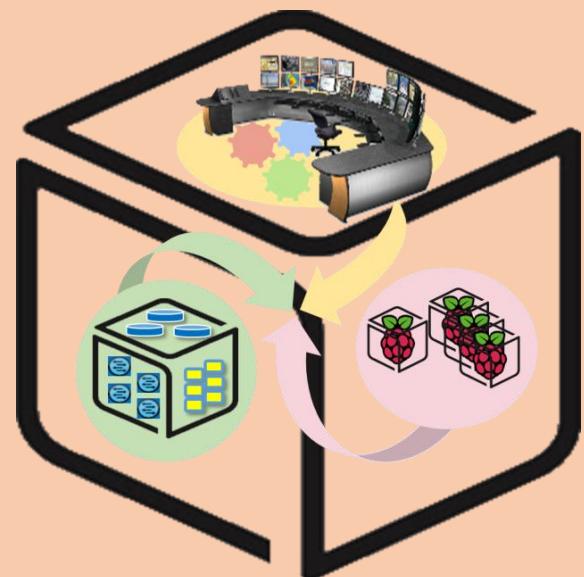


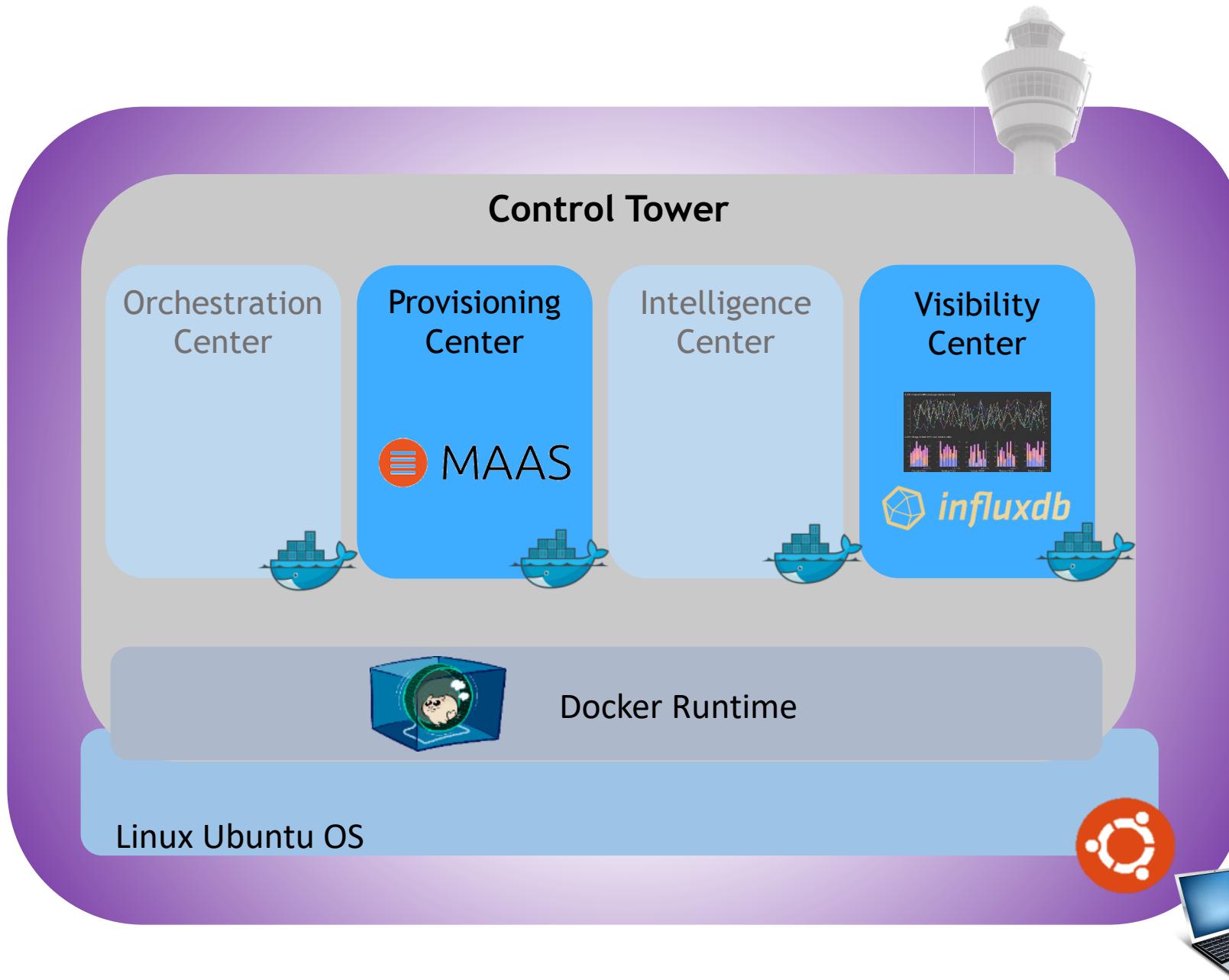
Computer Systems For AI-inspired Cloud Theory & Lab.

Lab #3: Tower



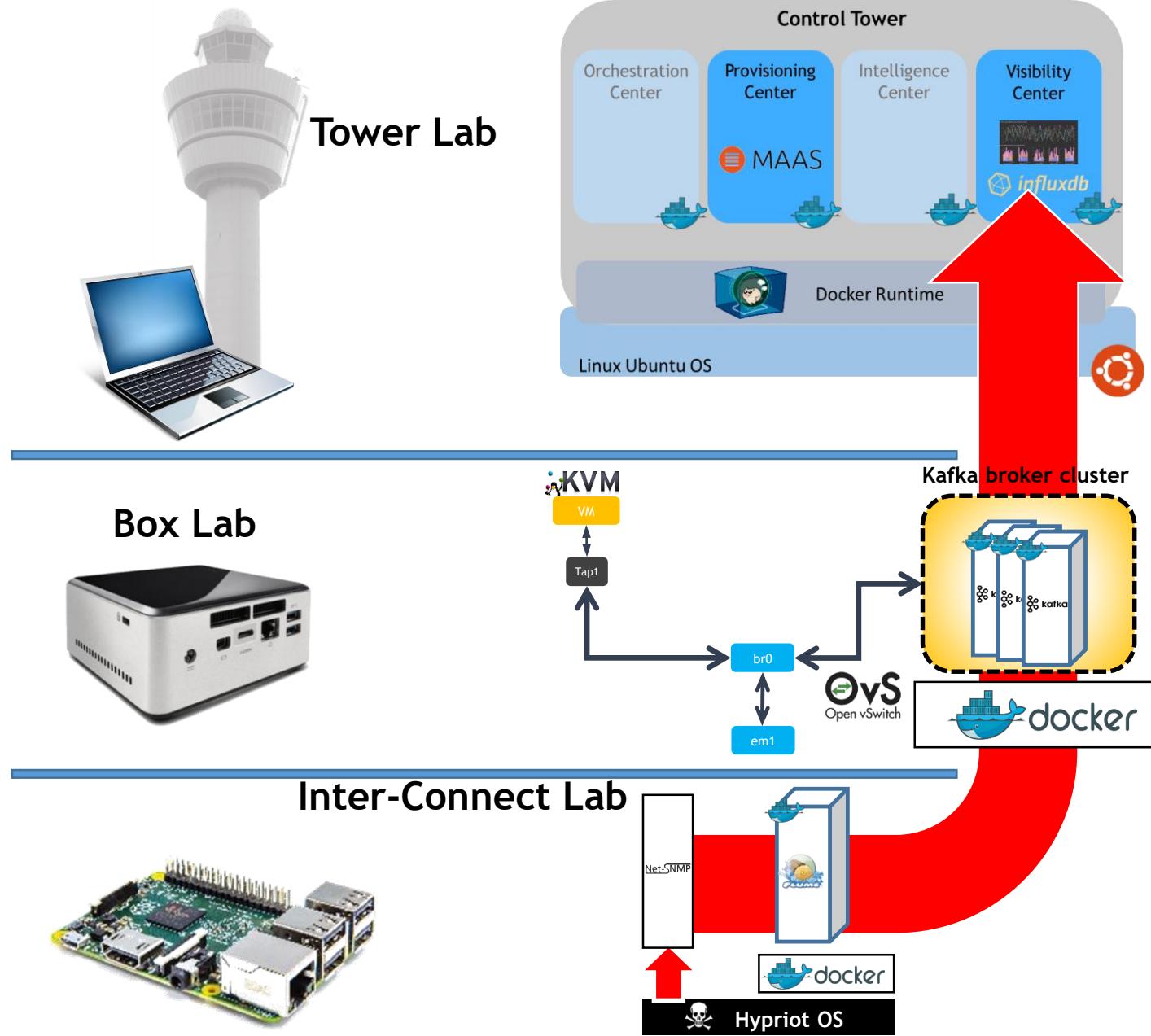
<https://github.com/SmartX-Labs/SmartX-Mini-MOOC>

Tower Lab: Concept



SmartX Labs #1~#3: Relationship

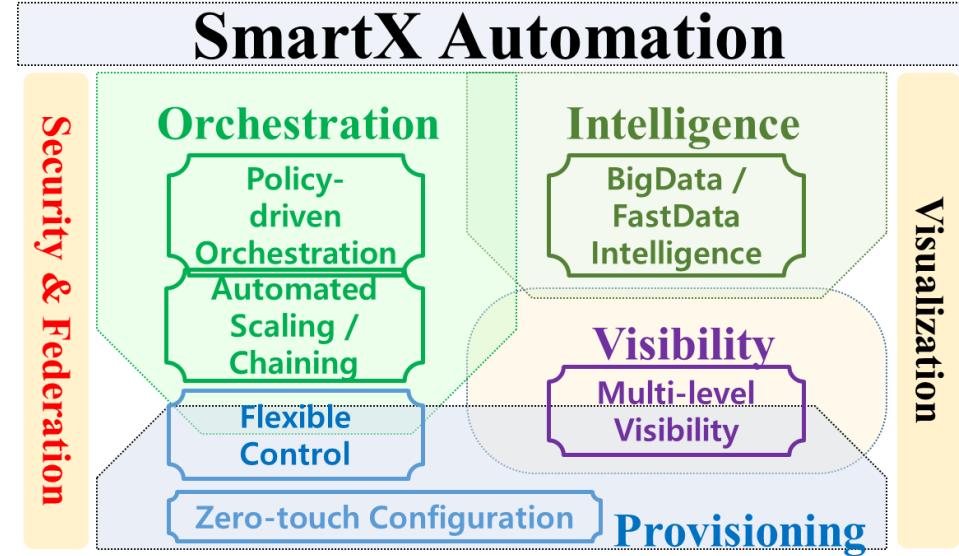
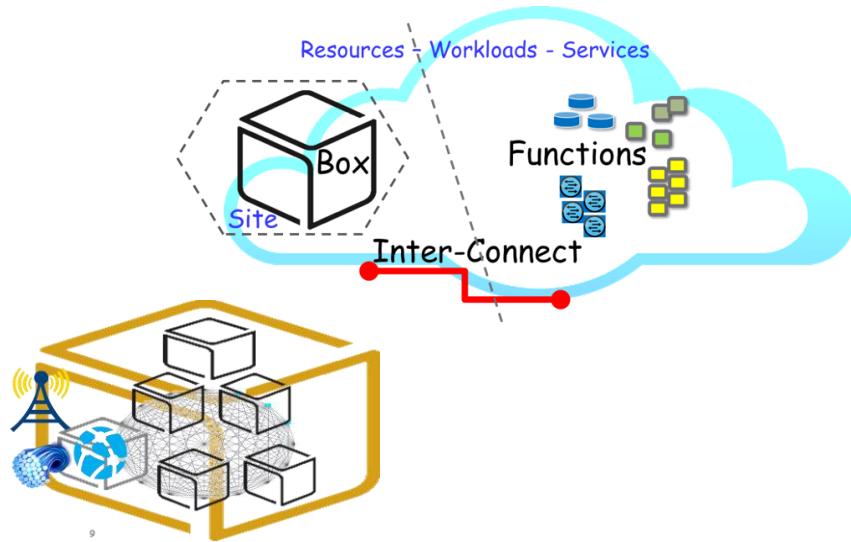
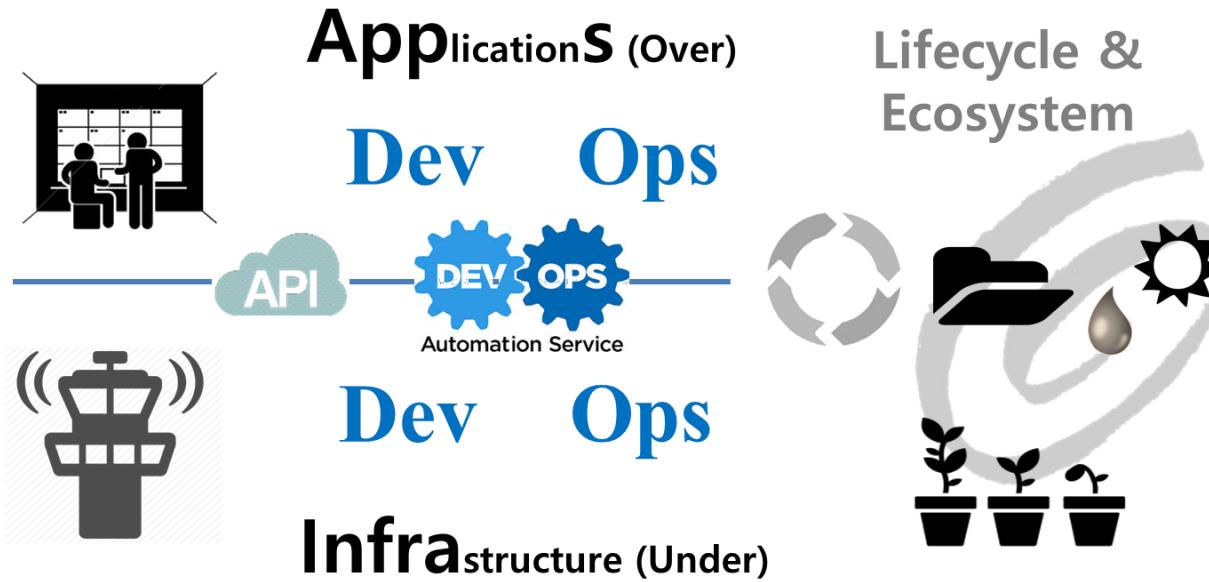
Lab #3: Tower 3



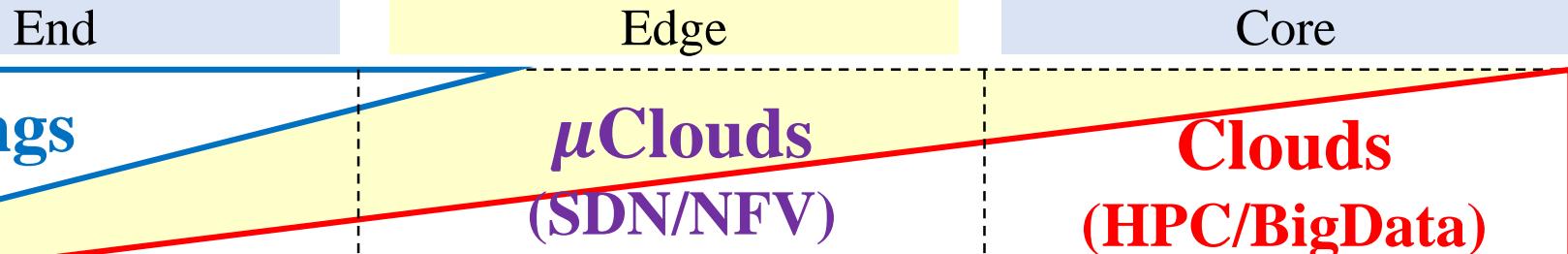
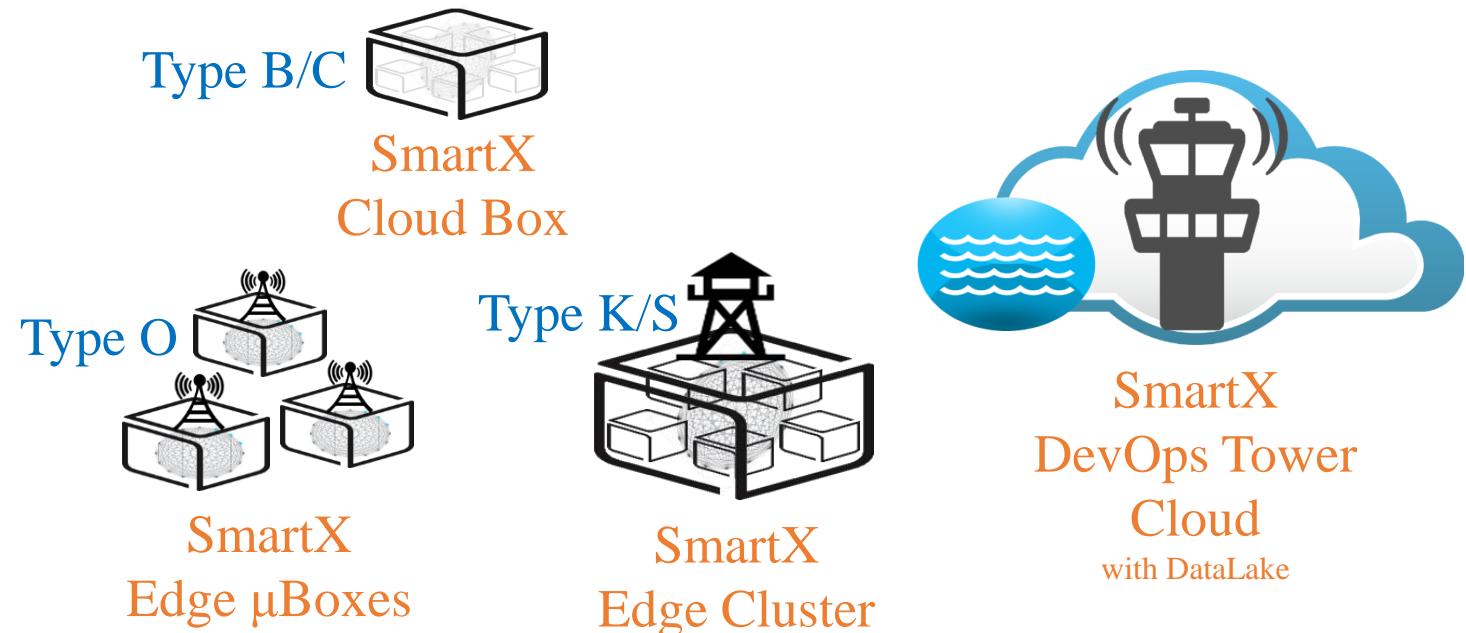
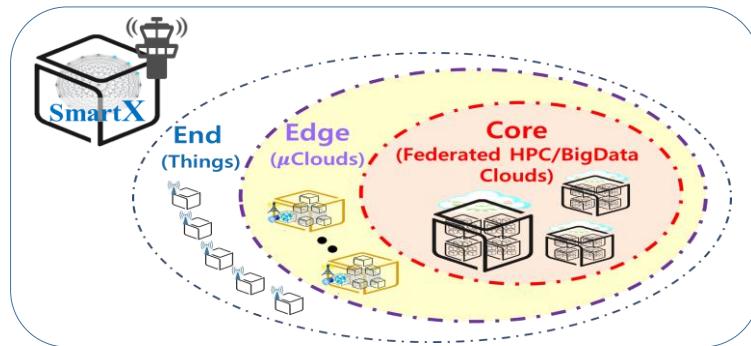
Theory



SmartX Automation Framework

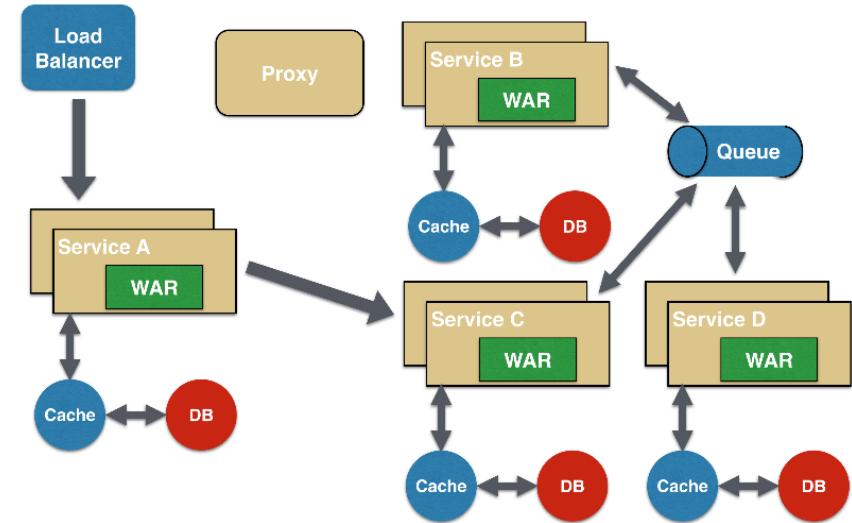
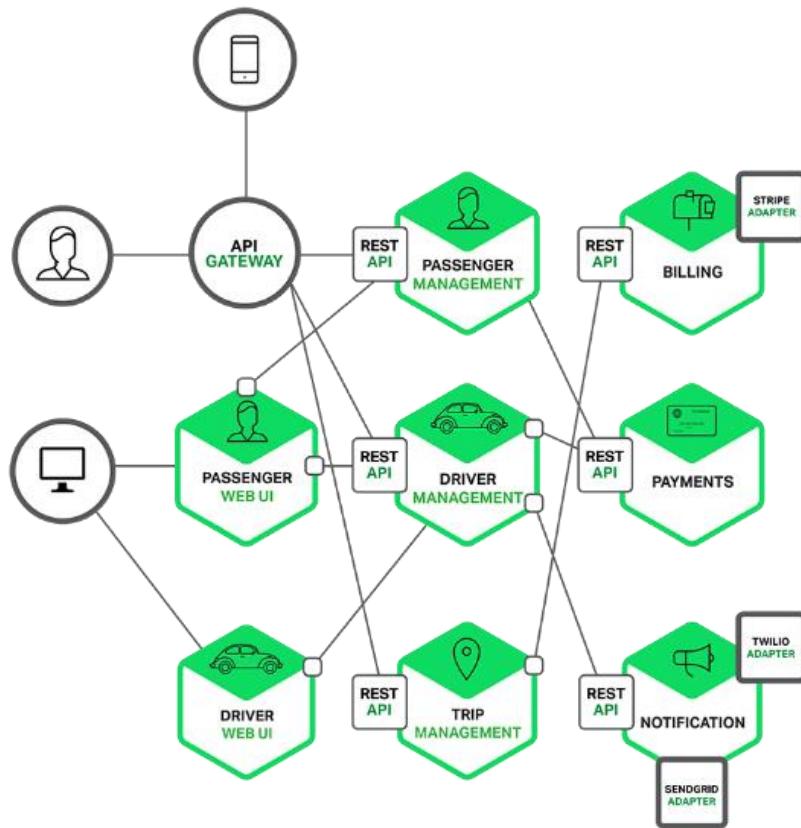


SmartX Composable Playground & Boxes



Container-based MSA (MicroServices Architecture)

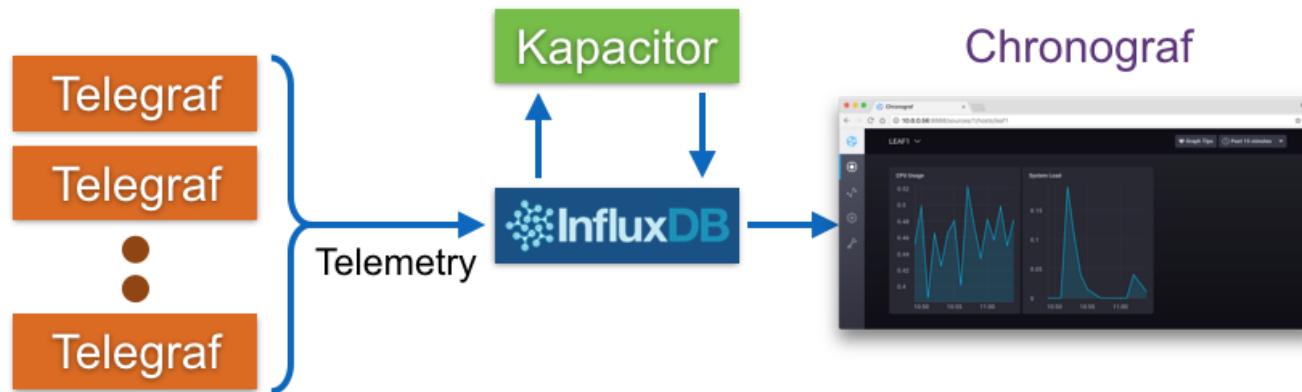
- Software development technique based on *Collection of loosely coupled small-size services (i.e., functions)*
- Fine-grained services and lightweight protocols to improve modularity, create applications easier, and helps resiliency against architecture erosion



Visibility: TSDB (Time Series Database)



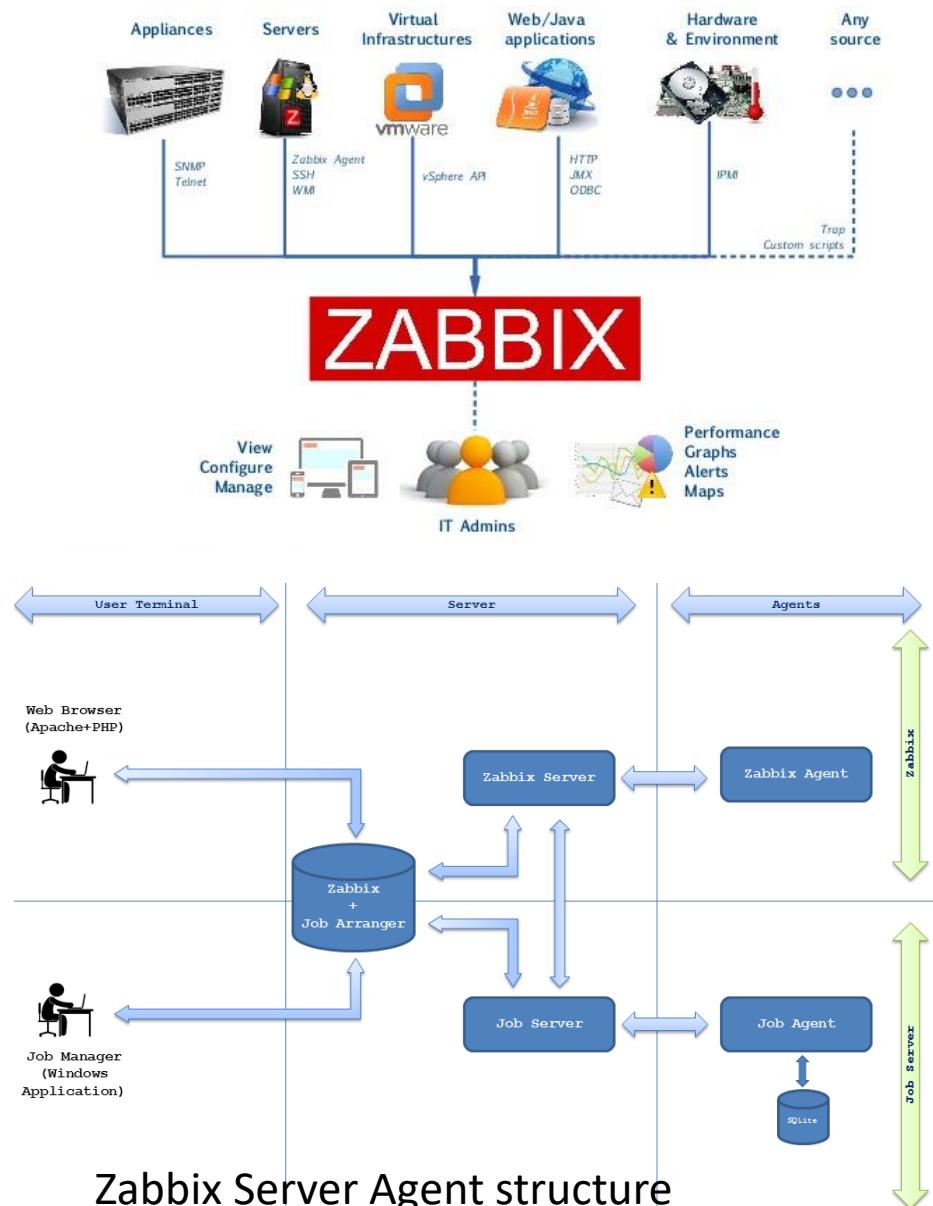
- Time series data is arrays of numbers indexed by time.
- In some fields these time series are called profiles, curves, or traces.



Visibility: Zabbix Distributed Monitoring

- Adopts a flexible notification mechanism
- User can configure & watch graph easily via Web GUI
- Consists of structured server and client
 - Client collects the monitoring data and send it to the Zabbix server
 - Server visualizes the data that is collected by the Zabbix Agent

ZABBIX



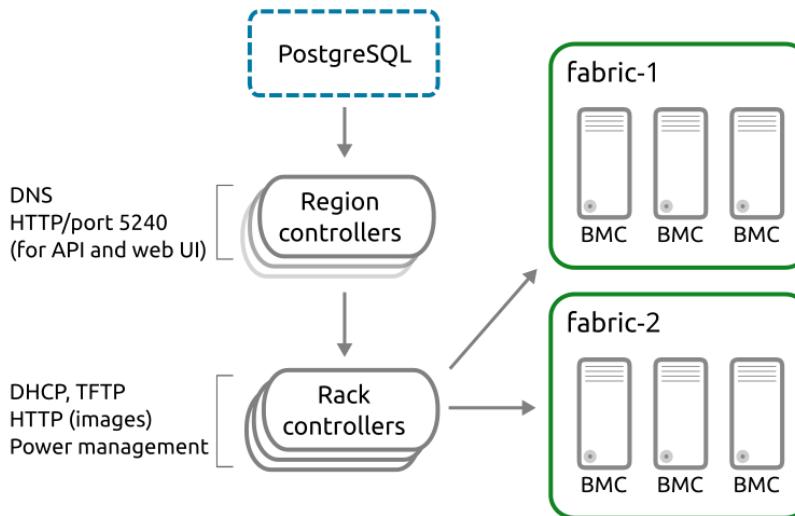
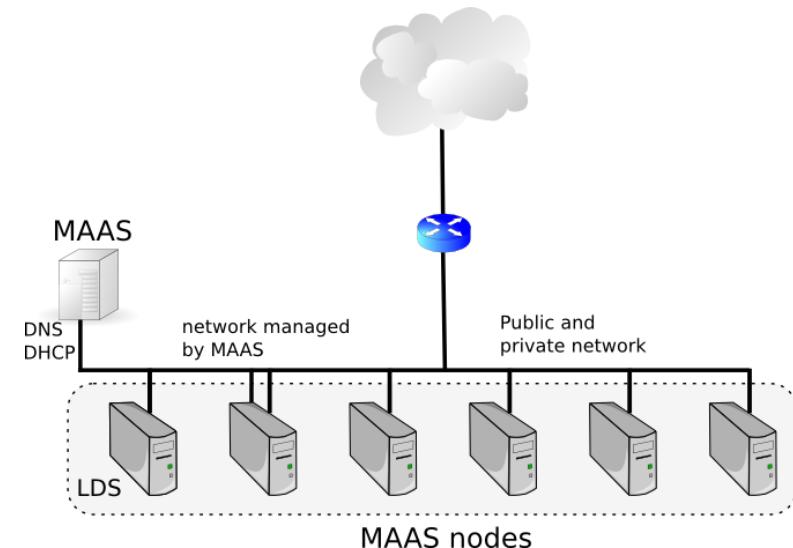


Provisioning: Ubuntu MAAS (Metal as a Service)

Lab #3: Tower 10



- Bear-metal machines can be quickly provisioned and destroyed; MAAS provides management of a large number of physical machines by creating a single resource pool
- MAAS can act as a standalone PXE services, provides Web GUI, supports various Linux distribution installation, ...



MAAS Controller Architecture

- Region Controller: Deals with operator requests
- Rack Controller: Provide the high bandwidth services to multiple server racks + Cache OS install images

Ubuntu MAAS: Automated Installation

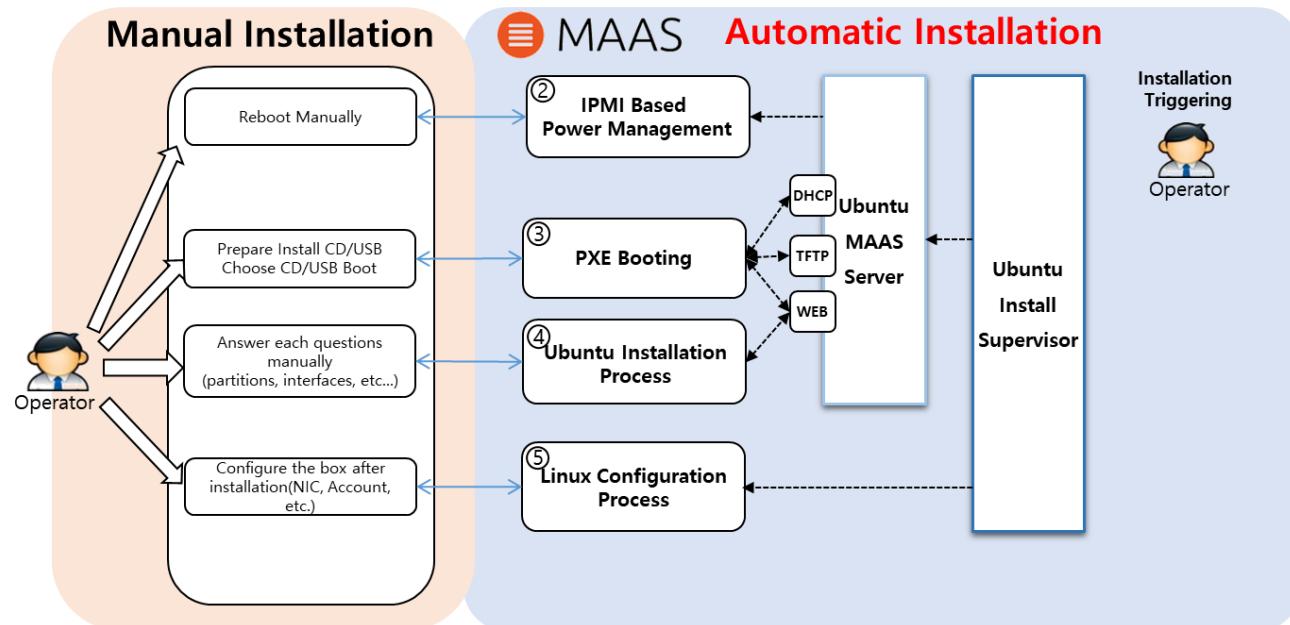
MAAS: Automated installation procedure

- Enlistment process to register a new machine
- Commissioning process to collect H/W information (CPU, RAM, Storage, etc.): All enlisted nodes must be accepted and finish this process in order to change state from 'declared' to 'ready' state
- Deployment process can happen with both Ubuntu juju and/or Ubuntu MAAS (WebUI)

Box Hardware Requirements for Automated Installation

- IPMI, Intel AMT, IBM HMC, ...
- PXE bootable with DHCP option
 - Two Ethernet interfaces

Warning!



Practice



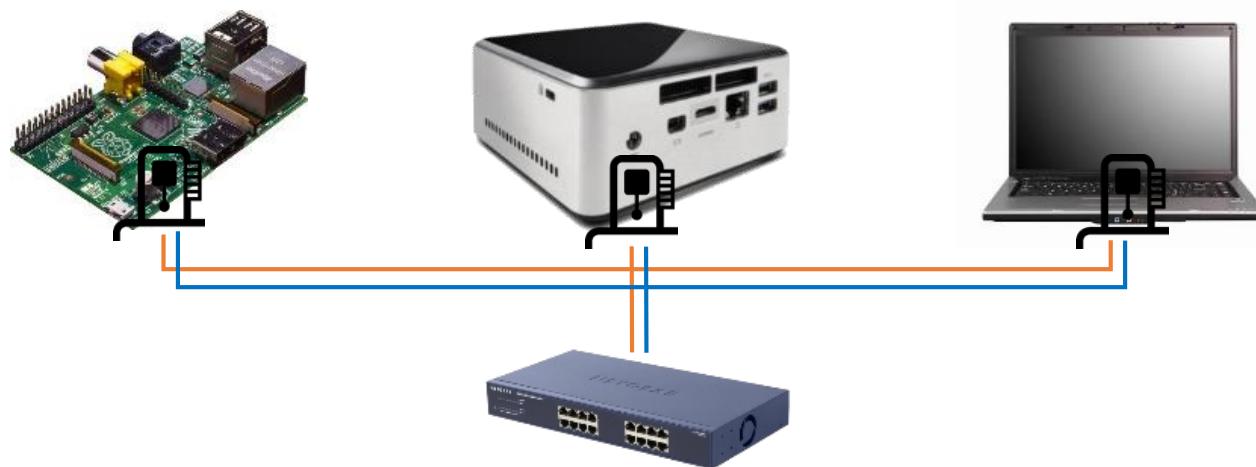
#0 - Lab Preparation (1/2)

Wired connection

NAME: Raspberry Pi Model B (Pi)
CPU: ARM Cortex A7 @900MHz
CORE: 4
Memory: 1GB
SD Card: 32GB

NAME: NUC5i5MYHE (NUC PC)
CPU: i5-5300U @2.30GHz
CORE: 4
Memory: 16GB DDR3
HDD: 94GB

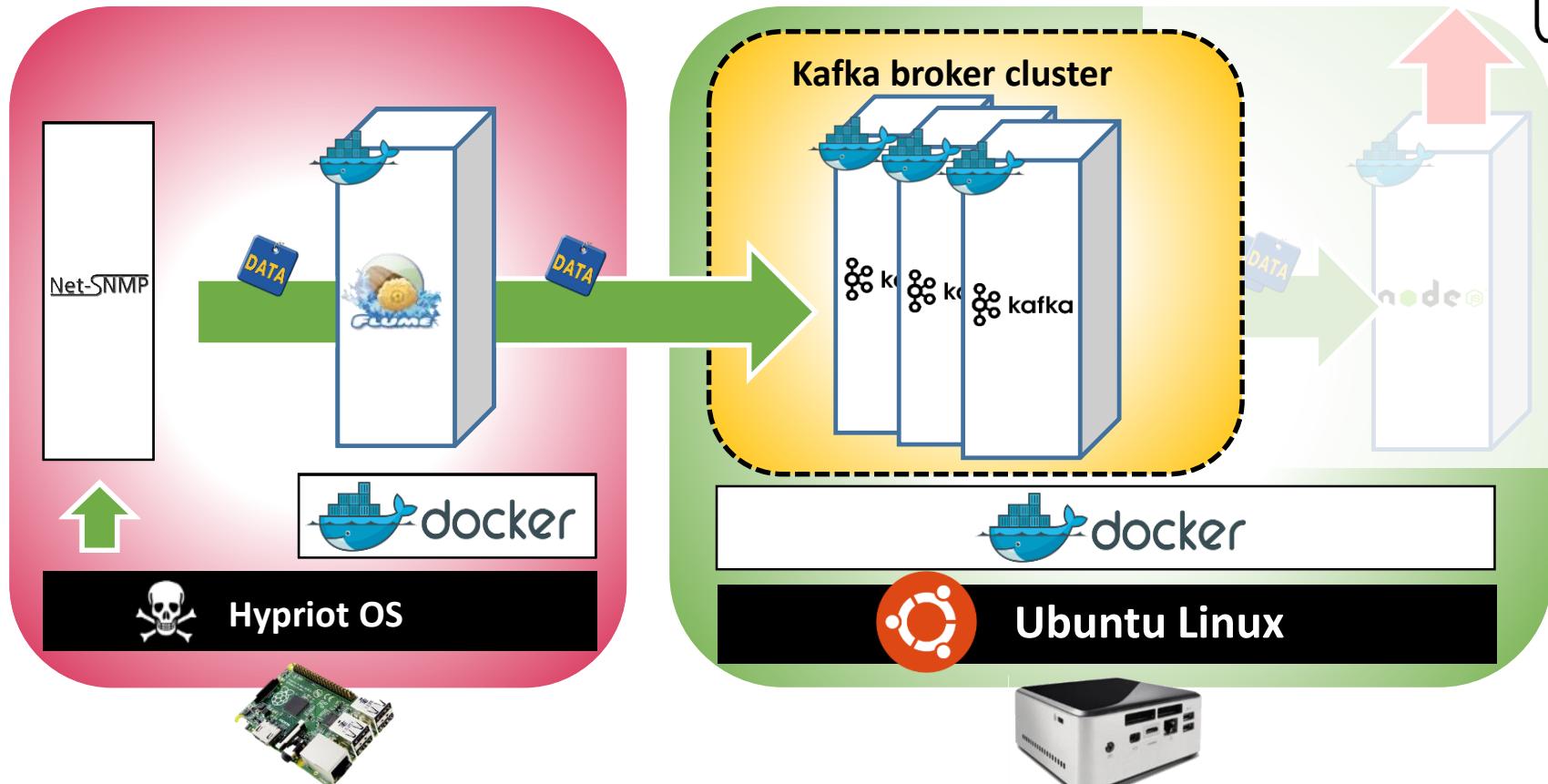
NAME: NT900X3A
CPU: i5-2537U @1.40GHz
CORE: 2
Memory: 4GB DDR3
HDD: 128GB



NAME: netgear prosafe 16 port gigabit switch(Switch)
Network Ports: 16 auto-sensing 10/100/1000 Mbps Ethernet ports

#0 - Lab Preparation (2/2)

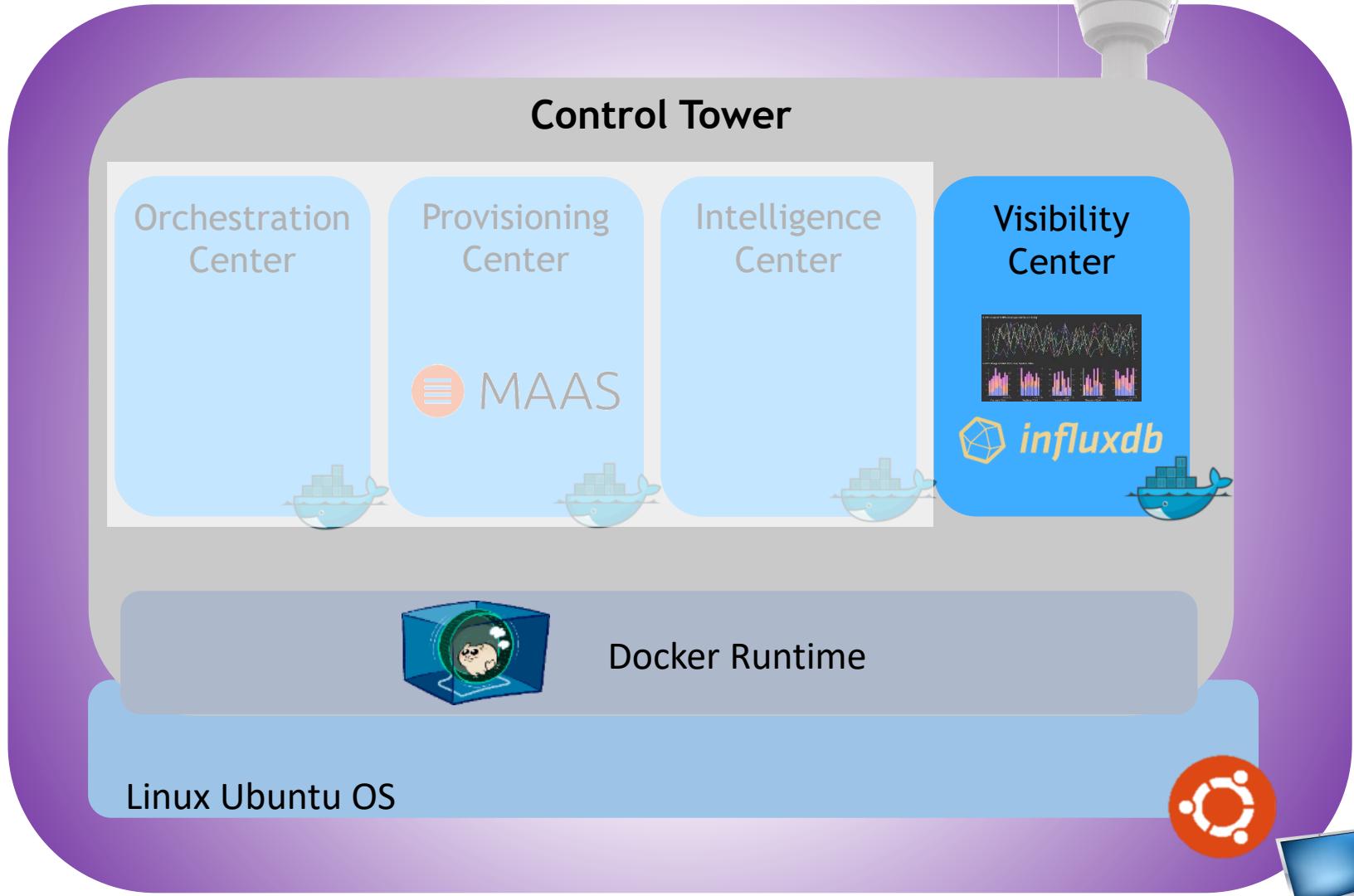
- Verify Inter-Connect Lab's configuration



Are they working?

If you can see logs of resource status on console consumer, go ahead!

Visualization of Resource Visibility





#1 Run InfluxDB & Chronograf Containers on NUC

- Run InfluxDB Container

```
$ docker run -d --name=influxdb --net=host influxdb
```

- Make and run Chronograf container

```
$ docker run -p 8888:8888 --net=host chronograf --influxdb-url=http://<NUC IP>:8086
```



#2 Install python packages: for python Kafka consumer

- Install python-pip

```
$ sudo apt-get install -y libcurl3 openssl curl  
$ sudo apt-get install -y python2.7 python-pip  
$ sudo apt-get install -y python3-pip
```

- Install python package

```
$ sudo pip install requests  
$ sudo pip install kafka-python  
$ sudo pip install influxdb  
$ sudo pip install msgpack
```



#3 Broker to InfluxDB code: Modify & Run python code

- Open ‘broker_to_influxdb.py’ code

```
$ vi ~/SmartX-mini/ubuntu-kafkatodb/broker_to_influxdb.py
```

```
cmd ="curl -XPOST 'http://localhost:8086/query' --data-urlencode 'q=CREATE DATABASE 'Labs'''  
subprocess.call([cmd], shell=True)  
  
timeout = 100  
actual_data=[]  
  
consumer = KafkaConsumer('resource',bootstrap_servers=[ '192.168.1.10:9091'])  
partitions = consumer.poll(timeout)  
while partitions == None or len(partitions) == 0:  
    consumer = KafkaConsumer('resource', bootstrap_servers=[ '192.168.1.10:9091'])  
    message = next(consumer)  
    print(message.value)
```

Modify to your nuc IP Address

Modify to your nuc IP Address

- Run python code

```
$ sudo sysctl -w fs.file-max=100000
```

```
$ ulimit -S -n 2048
```

```
$ python ~/SmartX-mini/ubuntu-kafkatodb/broker_to_influxdb.py
```

#4 Configure Chronograf Dashboard (1/4)



- Open Web browser and connect to Chronograf Dashboard
<http://<NUC IP>:8888>

The screenshot shows the Chronograf dashboard interface. On the left, there's a vertical sidebar with icons for Status, Alerts, News Feed, Getting Started, and Guides. The main area has two sections: "Status" at the top and "Alerts - Last 30 Days" below it. In the "Status" section, the title is "Alert Events per Day – Last 30 Days" and it displays "No Results". In the "Alerts" section, there's a message: "The current source does not have an associated Kapacitor instance" and a blue "Configure Kapacitor" button. To the right, there's a "News Feed" section with a single item: "Press Release: InfluxData Announces InfluxDays SF 2018 – Industry's Only Event Focused on Impact of Collected Time Series Data for Real-Time Decision Making" by Mark Herring, dated Oct 23. Below the news feed is a "Getting Started" sidebar with a blue header "Welcome to Chronograf!". It includes links to "Install the TICK Stack", "TICK Sandbox", and "Guides" which lists "Create a Dashboard", "Create a Kapacitor Alert", "Configure Kapacitor Event Handlers", "Transition from InfluxDB's Web Admin Interface", "Dashboard Template Variables", and "Advanced Monitor Flows".

#4 Configure Chronograf Dashboard (2/4)



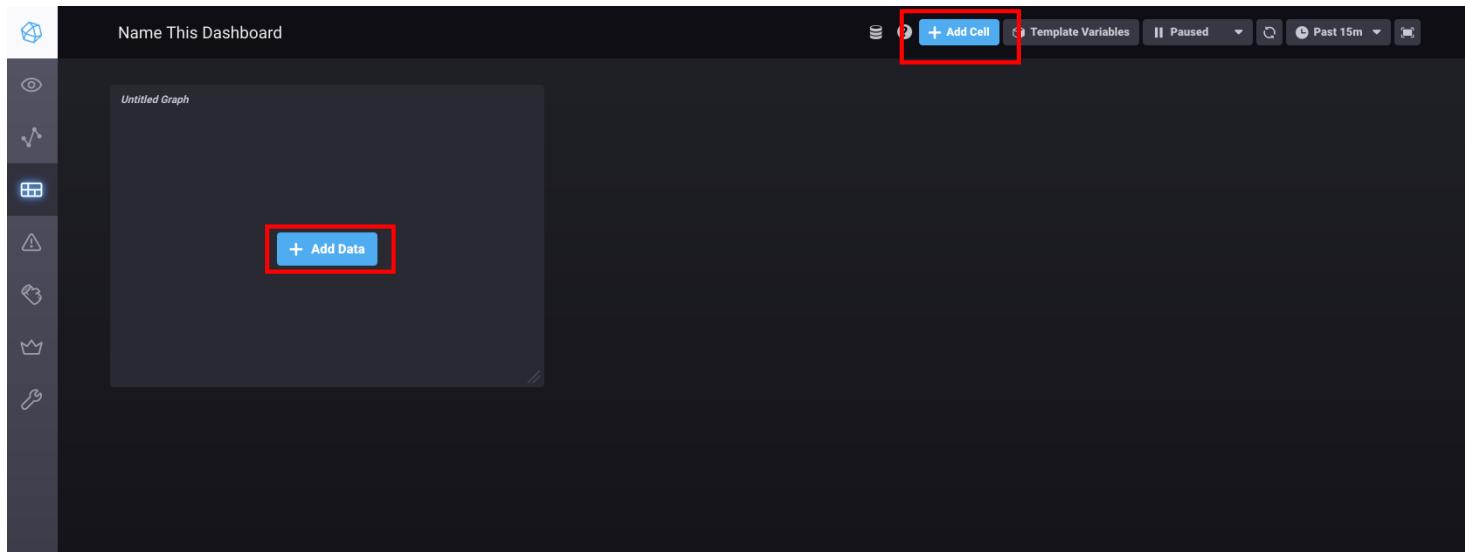
- Create Dash board

A screenshot of the Chronograf interface, specifically the 'Dashboards' section. The left sidebar features several icons: a blue hexagon (Dashboards), a blue eye (Dashboard View), a blue triangle (Alerts), a blue hand (Edit), a blue crown (Settings), and a blue key (Security). The 'Dashboards' icon is highlighted with a red box. The main area is titled 'Dashboards' and shows '0 Dashboards'. It includes a search bar labeled 'Filter by Name...', a 'Import Dashboard' button, and a prominent blue 'Create Dashboard' button with a plus sign. Below the button, a message reads 'Looks like you don't have any dashboards'.

#4 Configure Chronograf Dashboard (3/4)



- Add Cell & Data



#4 Configure Chronograf Dashboard (4/4)



- Add Data

The screenshot shows the Chronograf interface. At the top, there are tabs for "Queries" and "Visualization". Below the tabs, a query editor window displays the following code:

```
SELECT "Memory" FROM "Labs"."autogen"."la... +  
SELECT "Memory" FROM "Labs"."autogen"."labs" WHERE time > :dashboardTime:  
✓ Success!
```

To the right of the query editor, there is a configuration panel with the following sections:

- DB.RetentionPolicy**: Shows the selected database as `_internal.monitor`.
- Measurements & Tags**: Shows the selected measurement as `labs`. Under this section, there is a "Filter" input field.
- Fields**: Shows the available fields: `CPU_Load`, `CPU_usage`, and `Memory`. The `Memory` field is selected.
- Functions**: Shows a grid of functions: `mean`, `median`, `count`, `min`, `max`, `sum`, `first`, `last`, `spread`, `stddev`, `storage_usage`, and `timestamp`. An "Apply" button is located at the bottom right of this section.

#5 – Check Chronograf Dashboard

Lab #3: Tower 23



- We can see the changes of values from database

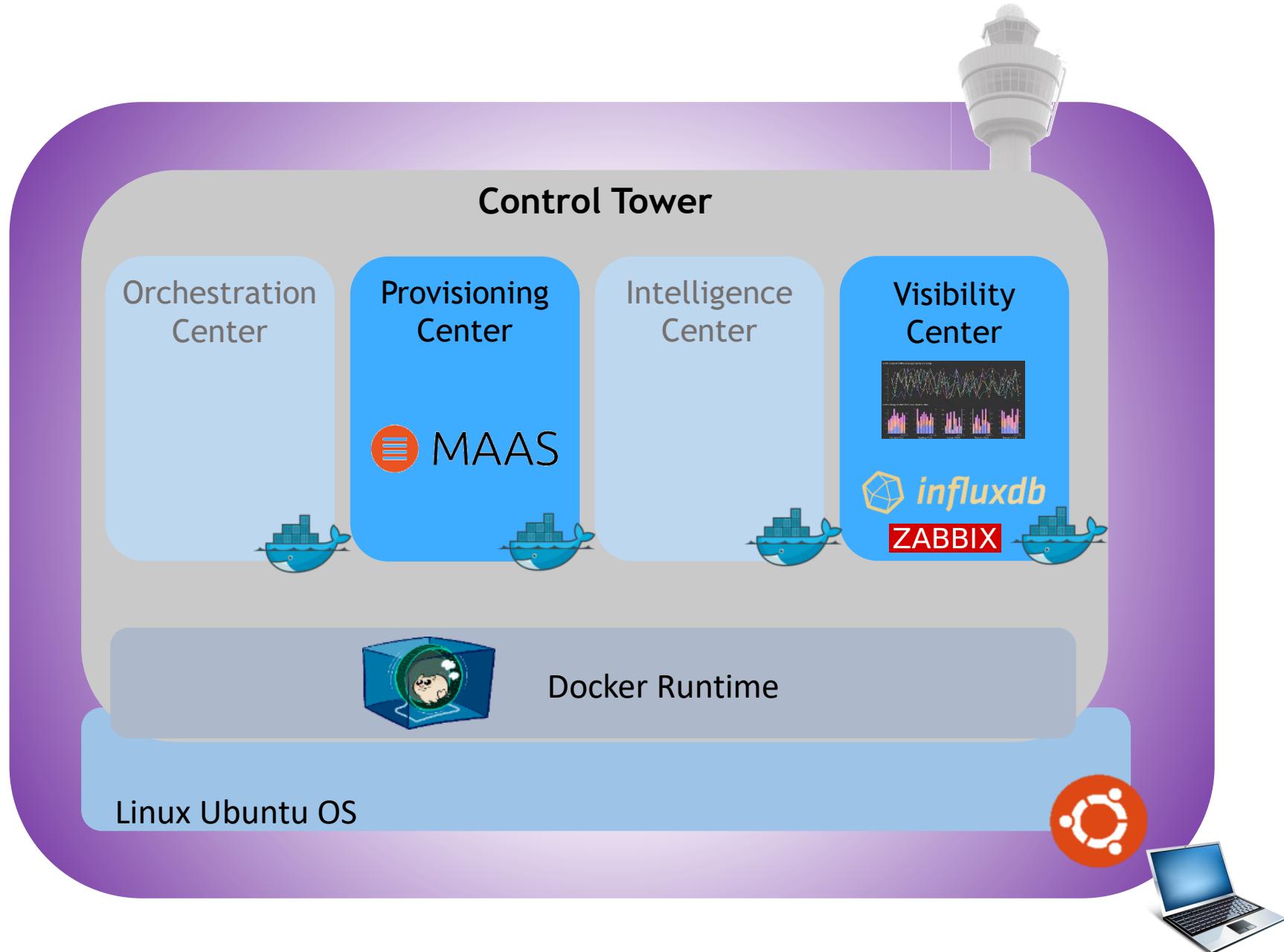


Note:

Remaining Lab practice requires
special Box resources with
IPMI or similar Remote Power Management,
PXE Boot support

Automated Provisioning & Visibility

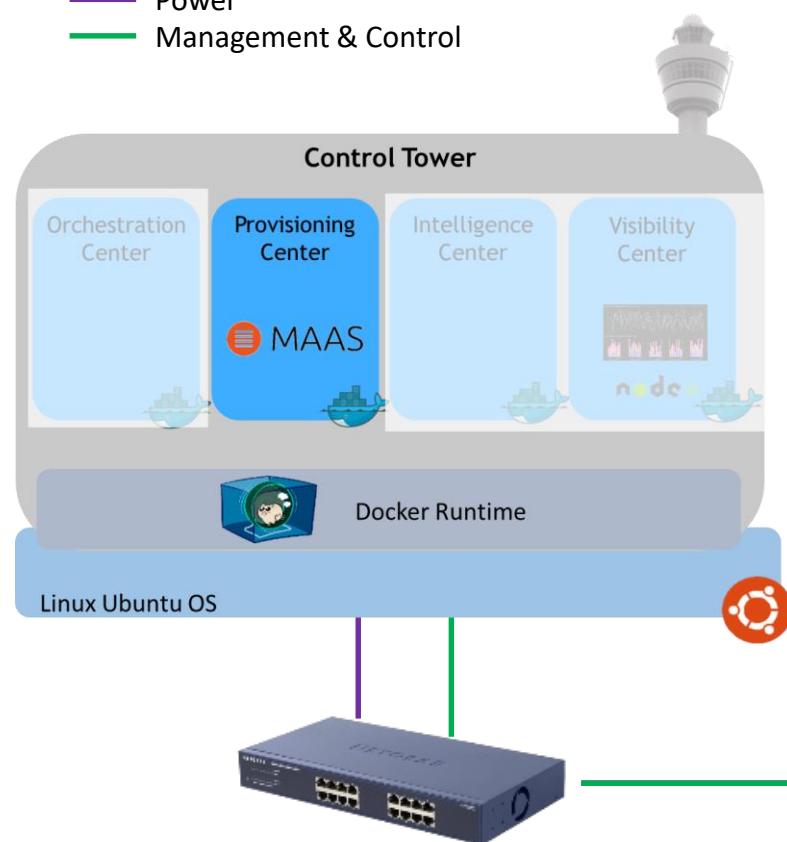
Lab #3: Tower 25



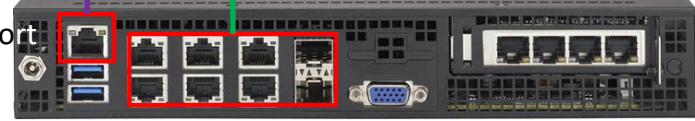
#6-1 OS Automated Installation: Establish physical interconnect

Power

Management & Control



IPMI port



Any Ethernet port
that support PXE

Requirements for Automated Installation

- IPMI, Intel AMT, IBM HMC … and so on.
- DHCP PXE bootable

Note: Typical NUC does not satisfy the above requirement!



#6-2 OS Automatic Installation: BIOS Configuration

Aptio Setup Utility - Copyright (C) 2017 American Megatrends, Inc.

Advanced

PCI Bus Driver Version	A5.01.12	Select which firmware type to be loaded for onboard LANs
PCI Devices Common Settings:		
Above 4G Decoding	[Disabled]	
SR-IOV Support	[Disabled]	
Maximum Payload	[Auto]	
Maximum Read Request	[Auto]	
ASPM Support	[Auto]	
ARI Forwarding	[Disabled]	
RSC-RR1U-E8 OPROM	[EFI]	
M.2 PCI-E 3.0 X4 OPROM	[EFI]	
Mini PCI-E OPROM	[EFI]	
Onboard LAN OPROM Type	[EFI]	
Onboard Video OPROM	[EFI]	
VGA Priority	[Onboard]	
Network Stack	[Enabled]	
IPv4 PXE Support	[Enabled]	
IPv6 PXE Support	[Disabled]	
PXE boot wait time	0	
Media detect count	1	

++: Select Screen
†!: Select Item
Enter: Select
+/-: Change Opt.
F1: General Help
F2: Previous Values
F3: Optimized Defaults
F4: Save & Exit
ESC: Exit

<BIOS PXE Configuration>

Aptio Setup Utility - Copyright (C) 2017 American Megatrends, Inc.

IPMI

BMC Network Configuration

IPMI LAN Selection [Failover]
IPMI Network Link Status: Dedicated LAN

Update IPMI LAN Configuration [No]
Configuration Address Source [Static]
Station MAC Address 0c-c4-7a-f5-46-90
Station IP Address 255.255.255.000
Subnet Mask 010.255.255.254
Gateway IP Address VLAN [Disable]

IPMI IP Configuration

++: Select Screen
†!: Select Item
Enter: Select
+/-: Change Opt.
F1: General Help
F2: Previous Values
F3: Optimized Defaults
F4: Save & Exit
ESC: Exit

Version 2.19.1266. Copyright (C) 2017 American Megatrends, Inc.

<BIOS IPMI Configuration>

- After then **Save Configuration and Exit**
- And then the PXE booting sequence is stated



#7-1 OS Automatic Installation: Enlistment

Booting under MAAS direction...

Check this message

```
[ OK ] Reached target Local Encrypted Volumes.
[ OK ] Started Commit a transient machine-id on disk.
[ OK ] Started ebtables ruleset management.
[ OK ] Started Network Time Synchronization.
[ OK ] Listening on Load/Save RF Kill Switch Status /dev/rfkill Watch.
[ OK ] Reached target System Time Synchronized.
[ 53.726408] cloud-init[570]: Cloud-init v. 18.3-9-g2e62cb8a-0ubuntu1~18.04.2 running "init-local"
at Tue, 16 Oct 2018 12:20:53 +0000. Up 53.25 seconds.
[ OK ] Started Initial cloud-init job (pre-networking).
[ OK ] Reached target Network (Pre).
Starting Network Service...
[ OK ] Started Network Service.
Starting Wait for Network to be Configured...
Starting Network Name Resolution...
[ OK ] Started Network Name Resolution.
[ OK ] Reached target Host and Network Name Lookups.
[ OK ] Reached target Network.
[ *** ] A start job is running for Wait for Network to be Configured (2min 4s / no limit)
```

<Succeed PXE booting>

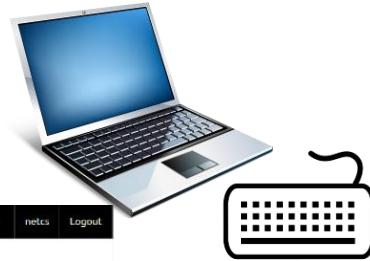
<Enlistment after PXE booting>



FQDN / MAC	Power	Status	Owner	Cores	RAM (GB)	Disks	Storage (GB)
fleet-lynx.maas	Unknown	New		0	0.0	0	0.0
K1-GJ1-Cube1.maas	On	Ubuntu 18.04 LTS	netcs	8	32.0	1	512.1
K1-GJ1-Cube2.maas	On	Ubuntu 18.04 LTS	netcs	8	32.0	1	512.1
K1-GJ1-Cube3.maas	On	Ubuntu 16.04 LTS	netcs	8	32.0	1	512.1
K1-GJ1-Cube4.maas	On	Ubuntu 16.04 LTS	netcs	8	32.0	1	512.1

<After Enlistment (you can check on Web GUI) >

#7-2 OS Automatic Installation: Commissioning



Click this button to commission

All the hardware information shown as 'Unknown' before Commissioning

Do not check anything

Click this button to commission

- It takes about **10 minutes**

#7-3 OS Automatic Installation: Deployment (1/2)



You can see Ready state and power on/off state after commissioning

Click this button to Deploy

Deploy

You can see all the Hardware information After commissioning

CPU
12 cores (amd64/generic)
No tests have been run

Memory
32.0 GiB
No tests have been run

Storage
512.1 GB over 1 disks
1x512GB (ssd)
1 test has passed.
See storage >

Choose your image Ubuntu Ubuntu 18.04 LTS "Bionic Beaver" Default kernel Deploy machine

Click this button to Deploy

- It is a **OS Installation procedure**
- It takes about 15 minutes

#7-3 OS Automatic Installation: Deployment (2/2)



<Complete Deploying (OS booted)>



fleet-lynx.maas

Deployed Power on check now

Machine state is changed to
'Deployed'

Take action

Machine summary Interfaces Storage Commissioning Hardware tests Logs Events Configuration

A screenshot of a web-based management interface for a machine named "fleet-lynx.maas". The status bar at the top indicates the machine is "Deployed" and provides a "Power on" button. A message below states "Machine state is changed to 'Deployed'". The navigation menu includes links for Machine summary, Interfaces, Storage, Commissioning, Hardware tests, Logs, Events, and Configuration. The "Machine summary" link is underlined, indicating it is the active tab. The "Interfaces" and "Storage" tabs are also visible in the menu.

<Complete Deploying (On Web GUI)>



#8 Zabbix Server installation (1/2)

- Add PPA (for Zabbix Server & Zabbix Agent)

```
$ wget https://repo.zabbix.com/zabbix/4.0/ubuntu/pool/main/z/zabbix-
release/zabbix-release_4.0-2+xenial_all.deb
$ dpkg -i zabbix-release_4.0-2+xenial_all.deb
$ sudo apt-get update
```

- Zabbix package installation

```
$ apt install zabbix-server-mysql zabbix-frontend-php zabbix-agent
```

- Creating initial database

```
$ mysql -uroot -p<your root_password>
mysql> create database zabbix character set utf8 collate utf8_bin;
mysql> grant all privileges on zabbix.* to zabbix@localhost identified by
'<your root_password>';
mysql> quit;
```

#8 Zabbix Server installation (2/2)



- Make new mysql user & password

```
$ zcat /usr/share/doc/zabbix-server-mysql*/create.sql.gz | mysql -uzabbix -p zabbix
```

Enter password: <Zabbix DB password>

- Fix configuration file (add 4 lines)

```
$ sudo vi /etc/zabbix/zabbix_server.conf
```

DBHost=localhost

DBName=zabbix

DBUser=zabbix

DBPassword=<Zabbix DB password>

#9 Zabbix Service & Apache activation



- Zabbix server service activation

```
$ sudo systemctl start zabbix-server
```

```
$ sudo systemctl enable zabbix-server
```

- Apache activation

```
$ systemctl start apache2
```

#10 Connect to Zabbix Web GUI

Lab #3: Tower 35



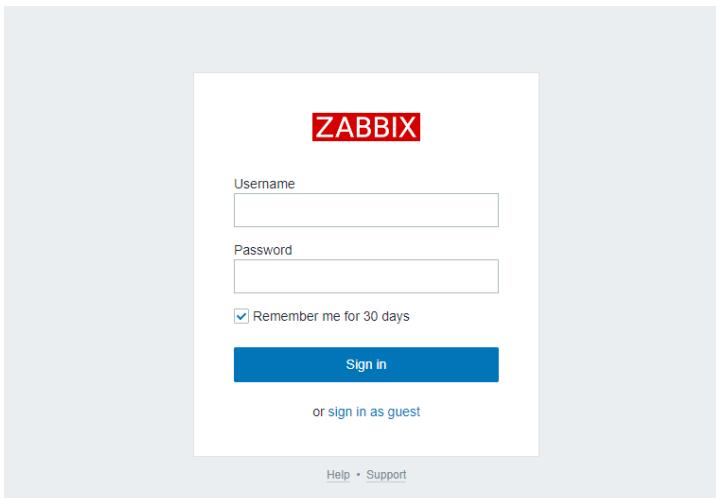
- Connect to Web GUI (On the Web browser)

<http://localhost/zabbix>

- Login via default Username / Password

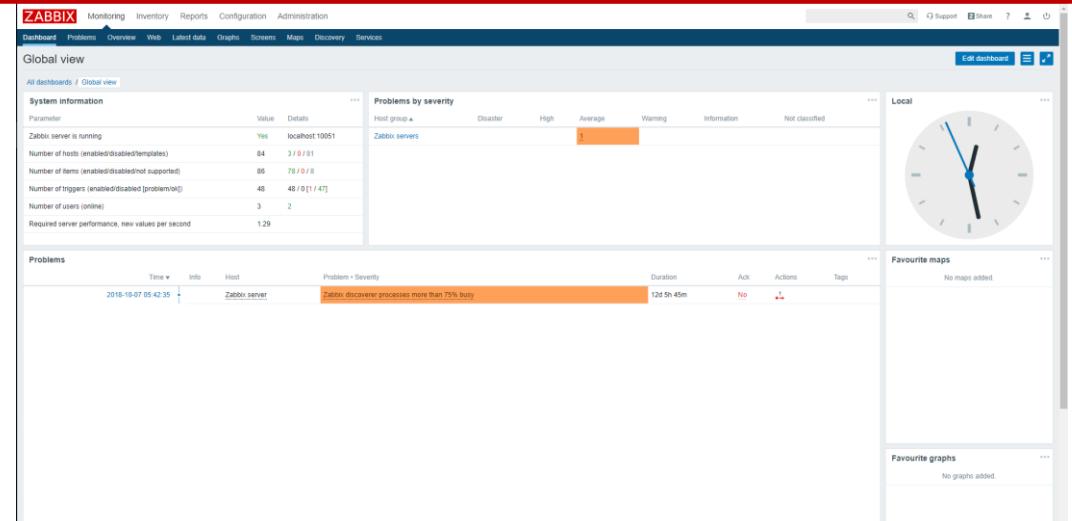
Username: Admin

Password: zabbix



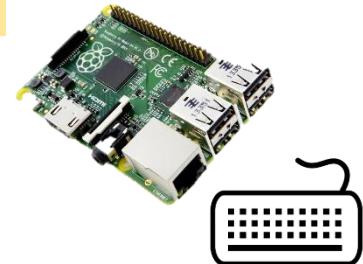
The screenshot shows the Zabbix login interface. It features a red header bar with the word "ZABBIX" in white. Below it is a white form with two input fields: "Username" and "Password", both with placeholder text. A "Remember me for 30 days" checkbox is checked. At the bottom are a "Sign in" button and a link to "sign in as guest". At the very bottom, there are "Help" and "Support" links.

<Zabbix login page>



<Zabbix Web GUI>

#11-1 Zabbix Agent: Agent installation



- Download and install Zabbix Agent

```
$ wget http://repo.zabbix.com/zabbix/2.4/debian/pool/main/z/zabbix/zabbix-
agent_2.4.0-1+wheezy_amd64.deb
$ sudo dpkg -i zabbix-agent_2.4.0-1+wheezy_amd64.deb
```

- Zabbix Agent configuration (Add 3 lines)

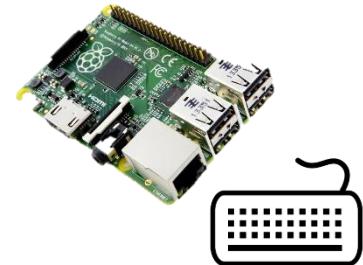
```
$ vi /etc/zabbix/zabbix_agentd.conf
```

Server=IP of Zabbix Server

ServerActive=IP of Zabbix Server

Hostname=use the FQDN of the node where the agent runs

#11-2 Zabbix Agent: Start Agent and open TCP port



- Start Zabbix Agent service

```
$ systemctl restart zabbix-agent
```

- Verify start daemon

```
$ sudo netstat -tulpn | grep zabbix
```

(if result is 10050/tcp, it is successfully worked)

- Open TCP port using ufw tool

```
$ sudo apt-get install ufw
```

```
$ sudo ufw allow 10050/tcp
```

#12-1 Zabbix Monitoring: Add Zabbix monitored Host on Server



ZABBIX Monitoring Inventory Reports Configuration Administration

Host groups Templates **Hosts** Maintenance Actions Event correlation Discovery Services

Name DNS
Monitored by Any Server Proxy IP
Port

Group all Import Filter

<Host Configuration menu on Web GUI>

Host Templates IPMI Macros Host inventory Encryption

* Host name * Groups

* At least one interface must exist.

Agent Interfaces DNS name Connect to IP DNS Port Default Remove

SNMP interfaces

JMX interfaces

IPMI interfaces

Description

Monitored by proxy Enabled

<Host Configuration>

Host Templates IPMI Macros Host inventory Encryption

Linked templates Name Action

Link new templates

Host groups Temples

Hosts

Host Templates

Linked tem

Link new tem

Templates

Template Net QTech QSW SNMPv2
Template Net TP-LINK SNMPv2
Template Net Ubiquiti AirOS SNMPv1
Template OS AIX
Template OS FreeBSD
Template OS HP-UX
 Template OS Linux
Template OS Linux SNMPv2
Template OS Mac OS X

<Select Template>

#12-2 Zabbix Monitoring: Adding host result & Add items



ZABBIX Monitoring Inventory Reports Configuration Administration Support Share ?

Host groups Templates Hosts Maintenance Actions Event correlation Discovery Services

Host updated

Hosts Group all Filter

Name DNS
Monitored by IP
Port

<input type="checkbox"/> Name ▲	Applications	Items	Triggers	Graphs	Discovery	Web	Interface	Templates	Status	Availability	Agent encryption	Ir
<input type="checkbox"/> new host	Applications	Items 2	Triggers	Graphs 1	Discovery	Web		Enabled	ZBX SNMP JMX IPMI	NONE		

<Adding host result on Web GUI>

#12-3 Zabbix Monitoring: Adding item to monitoring (1/2)



ZABBIX Monitoring Inventory Reports Configuration Administration

Host groups Templates **Hosts** Maintenance Actions Event correlation Discovery Services

Hosts Group: all Create host Import

Name: [] DNS: []
 Monitored by: Any Server Proxy IP: []
 Port: []

Apply Reset

	Name	Applications	Items	Triggers	Graphs	Discovery	Web	Interface	Templates	Status	Availability	Agent encryption	Info
<input type="checkbox"/>	new host	Applications	Items 2	Triggers	Graphs 1	Discovery	Web	[REDACTED]	10050	Enabled	ZBX SNMP JMX IPMI	NONE	

Displaying 1 of 1 found

ZABBIX Monitoring Inventory Reports Configuration Administration

Host groups Templates **Hosts** Maintenance Actions Event correlation Discovery Services

Items Create item

All hosts / new host Enabled ZBX SNMP JMX IPMI Applications Items 2 Triggers Graphs 1 Discovery rules Web scenarios Filter

Host group: <input type="text"/> Select	Type: all	Type of information: all	State
Host: new host <input type="button" value="Select"/>	Update interval: []	History: []	Status
Application: <input type="text"/> Select		Trends: []	Triggers
Name: []			Template
Key: []			Discovery

Subfilter affects only filtered data



#12-3 Zabbix Monitoring: Adding item to monitoring (2/2)

Configure three things

Item Preprocessing

Name: test-CPU

Type: Zabbix agent

* Key: system.cpu.load[percpu,avg1]

* Host interface: [REDACTED] 10050

Type of information: Numeric (float)

Units:

* Update interval: 30s

Custom intervals

Type	Interval	Period	Action
Flexible	50s	1-7,00:00-24:00	Remove

Add

* History storage period: 90d

* Trend storage period: 365d

Show value: As is

New application:

Applications: -None-

Populates host inventory field: -None-

Description:

Enabled

Click Add to add item

#12–4 Zabbix Monitoring: Adding Monitoring Graph

You must configure filter and then click apply

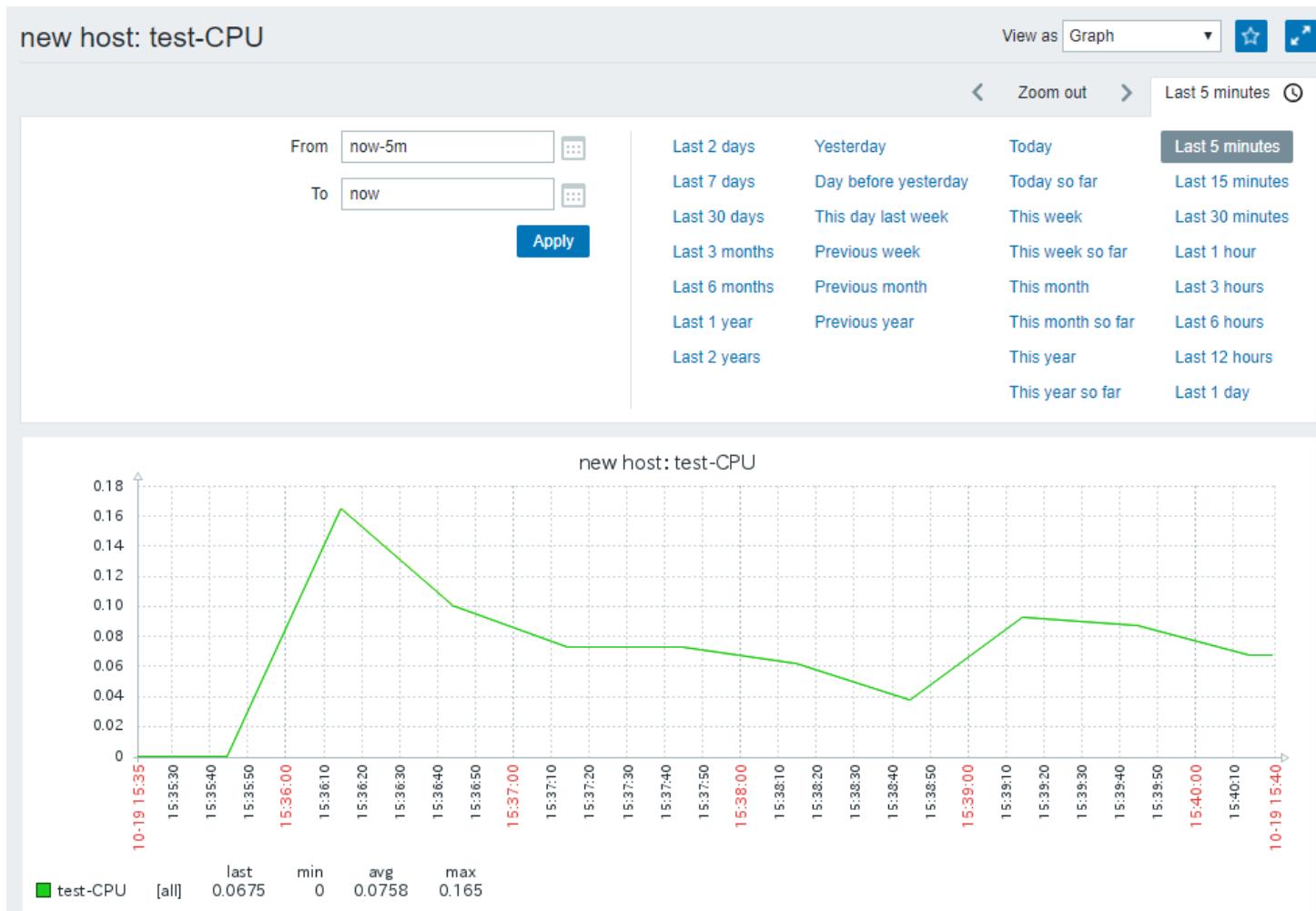
Latest data

Host	Name	Last check	Last value	Change
new host	- other - (2 items)			
	test-CPU	2018-10-19 15:38:15	0.06	-0.01
	Test-network	2018-10-19 15:23:44	152.68 Gbps	+2.27 Mbps

Click graph to show graph

#12-5 Zabbix Monitoring: Result

Lab #3: Tower 43



Review

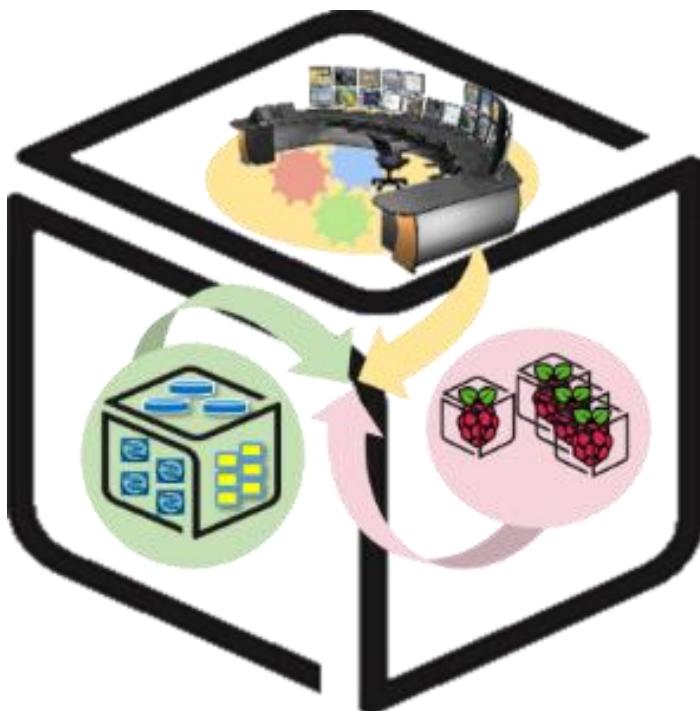


Lab Summary

With Tower Lab, you have experimented selected roles of Monitor/Control (관제) Tower

1. Visibility Center function to **enable ‘distributed monitoring’** over remote Boxes and to **store ‘monitoring information’** to time-size DB.
2. Provisioning Center function to **enable remote ‘installation & configuration** (of OS and others)’ of distributed Boxes.

Thank You for Your Attention Any Questions?



mini@smartx.kr