Session: OpenShift and CloudPak for Data on PowerVS

Antony Steel

Thanks to:
IBM Singapore and US PoC team

#include <std_disclaimer.h>
These notes have been prepared by an Australian, so beware of unusual spelling and pronunciation.
All comments regarding futures are probably nothing more than the imagination of the speaker and are IBM Confidential till after GA.

Agenda

Belisama

- Introduction to components
  - OpenShift (Red Hat OpenShift Container Platform – OCP)
    - IBM Cloud Private (ICP)
    - Kubernetes
  - PowerVS
- Installation and operation
- Overview
- Installing Cloud Paks
  - CP4D and CP4MCM
- CP4D
  - Demonstration / AutoAI and detection of insurance fraud.

Announcement of support of OpenShift on PowerVS may be Dec ‘20

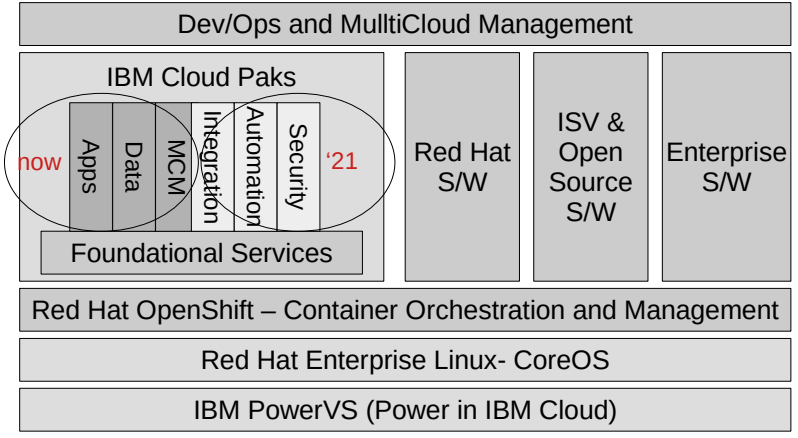
https://www.meetup.com/Singapore-AIX-IBM-i-Linux-on-Power-Meetup-Group/events/269942249/

Overview

Belisama

- IBM Power Systems Virtual Server on IBM Cloud (PowerVS)
  - Modernize Traditional Workloads AIX, IBM I and Linux
  - AIX and IBM i Virtual Servers available on-demand, hourly metered, billed monthly
  - Integrated into the IBM Cloud Catalogue for a consistent user experience and access to the cloud offering portfolio
  - Infrastructure equivalence to on-prem with FC-SAN, VIOS, PowerVM etc
  - Runs on Power Systems E880's, E980's and S922's
  - API enabled with Ansible automation
  - DC's
    - Dallas, Washington, Frankfurt, London, Toronto Sydney...

- New Capability: Full OpenShift Experience in PowerVS (Initial Support Targeted for Q4 2020)
  - Automation to install OpenShift in Power Virtual Server
  - Deploy IBM Cloud Paks, ISV and Open Source SW
  - Kickstart your projects to extend and modernize workloads
  - Bring your own (Red Hat OCP) License model
  - Preview tutorials and learning path: ibm.biz/Bdq3aV

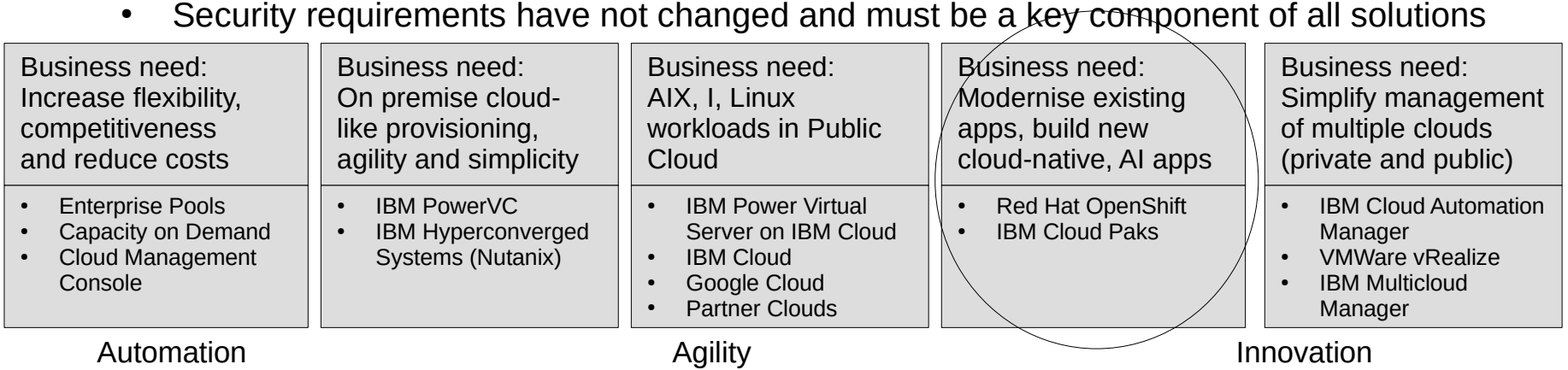


Heterogeneous Cluster - '21

Setting expectations

Belisama

- Journey to the cloud
  - Cloud is not a place
    - Management of resources and applications and managing more with less
    - Blurring lines between / merging different infrastructure teams
    - Application owners developers greater flexibility
    - Move from 1 all encompassing application to managing some tasks performed by smaller applications
      - Organisations often form view that core applications can be left on ageing infrastructure and cannot be migrated / moved to new method of management
    - Combining / managing / moving applications between company DC / commercial DS / flexible choice of cloud providers
    - Security requirements have not changed and must be a key component of all solutions



Background for OpenShift on PowerVS

Belisama

- Implementation of OpenShift on PowerVS
  - PowerVM is the virtualization product created by IBM to virtualize the resources to create micro-partitions on a ppc64le (POWER 7,8,9) server. Once deployed, it is configured and controlled by an HMC external server or a Novalink instance running on the Power server. No interaction between RHEL/Openshift is supported with these control planes. A static environment created by the PowerVM is supported and once this environment is created, each LPAR will be treated as if it were a bare metal resource.
    - For this space look at CloudPak for Multicloud Management / CloudForms
  - You can have multiple LPARs from the same PowerVM server configured to create a high availability environment. You could also use multiple PowerVM servers so the nodes are on different physical servers. Just think of the LPAR as a bare metal resource and configurations that support that model are valid.

- IBM have been partners for over 20 years and major contributors to many Open Source projects with recent merger focusing Red Hat optimising for Power
- IBM and Red Hat delivering first complete open hybrid cloud platform
  - Develop, run and manage Applications and Workloads in consistent manner across **all clouds (private and public)**
- Built on:
  - Approach that is Hybrid, Open and Secure
  - Red Hat OpenShift built on Red Hat's open container and Kubernetes technology – portability choice across any cloud with consistent set of common services (access, identity, monitoring etc) with OpenShift on Power, x86 and Z.
  - Principle to allow development and modernisation of applications with consistent automation / tools and access to data anywhere.
  - Goal to manage, configure and monitor performance across all clouds and clusters from one point

Containers key features and benefits

Belisama

- Portability:
  - Single executable package with all code, configuration files, dependencies, and required libraries
  - Bundle must not include operating system-related files
  - Open runtime engine is a prerequisite
  - Common bins and libraries can be shared across multiple containers
- Agility:
  - Container system is managed by Open Container Initiative
  - DevOps tools and process are used for rapid code deployment by using continuous integration and continuous deployment (CI/CD)
  - Open Source Docker engine works for Linux and Windows platforms
- Performance:
  - Multiple containers share operating system kernel for lightweight execution mode
  - Improves service usage, which results in reduced software license costs
  - Container start time is much faster than VM start time
- Fault isolation:
  - During concurrent execution, each container runs independently. A fault in one container does not affect other container's execution.
  - Container engine takes advantage of operating system security isolation technique.
- Ease of management:
  - Container orchestration manages installation, scalability, availability as defined
  - Application version upgrade, monitoring, and debugging managed centrally through container orchestration system
- Security:
  - Encapsulation and isolation is the first level of security for any containerized application. A rogue application does not affect other applications of the hosting environment
  - Container engine inherits default security features from hosting platform
  - Namespace provides an isolated view; for example, file system, mount point, network,process ID, and User ID

History

- Docker emerged in 2013 after over 13 years of container development and popularity quickly increased
- Docker is the virtualised platform to host containerised applications, starting from software development until execution. Docker engine is a client/server application that features the following components:
  - Daemon: These processes run in the background and are designed to receive instructions from other applications to perform specific tasks. Daemons plays a key role to manage Docker objects, such as images, containers, networks, and volumes.
  - REST API: These user interfaces are used in code to send instructions to Daemon jobs.
  - CLI: These commands are run in terminal sessions to send instructions to Daemon. Script-based batch jobs are created by using CLI.
- Kubernetes to “orchestrate” containers
  - Container orchestration is the process of organising to achieve the wanted performance. These loosely coupled containerised objects must be organised and coordinated to meet functional requirement, such as starting and stopping an application, grouping and coordinating applications in a cluster. It is primarily focused on managing the life-cycle of containers for automated deployment, management of nodes, scalability and availability of services based on workload, and networking among distributed containers in large systems.

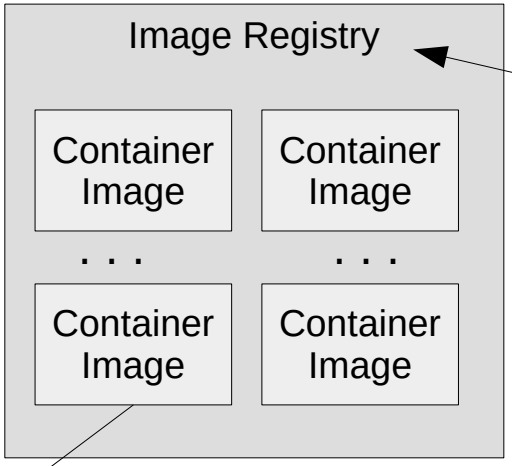
Kubernetes architecture

Belisama

- Master node
  - This node runs multiple controllers that are responsible for the health of the cluster, replication, scheduling, endpoints (linking Services and Pods), Kubernetes API. It interacts with the underlying cloud providers and others. Generally, it ensures that everything is running and monitors worker nodes.
- Worker node
  - This node runs the Kubernetes agent that is responsible for running Pod containers by way of Docker, requests secrets or configurations, mounts required Pod volumes, performs health checks, and reports the status of Pods and the node to the rest of the system.
- Pod
  - Within a cluster, a pod encapsulates an application that is composed of one or more processes from one or multiple containers. Every pod includes dedicated I/O resources, such as storage, a unique IP, and a set of configuration properties for the runtime environment. These features make pod the smallest unit of deployment and basic unit of execution. Pods have the following types:
    - Pod with a single container: This configuration is the most common.
    - Pod with multiple containers: Must be co-located containers to serve a functional requirement.
    - Networking: Each pod shares its namespace, IP, and port. However, for optimal performance, containers in same Pod communicates with the localhost identity.
    - Storage: A pod specifies shared storage volume. All containers in a pod can share persistent data through this volume.

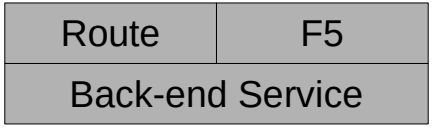
Containers

Belisama

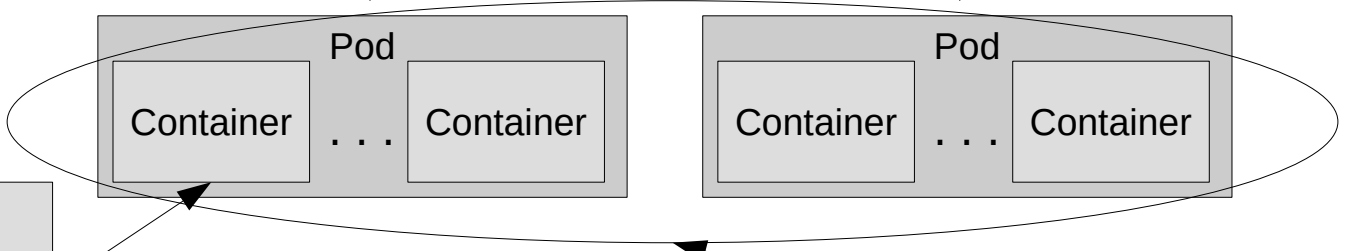


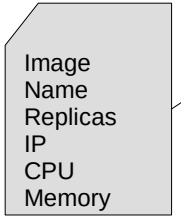
OpenShift registry, or upstream like Docker Hub and Red Hat registry

A container – the smallest compute unit.
The object for encapsulating or packaging up software code, along with all required dependencies, so that it can run uniformly and consistently on any infrastructure.



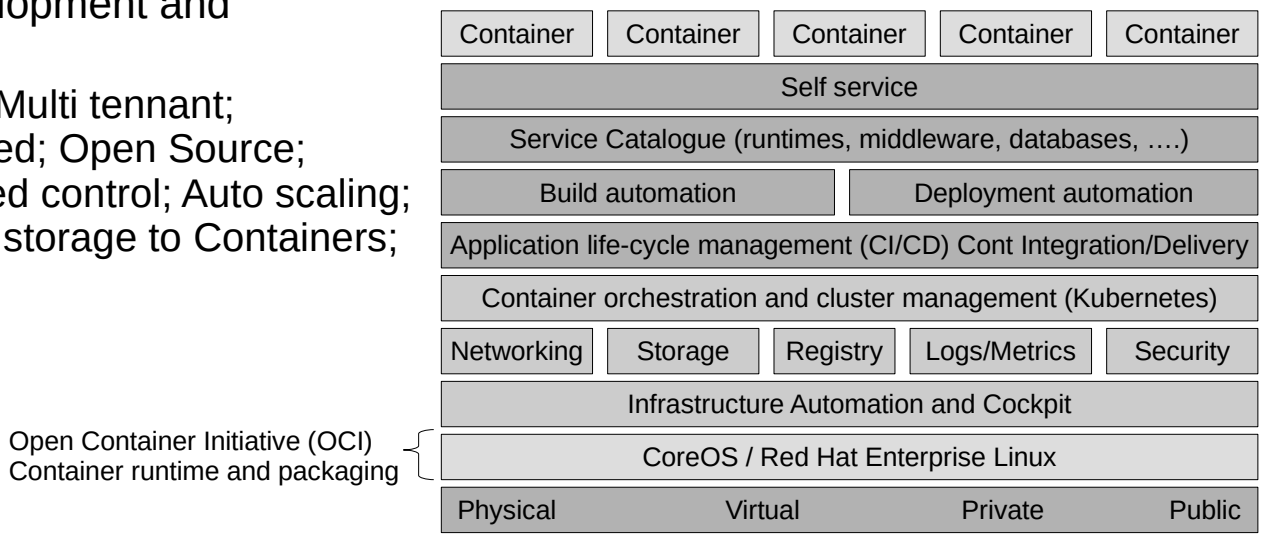
Pod – a unit for management and deployment





Project – to isolate applications by environments, teams, groups etc.

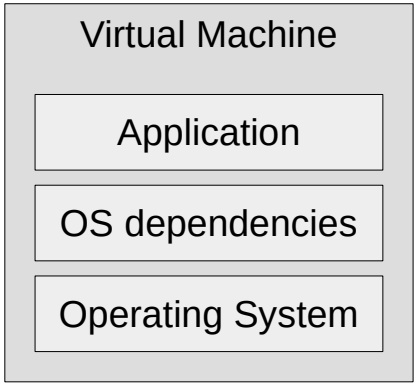
- Kubernetes (K8s)
  - An open source application container orchestration software that makes it easier to configure, deploy and manage containerized applications
  - For many years Red Hat and IBM have been contributing to the Kubernetes project and been the basis of both companies Hybrid Cloud Strategy
- OpenShift
  - Red Hat's open source container application platform based on the Kubernetes container orchestrator for Enterprise level application development and deployment
    - Self Service; Multi language; Multi tennant; Collaboration; Standards based; Open Source; Enterprise Grade; Policy based control; Auto scaling; Can directly attach persistent storage to Containers; Secure…

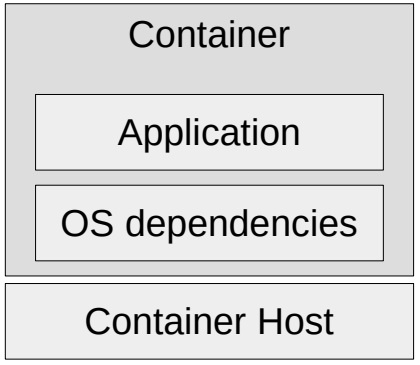


- Options for Worker Nodes
  - Red Hat Enterprise Linux
    - 10+ year enterprise life cycle
    - Industry standard security
    - High performance on any infrastructure
    - Customizable and compatible with wide ecosystem of partner solutions
    - For when customisation and integration with additional solutions is required
  - Red Hat Enterprise Linux CoreOS
    - Self-managing, over-the-air updates
    - Immutable and tightly integrated with OpenShift
    - Host isolation is enforced via Containers
    - Optimized performance on popular infrastructure
    - For when cloud-native, hands-free operations are a top priority
      - CoreOS is tested and shipped in conjunction with the platform. Red Hat runs thousands of tests against these configurations.
      - RHEL CoreOS admins are responsible for …. Nothing.

Virtual Machines / Containers

Belisama



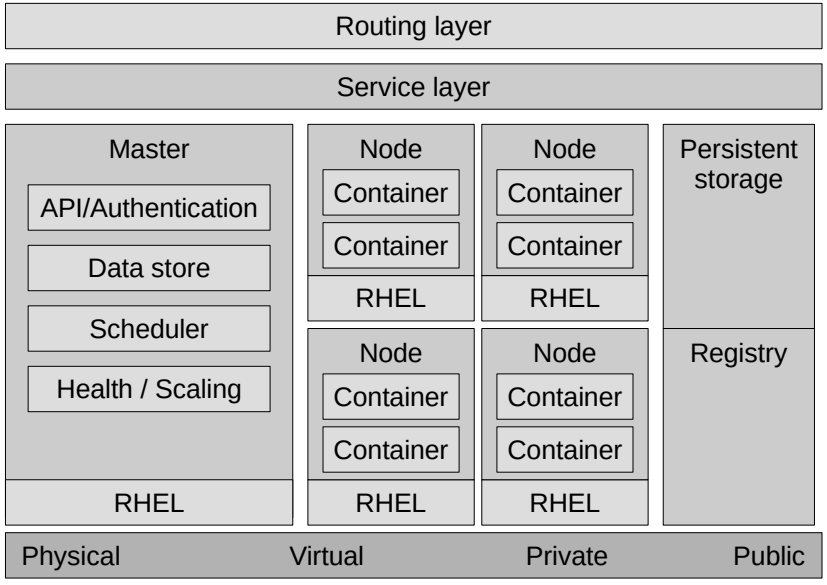


- √ VM Isolation
- ✘ Complete OS
- ✘ Less flexible compute
- ✘ Less flexible memory
- ✘ High resource usage

- √ Container Isolation
- √ Shared kernel
- √ Burstable compute
- √ Burstable memory
- √ Low resource usage

OpenShift Component view

Belisama



CRI-O - Container Runtime Interface – Open Container Initiative

Belisama

- Kubernetes introduced Container Runtime Interface (CRI), a plugin interface that gives Kubernetes the ability to use different Open Container Initiative (OCI) compliant container runtimes.
- CRI-O provides a lightweight runtime for Kubernetes. Prior to the introduction of CRI, Kubernetes was tied to specific container runtimes while with CRI, Kubernetes can be container runtime agnostic
- CRI uses the OCI-compatible environment for running Pods. CRI-O's compatibility with OCI enables it to be pulled in from any Container registry. This feature is a lightweight alternative for Docker and similar services. CRI-O is available with Red Hat OpenShift.

OpenShift Container Catalogue packages for Cloud Native apps

Belisama

- Databases
  - MongoDB, ProstgreSQL, MariaDB, MySQL, Redis, Cassandra
- Application Servers
  - Apache, NGINX
- Development tools
  - Node.js, OpenJDK, Jenkins
- Programming languages
  - Python, Perl, Go, Ruby, PHP, Golang
- Automation tools
  - Ansible
- ** Extend with your own Images! ….. and...

IBM Cloud Paks – IBM Middleware in a container

Belisama

- Enterprise-ready, containerized software solutions that give you an open, faster, more secure way to move core business applications to any cloud
- IBM containerized software
  - Packaged with Open Source components, pre-integrated with the common operational services and secure by design
- Operational services
  - Logging, monitoring, metering, security, identity access management, image registry
- Container platform
  - Kubernetes-based and portable
- Complete yet simple
  - Application, data and AI services, fully modular and easy to consume
- IBM certified
  - Full software stack support and ongoing security, compliance and version compatibility
- Run anywhere
  - On-premises, on private and public clouds and in pre-integrated systems

IBM Cloud Paks

Belisama

- Six IBM Cloud Paks cover key workloads on your journey to the cloud.
- Applications
  - IBM Cloud Pak™ for Applications helps to accelerate the build of cloud-native apps by leveraging built-in developer tools and processes, including support for microservices functions and serverless computing. Customers can quickly build apps on any cloud, while existing IBM middleware clients gain the most straightforward path to modernization.
- Build, deploy and run applications →
- Data
  - IBM Cloud Pak™ for Data helps to unify and simplify the collection, organization and analysis of data. Enterprises can turn data into insights through an integrated cloud-native architecture. IBM Cloud Pak for Data is extensible and easily customized to unique client data and AI landscapes through an integrated catalog of IBM, open source and third-party microservices add-ons.
- Collect, organize and analyze data →
- Integration
  - IBM Cloud Pak™ for Integration helps support the speed, flexibility, security and scale required for all of your integration and digital transformation initiatives, and comes pre-integrated with a set of capabilities including API lifecycle, application and data integration, messaging and events, high-speed transfer and integration security.
- Integrate app, data, cloud services and APIs →
- Automation
  - IBM Cloud Pak™ for Automation helps you deploy on your choice of clouds anywhere Kubernetes is supported, with low-code tools for business users and real-time performance visibility for business managers. Customers can migrate their automation runtimes without application changes or data migration, and automate at scale without vendor lock-in.
- Transform business process, decisions and content →
- Multicloud management
  - IBM Cloud Pak™ for Multicloud Management helps to provide consistent visibility, automation and governance across a range of hybrid, multicloud management capabilities such as event management, infrastructure management, application management, multicluster management, edge management and integration with existing tools and processes.
- Improve cloud visibility, governance and automation →
- Security
  - IBM Cloud Pak® for Security helps to uncover hidden threats, make informed decisions about the risks they pose, and then respond faster to those threats — while leaving data where it is. Customers can integrate tools and connect workflows across hybrid, multicloud environments, using a security platform that runs anywhere.
  - Connect security data, tools and workflows

Cloud Paks on Red Hat OpenShift on Power Systems

Belisama

Cloud Pak for Applications	Cloud Pak for Data	Cloud Pak for Integration	Cloud Pak for Automation	Cloud Pak for Multicloud Management	Cloud Pak for Security
Open liberty Swift WAS Liberty Spring Kabanero Enterprise JBoss istio jenkins node.j Transformantion UrbanCode Deploy Developer Team Orchestrator Developer Team Governance Red Hat OpenShift Application Runtimes (RHOAR)	Cloudant DB2 Cognos Streams MariaDB Redis MongoDB etcd PostgressSQL NetApp Persistent Storage Wand Taxonomies Knowis for Banking Lightbend Reactive Microservices Prolifics Prospecting Accelerator + Watson AI	App Connect Enterprise Event Streams MQ RabbitMQ Aspera Inegration Explorer Watson Voice Gateway Datapower	Business Automation workflow Operational Decision Magement Business Automation Insight Business Automation Content Analyser File Connect Manager Robotic Process Automation	MultiCloud Manager Cloud Application Manager Cloud Event Manager Cloud Automation Manager Ansible CloudForms	Federated Search Case Management Universal Data Insights Security Orchestration and Automation Developer framework
Build, deploy, and run applications	Collect, organise, and analyse data	Integrate applications, data, cloud services, and APIs	Transform business processes, decisions and content	Multicloud visibility, governance and automation	Connect security data, tools and teams

Open hybrid Multi-cloud platform

Red Hat OpenShift Red Hat Enterprise Linux

IBM Public Cloud	AWS	Microsoft Azure	Google Cloud	Edge	Private	IBM Z IBM LinuxOne IBM Power

PowerVC / PowerVM (AIX, i, Linux)	Red Hat Enterprise Virtualisation	Bare Metal

OpenShift on Power (roadmap)

Belisama

- Roadmap progressing from

ICP Workloads	IBM Cloud Paks
ICP Core Services	Cloud Pak Core Services
Kubernetes	OpenShift

- OpenShift on OpenStack
- Container native virtualisation – convergence of VM and container management
- CoreOS integration
  - Container linux / Fedora CoreOS
  - Docker → CRI-O
  - Registry → Quay
- Heterogeneous clusters
- Operators
- Serverless (Knative)

OpenShift on Power (roadmap) (Cont)

Belisama

- 2020 H1
  - IBM Multicloud Manager
  - IBM Cloud Paks – Multicloud Management (GA) with OpenShift Container Catalogue
  - Red Hat OpenShift 3.11
  - Advanced Container Management (included in Cloud Pak for Multicloud Management)
  - IBM Cloud Paks – Data (04/20), Apps (06/20), Multicloud Management (manage Power from X86) (06/20) and OpenShift Container Catalogue
  - Red Hat OpenShift 4.3
- 2020 H2
  - Advanced Container Management (included in Cloud Pak for Multicloud Management)
  - IBM Cloud Paks – added Integration (Q3 20), Automation (Q4 20), Security (Q1 21) and OpenShift Container Catalogue
  - Red Hat OpenShift 4.x
- Planned
  - Mixed x86, Power and X/LinuxONE
  - Additional Cloud Pak and ISV content
  - More Power container images in OpenShift Catalogue (>450 on 2/20)
  - OpenShift in IBM Cloud on Power h/w
- Built to run on
  - Any POWER8 or POWER9 System
  - PowerVC / PowerVM (AIX, Linux, i)
  - Linux KVM
  - Linux Bare metal

Power Virtual Server offerings

Belisama

- Multi-tenant, self managed, Power compute as-a-service in IBM Cloud with consumption-based OPEX pricing

https://cloud.ibm.com/docs/power-iaas?topic=power-iaas-about-virtual-server

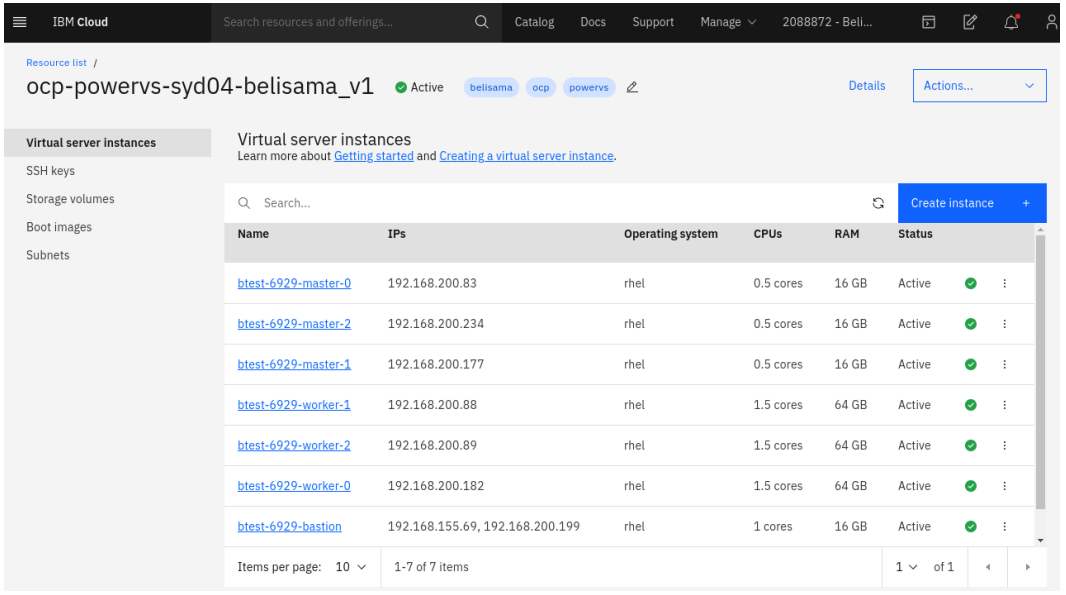
- Details
  - Systems: S922, E880 (Dallas, Washington only), E980 (all except Dallas, Washington)
  - Compute: 0.25-153 cores (15 for S922, 153 for E880/980)
    Dedicated or Shared option (capped or uncapped)
  - Memory: 8-64 GB per core (can go above 64GB at a higher rate)
  - Storage: Tier 3 (SSD) or Tier 1 (NVMe)
    10 GB minimum / 2 TB maximum per disk, 10 GB increments
  - Network: Public and/or Private IP
  - OS: AIX / IBM i, Linux
  - Location: Dallas, Washington, Frankfurt, London Torronto, Sydney
- Pricing
  - https://cloud.ibm.com/docs/power-iaas?topic=power-iaas-pricing-virtual-server

- Note: My steps part of PoC, portions changing / scripted as we speak
- PowerVS Prerequisites
  - IBM Cloud Account
  - Create Power Systems Virtual Server Service Instance
  - Create Private Network
  - Create OVA images (RHCOS and RHEL 8.2)
  - Uploading to IBM Cloud Object Storage and import into PowerVS (Console / Aspera / API)

Installation

Belisama

- Install IBM cloud CLI
- Configure Automation Host (Linux, Mac, Windows)
  - Install / Configure terraform
- Run Install script
- Tidy-up (delete bootstrap Node, configure DNS, Security/Authentication/keys etc)



Terraform configuration

Belisama

- File var.tfvars

```
<snip>
### IBM Cloud details

ibmcloud_api_key            = "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
ibmcloud_region             = "syd"
ibmcloud_zone               = "syd04"
service_instance_id         = "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
. . .
### Minimalistic config suitable for trying out Cloud Pak for Data (CP4D)
bastion                     = {memory      = "16",   processors  = "1"}
bootstrap                   = {memory      = "16",   processors  = "0.5",  "count"   = 1}
master                      = {memory      = "16",   processors  = "0.5",  "count"   = 3}
worker                      = {memory      = "64",   processors  = "1.5",  "count"   = 3}
. . .
rhel_image_name             = "rhel-82_ppc64le-001"
rhcos_image_name            = "rhcos-454-openshift.ppc46le-001"
processor_type              = "shared"
system_type                 = "s922"
network_name                = "ocp-powervs-syd04net"

rhel_username               = "root"
public_key_file             = "data/id_rsa.pub"
private_key_file            = "data/id_rsa"
rhel_subscription_username  = "antony.steel@belisama.com.sg"
rhel_subscription_password  = "xxxxxxx"
rhel_smt                    = 4
<\snip>
```

- Final output

```
bastion_private_ip = 192.168.xxx.bpriv
bastion_public_ip = 130.198.xxx.bpub
bastion_ssh_command = ssh -i data/id_rsa root@130.198.xxx.bpub
bootstrap_ip = 192.168.xxx.bs
cluster_authentication_details = Cluster authentication details are available
                            in 130.198.xxx.bpub under ~/openstack-upi/auth
cluster_id = btest-48c8
etc_hosts_entries =
130.198.xxx.yy api.btest-48c8.ibm.com console-openshift-console.apps.btest-48c8.ibm.com
         integrated-oauth-server-openshift-authentication.apps.btest-48c8.ibm.com
         oauth-openshift.apps.btest-48c8.ibm.com
         prometheus-k8s-openshift-monitoring.apps.btest-48c8.ibm.com
         grafana-openshift-monitoring.apps.btest-48c8.ibm.com example.apps.btest-48c8.ibm.com

install_status = COMPLETED
master_ips = [
  "192.168.xxx.y1",
  "192.168.xxx.y2",
  "192.168.xxx.y3",
]
oc_server_url = https://api.btest-48c8.ibm.com:6443
storageclass_name = nfs-storage-provisioner
web_console_url = https://console-openshift-console.apps.btest-48c8.ibm.com
worker_ips = [
  "192.168.xxx.y4",
  "192.168.xxx.y5",
  "192.168.xxx.y6",
]
```

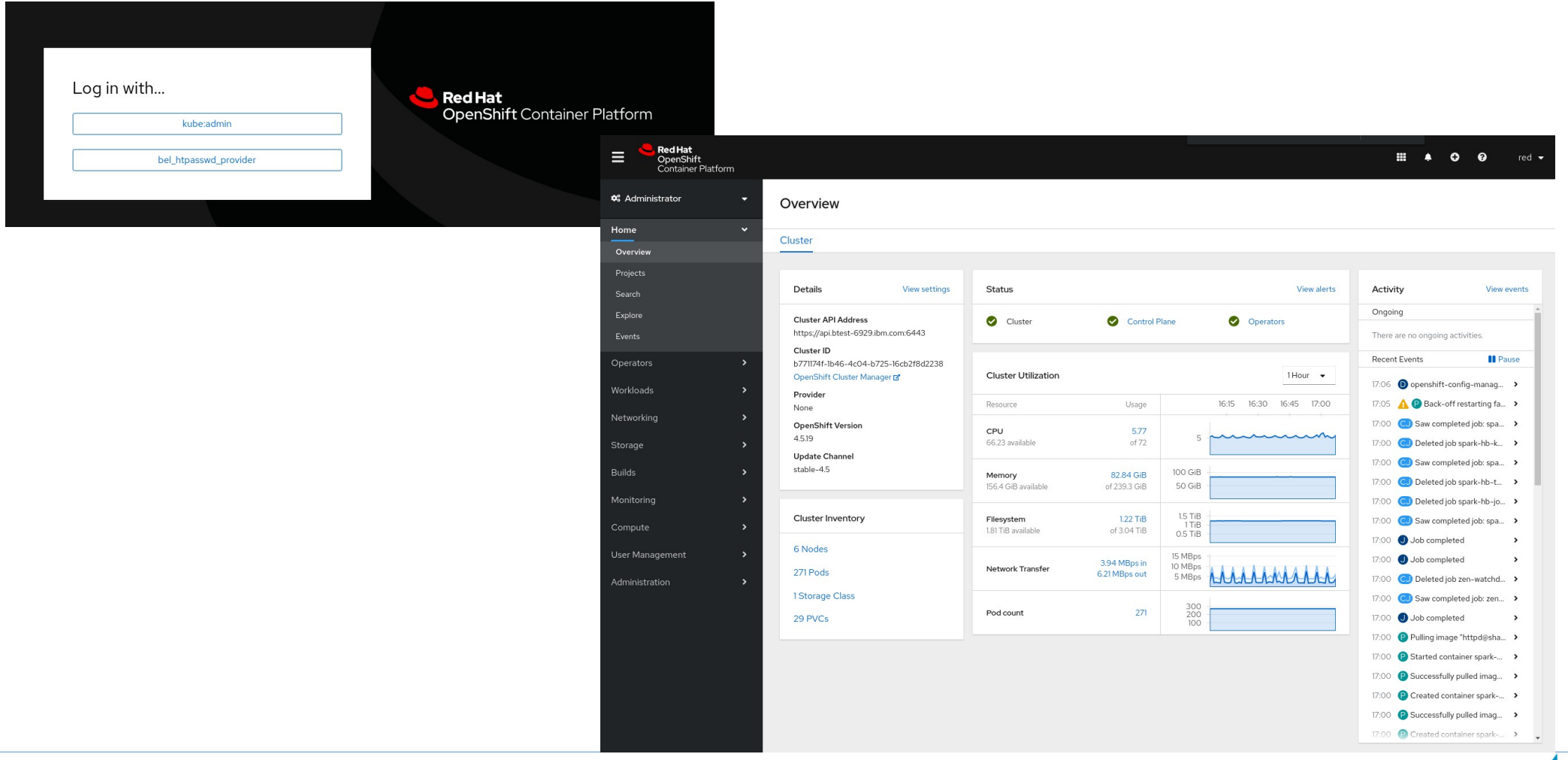
- OpenShift CLI (oc)

```
oc login https://api.btest-6929.ibm.com:6443 -u kubeadmin -p $(cat ./kubeadmin-password)
Login successful.

You have access to 66 projects, the list has been suppressed. You can list all projects with 'oc projects'

Using project "default".
vali:/data/proj/belisama/powervs/auth: oc get nodes
NAME       STATUS   ROLES    AGE   VERSION
master-0   Ready    master   48d   v1.18.3+10e5708
master-1   Ready    master   48d   v1.18.3+10e5708
master-2   Ready    master   48d   v1.18.3+10e5708
worker-0   Ready    worker   48d   v1.18.3+10e5708
worker-1   Ready    worker   48d   v1.18.3+10e5708
worker-2   Ready    worker   48d   v1.18.3+10e5708

oc projects
You have access to the following projects and can switch between them with 'oc project <projectname>':

    abhi-project
    abhi2
  * default
    demo
    demo2
    httlp
    kube-node-lease
    kube-public
    . . .
```

Next Steps

Belisama

- Explore GUI
- Update OpenShift
- Create users and authentication methods (used httpassword for demo)
- Install Cloud Paks


Log in with...
kube:admin
bel_htpasswd_provider
Red Hat
OpenShift Container Platform
Red Hat
OpenShift
Container Platform
red
Administrator
Home
Overview
Projects
Search
Explore
Events
Operators
Workloads
Networking
Storage
Builds
Monitoring
Compute
User Management
Administration
Overview
Cluster
Details
View settings
Cluster API Address
https://api.btest-6929.ibm.com:6443
Cluster ID
b771174f-1b46-4c04-b725-16cb2f8d2238
OpenShift Cluster Manager
Provider
None
OpenShift Version
4.5.19
Update Channel
stable-4.5
Cluster Inventory
6 Nodes
271 Pods
1 Storage Class
29 PVCs
Status
View alerts
Cluster
Control Plane
Operators
Cluster Utilization
1 Hour
Resource
Usage
16:15
16:30
16:45
17:00
CPU
66.23 available
5.77
of 72
Memory
156.4 GiB available
82.84 GiB
of 239.3 GiB
Filesystem
1.81 TiB available
1.22 TiB
of 3.04 TiB
Network Transfer
3.94 MBps in
6.21 MBps out
Pod count
271
Activity
View events
Ongoing
There are no ongoing activities.
Recent Events
Pause

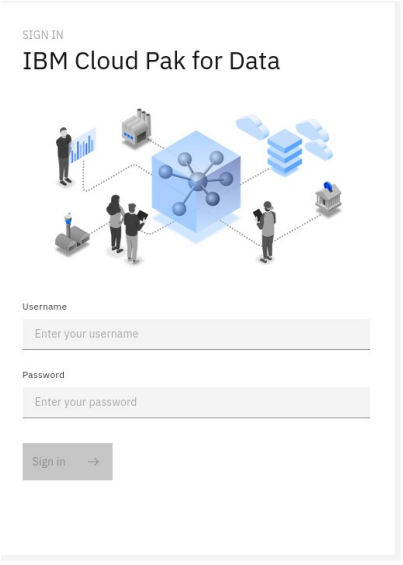
Installing CloudPak for Data

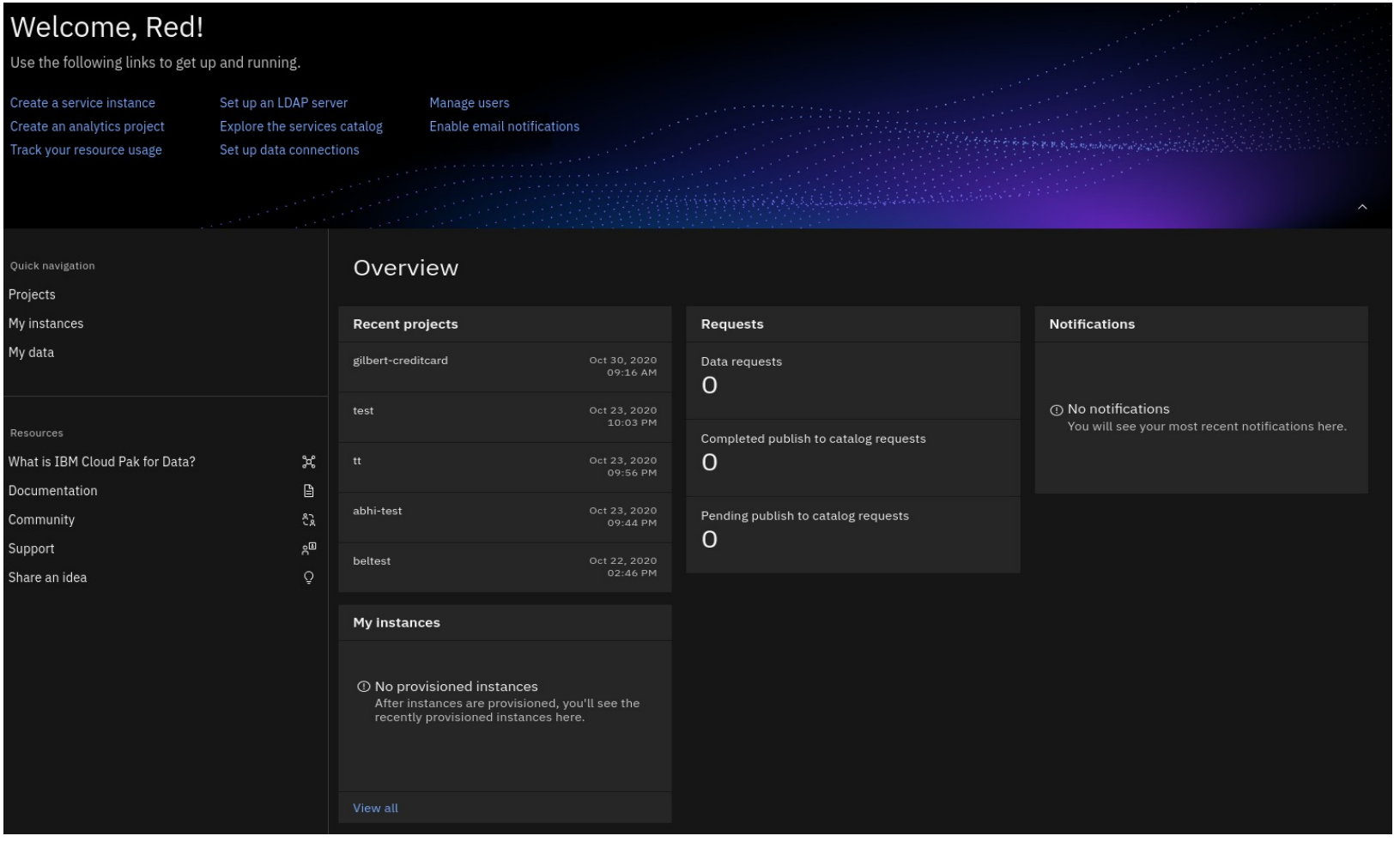
Belisama

- Steps
  - Create user and log in as user
  - Log into kubernetes cluster as kubeadmin
  - Download and apply some patches
  - Download IBM CloudPak for Data installation utility from Git Hub
  - Edit config file and create a new project
  - Install the CP4D components (Control plane, Watson Studio local, Watson Machine Learning and their components)
  - Install patches

CP4D GUI

Belisama





Session: Session: OpenShift and CloudPak for Data on PowerVS

¿ Questions ?

Thanks!

Please contact us if you wish a more detailed demonstration or PoC using our environment

For further information….
Contact:

Gilbert Thomas gilbert@sg.ibm.com
+65 94311518

Antony (Red) Steel antony.steel@belisama.com.sg
+65 9789 6663

Power Systems

Backup Slides

Cloud Pak for Security

Belisama

- Why?
  - Opportunity to capitalize on the huge momentum around IBM Cloud Paks to position as attachment for ELAs
  - Flexible consumption model allows customers to deploy to meet their needs
  - New functionality will give you a competitive advantage to any deal
  - Add new use cases and differentiation to existing offerings
  - Applicable for growth, strategic and key accounts

Cloud Pak for Security

Belisama

- Unified Interface
  - Threat management
  - Threat hunting
  - Insider thread
  - Cloud workload protection
  - Incident response
  - Identity and access
  - Data security
  - Privacy compliance
  - Risk assessment

Universal data service Security Orchestration and Automation DevApp framework

- Connectors at launch
  - Qradar and Qroc
  - Splunk Security SIEM
  - Carbon Black CB Response
  - Guardium Data Protection
  - Tenable.io
  - Elastic
  - BigFix Lifecycle and Compliance
  - Azure
  - IBM Cloud Security Advisor
  - Microsoft ATP
- Connectors 1H 2020
  - McAfee ePO
  - CrowdStrike
  - AWS CloudWatch

IBM Cloud Pak for Security 1H 2020

Belisama

- Applications
  - Data Explorer (Federated Search and Investigation
  - Incident response orchestration (Resilient)
- Services
  - Universal Data Service
  - Connected Asset and Risk
  - Cases
  - Dev Framework
- All on OpenShift

Security disconnected and fragmented

Belisama

- As security fragmentation and disconnect is exacerbated by cloud
  - Connect data (Complete insights)
    - Uncover hidden threats
    - Make better risk based decisions
    - Leave data where it is
    - maximise investments
  - Connect workflows (streamline operations)
    - Respond faster as a team and business
    - Orchestrate across security cases
    - Drive security into lines of your business
    - Reduce integration costs
  - Connect anywhere (scale)
    - Run anywhere
    - Reduced vendor lock-in
    - Increase and shift investment as needed
    - Extend your team’s capabilities

IBM Security focus

Belisama

- IBM Security strategy focuses on:
  - Strategy and Risk
    - We help implement security strategies and improve compliance with a global network of seasoned advisors.
  - Threat Management
    - We help stop threats with teams that can run your SOC and respond at a moment's notice with some of the most powerful tools in the business.
  - Digital Trust
    - We help grow the business by protecting critical data, users, and assets wherever they exist.
- Threat management consists of 4 pillars
  - Visibility; Detect; Investigate; Respond
- Uses
  - QRadar’s complete visibility and advanced security analytics
  - QRadar Advisor’s automated investigation and root cause analysis
  - Resilient’s leading orchestration and automation response platform
  - i2’s human-led, investigation and hunting, using cyber and non-cyber data
  - X-Force Exchange’s threat intelligence sharing platform

Summary

Belisama

- 2019
  - Federated Search
    - Investigate faster with federated search across QRadar, Splunk, Elastic, Guardium, AWS, Azure, Carbon Black, BigFix, Data Lake and McAfee
  - Orchestration and Automation
    - Respond faster and more thoroughly with robust orchestration and automation capabilities
  - Run anywhere
    - Deploy anywhere across hybrid multicloud architecture
  - SDK and Ecosystem
    - Expand data sources and capabilities with SDK for partners and customers to create new connectors and applications
- 2020
  - Thread Intelligence insights
    - Operationalise threat intelligence information across a hybrid multicloud infrastructure lowering risk and response times
  - Security operations dashboard
    - Create new security threat and risk views for analysts and managers
  - Data Lake
    - Open flexible data lake enabling users to bring their data to the CP4SEC (only if they require it)
  - Watson Advisor
    - Automated investigations across multiple SIEM, Cloud and EDR platforms, lowering risk and speeding up investigations
  - Threat management
    - Seamless, simple workflow across threat detection, response and investigation

- With the 1.3.0 release, you can manage your hybrid applications across containers and VMs no matter where they run.
- The Cloud Pak is a set of open, pluggable tools built around a core application and governance model. The reason this application and governance model is the centre of the Cloud Pak is that it enables disparate teams to come together around a common understanding of what the application is and how it is behaving.

- The Cloud Pak for Multicloud Management as four pillars:
  - Application Lifecycle Management
    - provides unified and simplified options for constructing and deploying applications and application updates. With these functions, developers and DevOps personnel can create and manage applications across environments through channel and subscription-based automation.
  - Infrastructure Lifecycle Management
    - enables two distinct capabilities:
      - deploying and managing VMs and Kubernetes clusters
      - self-service capabilities to orchestrate resources.
    - For Kubernetes, we enable enterprises to create, import, and manage clusters across different cloud providers. For VMs, we can extend the current VM management tools that enterprises use to improve visibility and control and help those getting started with hybrid cloud VM management. Across VMs and Kubernetes, we leverage Terraform and Service Automation for multicloud, a self-service management tool that empowers developers and administrators to meet business demands.
  - Performance Monitoring
    - gives Site Reliability Engineers (SREs) a consistent monitoring method across the enterprise to any public or private cloud. Our monitoring capabilities puts our core application model to work by enhancing application resiliency for microservices based applications in addition to monitoring for traditional resources across enterprises.
  - Governance Risk and Compliance Management
    - ensures that applications meet the security and compliance requirements of the enterprise, which is critical in managing a hybrid application. With IBM Cloud Pak for Multicloud Management v1.3, we deliver system dashboards that provide context to security and compliance findings for both applications deployed to Kubernetes and on traditional VMware/VM environments.

Summary of new features

Belisama

- Now runs on OpenShift 4.3 (in addition to 3.11 and 4.2)
- Main page has an Overview showing high level details across clouds, resources and applications
- The user interface now supports deploying a template to multiple target clouds by showing a list of all the supported cloud connections
- Monitor your application performance and availability with a default Synthetic test
- Using Red Hat Ansible Tower to automate and streamline greenfield deployments (an agent that can be installed on VMs on premise or in Kubernetes Clusters) in the Cloud Pak
- Automate agent integration to the Cloud Pak
- A new JBoss data collector

- Mix AIX and i applications with Cloud agility / management
  - Cloud Pak for Applications
  - OpenShift for Cloud Services
  - Travis CI or Jenkins for Continuous Integration / Continuous Delivery
  - Cloud Automation Manager for VM management
  - Multicloud Manager for Containers