IBM Client Center Montpellier
February 2019

Private Cloud for AI Behind the Scenes
MOP AI Cloud - PowerAI Expert Event

Author
Benoit.Marolleau@fr.ibm.com Cloud & AI Solution Architect
Sebastien.chabrolles@fr.ibm.com Cloud & Linux Expert

MOP PowerAI CoC Team
Sdelabarre@fr.ibm.com AIX/Linux Expert
Regis.cely@fr.ibm.com PowerAI Expert

Agenda

- Drivers: Why a Kubernetes based AI Cloud on premise?
- Architecture Overview - Architecture (interesting) Details
- Demonstration
- Q & A

Business Drivers

Goal : Make the access of any AI & PowerAI demonstration & testing environments easier for any Sales / Tech Sales teams in EMEA, with minimal operational costs, maximal HW/SW usage, as fast as possible.

- Phase 1 **NOW
    - Provide a demonstration cluster with PowerAI Vision / Driverless AI on Power / PowerAI Base / Watson Studio,
    - can be used with remote access by any market to run PowerAI vision demonstrations for customers
        - Cluster located in IBM Cient Center Montpellier - support during the regular business hours
        - Demonstrations are provided by local Technical resources, trained on PowerAI / Vision / DAI demonstrations
    - Phase 1 is essentially a time-saver for MOP Tech Sales Support teams, allowing to absorb all the simple testing requests to focus on complex PoC & testing requests.

- Phase 2 **TO BE
    - Default model @ MOP = Cloud , and traditional dedicated model for complex projects (As is).
        - Moving a machine from the Cloud pool to dedicated pool takes a few minutes.
        - Additional AI / Storage components for more complex PoCs in cooperation with Storage and GBS experts
    - **WML Accelerator Integration (WMLA with ICP)**
    - **ICP for Data support**
    - Full automation and self-service.

    As a tech sales IBM / BP, connect, request, and access your AI env.

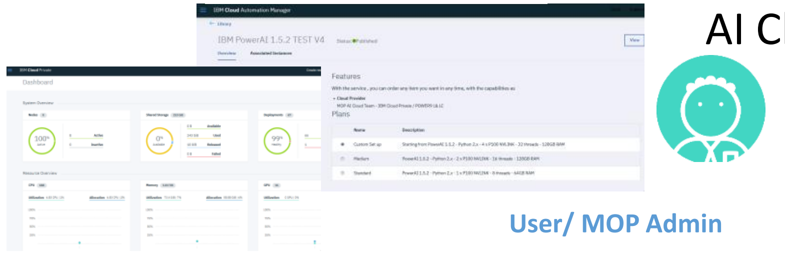
Other Drivers

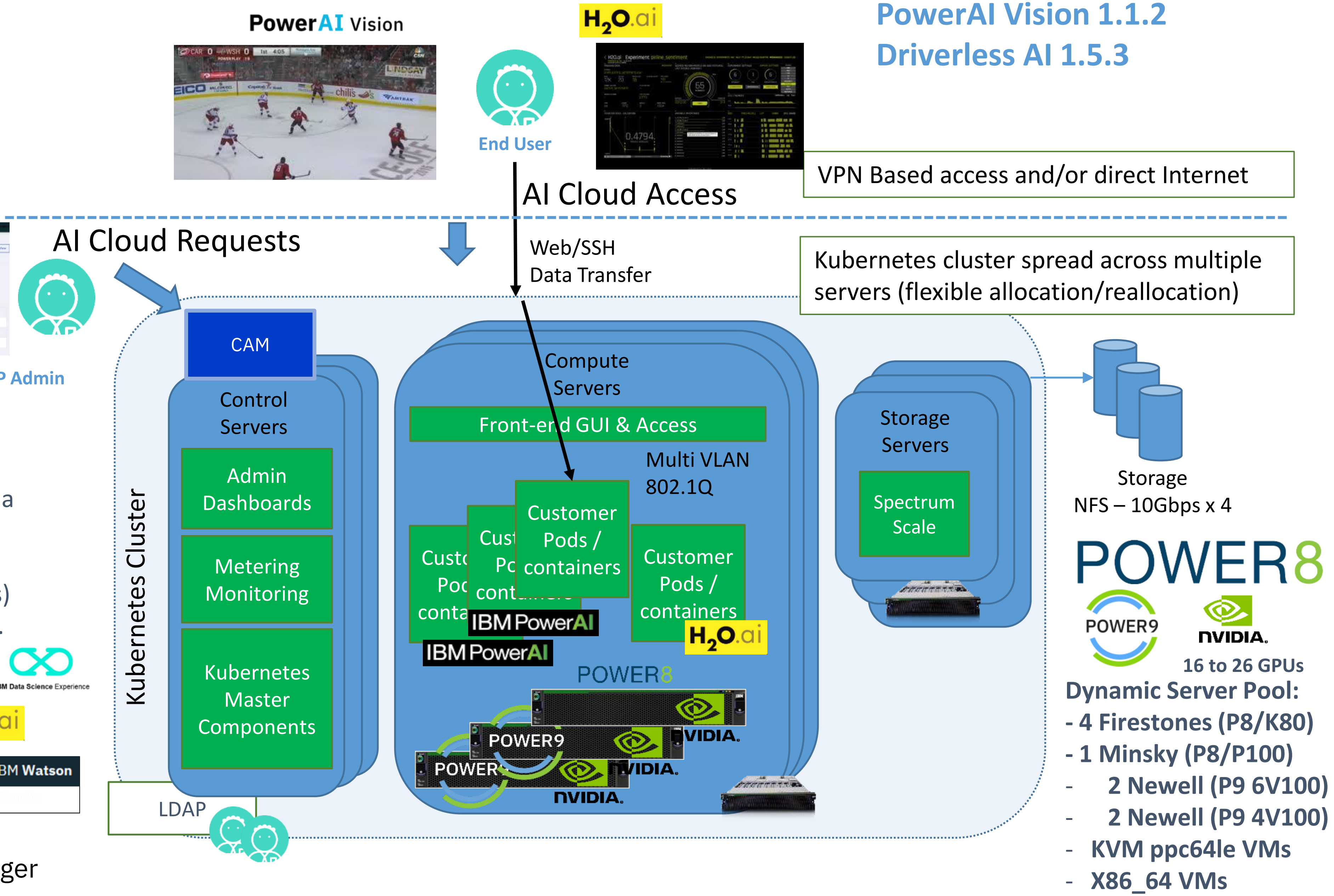
- Show our technology in action.
  - Open Source (K8s, Terraform, ansible, etc.)
  - Cloud Management
  - AI Technology – PowerAI , Watson , ICP For Data ...
- Real Showcase of Private AI Clouds – IBM Experience & giveback

PowerAI Experts: Lab Environments

Powered by IBM **Cloud** Private
& IBM **Cloud** Automation Manager

PowerAI Vision & Driverless AI
Lab Environments : 20 Teams
PowerAI Vision 1.1.2
Driverless AI 1.5.3





- ❑ Cloud Private & Kubernetes “GPU as a Service” Cluster
- ❑ Helm Chart Based Catalog:
  - PowerAI (several flavors/ versions)
  - AI Vision / Driverless AI / Watson…

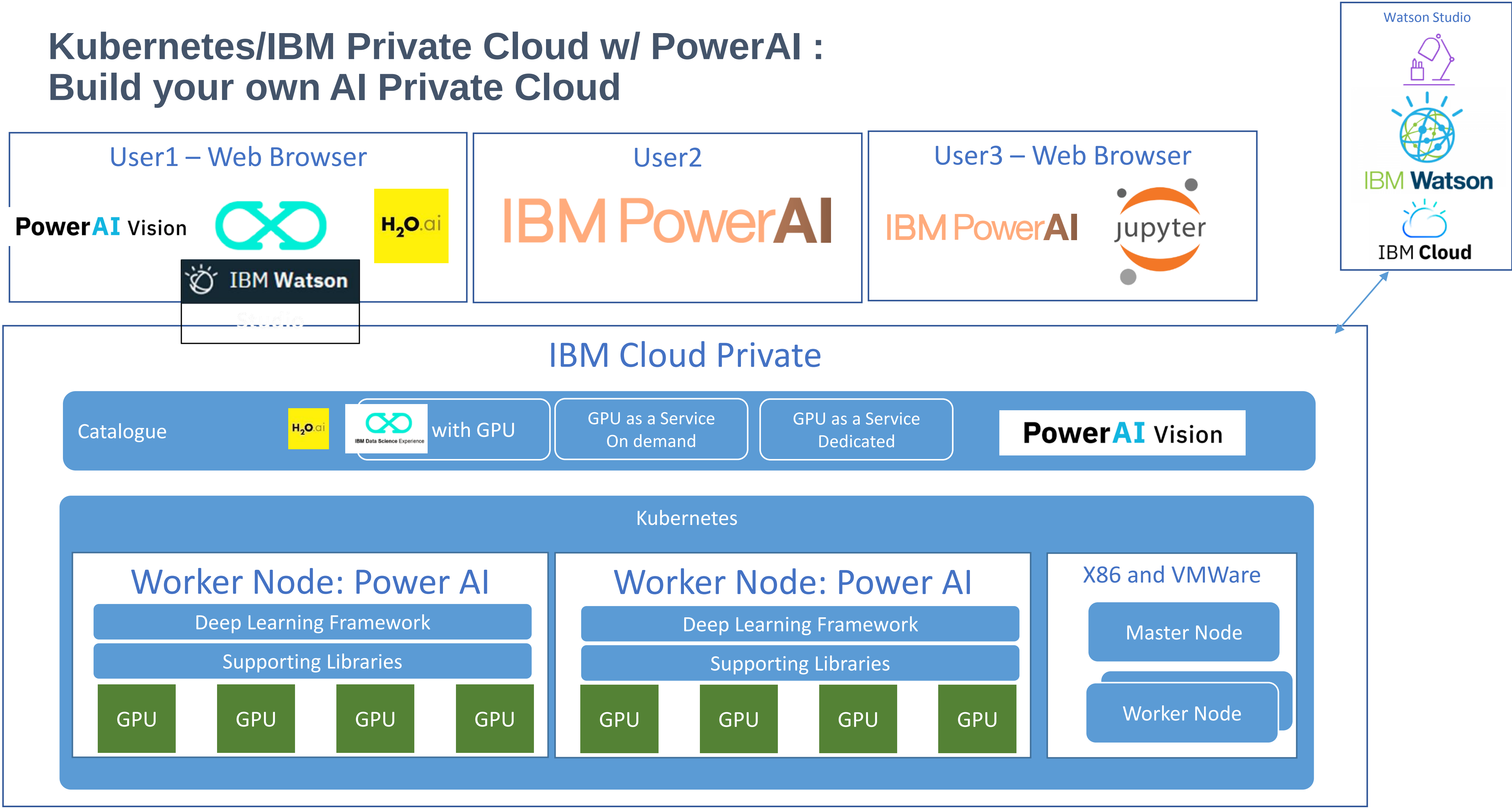
    PowerAI Vision · jupyter · IBM Data Science Experience · IBM PowerAI · H2O.ai

  - **Watson Studio Local** ppc64le
  - Others (based on requests)

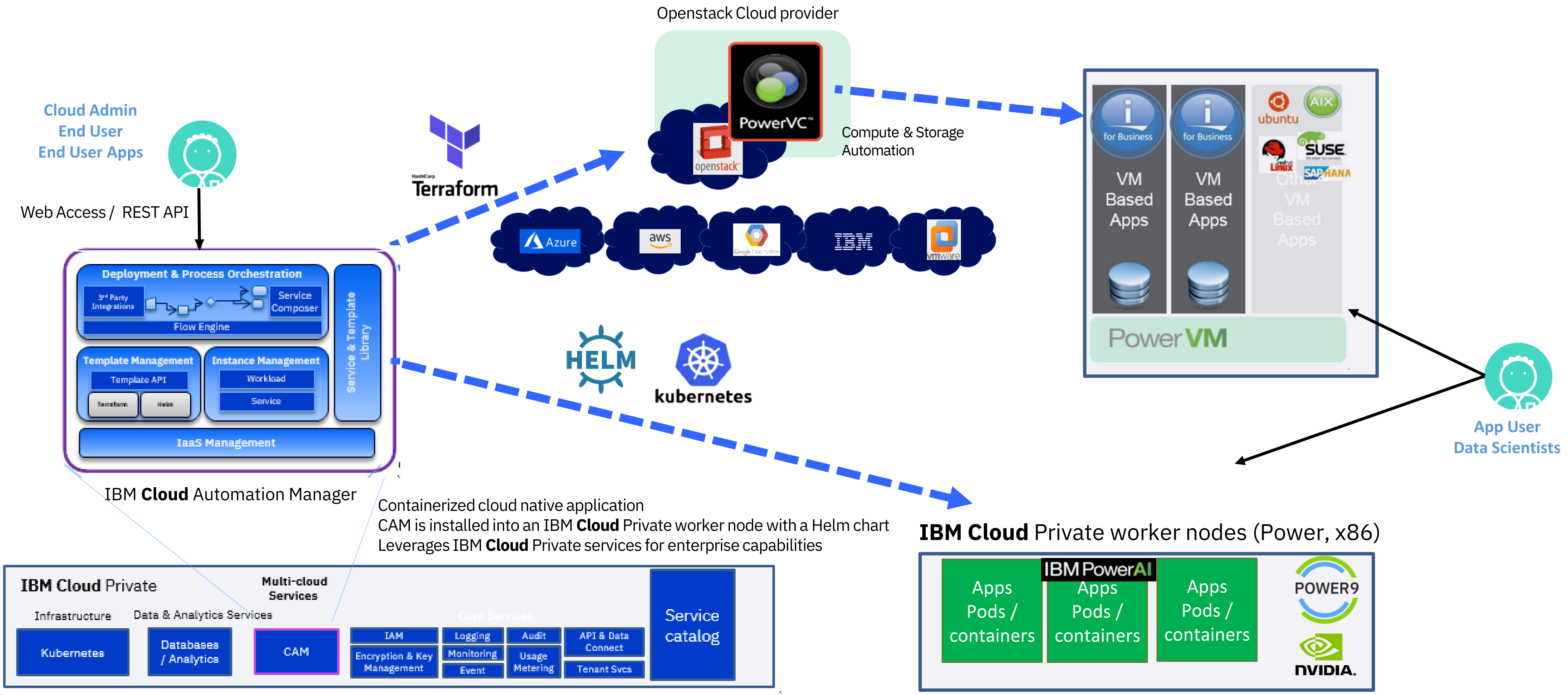
    IBM Watson

❑ Self-service portal based on IBM **Cloud** Automation Manager

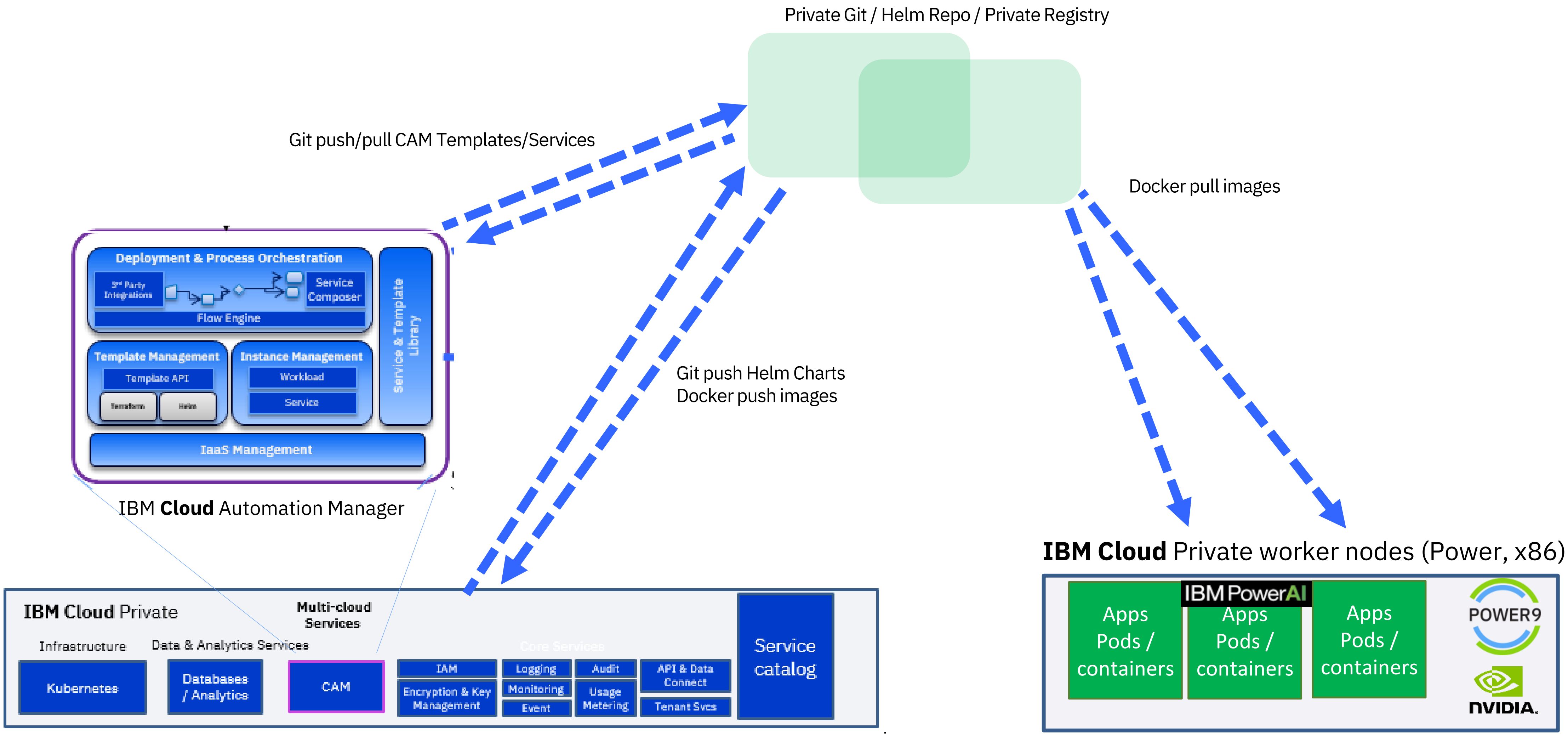
Kubernetes/IBM Private Cloud w/ PowerAI : Build your own AI Private Cloud



Architecture Overview



Architecture Overview 2/2 - DevOps & Asset versioning



Deep Dive: Why ICP & Kubernetes for AI

The main idea is to get bare metal performance with the flexibility of the virtualization i.e. allocate a subset of the GPUs in the machine for a single app. Ideally, add some HA / Clusterware and Network Isolation between Apps.

Kubernetes Clustering (ICP / OpenShift etc.) for AI coexists with existing K8s apps (Cloud Native apps, Modernization…) , simply by adding Power Based worker nodes in addition to (existing) x86 worker nodes.

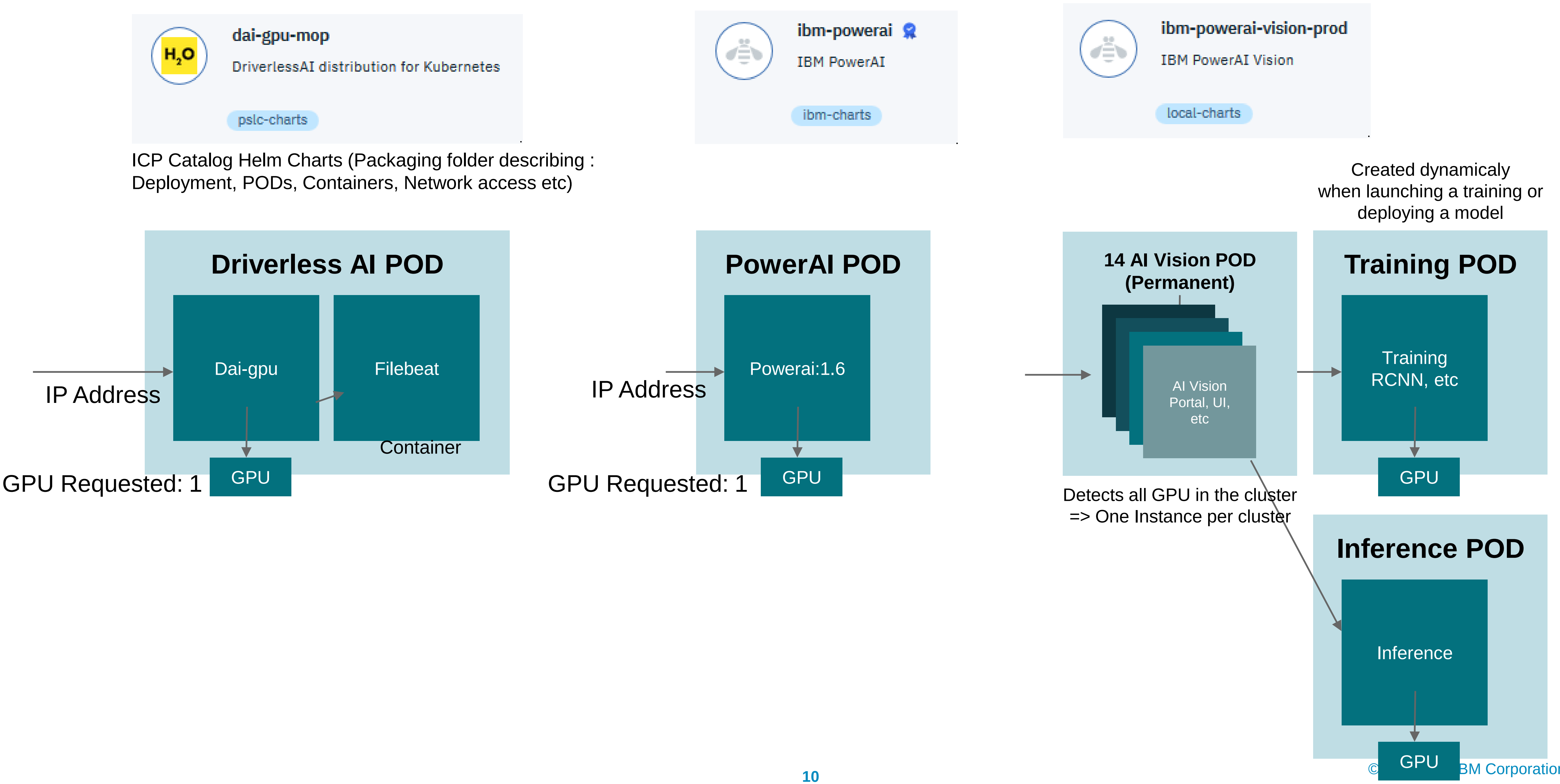
WML Accelerator uses more sophisticated Scheduling & Resource Management (EGO, Spark etc) components that are more powerful that the current K8s capabilities. Phase 2 will integrate WMLA.

- **GPU Support - Kubernetes + nvidia-docker plugin**
  - Resource Management – nvidia.com/gpu resource
  - Cuda libs & nvidia drivers auto loaded in the container
  - Ex: h2o.ai DAI Deployment allocates one GPU
  - ICP 3.1+ uses K8s 1.11+ which uses Nvidia-docker v2.
  - If heterogeous GPU types (K80, P100, V100 etc) , use of K8s Labels & Node Selectors.
    Ex: nvidia-tesla-v100-16gb

- **Cluster Management & App Resiliency**
  - Common logging, event management & monitoring, HA, etc.
  - Ex: Evacuate Apps, re-schedule it on another node. Planned maintenance, or unplanned outage

- **Network isolation**
  - Kubernetes works on flat networks (external and internal) . Cloud providers implement various
  Mechanisms (OSI Layer 3+) to expose apps. ICP uses calico as a Network plugin.
  => ICP Manage multi-VLAN proxy nodes using Kubernetes proxy, network policies, ingress controllers…

ICP & Data scientists Apps



Example with ICP & Watson Studio – Behind the scenes

Namespaces:
sysibmadm-data , sysibm-adm , dsxl-ml , ibm-private-cloud
PODs:
cloudant, redis, usermgmt, dsx-core, and ibm-nginx
Images:
27 images

Listing of key Components in DSX Local
(see under /wdp/k8s in the master node)

- **devtest-helpers** - Utility scripts to help with deployments
- **dsx-local-proxy** - the primary NGINX based server– serves up port 443 and reverse proxies to all other DSX Local service URLs
- **docker-registry** - Docker registry running as a Daemon Set in all hosts and service all needed docker images
- **cloudant-repo** - Cloudant repository database used to house metadata and projects etc.
- **redis-repo** - Redis in-memory Key value store – used for session storage in the web/UI micro services
- **swift-objectstore** - Openstack Swift container used to store csv data assets
- **usermgmt** - Supports management of users, authentication and working an external LDAP server
- **spark** - Spark cluster – master & worker daemon set
- **wdp-deploy-dashboard** – Backend and Front-end Admin components (IBM Data Platform Manager)
- **wdp-logs-elk** - Elastic Search, LogStash and Kibana – for Logging, Indexing
- **wdp-metrics-prometheus** - Monitoring metrics with Prometheus
- **dsx-local-k8s** - web-ui and api microservices (such as portal-main, projects api etc.)
- **docplexcloud-service** - Decision optimization / Deep Learning deployment

19

© 2017 IBM Corporation

Prefix/Suffix	image.repository	image.tag
cloudantRepo	privatecloud-cloudant-repo	v3.13.428
dsxConnectionBack	dsx-connection-back	1.0.4
dsxCore	dsx-core	v3.13.10
dsxScriptedML	privatecloud-dsx-scripted-ml	v0.01.2
filemgmt	filemgmt	1.0.2
hdpzeppelinDsxD8a2ls2x	hdpzeppelin-dsx-d8a2ls2x	v1.0.10
jupyterDsxD8a2ls2x	jupyter-dsx-d8a2ls2x	v1.0.11
jupyterDsxD8a3ls2x	jupyter-dsx-d8a3ls2x	v1.0.7
jupyterGpuPy35	jupyter-gpu-py35	v1.0.9
mlOnlineScoring	privatecloud-ml-online-scoring	v3.13.6
mlPipelinesApi	privatecloud-ml-pipelines-api	v3.13.4
mllib	ml-libs	v3.13.30
nginxRepo	privatecloud-nginx-repo	v3.13.6
pipeline	privatecloud-pipeline	v3.13.3
portalMachineLearning	privatecloud-portal-machine-learning	v3.13.20
portalMlaas	privatecloud-portal-mlaas	v3.13.17
redisRepo	privatecloud-redis-repo	v3.13.431
repository	privatecloud-repository	v3.13.2
rstudio	privatecloud-rstudio	v3.13.8
spark	**spark**	1.5.1
sparkClient	spark-client	v1.0.2
sparkaasApi	sparkaas-api	v1.3.14
spawnerApiK8s	privatecloud-spawner-api-k8s	v3.13.5
usermgmt	privatecloud-usermgmt	v3.13.5
utilsApi	privatecloud-utils-api	v3.13.5
wmlBatchScoring	wml-batch-scoring	v3.13.2
wmlIngestion	privatecloud-wml-ingestion	v3.13.2

Good news: ICP/K8s manages everything for you ☺

Deep Dive: Why CAM

- CAM is a multi cloud orchestrator - multi ICP, private x86 & Power based clouds, public clouds – running on top of ICP/K8s
  - Easy to deploy, manage & upgrade
- CAM Service Designer: Helm Chart deployment (ICP), Terraform scripts (IaaS – PowerVC/Openstack, Public Clouds etc)
- A service
  - Hide the complexity and contains a flow of tasks to perform (helm install, terraform, email notification etc)
  - can be published in the CAM GUI (see below) and a user can directly consume the service
  - is also a REST API that can be consumed by an external app (mobile, web, admin scripts…)

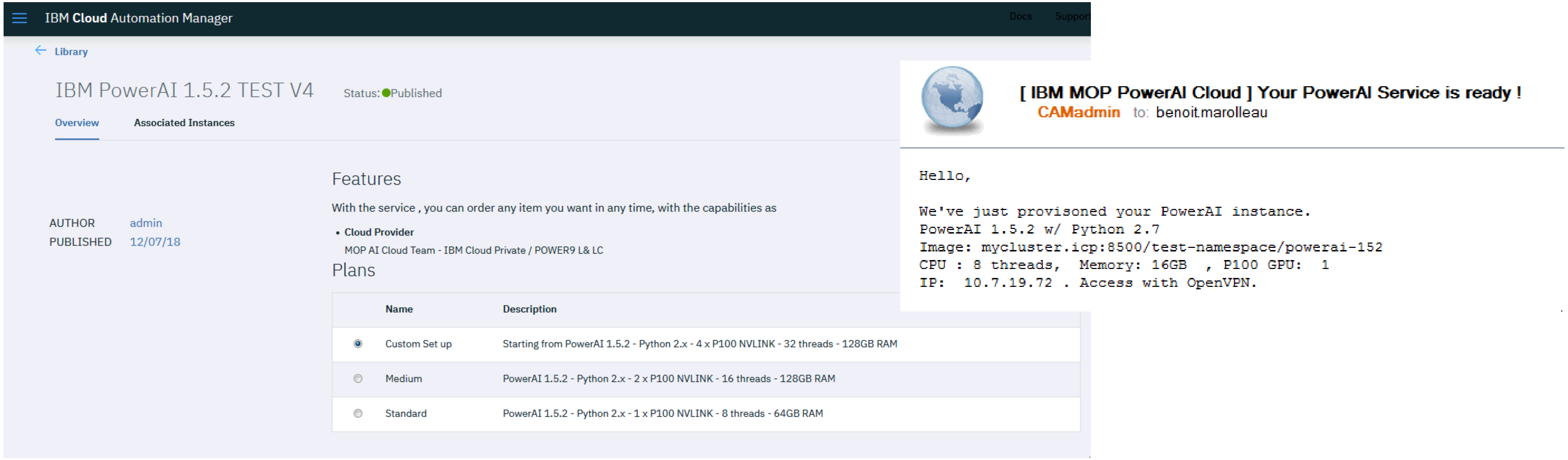


Fig.
Browse the catalog & choose the appropriate flavor & options for your AI project. Standardized Catalog - ex:
3 Flavors PowerAI Containers w/ GPU – Fixed PowerAI & frameworks versions

MOP AI Cloud – Need Access ? Next step

1/ Contact for PAIV / DAI / PAI / PAIE Access - Alain Roy , MOP Team

2/ Possible Enhancements – Phase 2

- PAIV - Work with the PAIV Labs – More Control on GPU allocation & management (infer / training) , user quotas , real admin user etc.

- Need more machines if the need is there.

If so, more things will be automated in CAM for self service.
Currently, no self service , semi-auto process with qualification.

- Aspera if needed for custom Dataset upload

- New Offering support
  - WML Accelerator support
  - ICP for Data support
- Full automation and self-service. As a tech sales IBM / BP, connect, request, and access your AI env.

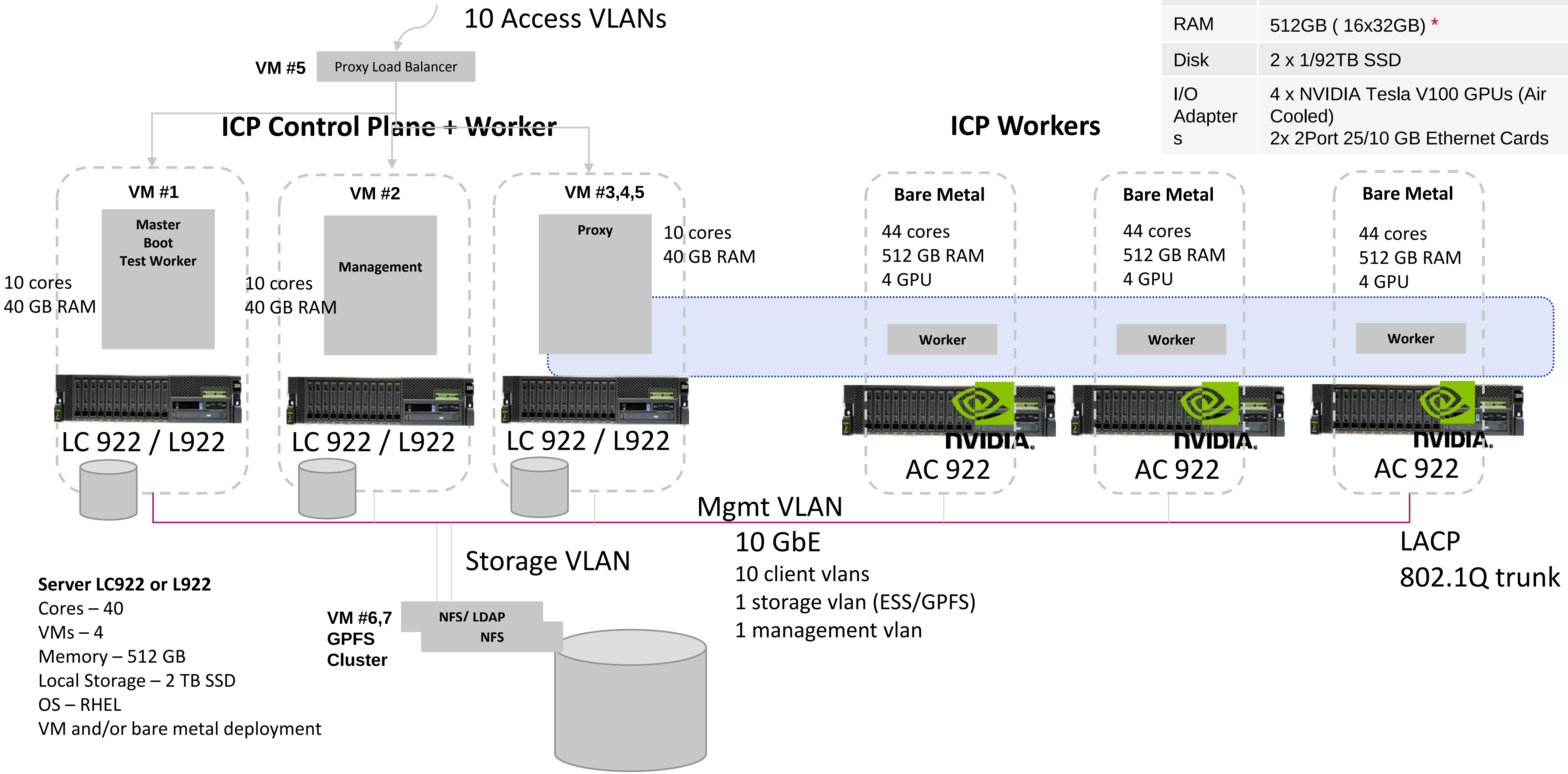
Demo

** INTRO DEMO DASHBOARD / MONITORING GRAPHANA / CAM
**DEMO 1 : Why CAM – todo Web App Or Ansible + CAM API
**DEMO 2 : PAIV Training Behind the scenes

Appendices - Notes
Sizing, HA for AI Apps

AI Cloud Architecture Overview – Physical Layer

System	AC922
Processor	2 x 22c@2.8Ghz
RAM	512GB (16x32GB) *
Disk	2 x 1/92TB SSD
I/O Adapters	4 x NVIDIA Tesla V100 GPUs (Air Cooled) 2x 2Port 25/10 GB Ethernet Cards



ICP Components - Basics

- **Master Node**: This type of node uses processes such as resource allocation and state maintenance to control worker nodes in a cluster. Master nodes primarily run Kubernetes core services such as apiserver, controller manager and scheduler. They also run light weight services such as auth service and catalog service.

- **Boot Node:** Ansible based installer and ops manager. Deploys IBM® Cloud private on master and worker nodes. The boot node is also used to scale the size of the cluster on demand, and for doing rolling updates.

- **Management Node**: This type of node is optional. It hosts management services such as monitoring, metering, and logging. When you implement management nodes, you prevent the master node from becoming overloaded.

- **Proxy Node**: This type of node is primarily used to run the ingress controller. Use of a proxy node enables you to access services inside IBM Cloud Private from outside of the cluster.

- **Worker Node**: This type of node works as a Kubernetes agent that provides an environment for running user applications in a container.

- Container networks managed by Calico
- One shared etcd for K8s and Calico (distributed key-value store that maintains configuration data.
- Helm (Tiller) runs on a single master node
- Docker Registry runs on each master node

General Guidance

- If running multiple ICP nodes on same physical server – use VMs (KVM guests) to isolate ICP Components from each other. For example to ensure ICP worker workload does not impact ICP master or management nodes
- Populate at least ½ of the available memory slots on Power servers – Invest in RAM for systems running multiple KVM guests, ICP Master, Management nodes
- Use 10 Gig networking to interconnect ICP nodes – data – separate 1Gig network for systems management
  - For “large” configurations – consider LAG of 10Gig or utilizing 40 Gig TOR Switches
- Utilize / Configure a POC system such that it can be “grown” to serve as one of the nodes in a production environment
- ICP requires at least 3 separate instances of a Master node (utilizes a voting algorithm) to ensure availability of the cluster – this implies at least three physical servers for any production ICP environment. An odd number of master nodes should always be used – so the scale up would be to 5 master nodes, again, ideally on separate physical systems
  - Key point – you will be starting (small) with three (3) physical nodes –
- At this point in time – a GPU enabled ICP worker node must run on bare metal – i.e., cannot be a KVM guest
- Fewer, larger (CPU & RAM) worker nodes is preferable to more – smaller (CPU & RAM) worker nodes
- Assuming multiple worker nodes – a distributed storage / file system will be required. The customer’s storage strategy is a key factor in any system design. We strongly recommend involving an architect from the Storage team when designing a specific customer’s configuration.

Sizing Methodology Utilized

1. Size the Business Applications & Associated middleware (needed to run the business applications) – using the existing application sizing guidance.
2. The output of step #1 becomes the number of cores / ram / disk needed on the worker nodes.
3. Apply a "formula" to derive the number of ICP Control Plane (Master/Boot, Management, Proxy, and Vulnerability Advisor) cores required to manage the number of worker node cores.
    1. This ratio of control place cores to worker cores will probably evolve over time as we are able to incorporate more field experience. Currently the rations are conservative – yielding more cores for control plan than what might be required for a particular customer environment.
4. Invest in RAM and SSDs for Master, Management, Proxy nodes.
5. Determine the topology - allocate the various ICP nodes / components across the number of physical servers needed to provide the necessary core counts. Allow 15%-20% headroom in core / ram estimates.
    1. Pay attention to not creating a single point of failure. Distribute ICP master nodes each on separate physical servers, etc.
    2. For production environments ICP requires three master nodes – ideally on three physically separate servers. Therefore this becomes the starting point for a small production capable ICP deployment.
6. A high performance distributed file system is required – see the storage section

Hardware Selection Rationale

Use Case Differentiators	Hardware Features
Focused is around customers working on ML/DL applications	
A few number of users – Data Scientists – but involving large data sets	
Applications will need to utilize GPUs	Select AC922 as system building block for ICP Worker Nodes
Key Applications – IBM's Data Science Experience, PowerAI, PowerAI Vision	Utilize L922 w/PowerVM or LC922 with KVM guests for "Foundation Block" to run ICP and DSX "control plane" components

Notes:

- Enterprise Systems – coming in 2H2018 – will be incorporated once they become GA
- See Also "AI Infrastructure Reference Architecture"

https://public.dhe.ibm.com/common/ssi/ecm/87/en/87016787usen/systems-hardware-ibm-spectrum-computing-white-paper-external-87016787usen-20180619.pdf

Cognitive Computing - Production Configuration

Control Plane Building Block
(can be used to run non-GPU Workers)

System	LC922 or L922
Processor	2 x 20c@2.7Ghz
RAM	512GB (16x32GB) *
Disk	2 x 128 GB SATA DOMs (boot/OS) 8 x 480 GB SSDs (storage pool)
I/O Adapters	(use built in 10G eth)

GPU Enabled Worker Building Block

System	AC922
Processor	2 x 22c@2.8Ghz
RAM	512GB (16x32GB) *
Disk	2 x 1/92TB SSD
I/O Adapters	4 x NVIDIA Tesla V100 GPUs (Air Cooled) 2x 2Port 25/10 GB Ethernet Cards

	per instanc	SMALL	MED	LARGE
# Control Plane Building Block Systems (LC922) (can also be used as worker nodes)		2	3	3
# Worker Building Block Systems (AC922)		1	2	4
Total # Cores Available ((n x LC922 cores) + (n x AC922 cores))		62	104	148
Total RAM Available (512 GB / System)	512	1536	2560	3584
Number of GPUs Available		4	8	16
ICP Worker Cores Required (from Application Sizing)		45	75	100
ICP Worker RAM Required (from Application Sizing)				
ICP Control Plane Components	% of Worker Cores			
o Master / Boot cores /RAM	8.66%	4	9	9
o Management	5.51%	2	4	6
o Proxy	8.66%	4	6	9
o Vunerability Advisor	5.51%	2	4	6
Total Control Plane		13	23	28
Total ICP		58	98	128
Total Cores Available - ICP Cores (if RED- Then system over allocated)		2	2	12

* Up to 4TB Supported!