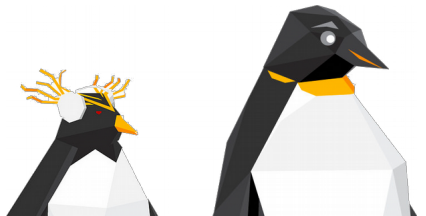
IBM Cloud Private and deployments to Secure Service Containers (Customer Experience)

Sébastien LLaurency
Linux & Cloud Architect - zATS
IBM Certified Expert Integration Architect
Member of Technical Expert Council France
IBM Client Center Montpellier, France

November 2019
Session CE



Are you in the right room ? (Session abstract)

In this session, you will heard about a unique solution, on the market, to take benefits of the speed of deployments for your cloud native applications in an hyper secure environment taking advantages of IBM LinuxONE.

Performing secure deployments of new releases of applications in an easy way is becoming increasingly important ! So how to keep the application and its related data secure ?

What are the existing solutions to keep development agility and security at the top of their capabilities !

Session Objectives

GUIDE SHARE EUROPE UK REGION

— After this session, you will be able to

- Position IBM Z & Hybrid Cloud Computing
- Discover the solutions used for Cloud private deployments
- Describe the solutions for Linux and z/OS on IBM Z
- Provide your team with alternative solutions for cloud
- Lead the discussion with your Dev & Ops for IBM Z

— After, or during, this session, we would like you to

- Ask any questions !
- Contact us to start a co-creative cloud or devops project for your IBM Z

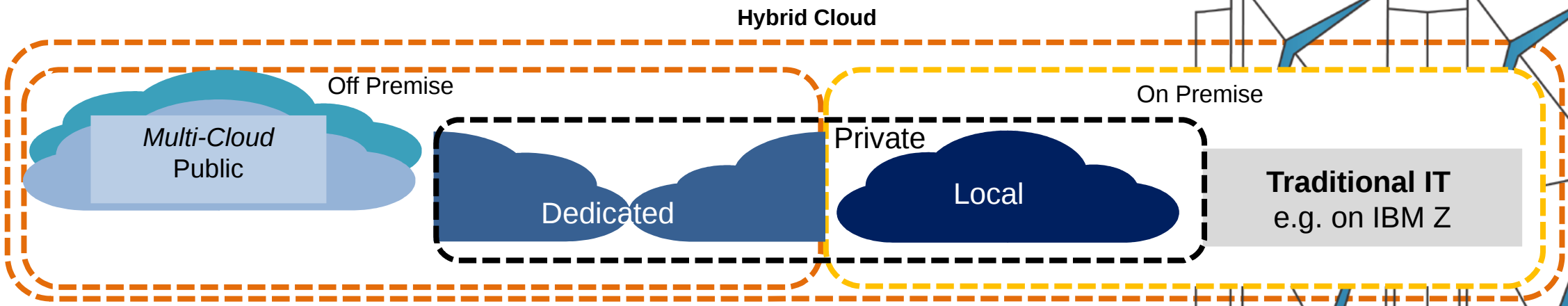
Agenda

- Introducing IBM Z Hybrid Cloud Journey
  - IBM Z:Integration In/Out, IBM Z:Experience, IBM Z:IBM Cloud
  - Application Modernization & Cloud Private
  - Solutions for an Agile Infrastructure
- Discovering the Technical Solution for Secured Deployments
  - Docker, Kubernetes
  - IBM Cloud Private with Linux on IBM Z,
  - IBM Secure Service Container,
- Deployment of IBM Cloud Private for Secure Service Container
- Performing Deployments to ICP for Secure Service Containers
  - Register existing s390x docker images in an ICP registry
  - Create helm chart for your application
  - Deploy your helm chart from the ICP Catalog
  - Work with your containers/logs/debug !
  - Content for some helm chart files
  - Monitor your cluster and your particular application !

Agenda

- Introducing IBM Z Hybrid Cloud Journey
  - IBM Z:Integration In/Out, IBM Z:Experience, IBM Z:IBM Cloud
  - Application Modernization & Cloud Private
  - Solutions for an Agile Infrastructure
- Discovering the Technical Solution for Secured Deployments
  - Docker, Kubernetes
  - IBM Cloud Private with Linux on IBM Z,
  - IBM Secure Service Container,
- Deployment of IBM Cloud Private for Secure Service Container
- Performing Deployments to ICP for Secure Service Containers
  - Register existing s390x docker images in an ICP registry
  - Create helm chart for your application
  - Deploy your helm chart from the ICP Catalog
  - Work with your containers/logs/debug !
  - Content for some helm chart files
  - Monitor your cluster and your particular application !

IBM Z is a Key Player in the Hybrid Cloud Journey.



Z: Cloud Integration – IN & OUT

Z: IBM Cloud!

Z: Cloud Experience

IBM Z provides the infrastructure to support **all dimensions** of cloud service delivery

Z: IBM Cloud

IBM provides a set of IBM Cloud solutions based on LinuxONE infrastructure leveraging its unmatched quality of services

Blockchain

IBM Blockchain Platform (Enterprise Plan)

Hyper Protect Family (based on Secure Service Container)

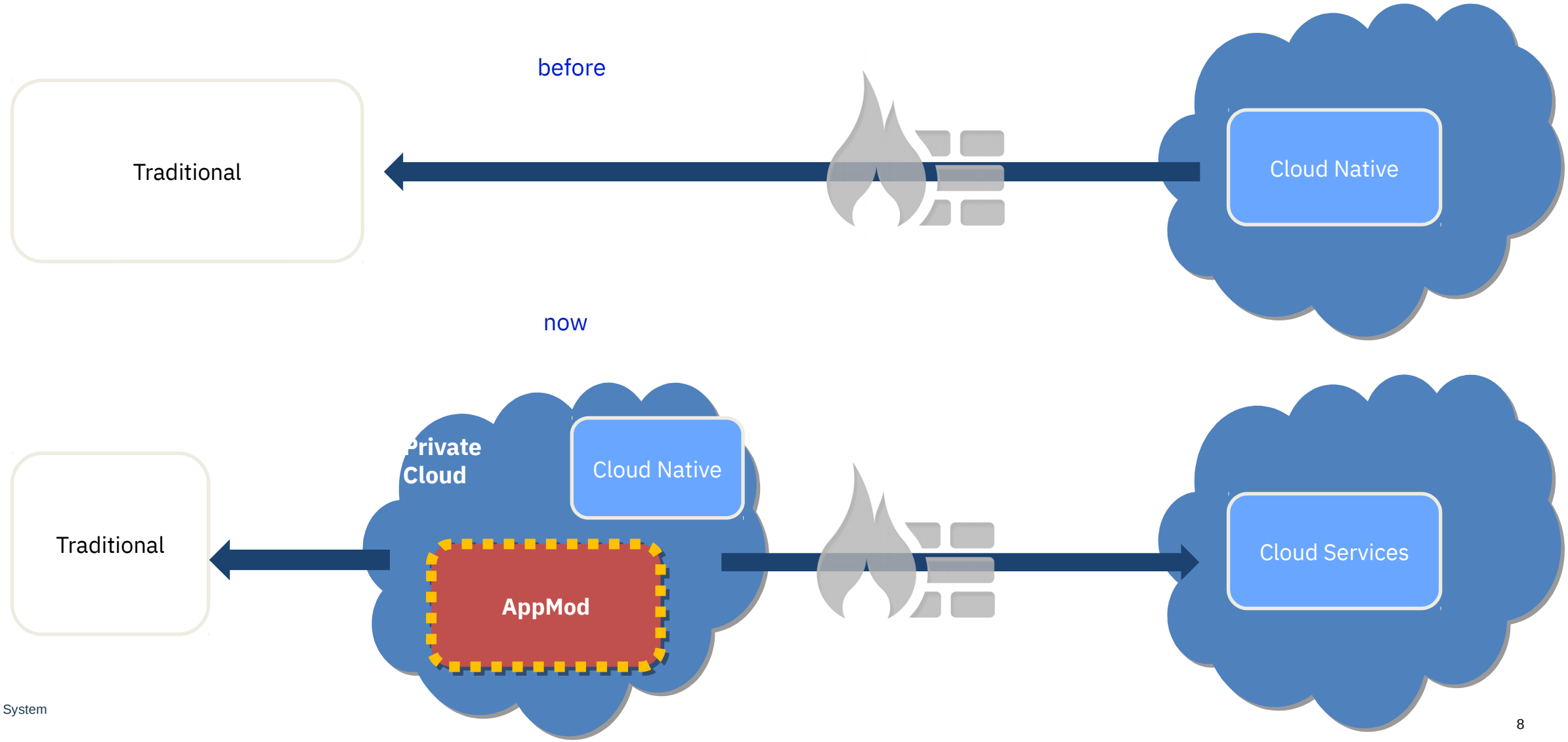
IBM Cloud Hyper Protect DbaaS (*)

IBM Cloud Hyper Protect Crypto Services

IBM Cloud Hyper Protect Virtual Servers (**)

(*) Beta program (**) Experimental

Application Modernization use case “unlocked” by Private Cloud



IBM Z Hybrid Cloud Architecture (current as of 5th of November 2019)

GUIDE SHARE EUROPE UK REGION

Using zCX*

zCX Virtual Docker Address Space

docker

Using Zowe

(ZOWE)

System APIs

Web UI

CLI

Using z/OS Provisioning Toolkit

(zOSPT)

System APIs

Web UI

CLI

using IBM Cloud Private

- ✓ IBM Z as a differentiating asset in ICP from services that span z/OS, Linux on Z, private and public cloud
- ✓ Cloud consumption for z/OS (DBz-aaS, WASz-aaS, MQ-aaS, CICS-aaS etc)
- ✓ DevOps, microservices and application life-cycle management for zOS

ICP Master on Intel/Power/z

Open Service Broker APIs

Kubernetes APIs

Brian
LOB/Executive Decision Maker

DB2

CICS IMS

MQ, WLP, z/OS Connect

z/OSMF

z/OS

service

service

service

service

service

service

service

ICP

Linux

Linux

Linux

IBM Secure Service Container for IBM Cloud Private (SSC4ICP)**

SSC

Linux

Linux

Linux

z/VM

KVM

LPAR

PowerVM etc.

VMWare etc.

Public Cloud

IBM Z

Power

Intel

More information and video available for the announcement of zOS 2.4: https://www.ibm.com/support/z-content-solutions/container-extensions/

IBM and Red Hat — Partners for 20 years

For over 20 years, IBM and Red Hat have collaborated with the Open Source community to drive innovation and empower businesses around the world.

Together, IBM and Red Hat will support our clients with forward-thinking, flexible, cloud-ready infrastructure solutions – **optimized for IBM Z System**.

With Red Hat Our Hybrid Multicloud Strategy reaches Next Level

Hybrid Multicloud for the Enterprise with IBM Z

Transform for Cloud	Cloud native experience	Private Cloud	Public Cloud
Transform infrastructure, applications and data by exposing and connecting existing assets with simplified and intelligent operations across infrastructure	A cloud-native ecosystem on IBM Z and LinuxONE for access and use by administrators, developers and architects with no special skills required	Integrate Z into hybrid multicloud environments and manage everything from behind the firewall.	Tailor your environment with a choice of Z-backed services delivered via IBM Cloud

No matter where you are, where you are going, or how you want to operate...

Build an efficient hybrid multicloud experience with IBM Z

and unlock the unmatched value of the platform for mission critical workloads

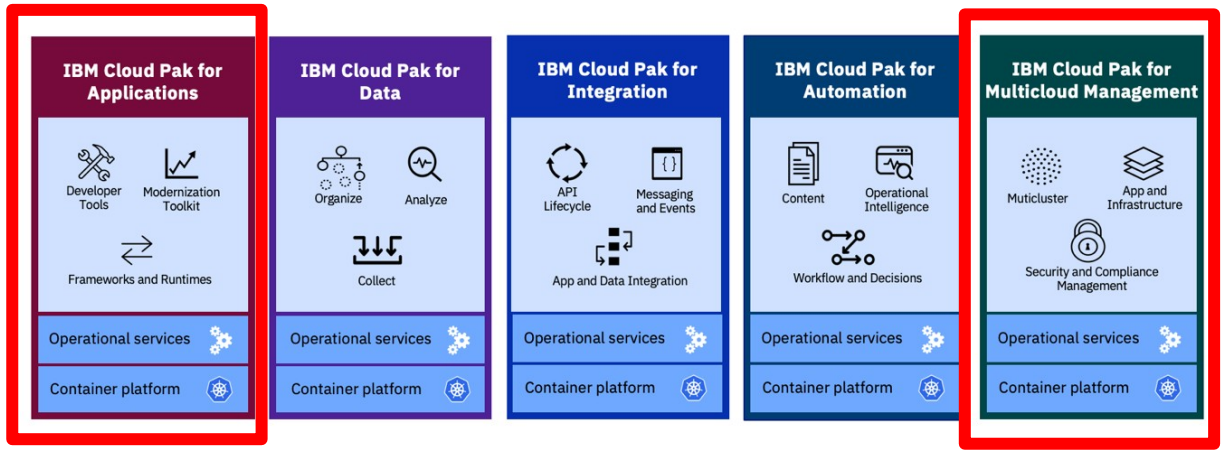
Roadmap for Private Cloud on IBM Z

IBM Cloud Paks On IBM Z/ LinuxONE

Move your core business applications to any cloud through enterprise-ready containerized software solutions

Built on Red Hat OpenShift *

*roadmap item



GA 1Q20

GA 1Q20

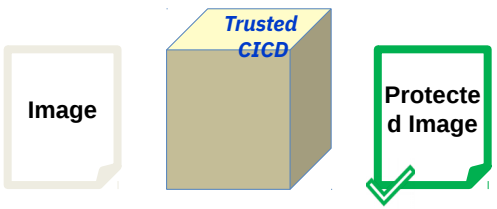
Hyper Protect Virtual Servers on IBM Z and LinuxONE

Support for Container or Non-Containerized workloads

Secure Build

GA 4Q19

RHEL Support*



IBM z/OS Cloud Broker

Integrate with business critical applications running on z/OS

Access and deploy z/OS resources and services

Support for OpenShift *

Cloud-like development experience integrates IBM Z and Non-Z technologies and services together

Available

OpenShift Beta

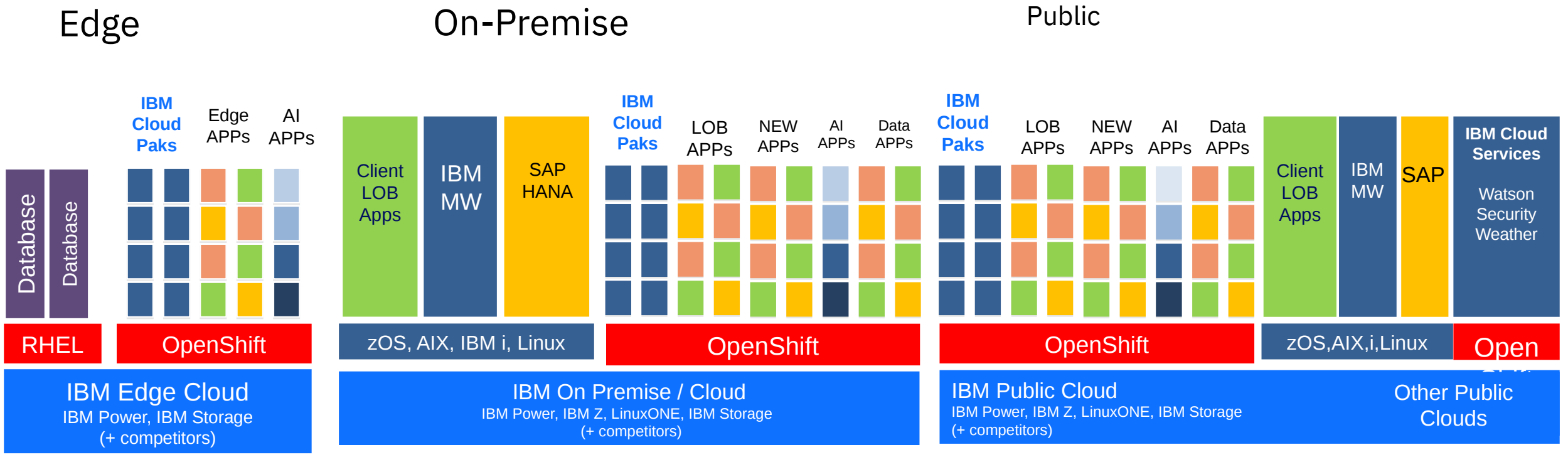
IBM Blockchain Platform on ICP on Z

Available

IBP on Z and LinuxONE Superior performance and highest security certifications that meet the strictest requirements for regulated industries

Our goal is to deliver flexible, cost effective and open-source cloud-based infrastructure solutions, designed for security and reliability and backed with support to our clients

The new Architecture



kubernetes

RED HAT OPENSHIFT Container Platform

OpenShift Container Platform

v3.11 → v4.x

IBM Cloud Paks = IBM SW on OpenShift

Multicloud Management

Applications

Data

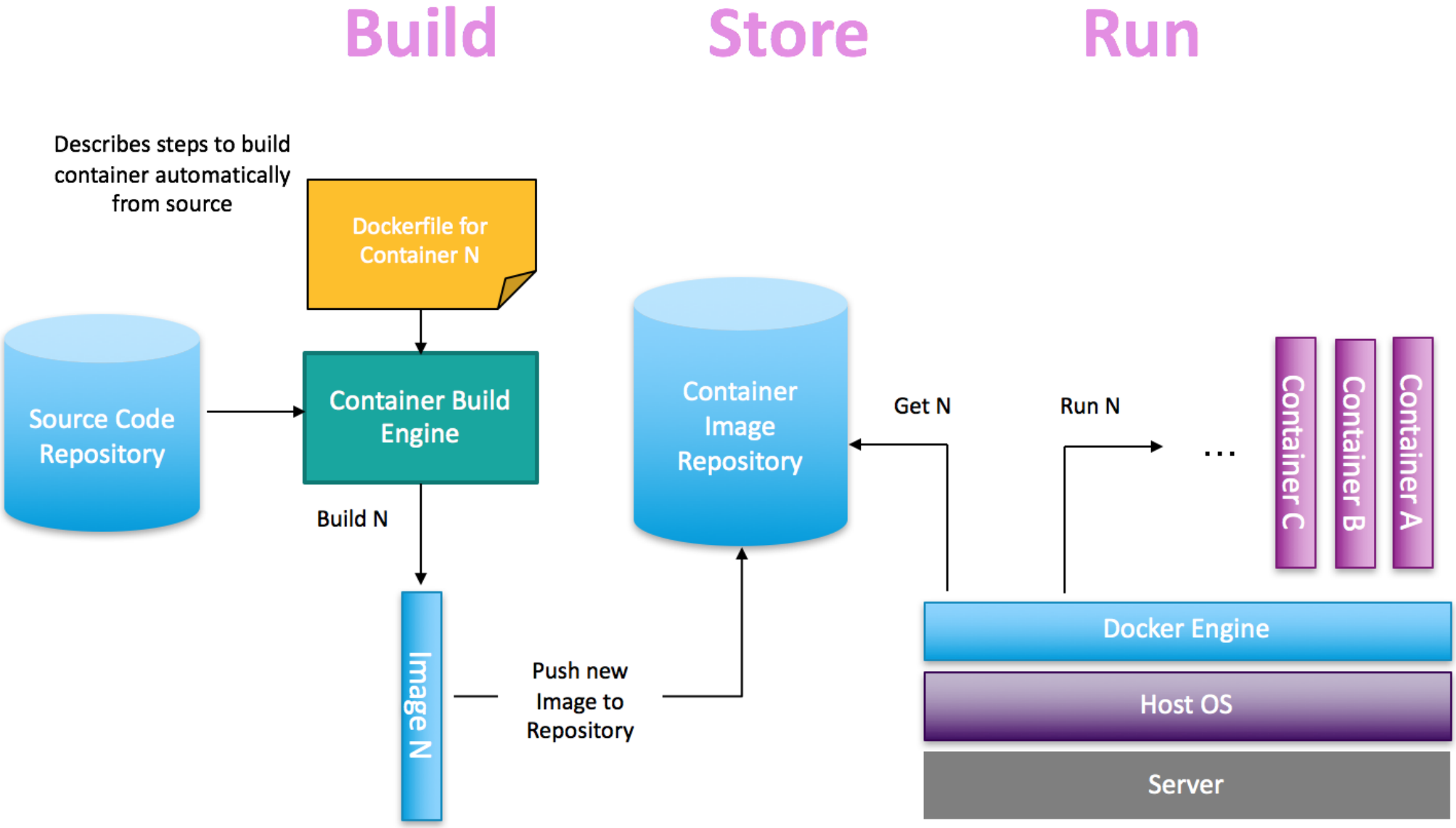
Integration

Automation

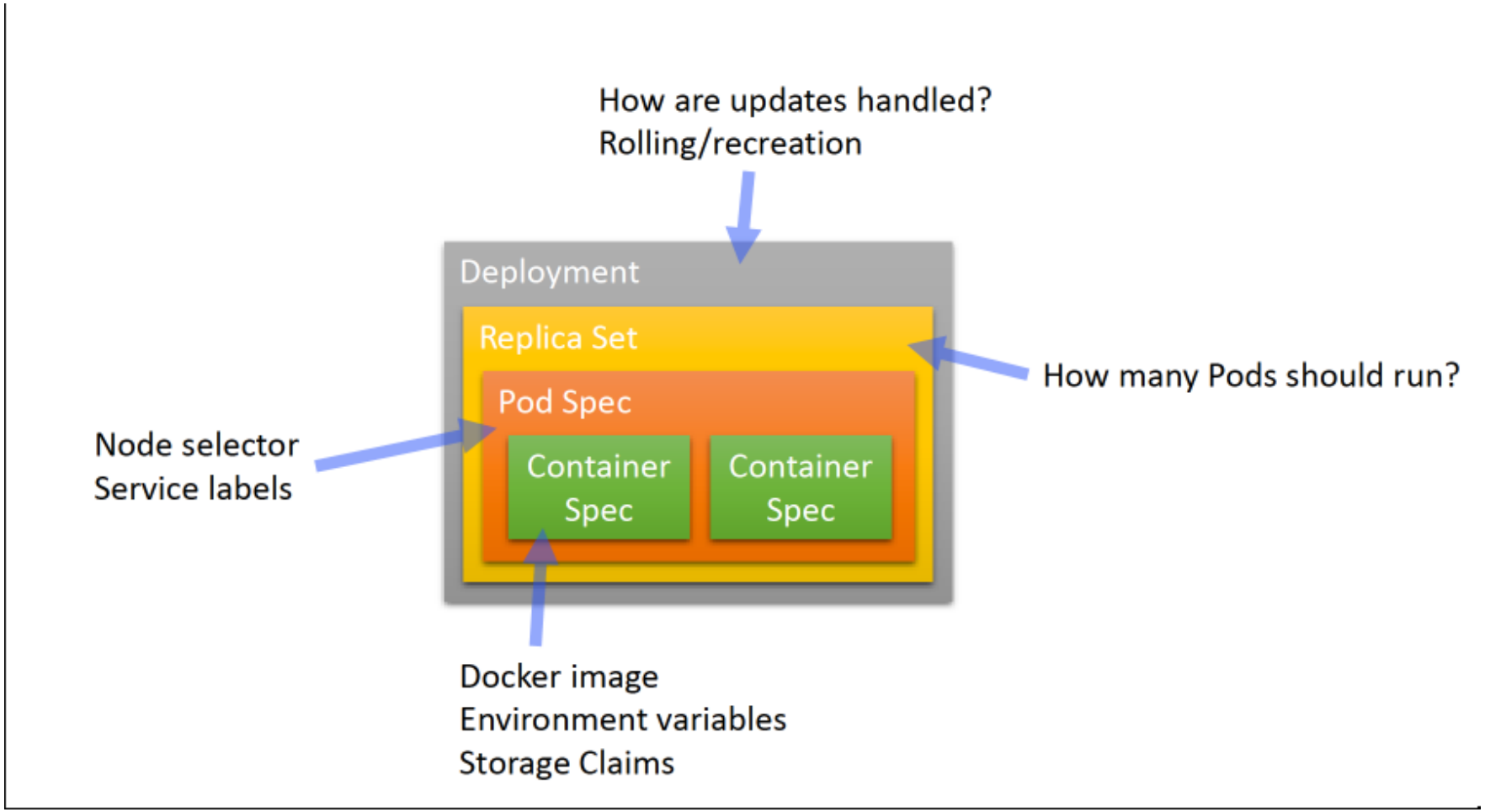
Agenda

- Introducing IBM Z Hybrid Cloud Journey
  - IBM Z:Integration In/Out, IBM Z:Experience, IBM Z:IBM Cloud
  - Application Modernization & Cloud Private
  - Solutions for an Agile Infrastructure
- Discovering the Technical Solution for Secured Deployments
  - Docker, Kubernetes
  - IBM Cloud Private with Linux on IBM Z,
  - IBM Secure Service Container,
- Deployment of IBM Cloud Private for Secure Service Container
- Performing Deployments to ICP for Secure Service Containers
  - Register existing s390x docker images in an ICP registry
  - Create helm chart for your application
  - Deploy your helm chart from the ICP Catalog
  - Work with your containers/logs/debug !
  - Content for some helm chart files
  - Monitor your cluster and your particular application !

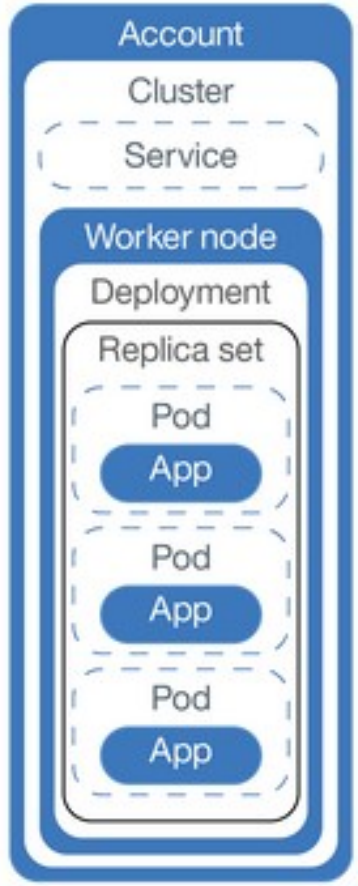
Docker Basic Functions



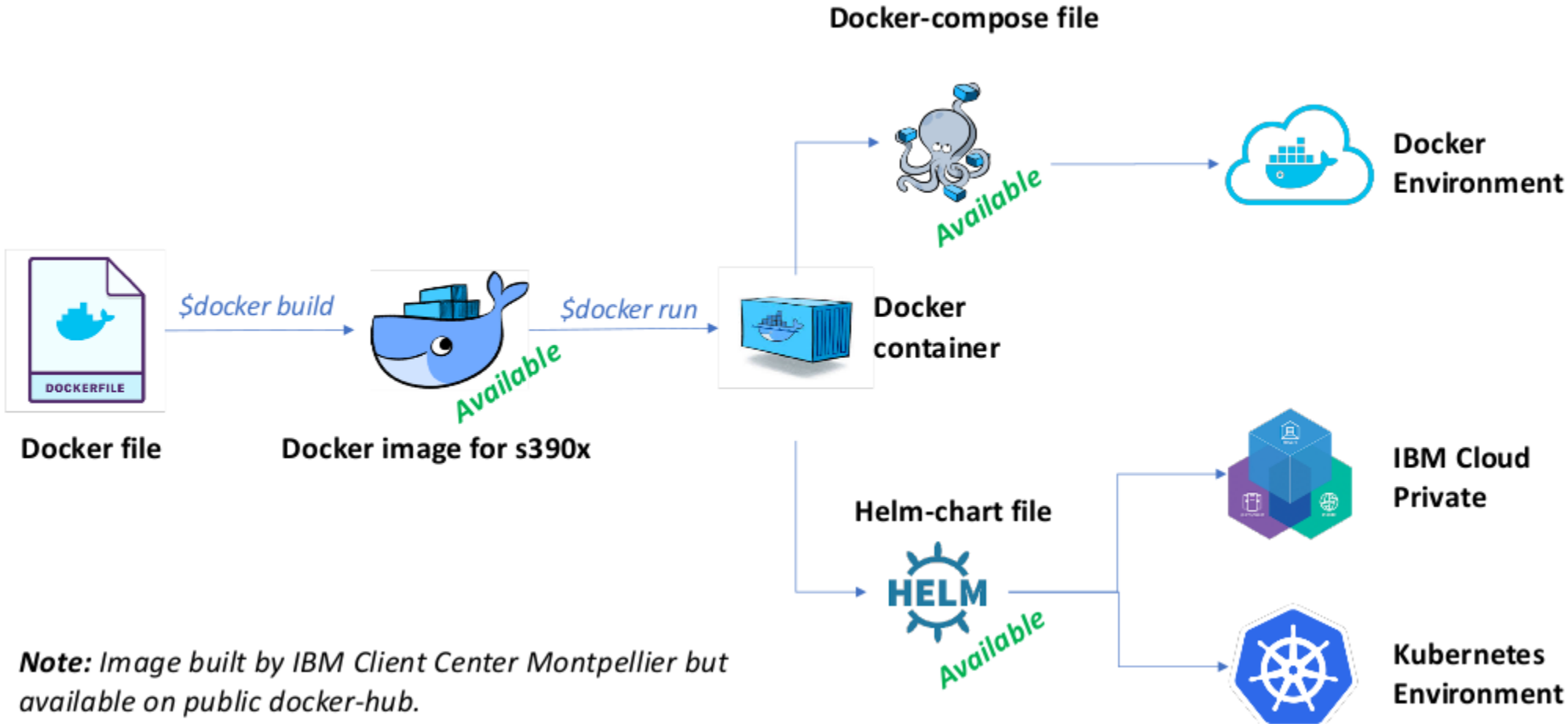
Kubernetes Concepts : Deployment, Replica, Pod



Overview



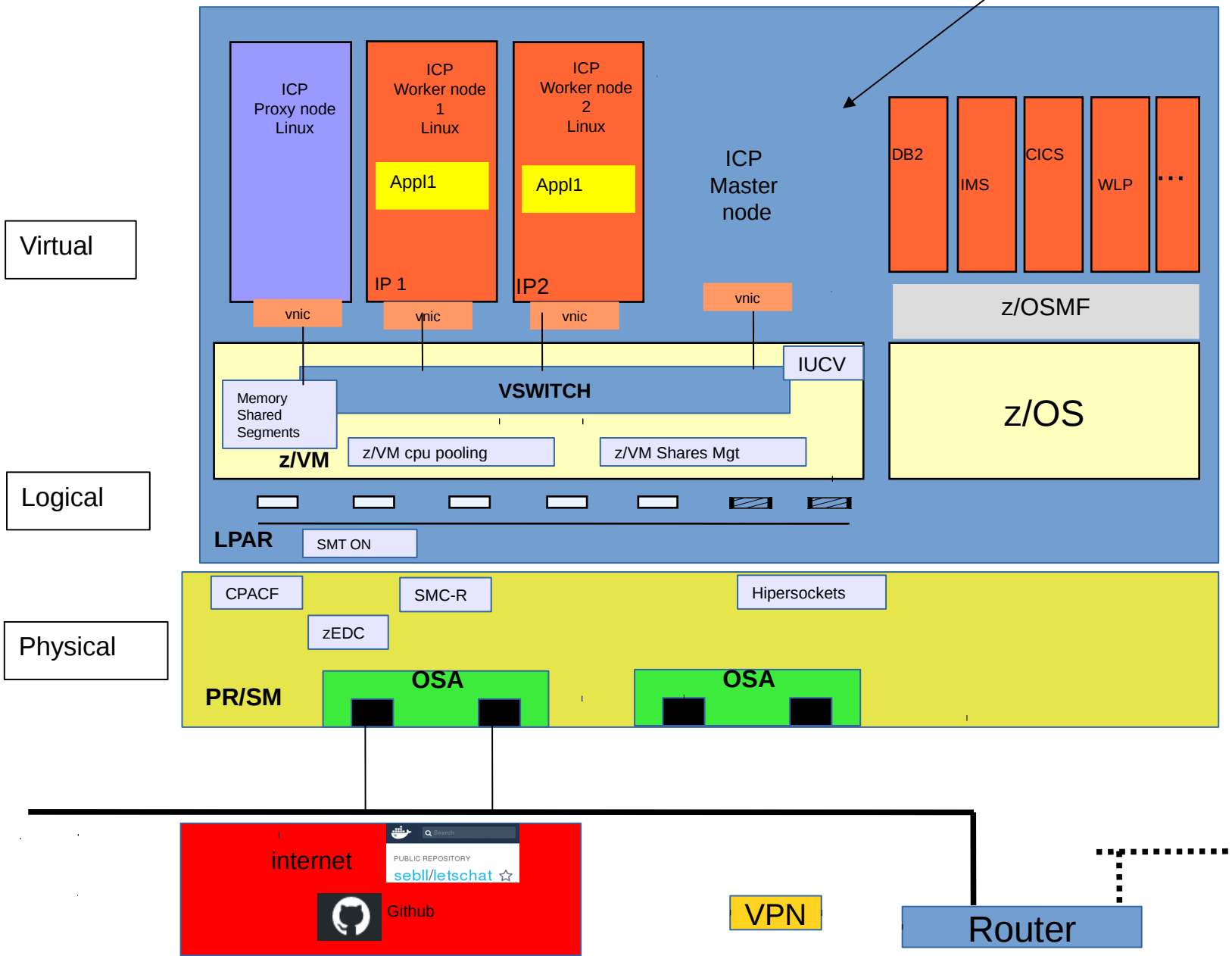
Deployment Model



Note: *Image built by IBM Client Center Montpellier but available on public docker-hub.*

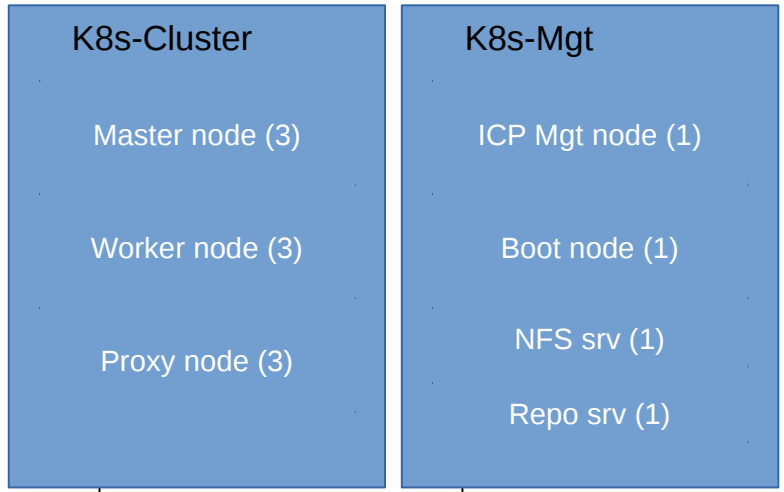
Example for IBM Cloud Private on IBM Z **without** Secure Service container

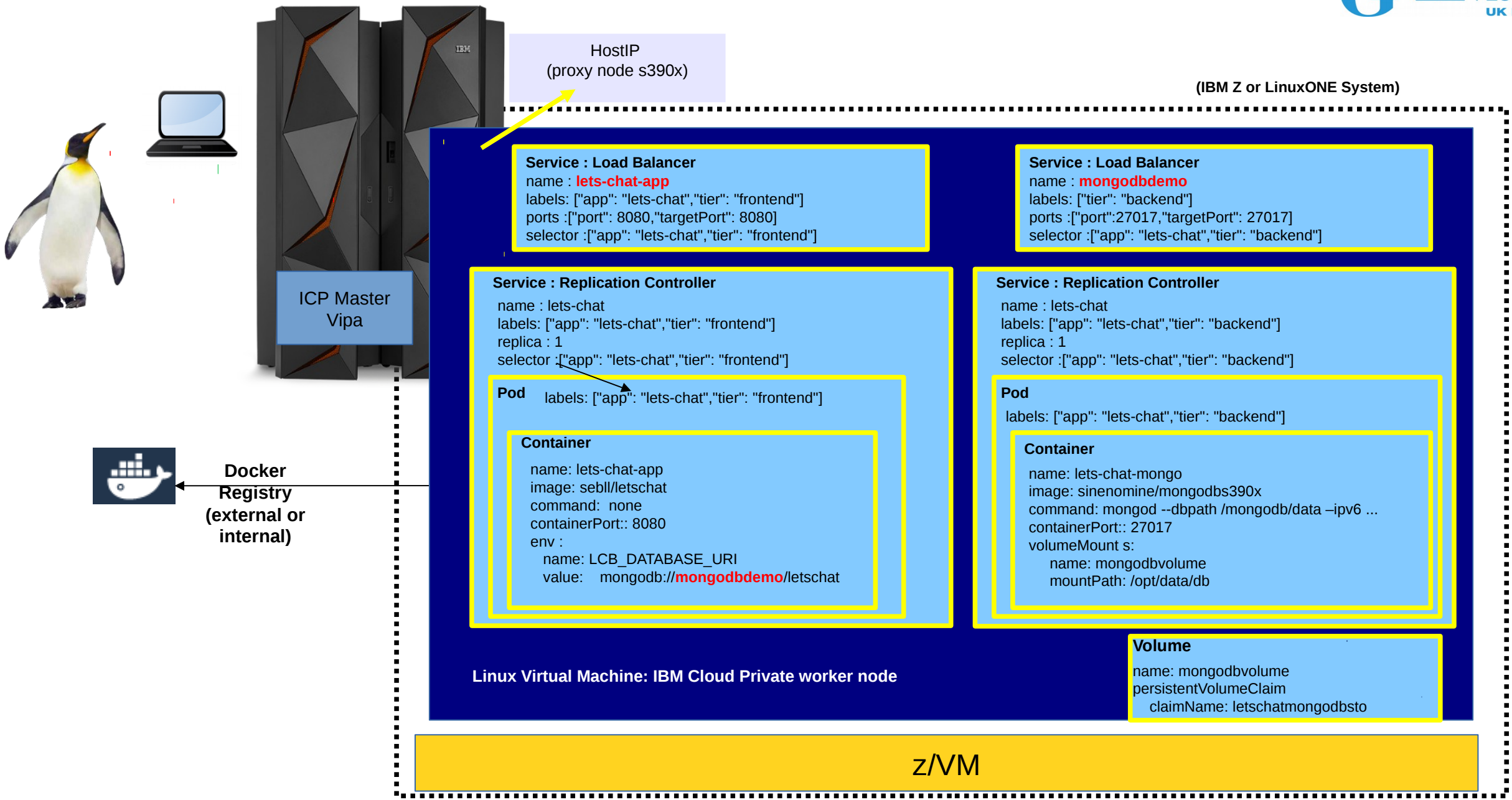
The IBM Cloud Private **Master** node can be deployed on IBM Z



If you have an existing x86 environment, an IBM Cloud Private master can be re-used.

vCenter infrastructure (x86)




HostIP
(proxy node s390x)
(IBM Z or LinuxONE System)
ICP Master
Vipa
Service : Load Balancer
name : lets-chat-app
labels: ["app": "lets-chat","tier": "frontend"]
ports :["port": 8080,"targetPort": 8080]
selector :["app": "lets-chat","tier": "frontend"]
Service : Load Balancer
name : mongodbdemo
labels: ["tier": "backend"]
ports :["port":27017,"targetPort": 27017]
selector :["app": "lets-chat","tier": "backend"]
Service : Replication Controller
name : lets-chat
labels: ["app": "lets-chat","tier": "frontend"]
replica : 1
selector :["app": "lets-chat","tier": "frontend"]
Pod
labels: ["app": "lets-chat","tier": "frontend"]
Container
name: lets-chat-app
image: sebll/letschat
command: none
containerPort:: 8080
env :
name: LCB_DATABASE_URI
value: mongodb://mongodbdemo/letschat
Service : Replication Controller
name : lets-chat
labels: ["app": "lets-chat","tier": "backend"]
replica : 1
selector :["app": "lets-chat","tier": "backend"]
Pod
labels: ["app": "lets-chat","tier": "backend"]
Container
name: lets-chat-mongo
image: sinenomine/mongodbs390x
command: mongod --dbpath /mongodb/data –ipv6 ...
containerPort:: 27017
volumeMount s:
name: mongodbvolume
mountPath: /opt/data/db
Volume
name: mongodbvolume
persistentVolumeClaim
claimName: letschatmongodbsto
Linux Virtual Machine: IBM Cloud Private worker node
Docker
Registry
(external or
internal)
z/VM

73%
Allow root access

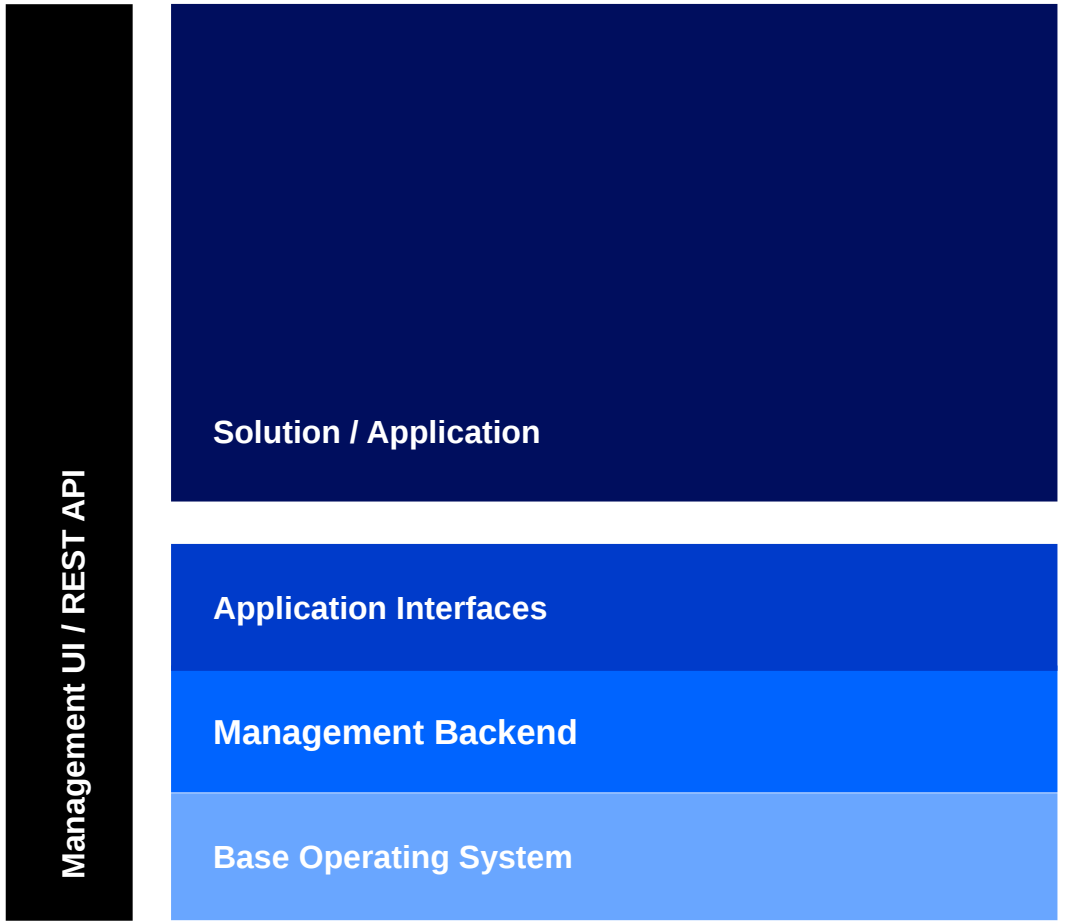
2%
Corporate data encrypted

58%
Threats from insiders

https://www-01.ibm.com/marketing/iwm/dre/signup?source=urx-17425&S_PKG=ov59678&
https://www.techrepublic.com/article/tesla-public-cloud-environment-hacked-attackers-accessed-non-public-company-data/
https://healthitsecurity.com/news/58-of-healthcare-phi-data-breaches-caused-by-insiders

IBM Secure Service Container - Appliance Concept

IBM Secure Service Container Appliance



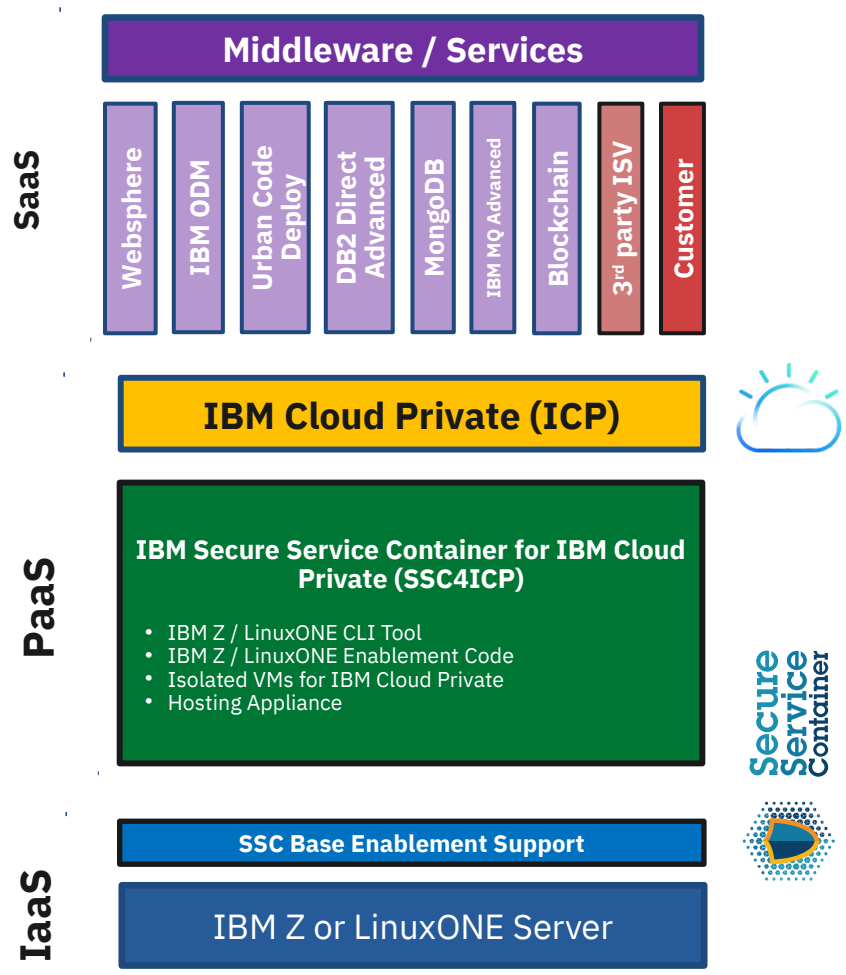
What is **IBM Secure Service Container for IBM Cloud Private**?

IBM Secure Service Container for IBM Cloud Private (SSC4ICP) is a software appliance built on the Secure Service Container framework that **securely hosts IBM Cloud Private Docker / Kubernetes based solutions** on IBM Z or LinuxONE Private and Hybrid cloud deployments.

SSC4ICP provides an encrypted environment (data at rest, data in flight), with peer to peer and **peer to host isolation protecting container applications from access via Hardware and Operating System admin credentials**, whether access is accidental or malicious, internal or external to an organization.

SSC4ICP provides these protections while integrating with **IBM Cloud Private, a Platform as a Service (PaaS)** management stack that delivers **rapid innovation and application modernization, investment leverage, enterprise integration, as well as management and compliance** to containerized applications.

Full Stack Solution



Differentiation: Security and Deployment

Protection from Misuse of Privileged Hardware & Operating System Credentials

Infrastructure management organizations can manage the physical IT infrastructure without having visibility to their end users' applications and customer data

Storage Admin

Automatic File System Encryption (LUKS) – Data at Rest

- Encryption keys stored within appliance, not accessible
- Key Management via appliance life cycle export/import
- Docker container data connected to disk also encrypted

Network Admin

Automatic Network Encryption (TLS) – Data in Flight

- Encrypted management REST API interfaces (i.e. storage, network configuration data, dumps, etc.)

System / Hardware Admin

Encrypted Diagnostic Data (ex: Debug Dump Logs)

- First Failure Data Capture – data required to fix problem
- Dump targets host kernel data (log message buffers, etc.)
- Dump data encrypted – only accessible by service teams
- Alternative to memory display alter – minimal access to customer data

Operating System Admin

No Operating System Access

- No direct Host or OS level interaction - SSH Disabled
- Prevent user traditionally with host OS access from having visibility to application or customer data

Software Appliance Form Factor for Simplified Deployment and Management

Avoid management of low level execution environment

- Appliance encapsulates operating system, virtualization layer, management UI, REST API interface components
- Agile CI/CD update flow of SSC4ICP platform for feature enhancements, security fixes (CVEs), etc.
- Avoid lifecycle management of individual components

Hybrid & Private Cloud Administrators

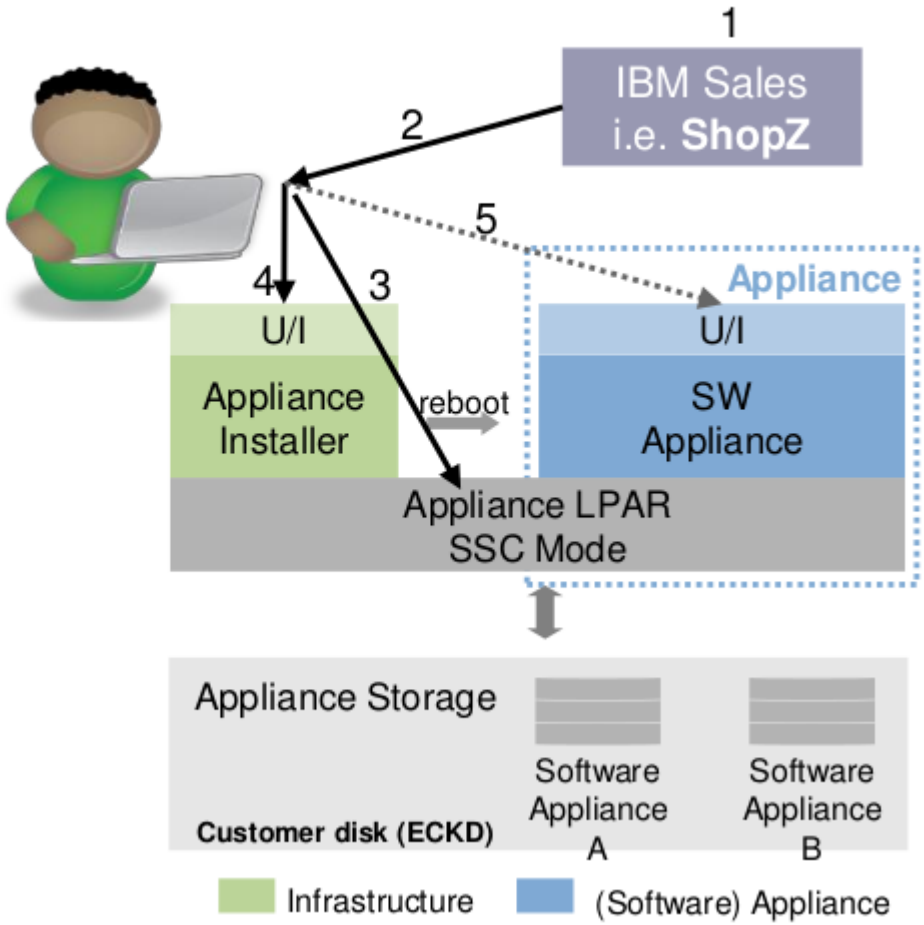
- Focus on deployment of k8s cluster to ICP worker / proxy nodes as infrastructure for containerized workloads

Solution Developers

- Focus on building containerized applications

How to deploy a Secure Service Container Appliance ?

Deployment in 5 steps



1) Buy a Software Appliance (e.g. IOAz, etc…)

2) Download the Appliance image

3) Create and activate an appliance (SSC) LPAR

4) Deploy Appliance using Appliance Installer

5)Configure and use Appliance through REST API or WebUI

IBM Hyper Protect Virtual Servers

Secure cloud evolution & offering migration from Secure Service Container for IBM Cloud Private

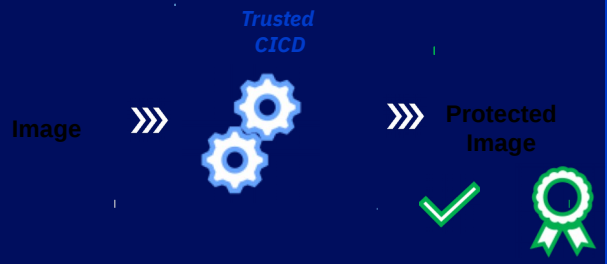
Protect critical Linux workloads during build, deployment, and management on-premise

The Solution

- IBM Hyper Protect Virtual Servers for on-premise deployment

Official Offering Name: TBD

Target Availability: 2H19



Solution Developers

Image Integrity

- Securely build your own images
- Enhance your images via Trusted CICD flow
  - Image signing
  - Image configuration encryption & signing
- Validate code used to build images

Infrastructure Providers

Least Privilege Principle

- Manage infrastructure without access to sensitive data, memory, decryption keys, secrets or application logs
- Manage images via RESTful APIs
- Protect production secrets during image deployment

Solution Admin & Users

Image Provenance

- Validate images originate from trusted source
- Check that no backdoor has been introduced in image builds
- Enable auditors to approve deployment of images

Agenda

- Introducing IBM Z Hybrid Cloud Journey
  - IBM Z:Integration In/Out, IBM Z:Experience, IBM Z:IBM Cloud
  - Application Modernization & Cloud Private
  - Solutions for an Agile Infrastructure
- Discovering the Technical Solution for Secured Deployments
  - Docker, Kubernetes
  - IBM Cloud Private with Linux on IBM Z,
  - IBM Secure Service Container,
- Deployment of IBM Cloud Private for Secure Service Container
- Performing Deployments to ICP for Secure Service Containers
  - Register existing s390x docker images in an ICP registry
  - Create helm chart for your application
  - Deploy your helm chart from the ICP Catalog
  - Work with your containers/logs/debug !
  - Content for some helm chart files
  - Monitor your cluster and your particular application !

IBM Secure Service Container Hills and Personas

SOLUTION DEVELOPER
Rocky

HYBRID CLOUD ADMINISTRATOR
Todd

APPLIANCE ADMINISTRATOR
Matt (former OS admin)

SYSTEM ADMINISTRATOR
Kelly

Hills

Rocky can **build and deploy an application secured by IBM Z / LinuxONE using standardized APIs and tools, without needing mainframe skills**.

Todd can **deploy software container infrastructure on IBM Z / LinuxONE protected against access from other HW/OS privileged user credentials**, **with a unified hybrid cloud experience in minutes.**

Matt can **provide a secure Docker environment** which **integrates with IBM Cloud Private on IBM Z / LinuxONE without Linux admin skills**.

Kelly can **provision the resources for and manage the physical infrastructure required** to host a software container platform while **assuring her end users that she will not have visibility to their application or data**

Skills

docker kubernetes kubernetes IBM Cloud Private Secure Service Container IBM LinuxONE

To Be: Initial Deployment Scenario

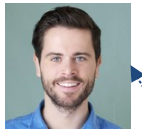
Manual ⟷ Automated ⟷ Scriptable

Todd can **deploy software container infrastructure on IBM Z / LinuxONE protected against access from other HW/OS privileged user credentials**, **with a unified hybrid cloud experience in minutes.**

Matt can **provide a secure Docker environment** which **integrates with IBM Cloud Private on IBM Z / LinuxONE without Linux admin skills**.

Kelly can **provision the resources for and manage the physical infrastructure required** to host a software container platform while **assuring her end users that she will not have visibility to their application or data**

Rocky



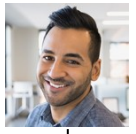
Download
(1) IBM Cloud Private
(2) SSC4ICP
- ✓ CLI Tool
- ✓ Isolated VM Docker Image
- • SSC4ICP Hosting Appliance

Download
(1) SSC4ICP
- • CLI Tool
- • Isolated VM Docker Image
- ✓ SSC4ICP Hosting Appliance

Actions

Todd

Matt

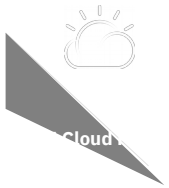


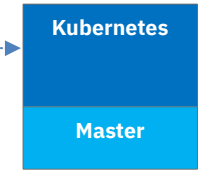
Kelly

1. **Identifies workload platform requirements**
2. **Receive workload platform requirements**
3. **Request HW Resources**
4. **Config / activate SSC Mode LPAR via HMC***
5. **Download & Deploy SSC4ICP appliance** [SSC Installer]
6. **Login to Appliance: Configure data pool and network**
7. **Create / share new admin credentials and SSC LPAR IP address**
8. **Prepare ICP Master** [Docker / CLI Tool]
9. **Configure cluster template**
10. **Setup cluster infrastructure** [CLI Tool]
11. **Config / Launch ICP install**
12. **Deploy solution to appliance**

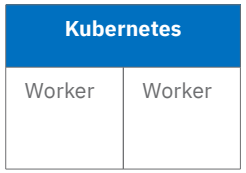
Resources

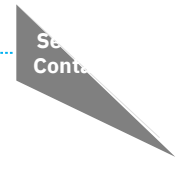
ICP Master
(x86 / POWER)





IBM Z or LinuxONE

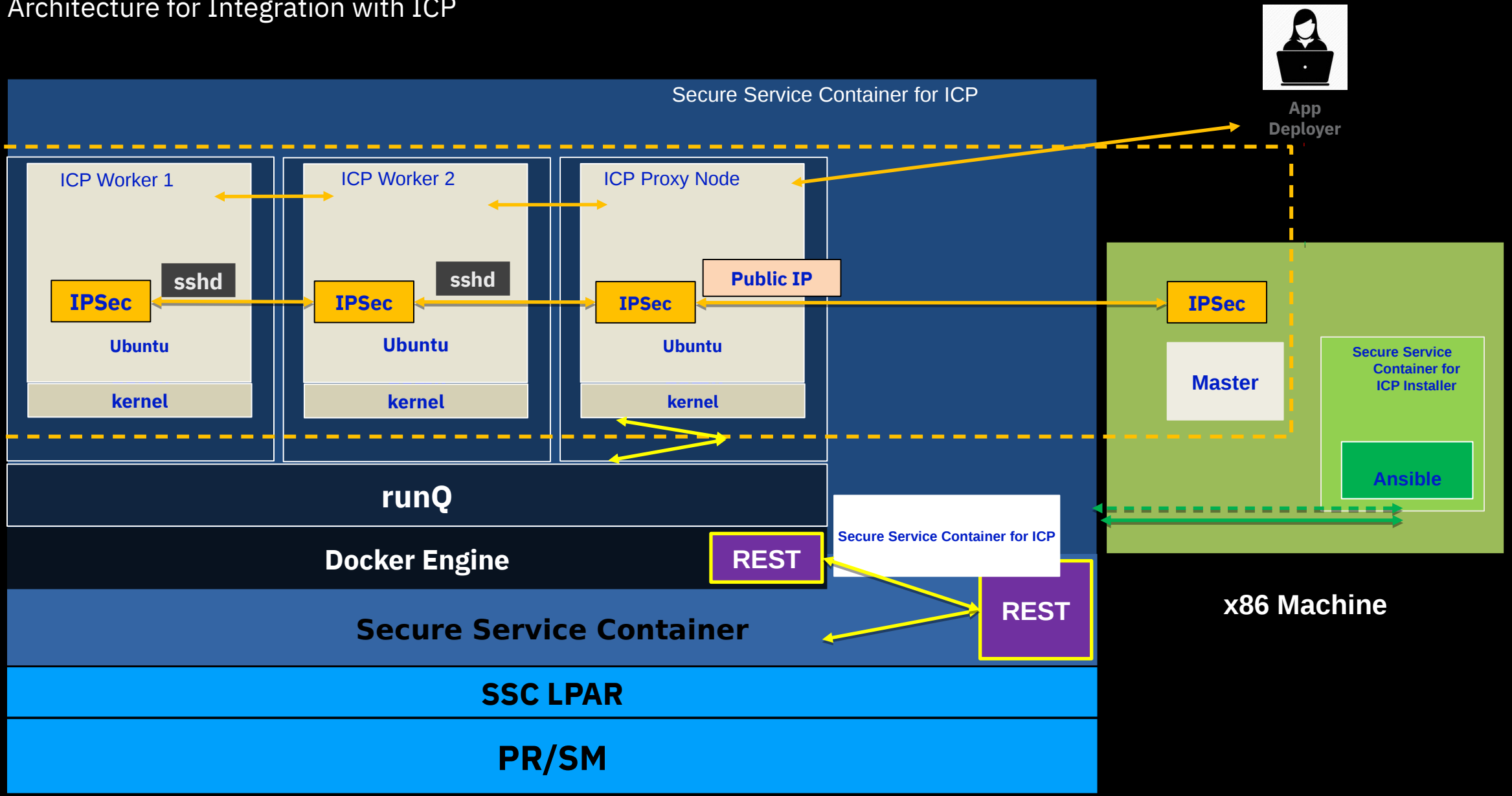




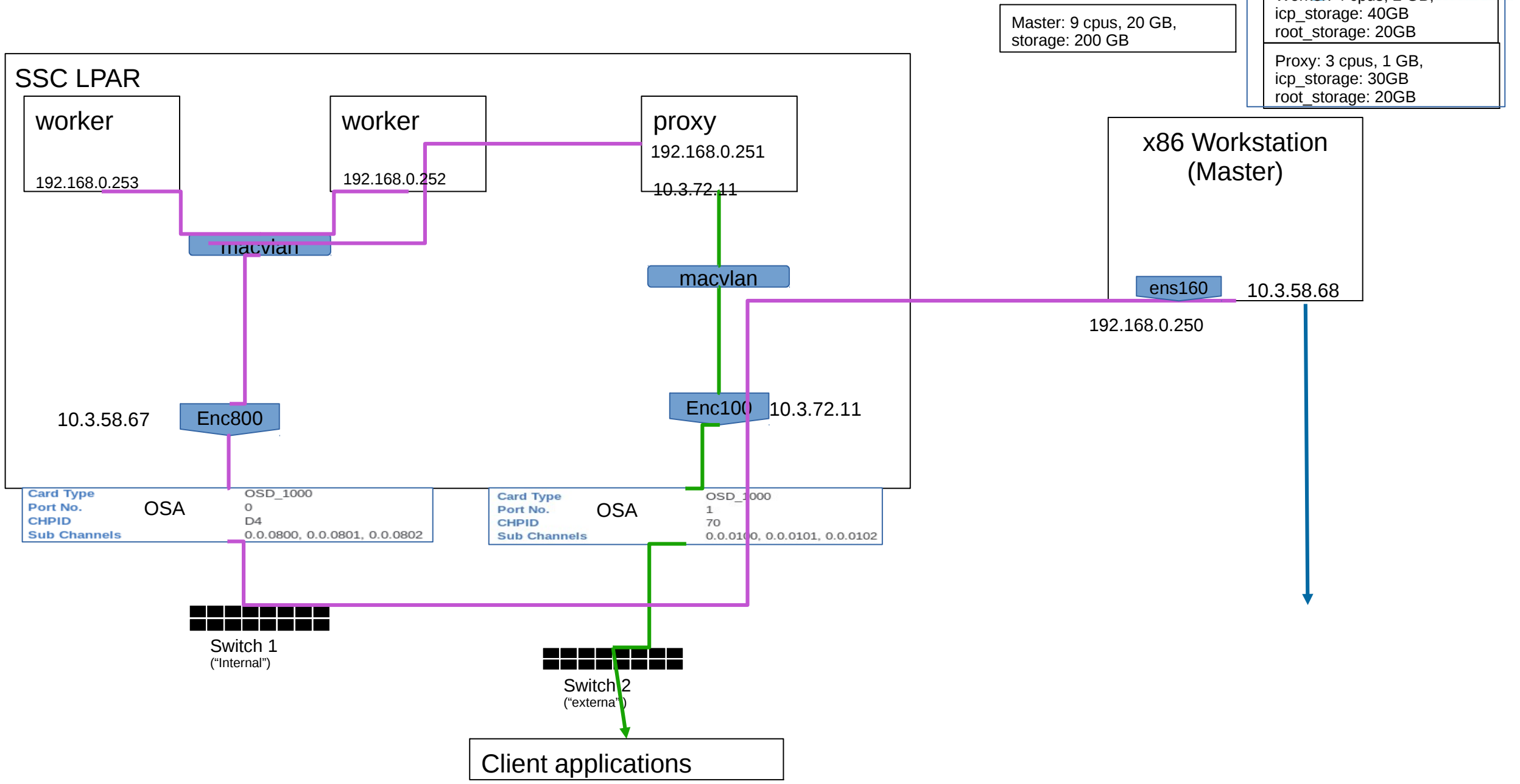
IBM Z or LinuxONE Server

*Classic or DPM mode

Secure Service Container for IBM Cloud Private V1.1.0
Architecture for Integration with ICP



Target architecture to be deployed



REST APIs calls with Secure Service Container 1.1.0 phase 2 code

==========

Generate a authentification token to work with the Appliance

Generic command :

```
curl --request POST --url https://<appliance_IP>/api/com.ibm.zaci.system/api-tokens \
-H 'accept: application/vnd.ibm.zaci.payload+json' -H 'cache-control: no-cache' \
-H 'content-type: application/vnd.ibm.zaci.payload+json;version=1.0' \
-H 'zaci-api: com.ibm.zaci.system/1.0' --insecure \
--data '{ "kind" : "request", "parameters" : { "user" : "<master_id>", "password" : "master_id_password" } }'
```

Filled with your information:

```
curl --request POST --url https://YourIPADDRESS/api/com.ibm.zaci.system/api-tokens --header 'accept:
application/vnd.ibm.zaci.payload+json' --header 'cache-control: no-cache' --header 'content-type:
application/vnd.ibm.zaci.payload+json;version=1.0' --header 'zaci-api: com.ibm.zaci.system/1.0' --insecure
--data '{ "kind" : "request", "parameters" : { "user" : "YourUSERID", "password" : "YourPASSWORD" } }'
```

Output of the command : a Token is generated . We will replace this value by **$token** in the next charts.

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJyb2xlcyI6WyJhZG1pbiJdLCJhZG1pbiI6dHJ1ZSwianRpIjo1MTk3MjMxMSwidXNlciI6InNzY2FkbWluIiwiZXhwIjoxNTUwMjI1NDA4LCJpYXQiOjE1NTAyMjM2MDh9.i72il-c-0hZFoRYT3RvaK3rXjjxkFEi9lt3l1fJlaTw

REST APIs calls with Secure Service Container 1.1.0 phase 2 code

==========

Create the LV Data Pool

Generic command :

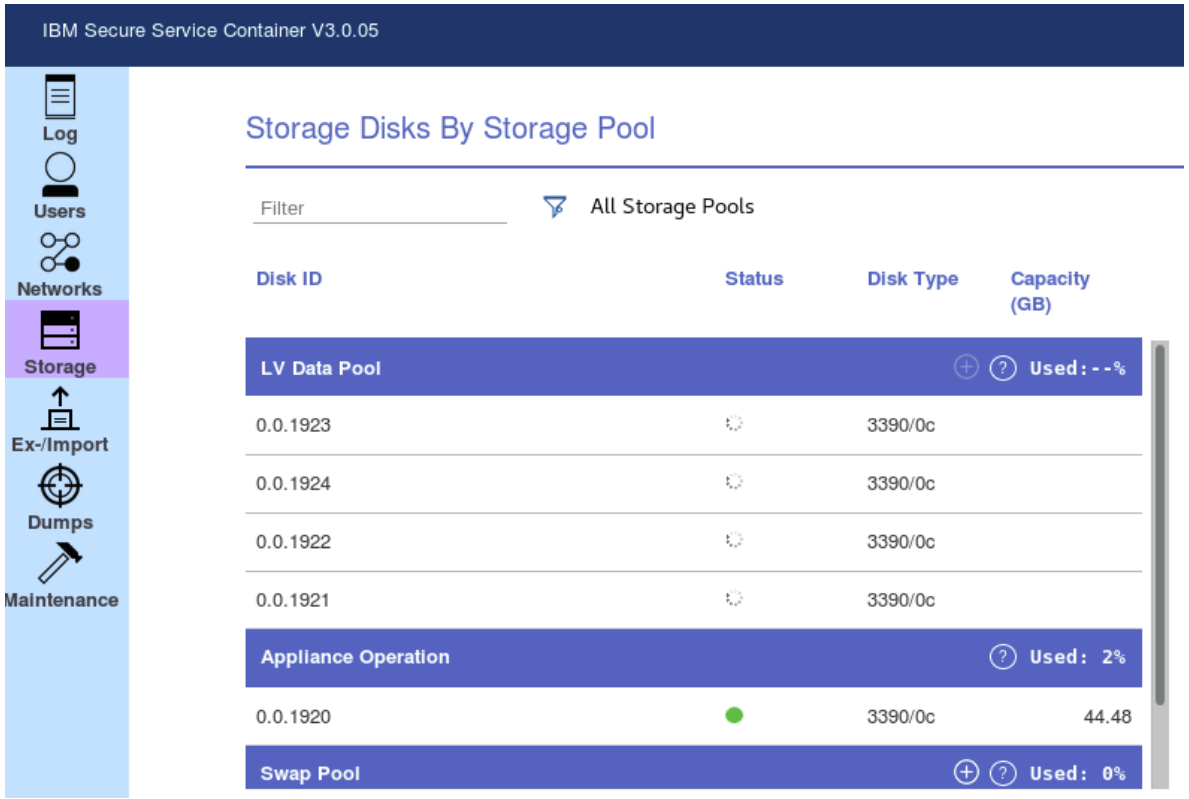
```
curl -X POST https://<appliance_IP>/api/com.ibm.zaci.system/storagepools/lv_data_pool/<subresource> \
-H 'accept: application/vnd.ibm.zaci.payload+json' -H 'cache-control: no-cache' \
-H 'content-type: application/vnd.ibm.zaci.payload+json;version=1.0' \
-H 'zaci-api: com.ibm.zaci.system/1.0' --insecure \
-H 'authorization: Bearer '<TOKEN>" -d '{ "kind": "request", "parameters": { "addDisks": [ "<disk_id1>", <"disk_id2"> ] } }'
```

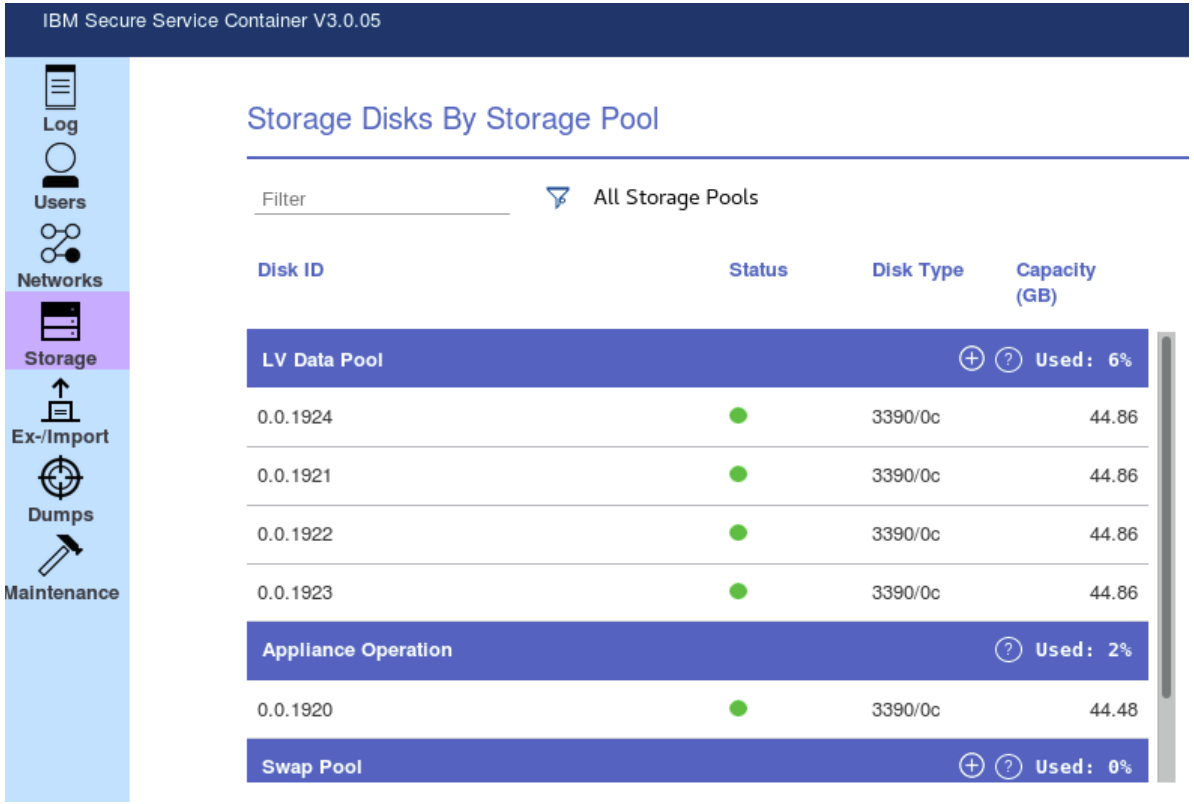
Filled with your information:

```
curl -X POST https://YourIPADDRESS/api/com.ibm.zaci.system/storagepools/lv_data_pool/storage-devices
--header 'accept: application/vnd.ibm.zaci.payload+json' --header 'cache-control: no-cache' --header 'content-
type: application/vnd.ibm.zaci.payload+json;version=1.0' --header 'zaci-api: com.ibm.zaci.system/1.0' --header
'authorization: Bearer ‘$token” -d '{ "kind": "request", "parameters": { "addDisks": [ "0.0.1921",
"0.0.1922","0.0.1923", "0.0.1924" ] } }' --insecure
```

Output of the command : ***cf. Next slide***

Creating the LV Data Pool





Installation & Configuration

The entire installation process is out of scope for this presentation, focusing on deployment of microservices in to IBM Cloud Private for Secure Service Container but you can access the detailed information here: https://www.ibm.com/support/knowledgecenter/en/SSUPZ7_1.1.0/topics/install_cli.html

- After the LV Data Pool created we can continue with the configuration of the solution (that is: IBM Cloud Private components).
- Mainly we have to start with :
  - docker load < ssc4icp-cli-installer.docker-image.tar
  - docker run --network=host --rm -v $(pwd):/data ibmzcontainers/ssc4icp-cli-installer:1.1.1 cp -r config /data
  - docker run --rm -it --net=host -v $(pwd)/config:/ssc4icp-cli-installer/config ibmzcontainers/ssc4icp-cli-installer:1.1.1 install

OUTPUT cf. Next slide.

Installation through Ansible

- Monday 18 February 2019 15:49:06 +0000 (0:00:00.033)   0:02:33.368 *******
- ok: [localhost]

- PLAY RECAP ***
- YourIPADDRESS   : ok=3   changed=2   unreachable=0   failed=0
- localhost   : ok=70   changed=4   unreachable=0   failed=0

- Monday 18 February 2019 15:49:07 +0000 (0:00:00.505)   0:02:33.873 *******
- ===
- cli-base : Load image https://YourIPADDRESS/api/com.ibm.zaas/images/load -- 35.49s
- master-node-setup : Docker check ------------------------------------ 8.71s
- cli-base : Create datapool https://YourIPADDRESS/api/com.ibm.zaas/quotagroups --- 8.54s
- master-node-setup : Check ping --------------------------------------- 7.74s
- cli-base : Create container https://YourIPADDRESS/api/com.ibm.zBlockchain/containers proxy-16001 --- 2.87s
- Check if IsolatedVM tar is available for loading ---------------------- 2.60s
- master-node-setup : SSH key generation ------------------------------- 1.76s
- Checking Python interpreter on Master node --------------------------- 1.39s
- cli-base : https://YourIPADDRESS/api/com.ibm.zaas/networks/enc100_proxy_network --- 0.95s
- cli-base : Create macvlan for containers https://YourIPADDRESS/api/com.ibm.zBlockchain/networks --- 0.78s
- cli-base : https://YourIPADDRESS/api/com.ibm.zaas/networks/enc800_network --- 0.72s
- cli-base : LPAR API token https://YourIPADDRESS/api/com.ibm.zaci.system/api-tokens --- 0.71s
- cli-base : Create macvlan for containers https://YourIPADDRESS/api/com.ibm.zBlockchain/networks --- 0.70s
- configuration : Validate IPs given in the configuration ---------------- 0.57s
- lpar-node-setup : Retrieve image tag for IsolatedVM ------------------- 0.55s
- Generate ipsec config for master ------------------------------------- 0.54s
- install-status : Update Installation status --------------------------- 0.51s
- configuration : Generate cluster configuration file ------------------- 0.50s
- lpar-node-setup : Retrieve image tag for IsolatedVM ------------------- 0.36s
- include_role : configuration --- 0.15s

- [sebll@oc0787150785 ICP312]$--
- --
- End of execution. Creation of 2 workers and 1 proxy

REST APIs calls with Secure Service Container 1.1.0 phase 2 code

==========

GET LIST OF WORKER NODES

Filled with your information:

```
curl -X GET https://10.3.58.67/api/com.ibm.zBlockchain/containers --header 'accept:
application/vnd.ibm.zaci.payload+json' --header 'cache-control: no-cache' --header 'content-type:
application/vnd.ibm.zaci.payload+json;version=1.0' --header 'zaci-api: com.ibm.zaci.system/1.0' --header
'authorization: Bearer '$token" --insecure
```

OUTPUT

{"e1de77bbd72e0e3f4ca58245353f00d2c3c879758a4dda3dad91a12e3490b46b": {"Image": "ibmzcontainers/isolated_vm_icp:1.1.1", "Names": ["/worker-15001"], "State": "running", "Status": "Up 6 days"}, "e81d0974e3a7dc211887d1ea6626099b95fc5e680c5e9169a27b674bcc5bebec": {"Image": "ibmzcontainers/isolated_vm_icp:1.1.1", "Names": ["/worker-15002"], "State": "running", "Status": "Up 6 days"}, "f4b8008910ffe14b44ddc0ec7ac381a669e20a35ceb41afbb2ebfd063ad7f8ff": {"Image": "ibmzcontainers/isolated_vm_icp:1.1.1", "Names": ["/proxy-16001"], "State": "running", "Status": "Up 6 days"}}

REST APIs calls with Secure Service Container 1.1.0 phase 2 code

==========

INCREASE DATA POOL SIZE

Filled with your information:

```
curl -X PUT https://10.3.58.67/api/com.ibm.zaas/quotagroups/appliance_data -H 'Content-Type:
application/vnd.ibm.zaci.payload+json;version=1.0' -H 'zACI-API: com.ibm.zaci.system/1.0' -H 'authorization:
Bearer '$token'' -d '{ "size": "20","size_unit": "GB"}' --insecure
```

QUERY QUOTAGROUPS

Filled with your information:

```
curl -X GET https://10.3.58.67/api/com.ibm.zaas/quotagroups -H 'Content-Type:
application/vnd.ibm.zaci.payload+json;version=1.0' -H 'zACI-API: com.ibm.zaci.system/1.0' -H 'authorization:
Bearer '$token'' --insecure
```

REST APIs calls with Secure Service Container 1.1.0 phase 2 code

==========

GET LIST OF WORKER NODES

Filled with your information:

```
curl -X GET https://10.3.58.67/api/com.ibm.zBlockchain/containers --header 'accept:
application/vnd.ibm.zaci.payload+json' --header 'cache-control: no-cache' --header 'content-type:
application/vnd.ibm.zaci.payload+json;version=1.0' --header 'zaci-api: com.ibm.zaci.system/1.0' --header
'authorization: Bearer '$token'' --insecure
```

START A WORKER NODE

Filled with your information:

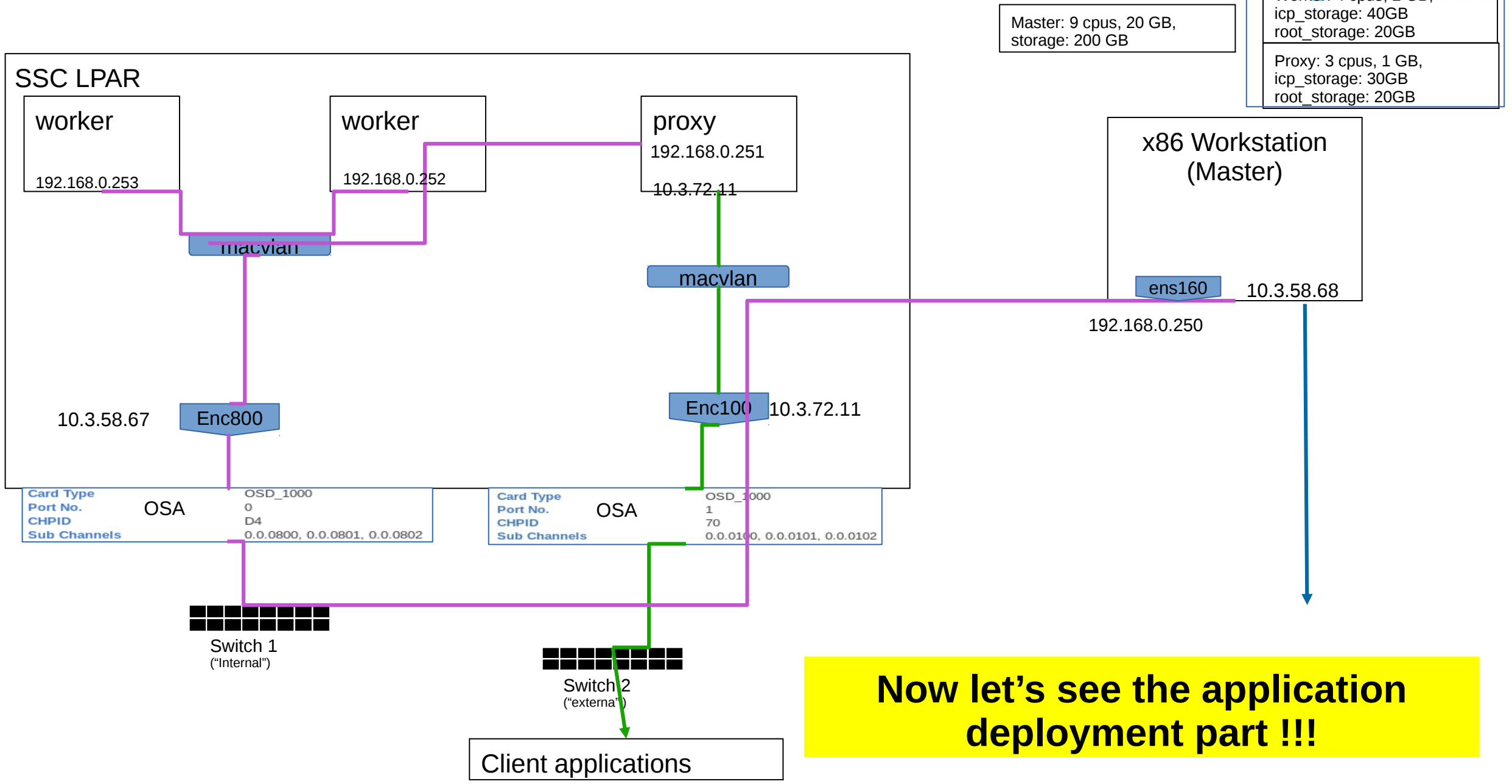
```
curl -X POST https://10.3.58.67/api/com.ibm.zBlockchain/containers/worker-15001/start --header 'accept:
application/vnd.ibm.zaci.payload+json' --header 'cache-control: no-cache' --header 'content-type:
application/vnd.ibm.zaci.payload+json;version=1.0' --header 'zaci-api: com.ibm.zaci.system/1.0' --header
'authorization: Bearer '$token'' --insecure
```

STOP A WORKER NODE

Filled with your information:

```
curl -X POST https://10.3.58.67/api/com.ibm.zBlockchain/containers/worker-15001/stop --header 'accept:
application/vnd.ibm.zaci.payload+json' --header 'cache-control: no-cache' --header 'content-type:
application/vnd.ibm.zaci.payload+json;version=1.0' --header 'zaci-api: com.ibm.zaci.system/1.0' --header
'authorization: Bearer '$token'' --insecure
```

Target architecture **deployed**

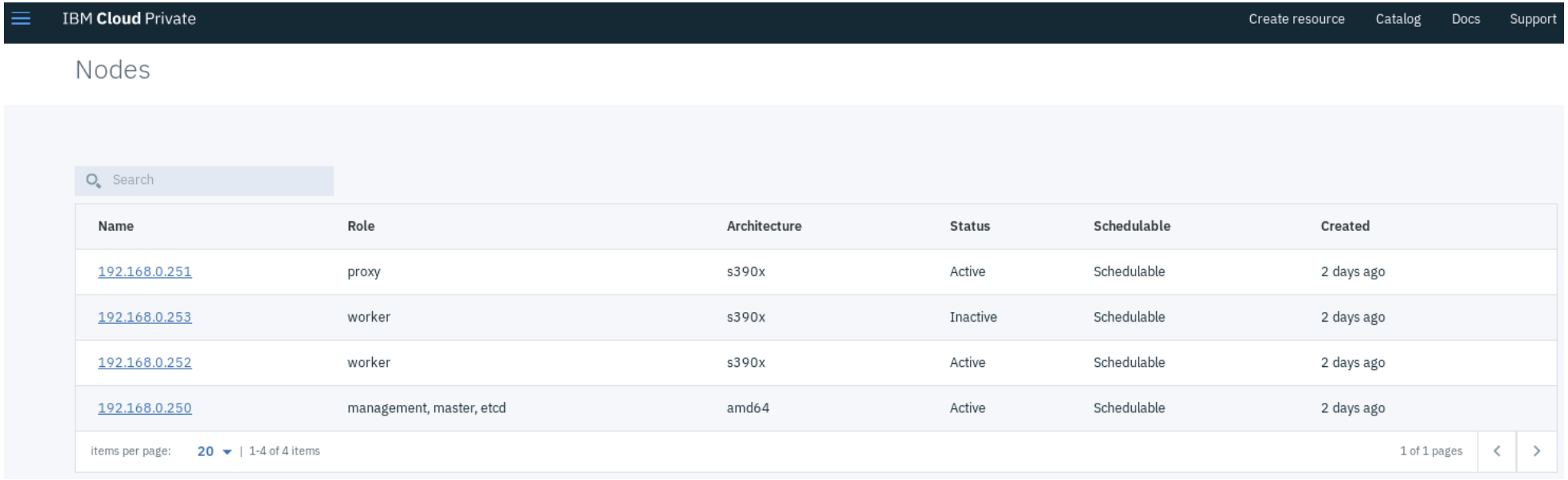


Now let's see the application deployment part !!!

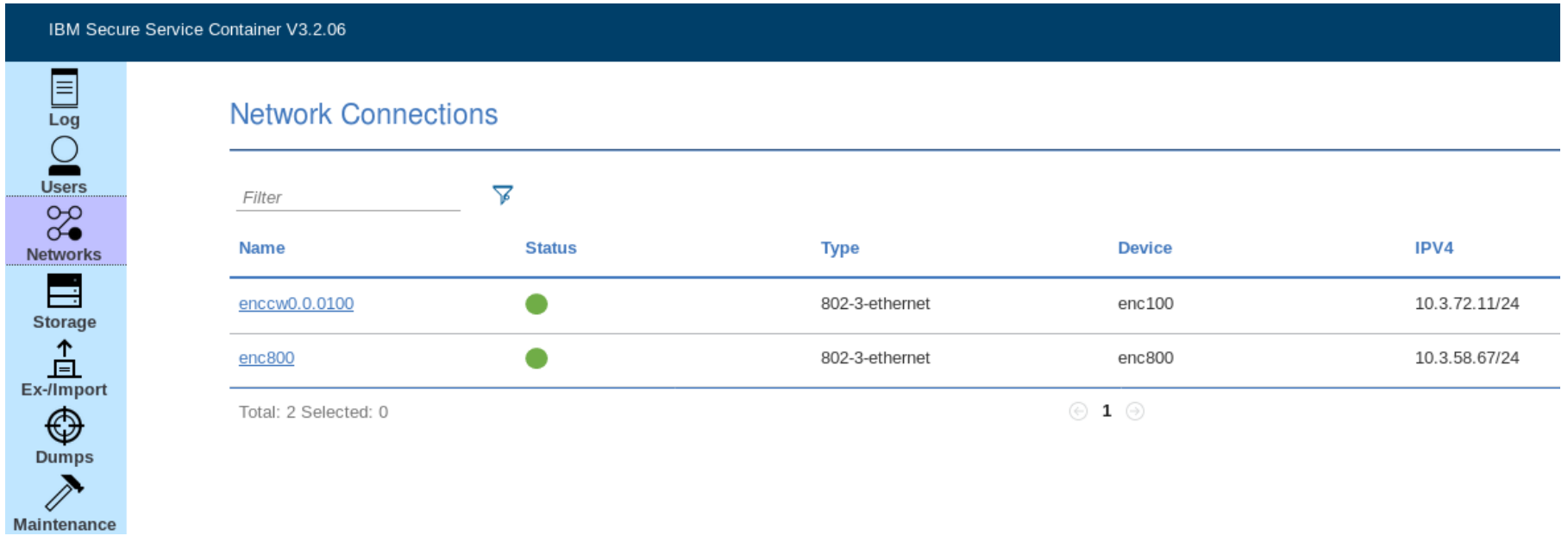
Live connection to the administration interfaces

GUIDE SHARE EUROPE UK REGION

ICP

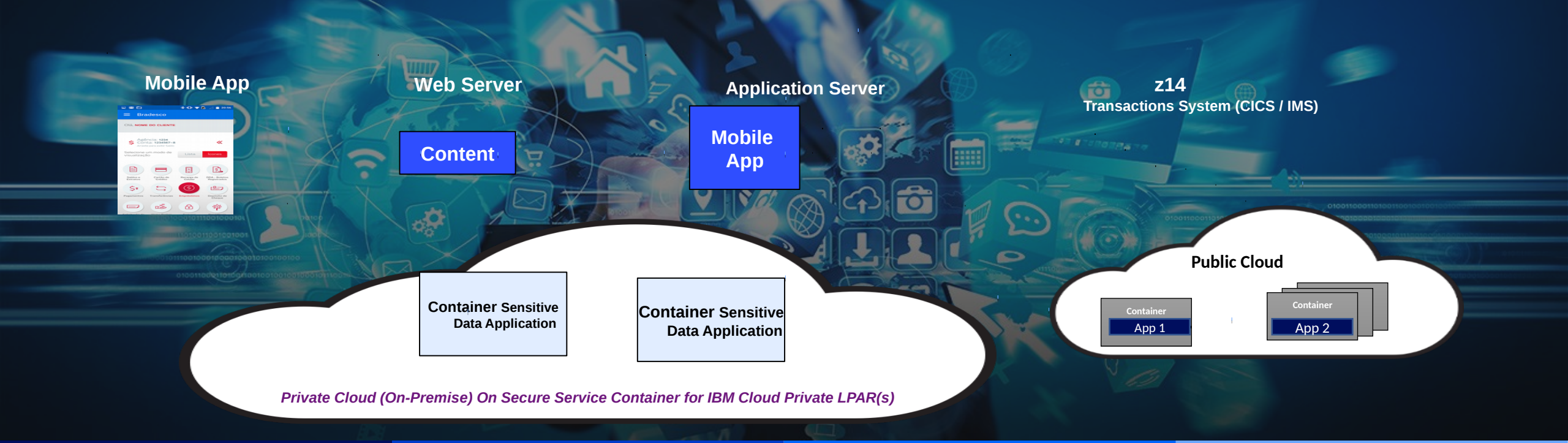


Secure Service Container Appliance



Agenda

- Introducing IBM Z Hybrid Cloud Journey
  - IBM Z:Integration In/Out, IBM Z:Experience, IBM Z:IBM Cloud
  - Application Modernization & Cloud Private
  - Solutions for an Agile Infrastructure
- Discovering the Technical Solution for Secured Deployments
  - Docker, Kubernetes
  - IBM Cloud Private with Linux on IBM Z,
  - IBM Secure Service Container,
- Deployment of IBM Cloud Private for Secure Service Container
- Performing Deployments to ICP for Secure Service Containers
  - Register existing s390x docker images in an ICP registry
  - Create helm chart for your application
  - Deploy your helm chart from the ICP Catalog
  - Work with your containers/logs/debug !
  - Content for some helm chart files
  - Monitor your cluster and your particular application !



Use Case: Application Modernization and Migration to Cloud

Problem

- Need to move some legacy workloads to on-premise cloud

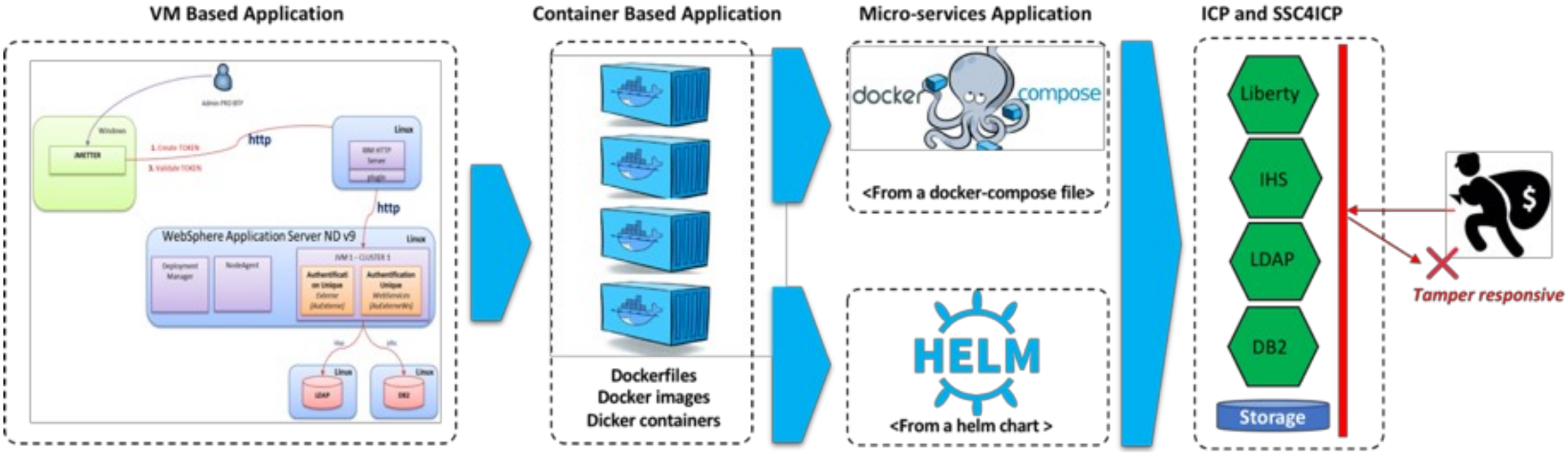
IBM Solution

- Containerize Sensitive Application and deploy in Encrypted Secure Service Container Environment

Target Outcome

- Modernize workload and deploy such that infrastructure teams cannot access

Proof of concept performed with IBM Cloud Private for Secure Service Container



1 - "ClientApp" Transformation & Modernization

Before being able to be deploy "Application" on ICP or ICP for Secure Service Container, we had, first, to dockerized it. This was a long journey requiring collaboration with Customer's team having knowledge of the "App" application.

2 - "ClientApp" Service orchestration

3 - Service Deployment

Secure Service Container

IBM Z Architecture capabilities serving Microservices

Domain	Feature / QoS	Benefits
Security & availability	IBM Z RAS features	Microservices are designed for failure and need HA and failover
Security and encryption	CPACF, Crypto-Express Cards, Post Quantum Algorithms	Massive number of secured Microservices on one physical server and and high speed encryption
Isolation (EAL5 LPARs)	Higest level for non-military environments	Most securable server & Highest certified multi-tenant security
Secure Service Containers	Unique offering on the market. Available for Private Cloud	Focusing on consumability and security for your containerized applications
HA/DR (Processes & features, GDPS)	GDPS Appliance	Easy failover in case of errors or service crash
Performance & throughput	Performance (Proc, Java optim., zEDC, …)	Improved quality of service with hardware innovations
Scale-up and scale-out	Both directions	Perfect fit for dynamic workloads, due to vertical scalability
Hipersocket network	Fast in-memory communication	Latency highly reduced as many interactions occur for micro-services
I/O performances and throughput	OSA Cards Express6S+, ROCe,	Memory to memory communication
Flexibility & optimisation	Capacity on Demand	Can support unpredicted Microservices growth and spin
High level of resource sharing (CPU pooling,..)	Weights on LPAR, priority of ressources	Fine granular capacity allocation & sharing through virtualization
Virtualization technologies (PR/SM, z/VM, KVM)	Choice for the most appropriate technology for your workloads	Useful for individual Microservice scalability
Ease of use (IBM Wave, DPM)	Visual & Simplified System Management	Manage easily your underlying infrastructure

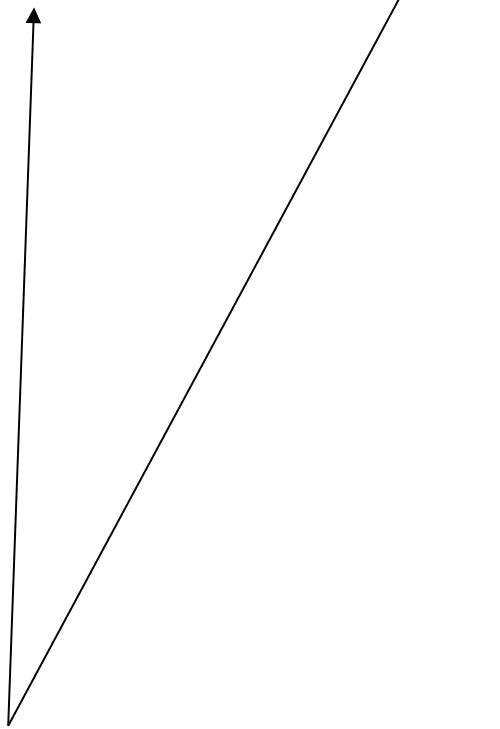
Steps for deployments

1. Register existing s390x docker images in an ICP registry
2. Create helm chart for your application
3. Deploy your helm chart from the ICP Catalog
4. Work with your containers/logs/debug !

1 - Register existing docker images in ICP registry

Starting from Docker-Compose files

```
version: '3'
services:
  isds:
    image: pocssc.auext.isds:latest
    container_name: ldappocsscv1
    ipc: host
    privileged: true
    stdin_open: true
    tty: true
    cap_add:
      - IPC_OWNER
    ports:
     - 9990:9990
    networks:
     - pocsscv1_nw
```



```
db2:
    image: pocssc.auextdb.db2express105:2.0
    container_name: db2pocsscv1
    hostname: db2
    environment:
    - LICENSE=accept
    - DB2INSTANCE=db2inst1
    - DB2INST1_PASSWORD=xxxxxx
    - DBNAME=db2inst1
    - BLU=false
    - ENABLE_ORACLE_COMPATIBILITY=false
    - UPDATEAVAIL=NO
    - TO_CREATE_SAMPLEDB=false
    - REPODB=false
    - IS_OSXFS=false
    - PERSISTENT_HOME=true
    - HADR_ENABLED=false
    - ETCD_ENDPOINT=
    - ETCD_USERNAME=
    - ETCD_PASSWORD=
    ports:
     - 5000:5000
    privileged: true
    networks:
     - pocsscv1_nw
```

Docker Images created for Linux s390x

Step 0 (starting point) : An initial set of docker images created for s390x

```
mplbank@mplbank:~$ sudo docker images | grep dipi-sit-syst
dipi-sit-syst/pocssc.auext.isds                  latest    8a73c46ff31e   3 days ago    4.89GB
dipi-sit-syst/pocssc.auextdb.db2express105       2.0       961bb2d54501   5 weeks ago   2.93GB
dipi-sit-syst/pocssc.auext.liberty18002          latest    3244b9ad1145   5 weeks ago   553MB
dipi-sit-syst/pocssc.auextws.was9009             latest    dbf4826deb26   6 weeks ago   1.73GB
dipi-sit-syst/pocssc.auext.ihs9008               latest    88e5da192760   6 weeks ago   581MB
mplbank@mplbank:~$
```

Step 1 : Tag your images for ICP local registry

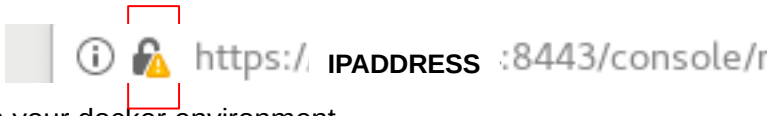
 # sudo docker tag dipi-sit-syst/pocssc.auext.liberty18002 mycluster.icp:8500/default/pocssc.auext.liberty18002:v1

Tag them in as a “default” namespace

Step 2 : Login* to the ICP local registry

```
mplbank@mplbank:~$ sudo docker login mycluster.icp:8500
Username (admin): admin
Password:
Login Succeeded
mplbank@mplbank:~$
```

Note, in order to login from a remote machine you may need to have the certificate
Open a browser to your ICP url and download the certifcate, click on the icon.



Copy this certificate in your docker environment

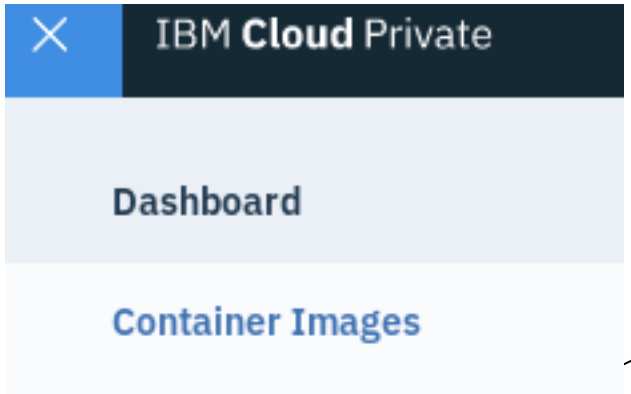
 # cp icp-router311MPLbank.crt /etc/docker/certs.d/mycluster.icp\:8500/ca.crt

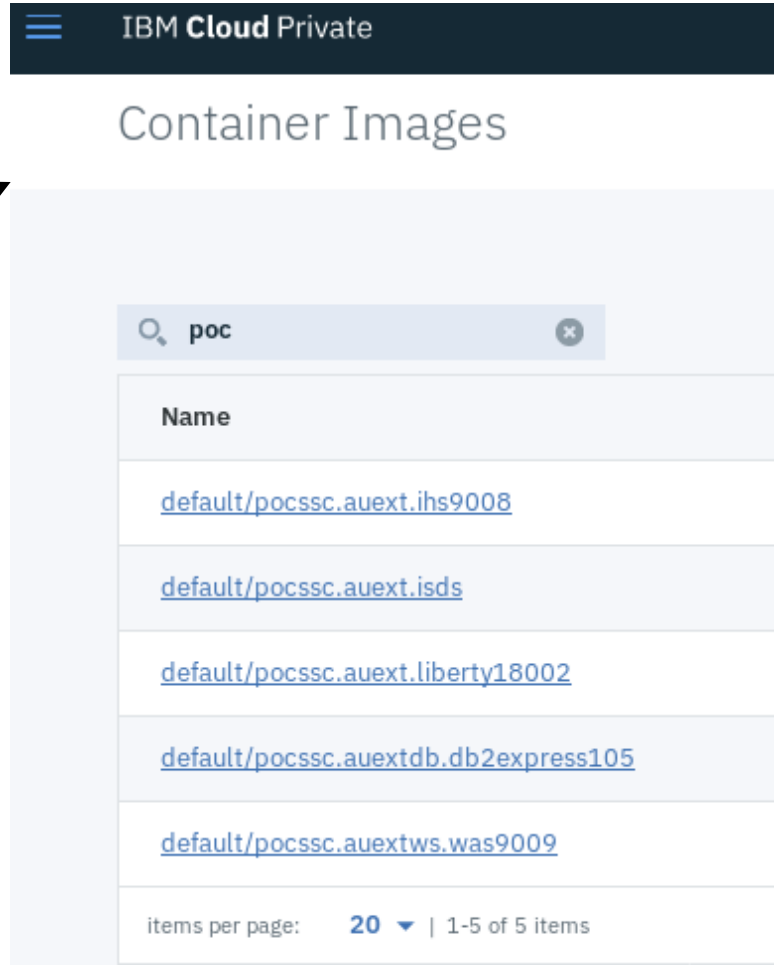
Step 3 : Push the docker images into the ICP local registry

 # sudo docker push mycluster.icp:8500/default/pocssc.auext.ihs9008:v1

Step 4 : Verify all the docker images are present in ICP

Login to the graphical Interface of ICP and select Container images

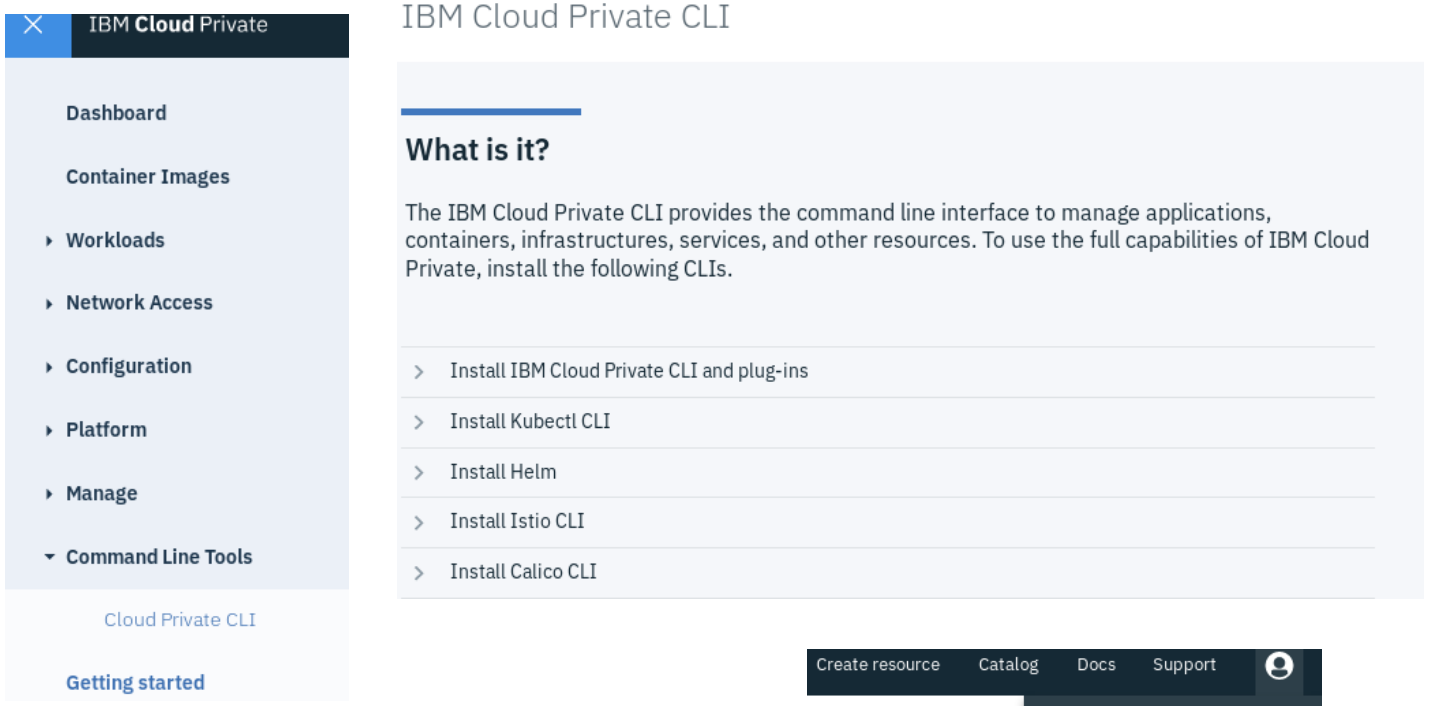




2 - Create helm chart for your application

Step 0: Prepare you remote machine to work with your ICP master

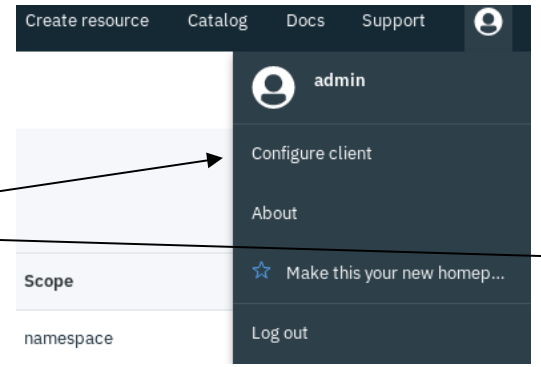
A- Connect to your ICP via a web browser, Download & install all the tooling
Go to Command Line Tools/ Cloud Private CLI



B- Intall the 3 first tools:

- ***ICP CLI & plug-ins***
- ***Kubectl CLI***
- ***Helm***

C- Once the tools installed, click on configure Client (right-top corner in your interface) & Execute all the command listed on your local machine.



Step 1 : Login to ICP API

 # cloudctl login -a https://YourIPADDRESS:8443 --skip-ssl-validation

Step 2 : Generate helm chart structure & build the content for your helm chart files

helm init ⟶

```
[sebll@oc0787150785 HelmChart]$ helm init
$HELM_HOME has been configured at /home/sebll/.helm.
Warning: Tiller is already installed in the cluster.
(Use --client-only to suppress this message, or --upgrade to upgrade Tiller to the current version.)
Happy Helming!
[sebll@oc0787150785 HelmChart]$
```

helm create aue

```
[sebll@oc0787150785 HelmChart]$ tree aue
aue-
├── charts
├── Chart.yaml
├── README.md
├── templates
│   ├── db2.yaml
│   ├── _helpers.tpl
│   ├── ihs.yaml
│   ├── ingress.yaml
│   ├── isds.yaml
│   ├── liberty.yaml
│   ├── NOTES.txt
│   ├── service.yaml
│   └── was.yaml
└── values.yaml

2 directories, 12 files
[sebll@oc0787150785 HelmChart]$
```

The content of each file is provided in the appendix section of this document,
so you can correct/adapt it for your needs.

Step 3 : Validate your helm chart & create the helm archive

Now you have all the files filled-in with your parameters, let's validate the helm chart before publishing it into the ICP catalog.

 # helm lint aue

```
[sebll@oc0787150785 HelmChart]$ helm lint aue
==> Linting aue.
Lint OK

1 chart(s) linted, no failures
[sebll@oc0787150785 HelmChart]$
```

 # helm package aue

```
[sebll@oc0787150785 HelmChart]$ helm package aue.
Successfully packaged chart and saved it to: /home/sebll/Documents/2018Migration/NTC/Innovate/CodePattern/HelmChart/aue .tgz
[sebll@oc0787150785 HelmChart]$
```

Step 3 : Load your helm chart in the ICP Catalog

 # cloudctl catalog load-chart --archive aue.tgz

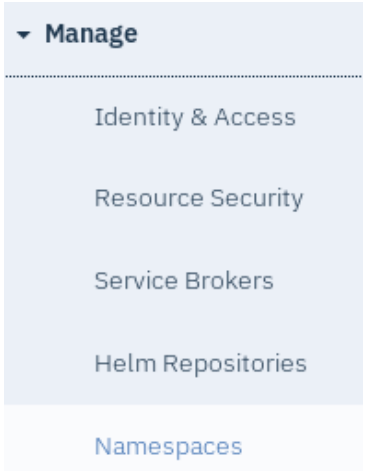
```
[sebll@oc0787150785 HelmChart]$ cloudctl catalog load-chart --archive aue .tgz
Loading helm chart
Loaded helm chart

Synch charts
Synch started
OK
[sebll@oc0787150785 HelmChart]$
```

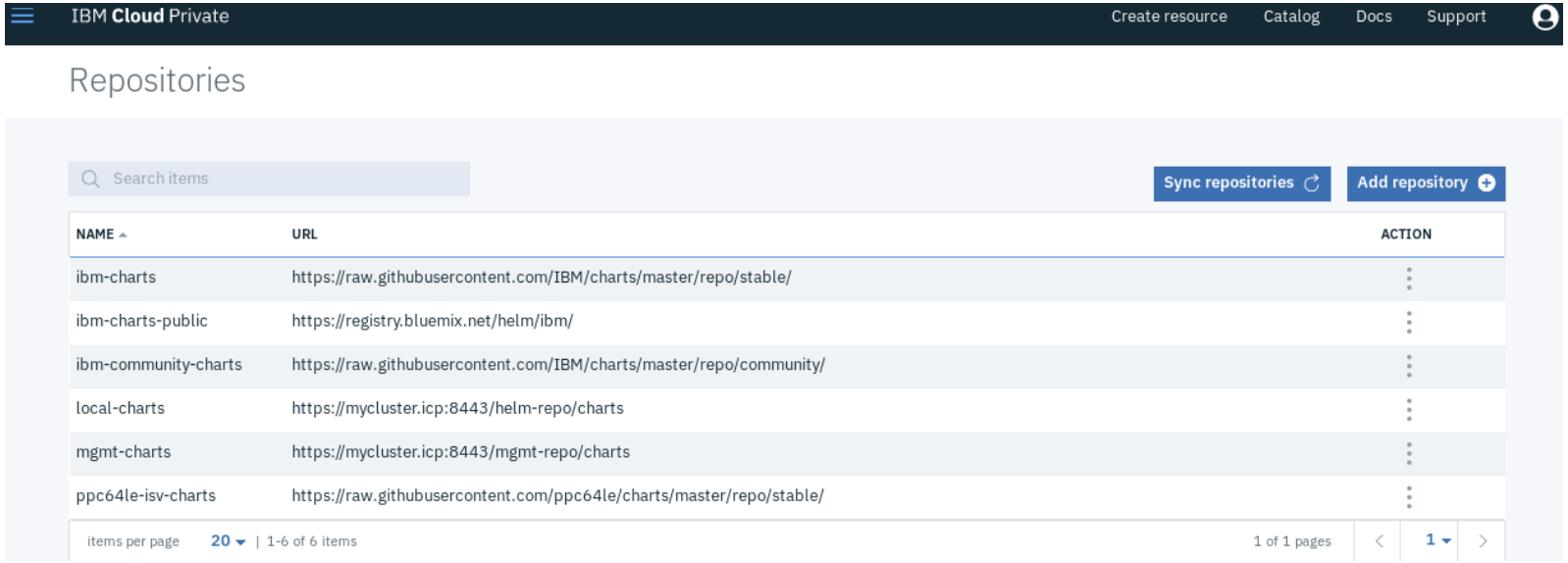
3 - Deploy your helm chart from the ICP Catalog

Step 1 : Ensure you synchronized the ICP repository

Go to Manage>Helm Repositories

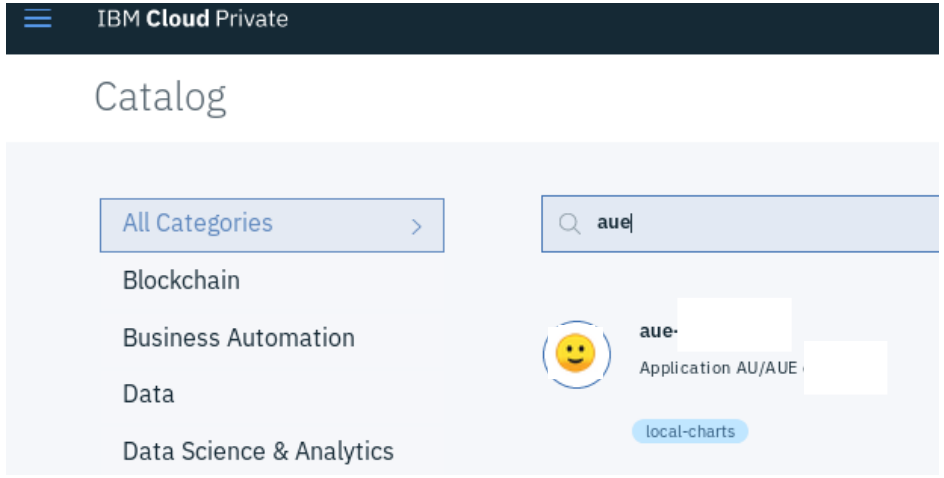


Click on Sync repositories

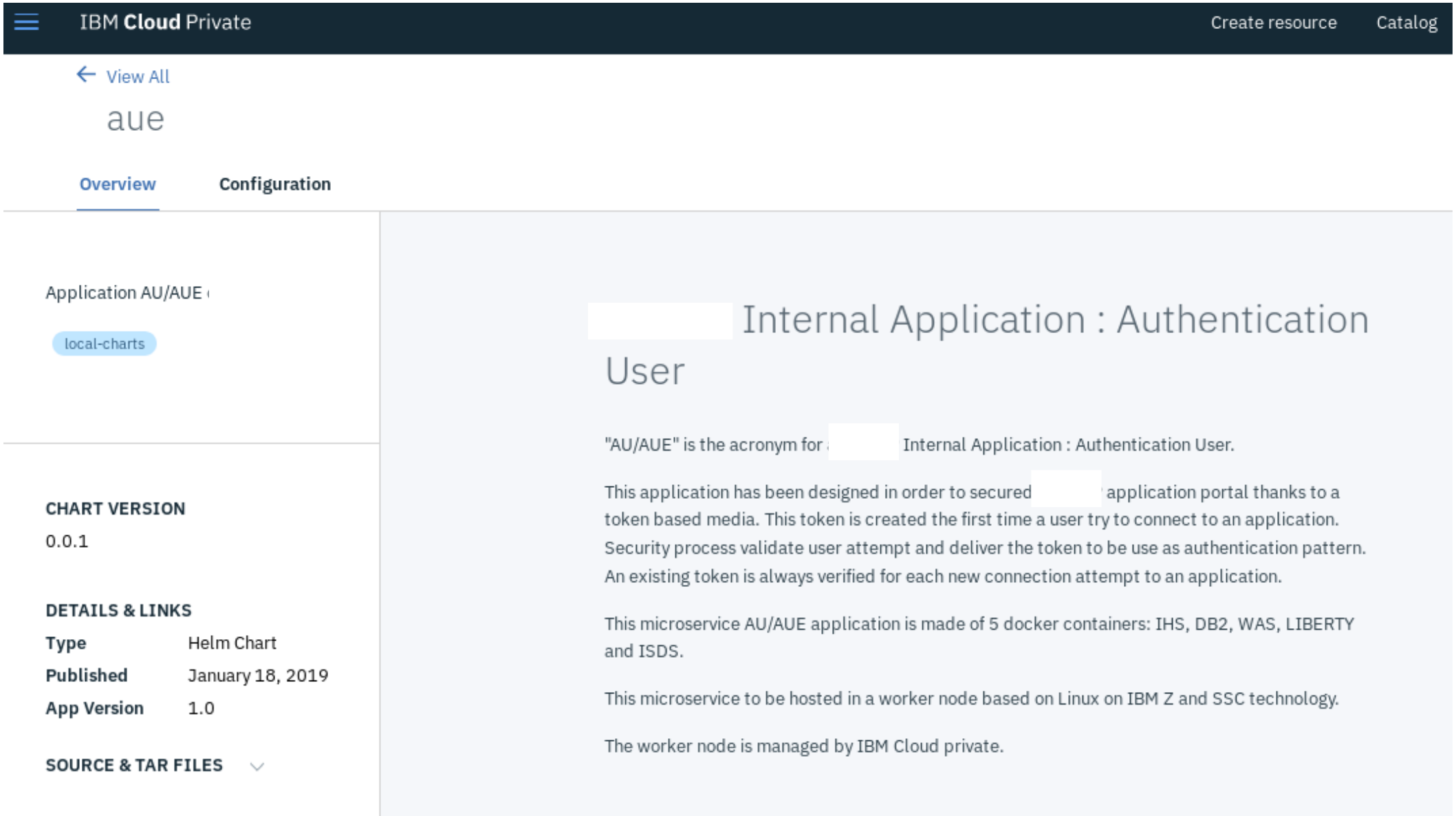


Step 2 : Go to the ICP Catalog

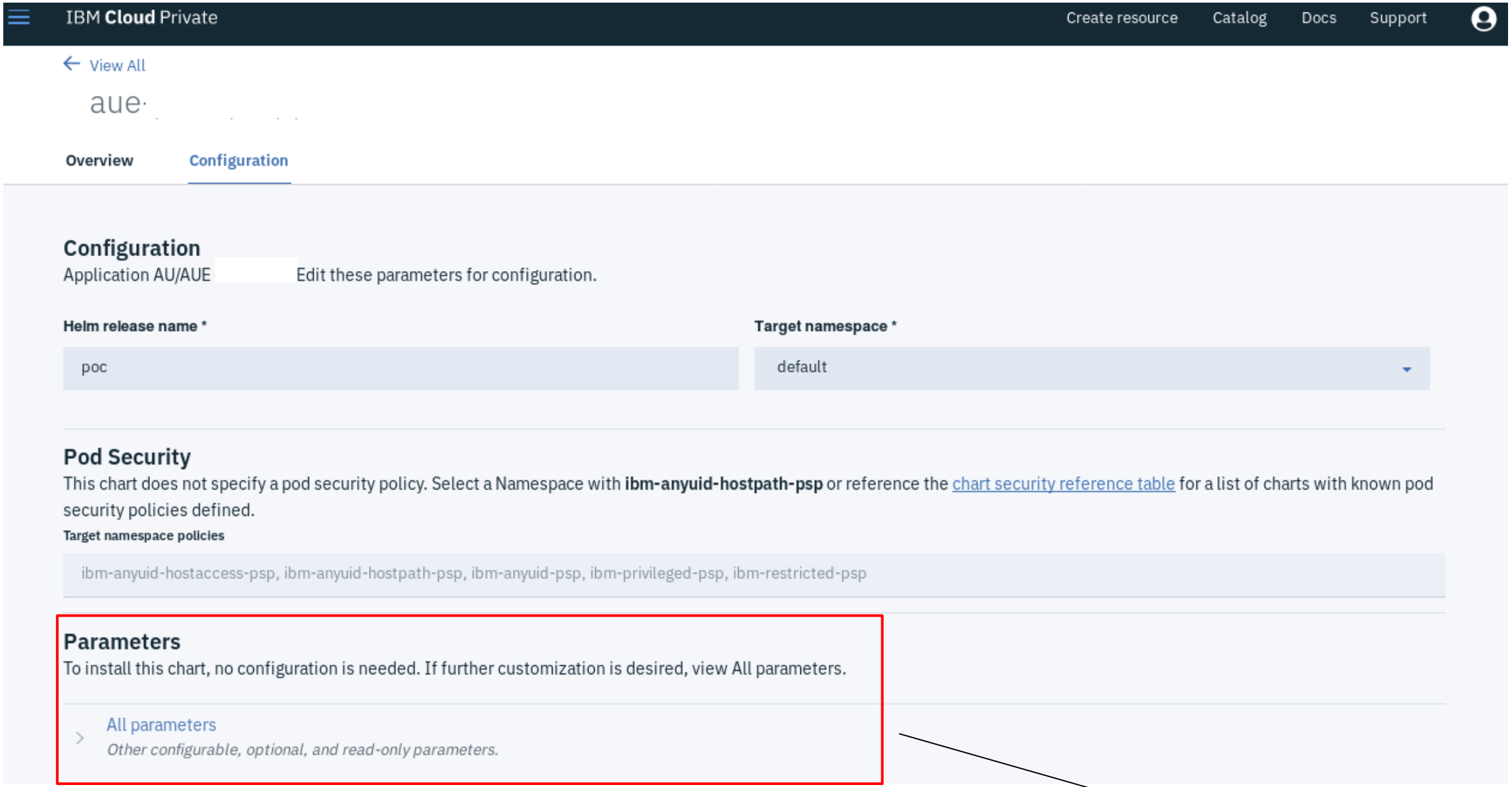
Look for your application



Step 3 : Select your app.& configure it



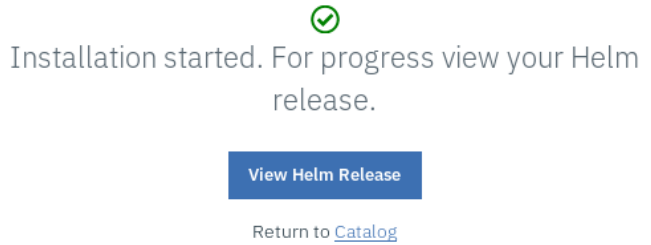
Step 4 : Provide helm release & select default* namespace

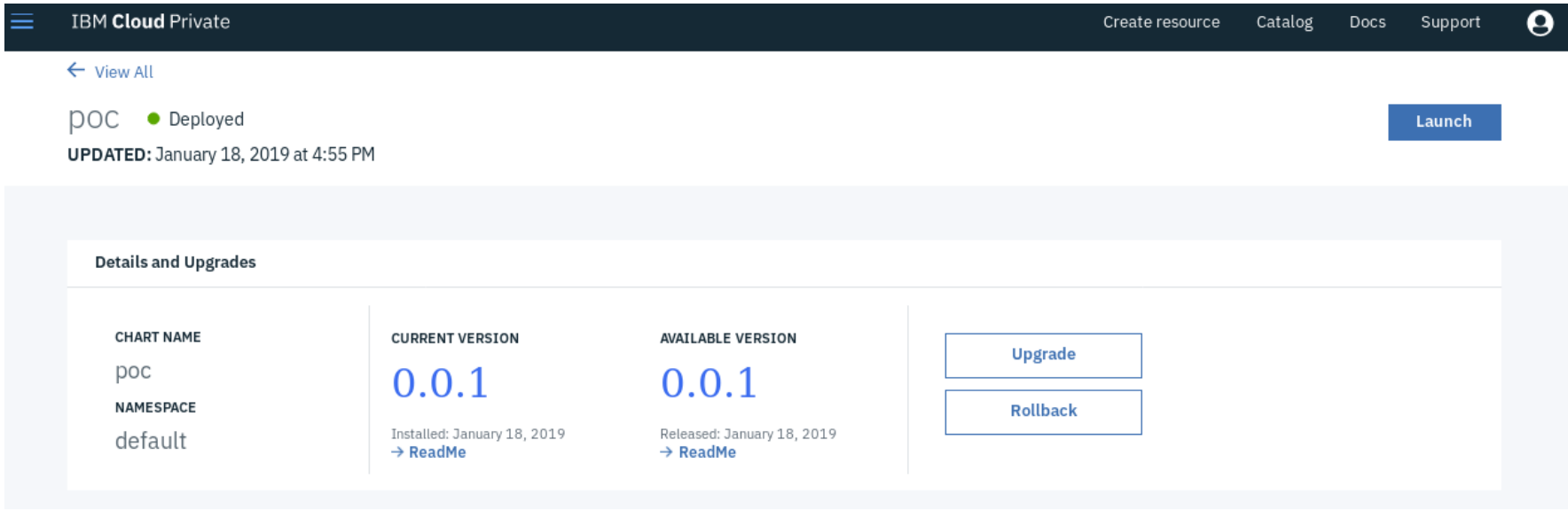


* you may choose another namespace for you project

See appendix for details view

Step 5 : Click “Configure” & then “View Helm Release”

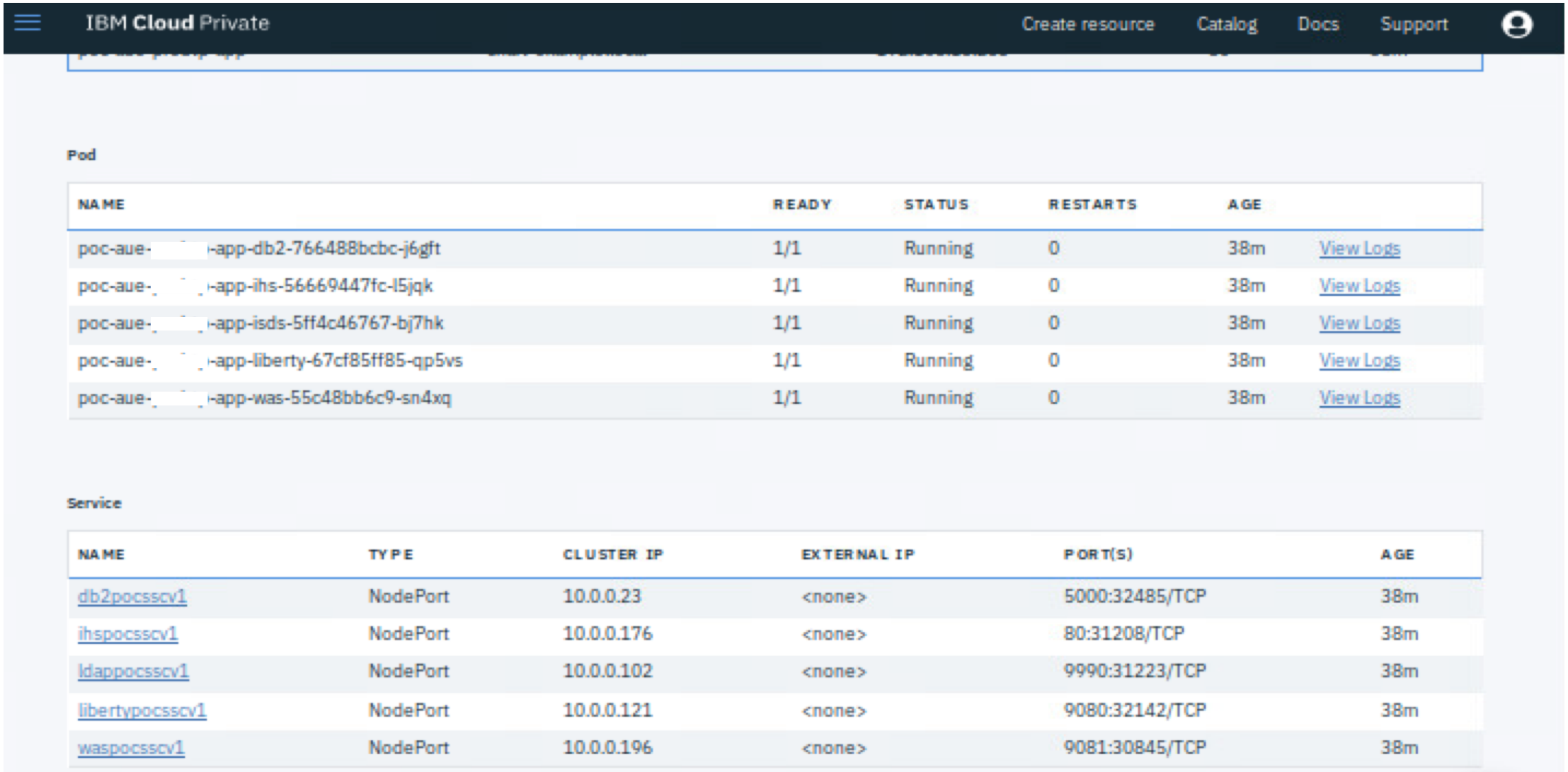




All the containers are created

Deployment

NAME	DESIRED	CURRENT	UP TO DATE	AVAILABLE	AGE
poc-aue-[redacted]-app-db2	1	1	1	0	2m
poc-aue-[redacted]-app-ihs	1	1	1	0	2m
poc-aue-[redacted]-app-isds	1	1	1	0	2m
poc-aue-[redacted]-app-liberty	1	1	1	0	2m
poc-aue-[redacted]-app-was	1	1	1	0	2m



Pod

NAME	READY	STATUS	RESTARTS	AGE	
poc-aue-[redacted]-app-db2-766488bcbc-j6gft	1/1	Running	0	38m	View Logs
poc-aue-[redacted]-app-ihs-56669447fc-l5jqk	1/1	Running	0	38m	View Logs
poc-aue-[redacted]-app-isds-5ff4c46767-bj7hk	1/1	Running	0	38m	View Logs
poc-aue-[redacted]-app-liberty-67cf85ff85-qp5vs	1/1	Running	0	38m	View Logs
poc-aue-[redacted]-app-was-55c48bb6c9-sn4xq	1/1	Running	0	38m	View Logs

Service

NAME	TYPE	CLUSTER IP	EXTERNAL IP	PORT(S)	AGE
db2pocsscv1	NodePort	10.0.0.23	<none>	5000:32485/TCP	38m
ihspocsscv1	NodePort	10.0.0.176	<none>	80:31208/TCP	38m
ldappocsscv1	NodePort	10.0.0.102	<none>	9990:31223/TCP	38m
libertypocsscv1	NodePort	10.0.0.121	<none>	9080:32142/TCP	38m
waspocsscv1	NodePort	10.0.0.196	<none>	9081:30845/TCP	38m

4 - Work with your containers /logs/debug…and go to 1 ^^, till success!

```
[sebll@oc0787150785 HelmChart]$ tree aue
aue
├── charts
├── Chart.yaml
├── README.md
├── templates
│   ├── db2.yaml
│   ├── _helpers.tpl
│   ├── ihs.yaml
│   ├── ingress.yaml
│   ├── isds.yaml
│   ├── liberty.yaml
│   ├── NOTES.txt
│   ├── service.yaml
│   └── was.yaml
└── values.yaml

2 directories, 12 files
[sebll@oc0787150785 HelmChart]$
```

Content for Chart.yaml

```
apiVersion: v1
appVersion: "1.0"
description: Application AU/AUE
name: aue
maintainers:
- email: guillaume_hoareau@fr.ibm.com
  name: Guillaume Hoareau
sources:
- https://github.com/guikarai/SSC4ICP
version: 0.0.1
icon: https://raw.githubusercontent.com/guikarai/SSC4ICP/master/aue.png
```

```
[sebll@oc0787150785 HelmChart]$ tree aue
aue
├── charts
├── Chart.yaml
├── README.md
├── templates
│   ├── db2.yaml
│   ├── _helpers.tpl
│   ├── ihs.yaml
│   ├── ingress.yaml
│   ├── isds.yaml
│   ├── liberty.yaml
│   ├── NOTES.txt
│   ├── service.yaml
│   └── was.yaml
└── values.yaml

2 directories, 12 files
[sebll@oc0787150785 HelmChart]$
```

Content for README.md

```
#  Internal Application : Authentication User

"AU/AUE" is the acronym for an Internal Application : Authentication User.

This application has been designed in order to secured  application portal thanks to a token based media.
This token is created the first time a user try to connect to an application. Security process validate user
attempt and deliver the token to be use as authentication pattern.
An existing token is always verified for each new connection attempt to an application.

This microservice AU/AUE application is made of 5 docker containers: IHS, DB2, WAS, LIBERTY and
ISDS.

This microservice to be hosted in a worker node based on Linux on IBM Z and SSC technology.

The worker node is managed by IBM Cloud private.
```

```
[sebll@oc0787150785 HelmChart]$ tree aue
aue
├── charts
├── Chart.yaml
├── README.md
├── templates
│   ├── db2.yaml
│   ├── _helpers.tpl
│   ├── ihs.yaml
│   ├── ingress.yaml
│   ├── isds.yaml
│   ├── liberty.yaml
│   ├── NOTES.txt
│   ├── service.yaml
│   └── was.yaml
└── values.yaml

2 directories, 12 files
[sebll@oc0787150785 HelmChart]$
```

Content for values.yaml

```
##################################################################################
# Licensed Materials - Property of IBM
# 5737-E67
# (C) Copyright IBM Corporation 2016, 2018 All Rights Reserved
# US Government Users Restricted Rights - Use, duplication or disclosure
# restricted by GSA ADP Schedule Contract with IBM Corp.
##################################################################################

# Default values for aue chart
# This is a YAML-formatted file.
# Declare variables to be passed into your templates.
# zATS. SebLL 17/01/2019.

##################################################################################
## Architecture Configuration Parameters
##################################################################################
# Specify architecture (amd64, ppc64le, s390x) and weight to be  used for scheduling as follows:
#  0 - Do not use
#  1 - Least preferred
#  2 - No preference
#  3 - Most preferred
arch:
  amd64: "0 - Do not use"
  ppc64le: "0 - Do not use"
  s390x: "3 - Most preferred"

##################################################################################
## Pre Validation Configuration Parameters
##################################################################################
#preValidation:
#  skipDiskValidation: true
#  image:
#    repository: "not needed"
#    tag: "none"
#    pullPolicy: "IfNotPresent"
```

```
[sebll@oc0787150785 HelmChart]$ tree aue
aue
├── charts
├── Chart.yaml
├── README.md
├── templates
│   ├── db2.yaml
│   ├── _helpers.tpl
│   ├── ihs.yaml
│   ├── ingress.yaml
│   ├── isds.yaml
│   ├── liberty.yaml
│   ├── NOTES.txt
│   ├── service.yaml
│   └── was.yaml
└── values.yaml

2 directories, 12 files
[sebll@oc0787150785 HelmChart]$
```

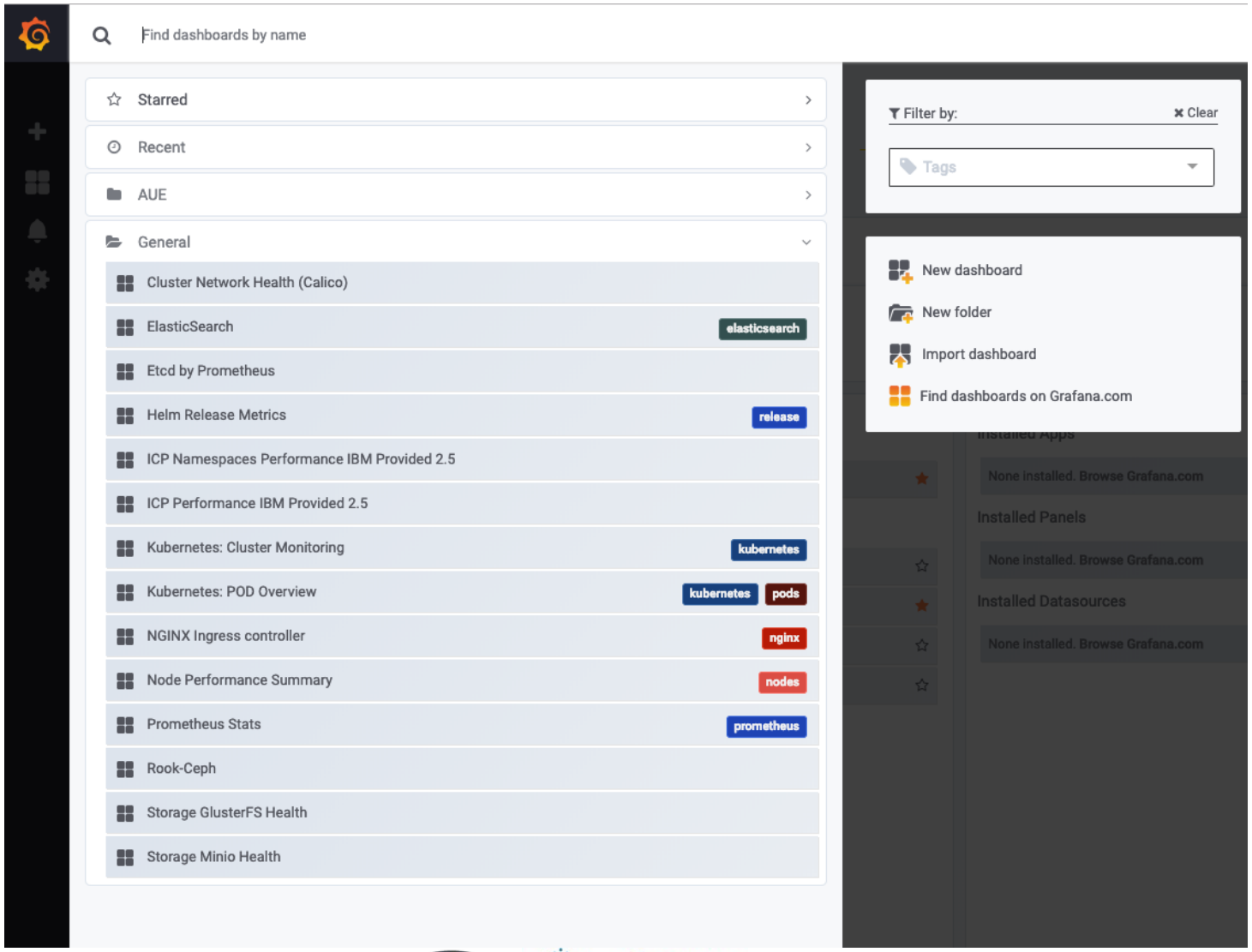
Content for values.yaml

```
################################################################################
## IBM Http Server - IHS Configuration Parameters
################################################################################
ihs:
  name: ihs-poc
  image:
    repository: "mycluster.icp:8500/default/pocssc.auext.ihs9008"
    tag: "v1"
    pullPolicy: "IfNotPresent"
  service:
    type: NodePort
    port: 80
#  resources:
#    requests:
#      cpu: "500m"
#      memory: "512Mi"
#    limits:
#      cpu: "1000m"
#      memory: "1Gi"

################################################################################
## DB2 Configuration Parameters
################################################################################
db2:
  name: db2-poc
  image:
    repository: "mycluster.icp:8500/default/pocssc.auextdb.db2express105"
    tag: "v1"
    pullPolicy: "IfNotPresent"
  service:
    name: ibm-db2-aue
    type: NodePort
    port: 5000
#  resources:
#    requests:
#      cpu: "500m"
#      memory: "512Mi"
#    limits:
#      cpu: "1000m"
```

5 - Monitor your cluster and your particular application !

Monitoring capabilities – General and default templates



3x Noticeable dashboards

- **Elasticsearch**

Text search engine powered by elasticsearch.
Usefull to exploit and to analyse logs activity.
Search can be turned in graph, graphas can be assembled as a dashboard with kibana.

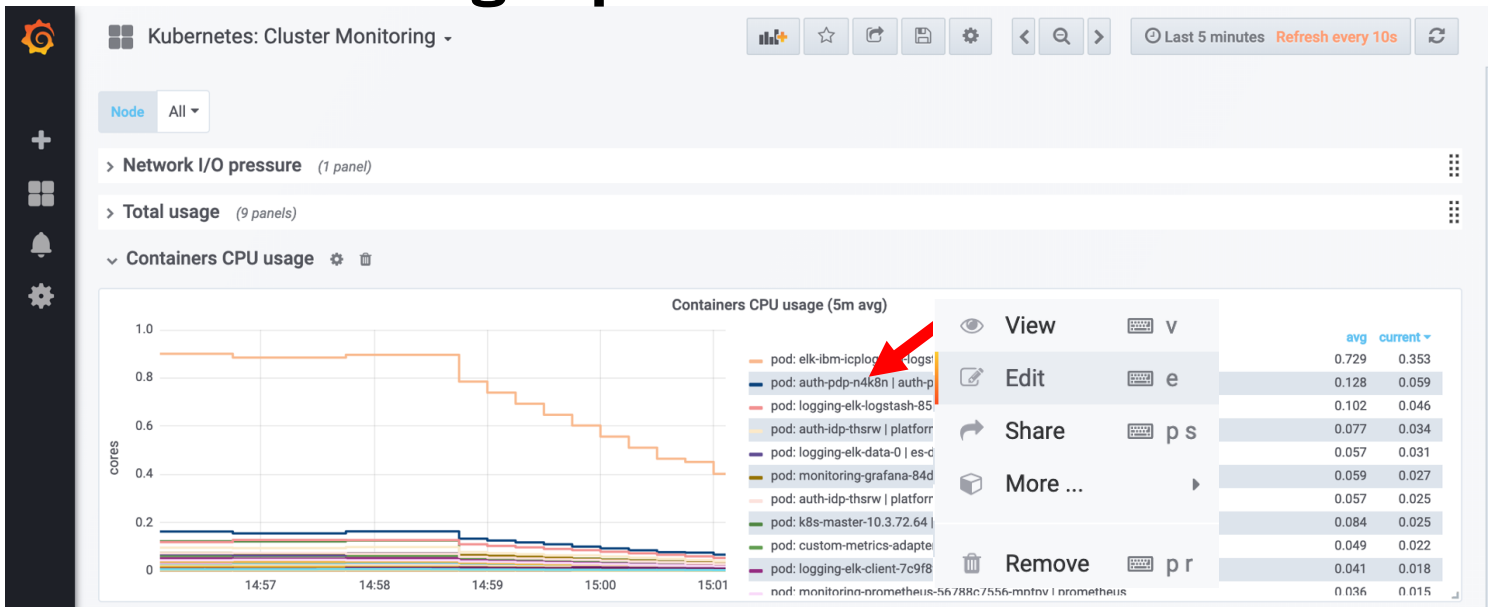
- **Kubernetes: Cluster Monitoring**

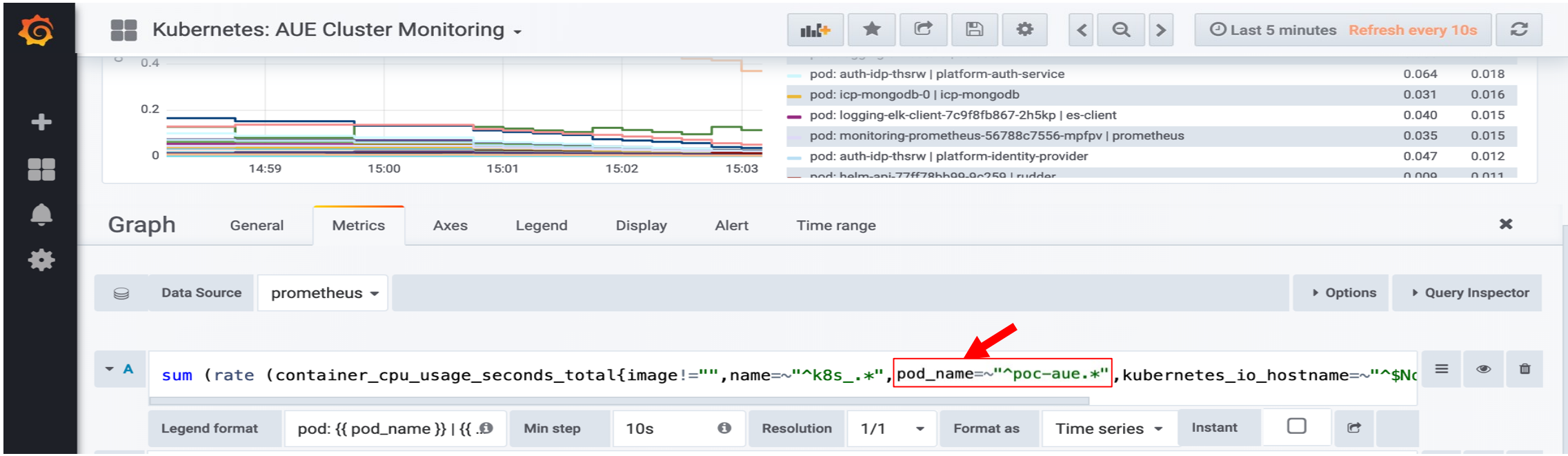
Overall monitoring at nodes level. Allow a top to bottom approach from nodes to pod and containers.

- **Kubernetes: POD Overview**

Eagle view from selection according to Namespace, POD, and container selection.

Monitoring capabilities – Kubernetes: Cluster Monitoring (Customizations)





Monitoring capabilities – Kubernetes: Cluster Monitoring (AUE CPU activity result)

IBM Z Architecture capabilities serving Microservices

Domain	Feature / QoS	Benefits
Security & availability	IBM Z RAS features	Microservices are designed for failure and need HA and failover
Security and encryption	CPACF, Crypto-Express Cards, Post Quantum Algorithms	Massive number of secured Microservices on one physical server and and high speed encryption
Isolation (EAL5 LPARs)	Higest level for non-military environments	Most securable server & Highest certified multi-tenant security
Secure Service Containers	Unique offering on the market. Available for Private Cloud	Focusing on consumability and security for your containerized applications
HA/DR (Processes & features, GDPS)	GDPS Appliance	Easy failover in case of errors or service crash
Performance & throughput	Performance (Proc, Java optim., zEDC, …)	Improved quality of service with hardware innovations
Scale-up and scale-out	Both directions	Perfect fit for dynamic workloads, due to vertical scalability
Hipersocket network	Fast in-memory communication	Latency highly reduced as many interactions occur for micro-services
I/O performances and throughput	OSA Cards Express6S+, ROCe,	Memory to memory communication
Flexibility & optimisation	Capacity on Demand	Can support unpredicted Microservices growth and spin
High level of resource sharing (CPU pooling,..)	Weights on LPAR, priority of ressources	Fine granular capacity allocation & sharing through virtualization
Virtualization technologies (PR/SM, z/VM, KVM)	Choice for the most appropriate technology for your workloads	Useful for individual Microservice scalability
Ease of use (IBM Wave, DPM)	Visual & Simplified System Management	Manage easily your underlying infrastructure

Summary
IBM Cloud Private Benefits with LinuxONE / Linux on Z

GUIDE SHARE EUROPE UK REGION

The most secure data serving platform in the world…

- Integrated cloud platform for enterprise workloads running in a customer controlled and secure environment
- Developers quickly start developing cloud-native services on x86 or LinuxONE with no change in tooling
- Ability to deploy private cloud for Intel, Power, Z and public cloud in under an hour from dozens of IBM and Open Source supported services
- Ability to deploy/manage Cloud Foundry on x86 – ICP master resides on x86/Power, worker nodes on z**
- Ability to deploy virtual machines and deploy/manage non-containerized workloads using IBM CAM in ICP

…To do more work with fewer servers at lower cost

- **One platform for entire business processes with highest Security rating & highest Cloud SLA availability of any commercially available server**
- **Support massive workloads with thousands of users in parallel and up to thousands of Linux servers – all in one box**
- **For IBM Z (not for LinuxONE) : Ability to deploy z/OS subsystems (DB2, WAS, CICS, IMS, etc.) ***
- **IBM Secure Service Container for IBM Cloud Private (announced 2nd of October 2018)**
  - **VM level isolation between containers,**
  - **Protection from privileged users to mitigate breaches/leaks from internal threats, ransom ware, malware,**
  - **Locked down platform (preventing modification of security policies, running of privileged containers etc.),**
  - **Pervasive Encryption for all data generated by workloads,**
  - **HW Key Protect Technology for tamper-proof encryption key storage**

Unique to ICP on IBM Z

* IBM Cloud Private using z/OS brokered Services.
** Master Node on IBM Z is **now (27th of Feb. 2019), available.**

Take away

GUIDE SHARE EUROPE UK REGION

- **Many ways to have IBM Z integrated in your cloud strategy** depending on your security, performance & governance requirements.
- The **efficiency** of your current architecture can take benefits of new solutions for **more agility**
- **Open Source solutions** are available with Linux on IBM Z
- Transform your IBM Z in an **innovative platform** reusing existing services
- Discover **DevOps solutions** with IBM Z for managing the continuous deployment on an **Hybrid Cloud**
- **To get started, contact us for:**
  - **IBM Z** Cloud Computing Workshop
  - **IBM Z** Open Source Discovery Session

IBM LinuxONE III

THANK YOU

Cloud, elevated.

Sébastien LLaurency
Linux & Cloud Architect - zATS
IBM Certified Expert Integration Architect
Member of Technical Expert Council France
IBM Client Center Montpellier, France

fr.linkedin.com/in/sebastienllaurency

@SLLaurency

Thank you!

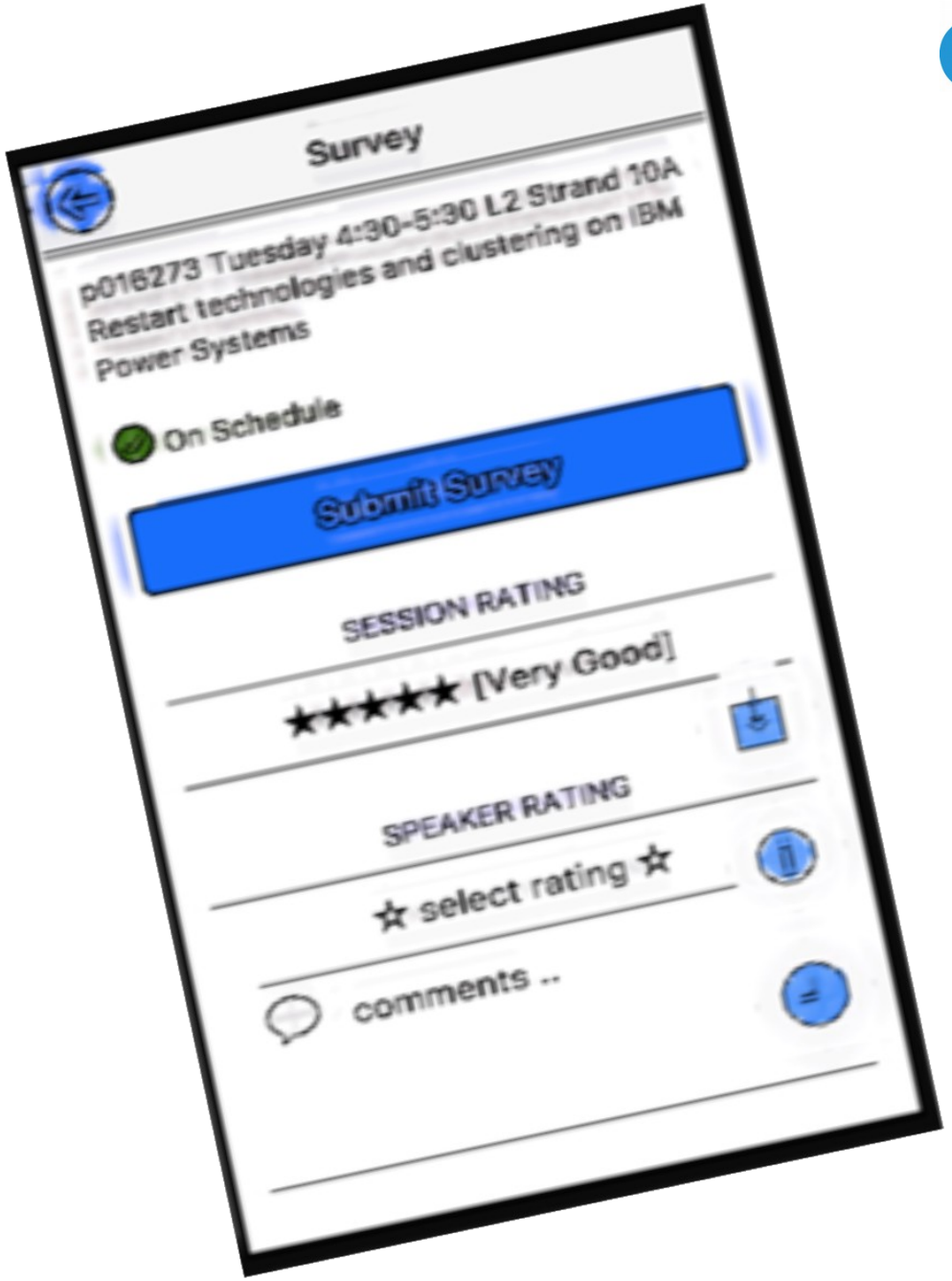
Sébastien LLaurency
Linux & Cloud Architect - zATS
IBM Certified Expert Integration Architect
Member of Technical Expert Council France
IBM Client Center Montpellier, France

firstlastname@us.ibm.com

+1-555-555-5555

ibm.com

Please complete the Session Evaluation!



Notices and disclaimers

-
- **U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.**
- Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. **This document is distributed "as is" without any warranty, either express or implied. In no event, shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity.**
  IBM products and services are warranted per the terms and conditions of the agreements under which they are provided.
- IBM products are manufactured from new parts or new and used parts.
  In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply."
- **Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.**
- Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those
- customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.
- References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.
- Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.
- It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer follows any law.

Notices and disclaimers continued

- Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products about this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. **IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a purpose.**
- The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.
- IBM, the IBM logo, ibm.com and [names of other referenced IBM products and services used in the presentation] are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml

Section header slide

Two content slide

Title only slide

Please submit your session feedback!

- Do it online at http://conferences.gse.org.uk/2019/feedback/nn

- This session is nn

Place your custom session
QR code here.
Please remove the border and text beforehand.