered application model will be checked with
the application meta-model.
After verifying the validity of the model, the Application Delivery Model will
be obtained, in the form of a Custom Resource in a YAML file.
Application Delivery Model correctly created! The new file is in the following
path: /home/issia/SectionIII/myapp.yaml
```



```
1 apiVersion: ehu.gcis.org/v1alpha1
2 kind: Application
3 metadata:
4   name: myapp
5 spec:
6   microservices:
7     - name: numbergeneratorbutton
8       service: IntegerNumbers
9       image: ekhurtado/gcis-issia-24:number-generator
10      customization: '{"custom_type": "random", "custom_initialvalue": 0}'
11      outPort:
12        name: numbergeneratorbutton-oport
13        dataType: TNumber
14        protocol: HTTP
15     - name: numberprocessor-1
16       service: IncreaseValue
17       image: ekhurtado/gcis-issia-24:number-processor
18       customization: '{"custom_step": 2}'
19       inPort:
20         name: numberprocessor-1-iport
21         dataType: TNumber
22         protocol: HTTP
23         number: '7000'
24       outPort:
25         name: numberprocessor-1-oport
26         dataType: TNumber
```



# DEPLOYMENT AND ORCHESTRATION OF FOG APPLICATIONS

## APPLICATION DEPLOYMENT

- 4 Create a new resource ( “Custom Resource Definitions” menu of the dashboard)

The screenshot shows the Kubernetes dashboard interface. On the left, a sidebar lists resources: Stateful Sets, Service, Ingresses (0), Ingress Classes, Services (0), and Config and Storage. The main area is titled "Custom Resource Definitions". Under "Custom Resource Definitions", there are two entries: "Microservice" and "Application". A modal window is open, titled "Create from file". It contains a code editor with the following YAML content:

```
1 apiVersion: ehu.gcis.org/v1alpha1
2 kind: Application
3 metadata:
4   name: myapp
5 spec:
6   microservices:
7     - name: numbergeneratorbutton
8       service: IntegerNumbers
9       image: ekhurtado/gcis-issia-24:number-generator
10      customization: '{"custom_type": "random", "custom_initialvalue": 0}'
11      outPort:
12        name: numbergeneratorbutton-oport
13        dataType: TNumber
14        protocol: HTTP
15      - name: numberprocessor-1
16        service: IncreaseValue
17        image: ekhurtado/gcis-issia-24:number-processor
18        customization: '{"custom_step": 2}'
19        inPort:
20          name: numberprocessor-1-iport
21          dataType: TNumber
22          protocol: HTTP
23          number: '7000'
```

At the bottom of the modal, there are "Upload" and "Cancel" buttons. The "Upload" button is highlighted with a red box. To the right of the modal, there is a large yellow arrow pointing towards the "Create new resource" button in the top right corner of the dashboard header. Below the modal, there is a "PASTE!" icon with a clipboard symbol.

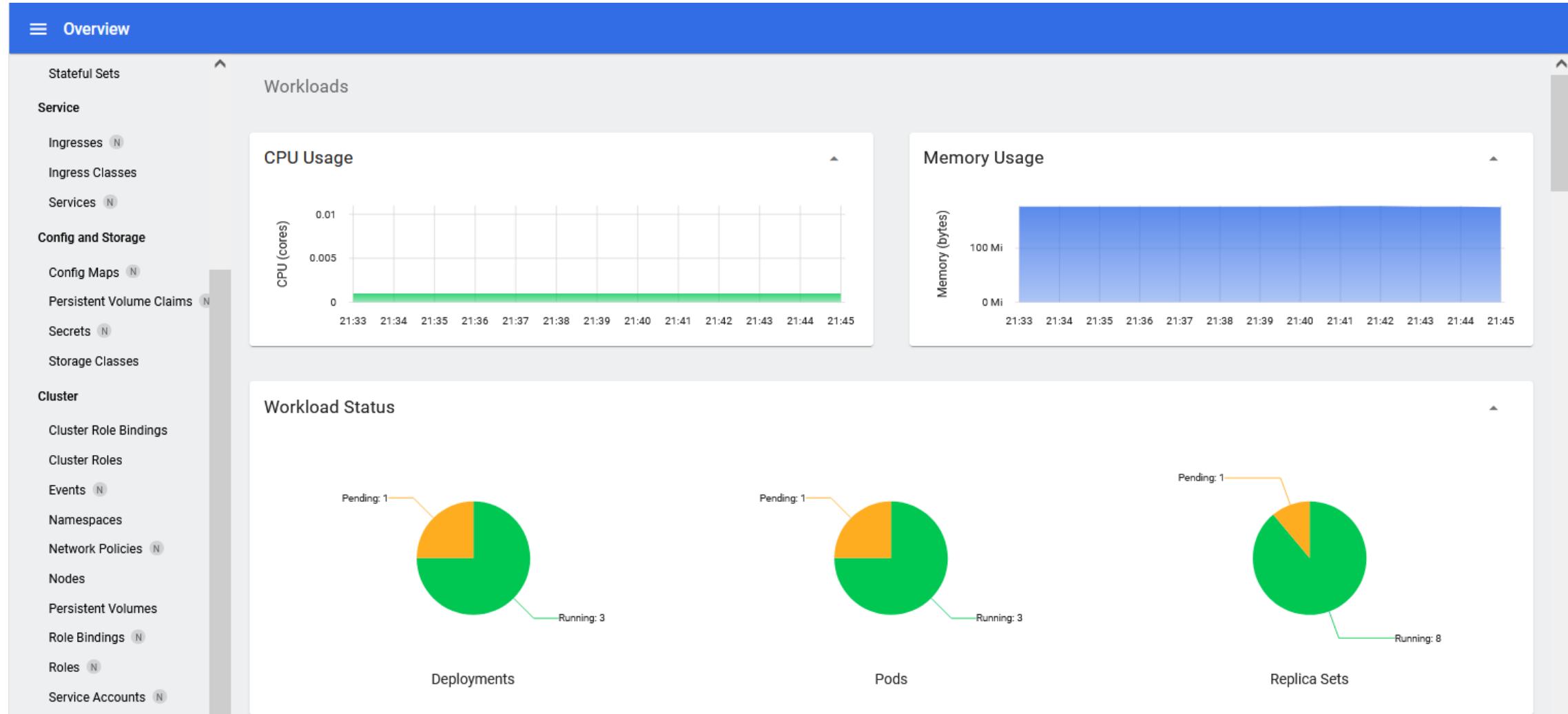
# DEPLOYMENT AND ORCHESTRATION OF FOG APPLICATIONS

INTERNATIONAL SUMMER SCHOOL ON  
INDUSTRIAL AGENTS 2024

STANDARDIZATION OF I4.0 SYSTEMS

## APPLICATION DEPLOYMENT

### Application Startup



# DEPLOYMENT AND ORCHESTRATION OF FOG APPLICATIONS

INTERNATIONAL SUMMER SCHOOL ON  
INDUSTRIAL AGENTS 2024

STANDARDIZATION OF 4.0 SYSTEMS

## APPLICATION DEPLOYMENT

### Application Running

The screenshot shows a Kubernetes dashboard interface. On the left, a sidebar lists various resources: Workloads, Cron Jobs, Daemon Sets, Deployments, Jobs, Pods, Replica Sets, Replication Controllers, Stateful Sets, Service, Ingresses, Ingress Classes, Services, Config and Storage, Config Maps, Persistent Volume Claims, Secrets, Storage Classes, Cluster, Cluster Role Bindings, Cluster Roles, Events, Namespaces, Network Policies, and Nodes. The main area is titled "Workload Status" and displays three large green circles representing the count of running workloads: Deployments (6), Pods (6), and Replica Sets (11). Below this, a table titled "Deployments" lists three entries:

Name	Images	Labels	Pods	Created
numberdisplay-myapp	ekhurtado/gcis-issia-24:number-display	applicationName: myapp microservice.name: numberdisplay resource.controller: microservice-controller Show all	1 / 1	5 minutes ago
numberprocessor-1-myapp	ekhurtado/gcis-issia-24:number-processor	applicationName: myapp microservice.name: numberprocessor-1 resource.controller: microservice-controller Show all	1 / 1	5 minutes ago
numbergeneratorbutton-myapp	ekhurtado/gcis-issia-24:number-generator	applicationName: myapp microservice.name: numbergeneratorbutton resource.controller: microservice-controller Show all	1 / 1	5 minutes ago

# DEPLOYMENT AND ORCHESTRATION OF FOG APPLICATIONS

INTERNATIONAL SUMMER SCHOOL ON  
INDUSTRIAL AGENTS 2024

STANDARDIZATION OF I4.0 SYSTEMS

## APPLICATION DEPLOYMENT

### Application Events

Events								
Name	Reason	Message	Source	Sub-object	Count	First Seen	Last Seen ↑	
numbergeneratorbutton-myapp-deployed-dpyb8	Running	The microservice has been extended correctly.	numbergeneratorbutton-myapp	Events	0	9 minutes ago	9 minutes ago	
numbergeneratorbutton-myapp-deploying-tf9mi	Deploying	Microservice deployment started.	numbergeneratorbutton-myapp	Events	0	9 minutes ago	9 minutes ago	
numbergeneratorbutton-myapp.17db6b4dac74b49f	ScalingReplicaSet	Scaled up replica set numbergeneratorbutton-myapp-7654dd97c6 to 1	deployment-controller	-	1	9 minutes ago	9 minutes ago	
numbergeneratorbutton-myapp-Created-zueou	Created	The new microservice has been successfully created.	numbergeneratorbutton-myapp	Events	0	9 minutes ago	9 minutes ago	

# DEPLOYMENT AND ORCHESTRATION OF FOG APPLICATIONS

INTERNATIONAL SUMMER SCHOOL ON  
INDUSTRIAL AGENTS 2024

STANDARDIZATION OF I4.0 SYSTEMS

## APPLICATION DEPLOYMENT

### Microservice Events

Events							
Name	Reason	Message	Source	Sub-object	Count	First Seen	Last Seen ↑
numberdisplay-myapp-deployed-0vvpe	Running	The microservice has been extended correctly.	numberdisplay-myapp	Events	0	11 minutes ago	11 minutes ago
numberdisplay-myapp-Created-mov9	Created	The new microservice has been successfully created.	numberdisplay-myapp	Events	0	11 minutes ago	11 minutes ago
numberdisplay-myapp-deploying-mhpca	Deploying	Microservice deployment started.	numberdisplay-myapp	Events	0	11 minutes ago	11 minutes ago
numberdisplay-myapp.17db6b4f95bc3b48	ScalingReplicaSet	Scaled up replica set numberdisplay-myapp-656d6d877d to 1	deployment-controller	-	1	11 minutes ago	11 minutes ago

# DEPLOYMENT AND ORCHESTRATION OF FOG APPLICATIONS

## APPLICATION DEPLOYMENT

*From Ubuntu terminal*

`kubectl get pods -o wide -w`

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED NODE	READINESS GATES
nodered-c5896c8f6-kjw65	1/1	Running	7 (63m ago)	3d11h	10.42.0.89	cluster	<none>	<none>
microservice-controller-8494b7dd8-jnvbb	1/1	Running	10 (17m ago)	3d10h	10.42.2.12	node2	<none>	<none>
application-controller-857bf46bdf-4qmvv	1/1	Running	0	3m24s	10.42.0.93	cluster	<none>	<none>
numbergeneratorbutton-myapp-7654dd97c6-cp47h	0/1	Pending	0	0s	<none>	<none>	<none>	<none>
numbergeneratorbutton-myapp-7654dd97c6-cp47h	0/1	Pending	0	1s	<none>	node1	<none>	<none>
numbergeneratorbutton-myapp-7654dd97c6-cp47h	0/1	ContainerCreating	0	1s	<none>	node1	<none>	<none>
numbergeneratorbutton-myapp-7654dd97c6-cp47h	1/1	Running	0	3s	10.42.1.19	node1	<none>	<none>
numberprocessor-1-myapp-7cf557bccc-w64xr	0/1	Pending	0	0s	<none>	<none>	<none>	<none>
numberprocessor-1-myapp-7cf557bccc-w64xr	0/1	Pending	0	0s	<none>	node1	<none>	<none>
numberprocessor-1-myapp-7cf557bccc-w64xr	0/1	ContainerCreating	0	0s	<none>	node1	<none>	<none>
numberprocessor-1-myapp-7cf557bccc-w64xr	1/1	Running	0	2s	10.42.1.20	node1	<none>	<none>
numberdisplay-myapp-656d6d877d-v69gq	0/1	Pending	0	0s	<none>	<none>	<none>	<none>
numberdisplay-myapp-656d6d877d-v69gq	0/1	Pending	0	0s	<none>	node2	<none>	<none>
numberdisplay-myapp-656d6d877d-v69gq	0/1	ContainerCreating	0	0s	<none>	node2	<none>	<none>
numberdisplay-myapp-656d6d877d-v69gq	1/1	Running	0	3s	10.42.2.21	node2	<none>	<none>

# DEPLOYMENT AND ORCHESTRATION OF FOG APPLICATIONS

## APPLICATION DEPLOYMENT

- 5 Check the application is correctly running

The screenshot shows the Kubernetes UI interface. The top navigation bar includes the Kubernetes logo, a dropdown for the namespace (set to 'default'), a search bar, and a red button labeled 'View Logs'. The main menu on the left has options: Workloads, Cron Jobs, Daemon Sets, Deployments, Jobs, **Pods** (highlighted with a red box), Replica Sets, and Replication Controllers. The current view is under 'Pods' for the 'numberdisplay-myapp-656d6d877d-ghgv' pod. A red box highlights the pod name in the breadcrumb. The interface displays two charts: 'CPU Usage' and 'Memory Usage'. Below the charts is a detailed log viewer with a blue header. The log content is as follows:

```
-> Resulting value: 698
-----
-----
New data:
-> Data time: 2024/06/22 20:05:47
-> Data type: integer
-> Resulting value: -282
-----
-----
New data:
-> Data time: 2024/06/22 20:05:52
-> Data type: integer
-> Resulting value: 999
-----
```

A yellow arrow points from the bottom right towards the log viewer. On the far right, there is a context menu with checkboxes for: Invert colors, Reduce font size, Show timestamps, Auto-refresh (every 5 s.) (which is checked), and Show previous logs.

## TABLE OF CONTENTS

### SECTION III



- Introduction
- Kubernetes Extension Mechanisms
- Application Deployment
- Application Orchestration
  - Node failure
- More complex application

## APPLICATION ORCHESTRATION

### NODE FAILURE

#### 1 Initial situation

Pods				
	Name	Images	Labels	Status
●	numberdisplay-myapp-656d6d877d-v69gq	ekhurtado/gcis-issia-24:number-display	pod-template-hash: 656d6d87 7d resource.name: numberdisplay-myapp	node2 Running
●	numberprocessor-1-myapp-7cf557bccc-w64xr	ekhurtado/gcis-issia-24:number-processor	pod-template-hash: 7cf557bc cc resource.name: numberprocessor-1-myapp	node1 Running
●	numbregeneratorbutton-myapp-7654dd97c6-cp47h	ekhurtado/gcis-issia-24:number-generator	pod-template-hash: 7654dd97 c6 resource.name: numbregeneratorbutton-myapp	node1 Running

# DEPLOYMENT AND ORCHESTRATION OF FOG APPLICATIONS

## APPLICATION ORCHESTRATION

### NODE FAILURE

- 2 Simulate node failure

```
multipass shell node1
```

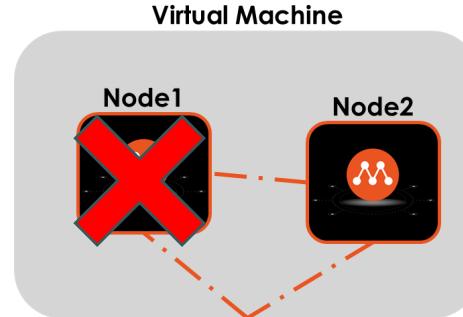
```
sudo systemctl stop k3s-agent.service
```

- 3 Exit multipass node1 and check if the node has fallen down

```
exit
```

```
Kubectl get nodes -o wide
```

```
ubuntu@node1:~$ sudo systemctl stop k3s-agent.service
ubuntu@node1:~$ exit
logout
issia@cluster:~/KubernetesExtension$ kubectl get nodes -o wide
NAME      STATUS    ROLES          AGE     VERSION   INTERNAL-IP    EXTERNAL-IP  OS-IMAGE       KERNEL-VERSION   CONTAINER-RUNTIME
node2     Ready     <none>        4d11h   v1.29.5+k3s1  10.128.11.17  <none>       Ubuntu 20.04.6 LTS  5.4.0-186-generic  containerd://1.7.15-k3s1
node1     Ready     <none>        4d11h   v1.29.5+k3s1  10.128.11.240   <none>       Ubuntu 20.04.6 LTS  5.4.0-186-generic  containerd://1.7.15-k3s1
cluster   Ready     control-plane,master 4d11h   v1.29.5+k3s1  192.168.237.130  <none>       Ubuntu 20.04.6 LTS  5.15.0-107-generic  containerd://1.7.15-k3s1
issia@cluster:~/KubernetesExtension$ kubectl get nodes -o wide
NAME      STATUS    ROLES          AGE     VERSION   INTERNAL-IP    EXTERNAL-IP  OS-IMAGE       KERNEL-VERSION   CONTAINER-RUNTIME
cluster   Ready     control-plane,master 4d11h   v1.29.5+k3s1  192.168.237.130  <none>       Ubuntu 20.04.6 LTS  5.15.0-107-generic  containerd://1.7.15-k3s1
node1     NotReady  <none>        4d11h   v1.29.5+k3s1  10.128.11.240   <none>       Ubuntu 20.04.6 LTS  5.4.0-186-generic  containerd://1.7.15-k3s1
node2     Ready     <none>        4d11h   v1.29.5+k3s1  10.128.11.17   <none>       Ubuntu 20.04.6 LTS  5.4.0-186-generic  containerd://1.7.15-k3s1
```



# DEPLOYMENT AND ORCHESTRATION OF FOG APPLICATIONS

## APPLICATION ORCHESTRATION

### NODE FAILURE

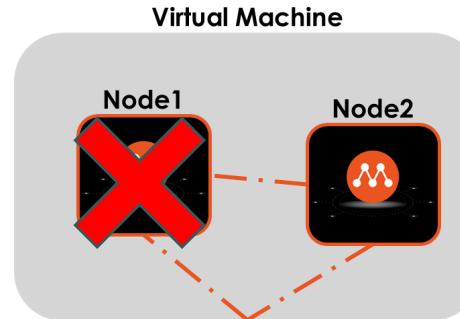
4 Kubernetes re-deploys affected pods

numbergeneratorbutton-myapp-7654dd97c6-cp47h	1/1	Running	0	16m	10.42.1.19	node1
numberprocessor-1-myapp-7cf557bccc-w64xr	1/1	Running	0	16m	10.42.1.20	node1
numbergeneratorbutton-myapp-7654dd97c6-cp47h	1/1	Running	0	21m	10.42.1.19	node1
numberprocessor-1-myapp-7cf557bccc-w64xr	1/1	Running	0	21m	10.42.1.20	node1
numbergeneratorbutton-myapp-7654dd97c6-cp47h	1/1	Terminating	0	21m	10.42.1.19	node1
numberprocessor-1-myapp-7cf557bccc-w64xr	1/1	Terminating	0	21m	10.42.1.20	node1
numbergeneratorbutton-myapp-7654dd97c6-gln28	0/1	Pending	0	0s	<none>	<none>
numbergeneratorbutton-myapp-7654dd97c6-gln28	0/1	Pending	0	0s	<none>	<none>
numberprocessor-1-myapp-7cf557bccc-wthh8	0/1	Pending	0	0s	<none>	node2
numberprocessor-1-myapp-7cf557bccc-wthh8	0/1	Pending	0	0s	<none>	node2
numbergeneratorbutton-myapp-7654dd97c6-gln28	0/1	ContainerCreating	0	0s	<none>	node2
numbergeneratorbutton-myapp-7654dd97c6-gln28	0/1	ContainerCreating	0	0s	<none>	node2
numbergeneratorbutton-myapp-7654dd97c6-gln28	1/1	Running	0	25s	10.42.2.22	node2
numberprocessor-1-myapp-7cf557bccc-wthh8	1/1	Running	0	25s	10.42.2.23	node2

The screenshot shows the Kubernetes dashboard interface. On the left, a sidebar lists various cluster components: Workloads, Services, Config and Storage, Cluster, and Network Policies. The main area displays three pie charts under the 'Workload Status' section: 'Deployments' (Failed: 2, Running: 4), 'Pods' (Failed: 2, Running: 5), and 'Replica Sets' (Failed: 2, Running: 9). Below these charts is a table titled 'Deployments' with columns for Name, Images, Labels, Pods, and Created. It lists three entries: 'numberdisplay-myapp' (Running, 1/1, 23 minutes ago), 'numberprocessor-1-myapp' (Failed, 2/1, 23 minutes ago), and 'numbergeneratorbutton-myapp' (Failed, 2/1, 23 minutes ago).



5 minutes!!



## APPLICATION ORCHESTRATION

### NODE FAILURE

- 5 Check the application is correctly running

The screenshot shows the Kubernetes dashboard interface. On the left, a sidebar menu lists Workloads, Cron Jobs, Daemon Sets, Deployments, Jobs, **Pods** (which is selected and highlighted with a red box), Replica Sets, and Replication Controllers. The main content area displays two charts: "CPU Usage" and "Memory Usage". The "CPU Usage" chart shows a single bar for a "Display" pod, with the text "Display" pod overlaid in red. The "Memory Usage" chart shows a blue line graph fluctuating around 40 Mi. A red box highlights the URL in the browser header: "Workloads > Pods > numberdisplay-myapp-656d6d877d-v69gq". A red arrow points to the "Logs" button in the top right corner of the main dashboard. A large yellow arrow points from the "Logs" button to a modal window titled "Logs from numberdisplay-myapp-656d6d877d-v69gq in numberdisplay-...". The modal contains a log output window showing three entries of data processing:

```
-----  
New data:  
-> Data time: 2024/06/22 23:35:24  
-> Data type: integer  
-> Resulting value: 406  
-----  
New data:  
-> Data time: 2024/06/22 23:35:28  
-> Data type: integer  
-> Resulting value: -388  
-----  
New data:  
-> Data time: 2024/06/22 23:35:29  
-> Data type: integer
```

## TABLE OF CONTENTS

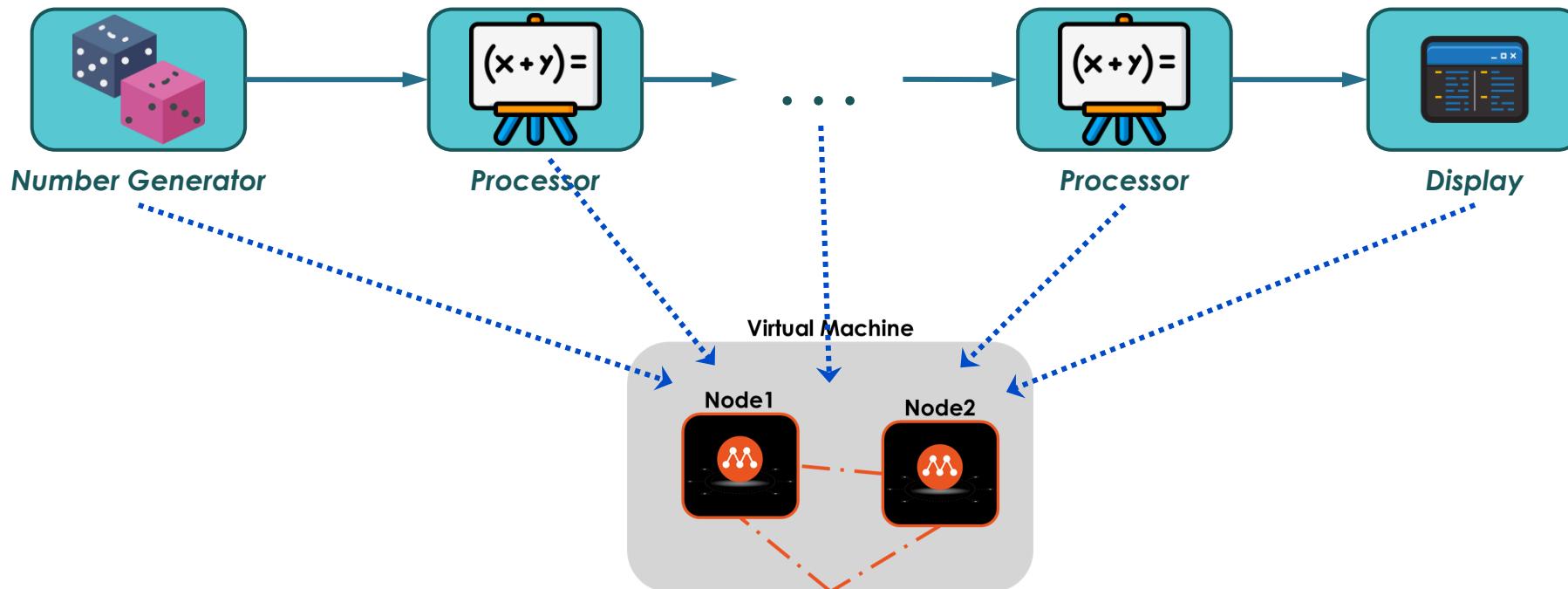
### SECTION III



- Introduction
- Kubernetes Extension Mechanisms
- Application Deployment
- Application Orchestration
  - Node failure
- More complex application

# DEPLOYMENT AND ORCHESTRATION OF FOG APPLICATIONS

## MORE COMPLEX APPLICATION



# GENERATION OF FOG COMPONENTS

## TABLE OF CONTENTS

### SECTION IV



# GENERATION OF FOG COMPONENTS

- 1 Write the component code ("customComponentCode.py" in SectionIV directory)

```
1 import json
2 import math
3 import os, requests
4 from threading import Thread
5
6 from flask import Flask, request
7
8 function = os.environ.get('SERVICE')
9 if function == "SquareRoot": import square_root
10
11 output = os.environ.get('OUTPUT')
12 output_port = os.environ.get('OUTPUT_PORT')
13
14 app = Flask(__name__)
15
16
17 @app.route('/', methods=['GET', 'POST'])
18 def main():
19     print(request)
20
21     # First, we'll get the data from the HTTP message
22     messageData = None
23     if request.method == 'GET':
24         # Handle GET requests
25         return "You submitted a GET method!"
26     elif request.method == 'POST':
27         # Handle POST requests
28         if "text/plain" in request.headers.get("Content-Type"):
29             messageData = request.get_data(as_text=True)
30
31     thread_func = None # We use the new thread to send the "OK" message to the previous component as soon as possible, without blocking this component.
32     match function:
33         case "SquareRoot":
34             thread_func = Thread(target=square_root, args=(messageData,))
35             # TODO: Place your code here
36         case "YOURFUNCTIONNAME":
37             thread_func = Thread(target=YOURFUNCTIONMETHOD, args=(messageData,)) # TODO: Change your functionality method name here
38         case _:
39             return "No function selected"
40     # The selected functionality is started in a new thread of execution
41     thread_func.start()
42     # Response to the previous component saying that everything went well
43     return "OK\n"
```

## Environment variable

Data reception

Example and instructions  
to customize

# GENERATION OF FOG COMPONENTS

- 1 Write the component code ("customComponentCode.py" in SectionIV directory)

```
74 def YOURFUNCTIONMETHOD(messageData): # TODO: Change your functionality method name here
75     # Get the value from the data sent in the HTTP message
76     jsonData = json.loads(messageData)
77     type = jsonData['type']
78     value = jsonData['value']
79
80     # TODO: Change it in case you want to use a custom variable
81     step = int(os.environ.get('CUSTOM_VARIABLE'))
82
83     # If the entered step is of a different type, the result may be different, so we will make sure
84     match type:
85         case "natural" | "integer":
86             value = int(value)
87         case "float":
88             value = float(value)
89         case _:
90             pass
91
92     # Perform the mathematical function
93     # TODO: Place your code here
94     result = None
95
96     # Having the result, it will be passed to the next component
97     sendData(type, result)←
```

*Data sending*

# GENERATION OF FOG COMPONENTS

- 2 Create the Docker image (the Dockerfile is available in SectionIV directory)

```
issia@cluster:~/SectionIV$ sudo docker build -t aarmentia/custom-comp:latest .
[+] Building 7.9s (10/10) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 199B
=> [internal] load metadata for docker.io/library/python:3.10-alpine
=> [auth] library/python:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load build context
=> => transferring context: 3.58kB
=> [1/4] FROM docker.io/library/python:3.10-alpine@sha256:69086c7682396e8f5a41e1e0bf12a3959600ef213380c22c21dd99ff488c698a
=> => resolve docker.io/library/python:3.10-alpine@sha256:69086c7682396e8f5a41e1e0bf12a3959600ef213380c22c21dd99ff488c698a
=> => sha256:2f8987cab5d3db1edb807b73a0e1d73354bb2f0d530467a8264f17793088b81 1.37kB / 1.37kB
=> => sha256:1ffbd78f3826e1be410b908226e2b7e8d898d42237dce1563b10b507050ab2a5 6.92kB / 6.92kB
=> => sha256:ec99f8b99825a742d50fb3ce173d291378a46ab54b8ef7dd75e5654e2a296e99 3.62MB / 3.62MB
=> => sha256:a68bf89b0030f704607f0fb77455bc17e2b3d817d5132d0315facff87d5a2d0c 463.14kB / 463.14kB
=> => sha256:5dc785ae77095338bd26aeb9b486c1cb456670bc4244b15dcbb0db8ddb41f255 12.22MB / 12.22MB
=> => sha256:69086c7682396e8f5a41e1e0bf12a3959600ef213380c22c21dd99ff488c698a 1.86kB / 1.86kB
=> => extracting sha256:ec99f8b99825a742d50fb3ce173d291378a46ab54b8ef7dd75e5654e2a296e99
=> => sha256:3da18bd7e738b12ca47370f71dd7c87996551981a7c29a8814058c386c5b6392 242B / 242B
=> => extracting sha256:a68bf89b0030f704607f0fb77455bc17e2b3d817d5132d0315facff87d5a2d0c
=> => sha256:a21274535ff3c6fff69ca090c5c8ffb0696a8ad9b8c8ced7920294ecba00dc54 3.08MB / 3.08MB
=> => extracting sha256:5dc785ae77095338bd26aeb9b486c1cb456670bc4244b15dcbb0db8ddb41f255
=> => extracting sha256:3da18bd7e738b12ca47370f71dd7c87996551981a7c29a8814058c386c5b6392
=> => extracting sha256:a21274535ff3c6fff69ca090c5c8ffb0696a8ad9b8c8ced7920294ecba00dc54
=> [2/4] RUN pip install Flask
=> [3/4] RUN pip install requests
=> [4/4] COPY customComponentCode.py /
=> exporting to image
=> => exporting layers
=> => writing image sha256:a45633047c0fd090394ee627262fdeb57759abf8a77df7502153d1bc1e40abc4
=> => naming to docker.io/aarmentia/custom-comp:latest
```

# GENERATION OF FOG COMPONENTS

## 3 Store the Docker image in DockerHub

```
issia@cluster:~/SectionIV$ sudo docker push aarmentia/custom-comp:latest
The push refers to repository [docker.io/aarmentia/custom-comp]
09a12d9eacf6: Mounted from aarmentia/issia2024
b9d5c3467cf8: Pushed
9147dcd3187e: Pushed
e8982d30e60b: Mounted from library/python
858b3aa272e1: Mounted from library/python
9e763289d00e: Mounted from library/python
699ede46ccf2: Mounted from library/python
94e5f06ff8e3: Mounted from library/python
latest: digest: sha256:1f12af6e0c80e10790a007312cf07d345177a91aa9782ca63286510687a3a483 size: 1998
```



# GENERATION OF FOG COMPONENTS

## 4 Design the Fog Application

The screenshot illustrates the process of generating a fog component. It shows two main windows: a configuration dialog for a 'customComponent' node and a Node-RED application editor.

**Configuration Dialog:** The top window is titled 'Edit customComponent node'. It contains three fields:

- Properties:** Add the component name: CustomComp
- Docker image:** aarmentia/custom-comp:latest
- Service:** AintzaneFunction

**Node-RED Application Editor:** The bottom window shows a flow diagram for 'MyCustomApp'. It includes nodes for 'IntegerNumbers', 'CustomComponent', and 'ConsoleDisplay'. A yellow arrow points from the configuration dialog to the 'CustomComponent' node in the Node-RED editor. Another yellow arrow points from the Node-RED editor to the generated XML code in the bottom right panel.

**Generated XML Code:**

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<application name="MyCustomApp">
    <microservice name="NumberGeneratorButton" service="IntegerNumbers" imgBase="ekhurtado/gcis-issia-24:number-generator" customization="{"custom_type": 'random', "custom_initialvalue": 0}">
        <outPort name="NumberGeneratorButtonPort" protocol="HTTP" dataType="TNumber"/>
    </microservice>
    <microservice name="CustomComp" service="AintzaneFunction" imgBase="aarmentia/custom-comp:latest">
        <inPort name="CustomCompiPort" protocol="HTTP" dataType="TZenbaki" number="7000"/>
        <outPort name="CustomCompoPort" protocol="HTTP" dataType="TZenbaki"/>
    </microservice>
    <microservice name="NumberDisplay" service="ConsoleDisplay" imgBase="ekhurtado/gcis-issia-24:number-display">
        <inPort name="NumberDisplayIPort" protocol="HTTP" dataType="TNumber" number="7000"/>
    </microservice>
    <channel from="NumberGeneratorButtonPort" to="CustomCompiPort"/>
    <channel from="CustomCompoPort" to="NumberDisplayIPort"/>
</application>
```

# GENERATION OF FOG COMPONENTS

5

Create the Application Delivery Model and deploy the application

Activities Text Editor jun 2 mycustomapp ~/\$

Open +

mycustomapp.yaml

```
1 apiVersion: ehu.gcis.org/v1alpha1
2 kind: Application
3 metadata:
4   name: mycustomapp
5 spec:
6   microservices:
7     - name: numbergeneratorbutton
8       service: IntegerNumbers
9       image: ekhurtado/gcis-issia-24:number-generator
10      customization: '{"custom_type": "random", "custom_initialvalue": 0}'
11      outPort:
12        name: numbergeneratorbutton-oport
13        dataType: TNumber
14        protocol: HTTP
15      - name: customcomp
16        service: AintzaneFunction
17        image: amenteria/custom-comp:latest
18        inPort:
19          name: customcomp-iport
20          dataType: TZenkaki
21          protocol: HTTP
22          number: '7000'
23        outPort:
24          name: customcomp-oport
25          dataType: TZenkaki
26          protocol: HTTP
27      - name: numberdisplay
28        service: ConsoleDisplay
29        image: ekhurtado/gcis-issia-24:number-display
30        inPort:
31          name: numberdisplay-iport
32          dataType: TNumber
33          protocol: HTTP
34          number: '7000'
35      channels:
36        - from: numbergeneratorbutton-oport
37          to: customcomp-iport
38        - from: customcomp-oport
39          to: numberdisplay-iport|
```

kubernetes default Search

Workloads

Workload Status

Cron Jobs

Daemon Sets

Deployments

Jobs

Pods

Replica Sets

Replication Controllers

Stateful Sets

Service

Ingresses

Ingress Classes

Services

Config and Storage

Config Maps

Persistent Volume Claims

Deployments

Pods

Replica Sets

Running: 9

Running: 9

Running: 14

Deployments

Pods

Replica Sets

Running: 9

Running: 9

Running: 14

Deployment

Name	Images	Labels	Pods	Created
numberdisplay-mycustomapp	ekhurtado/gcis-issia-24:number-display	applicationName: mycustomapp microservice.name: numberdisplay resource_controller: microservice_controller	1 / 1	4 minute

# GENERATION OF FOG COMPONENTS

## 6 Confirm the application is running properly

Pods			
Name	Images	Labels	Node
numberdisplay-mycustomapp-86d6f9c8b6-r72dn	ekhurtado/gcis-issia-24:number-display	pod-template-hash: 86d6f9c8b6 resource.name: numberdisplay-mycustomapp	node2
customcomp-mycustomapp-598d544748-2c27	aarmenia/custom-comp:latest	pod-template-hash: 598d544748 resource.name: customcomp-mycustomapp	node2
numbregeneratorbutton-mycustomapp-7fb97f9954-lgfqk	ekhurtado/gcis-issia-24:number-generator	pod-template-hash: 7fb97f9954 resource.name: numbregeneratorbutton-mycustomapp	node2
numbregeneratorbutton-myapp-7654dd97c6-gln28	ekhurtado/gcis-issia-24:number-generator	pod-template-hash: 7654dd97c6 resource.name: numbregeneratorbutton-myapp	node2
numberprocessor-1-myapp-7cf557bcc-wthh8	ekhurtado/gcis-issia-24:number-processor	pod-template-hash: 7cf557bcc resource.name: numberprocessor-1-myapp	node2
numberdisplay-myapp-656d6d877d-v69gq	ekhurtado/gcis-issia-24:number-display	pod-template-hash: 656d6d877d resource.name: numberdisplay-myapp	node2
numberprocessor-1-myapp-7cf557bcc-w64xr	ekhurtado/gcis-issia-24:number-processor	pod-template-hash: 7cf557bcc resource.name: numberprocessor-1-myapp	node1
numbregeneratorbutton-myapp-7654dd97c6-cp47h	ekhurtado/gcis-issia-24:number-generator	pod-template-hash: 7654dd97c6 resource.name: numbregeneratorbutton-myapp	node1

Workloads > Pods > numberdisplay-mycustomapp-86d6f9c8b6-r72dn > Logs

Logs from numberdisplay-... in numberdisplay-...

```
-----  
-----  
New data:  
-> Data time: 2024/06/23 01:10:48  
-> Data type: integer  
-> Resulting value: -876467493  
-----  
-----  
New data:  
-> Data time: 2024/06/23 01:10:53  
-> Data type: integer  
-> Resulting value: -183250432  
-----  
-----  
New data:  
-> Data time: 2024/06/23 01:10:58  
-> Data type: integer  
-> Resulting value: 531441  
-----
```



## INTERNATIONAL SUMMER SCHOOL ON INDUSTRIAL AGENTS 2024

STANDARDIZATION OF I4.0 SYSTEMS

# Microservice-based Industrial Agents: Managing manufacturing applications through the Edge-Fog-Cloud continuum

Aintzane Armentia, Oskar Casquero, Julen Cuadra, Ekaitz Hurtado