



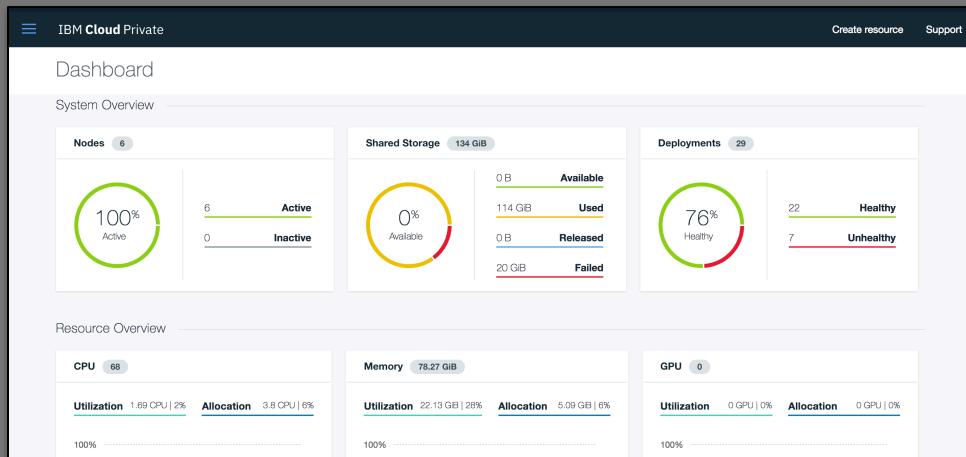
IBM Cloud Containers Workshop

Multi Cloud Management Metering, Monitoring and Logging



Remember: Try IBM Cloud Private

Free Community Edition



<http://ibm.biz/Try-IBMCLOUDPRIVATE>

Agenda

- Multi Cloud Management
- Metering
- Native monitoring and Dashboard
- Prometheus





Traditional
IT

Rewrite **some**,
Modernize **most**

Open architecture
by design

Innovate
with **guardrails**

>20%
> 5%

Open Hybrid Multi cloud

*multi cloud **by design***

The path of multicloud can accelerate rapidly . . .

IBM Multicloud Manager maintains your pace of innovation

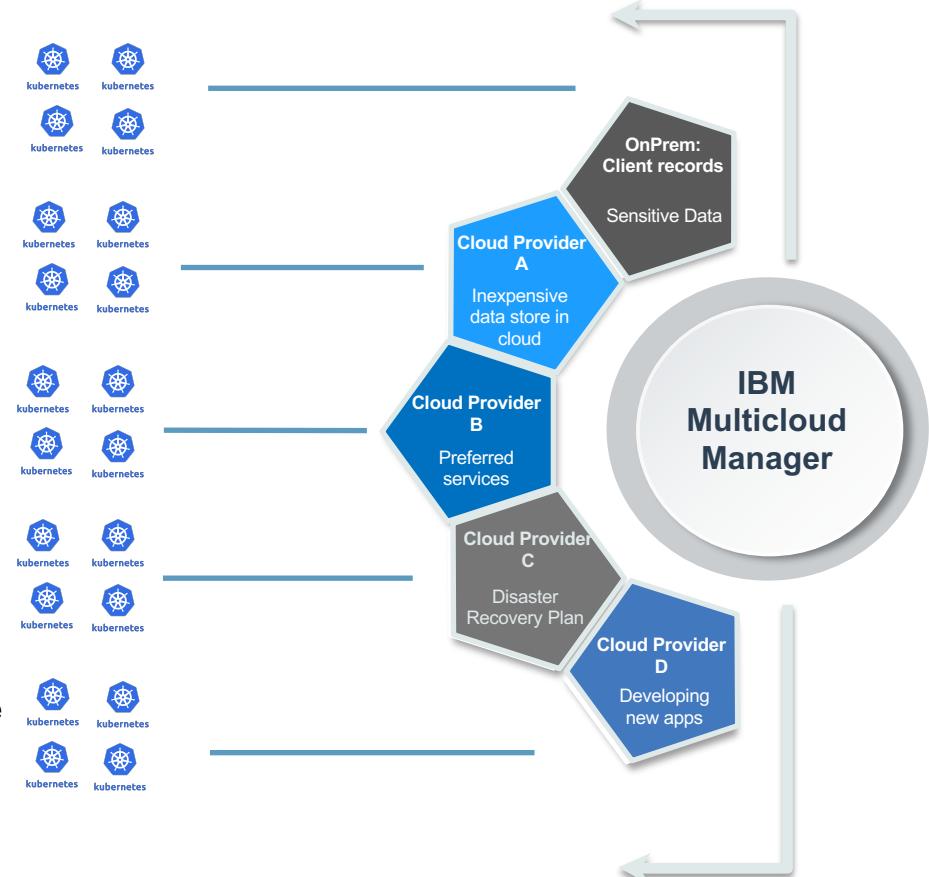
Multiple clusters to manage geographical data residency laws

Reduce on prem storage costs by storing non sensitive data in public cloud

Leverage most desired services in preferred cloud (AI/Analytics/object storage)

Load all applications on another provider maintain operations if one provider goes down (disaster recovery)

Build net new client experience cloud native apps on another cloud to posture against vendor lock-in



From 1 cluster
on 1 cloud



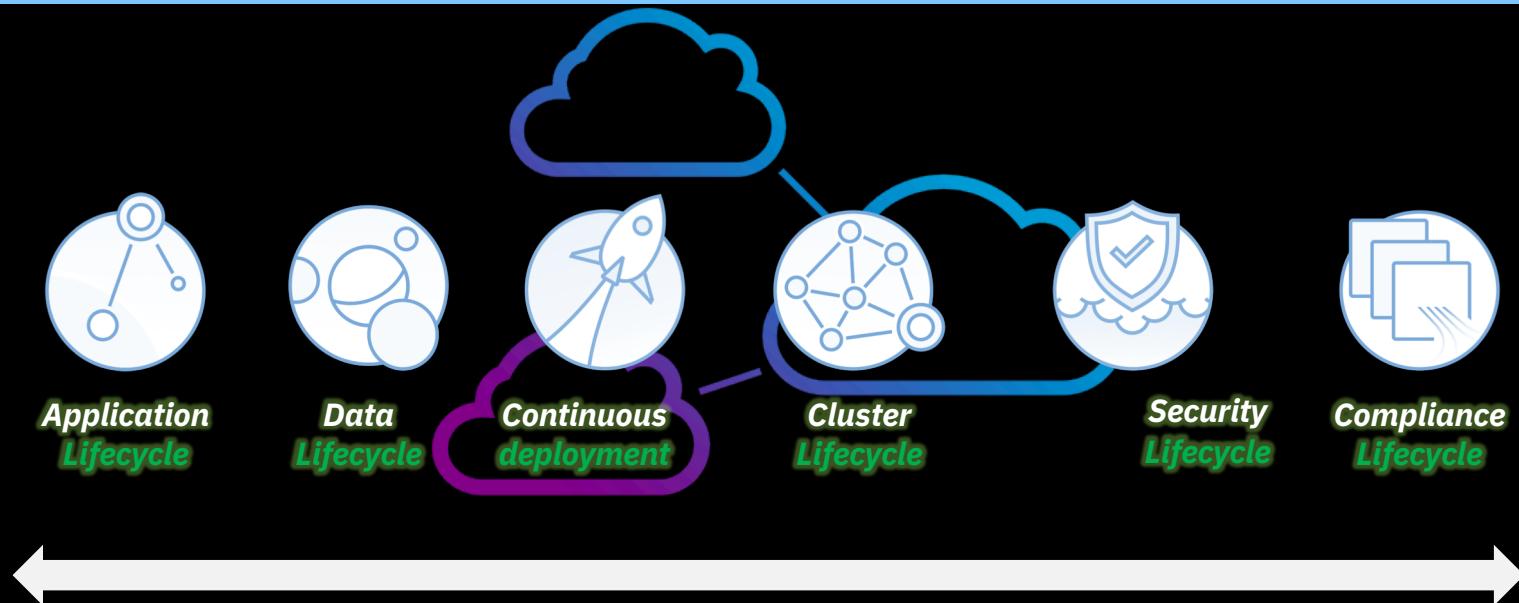
to many
clusters on
numerous
clouds



**IBM
Multicloud
Manager**

seamlessly
moves you
through the
journey

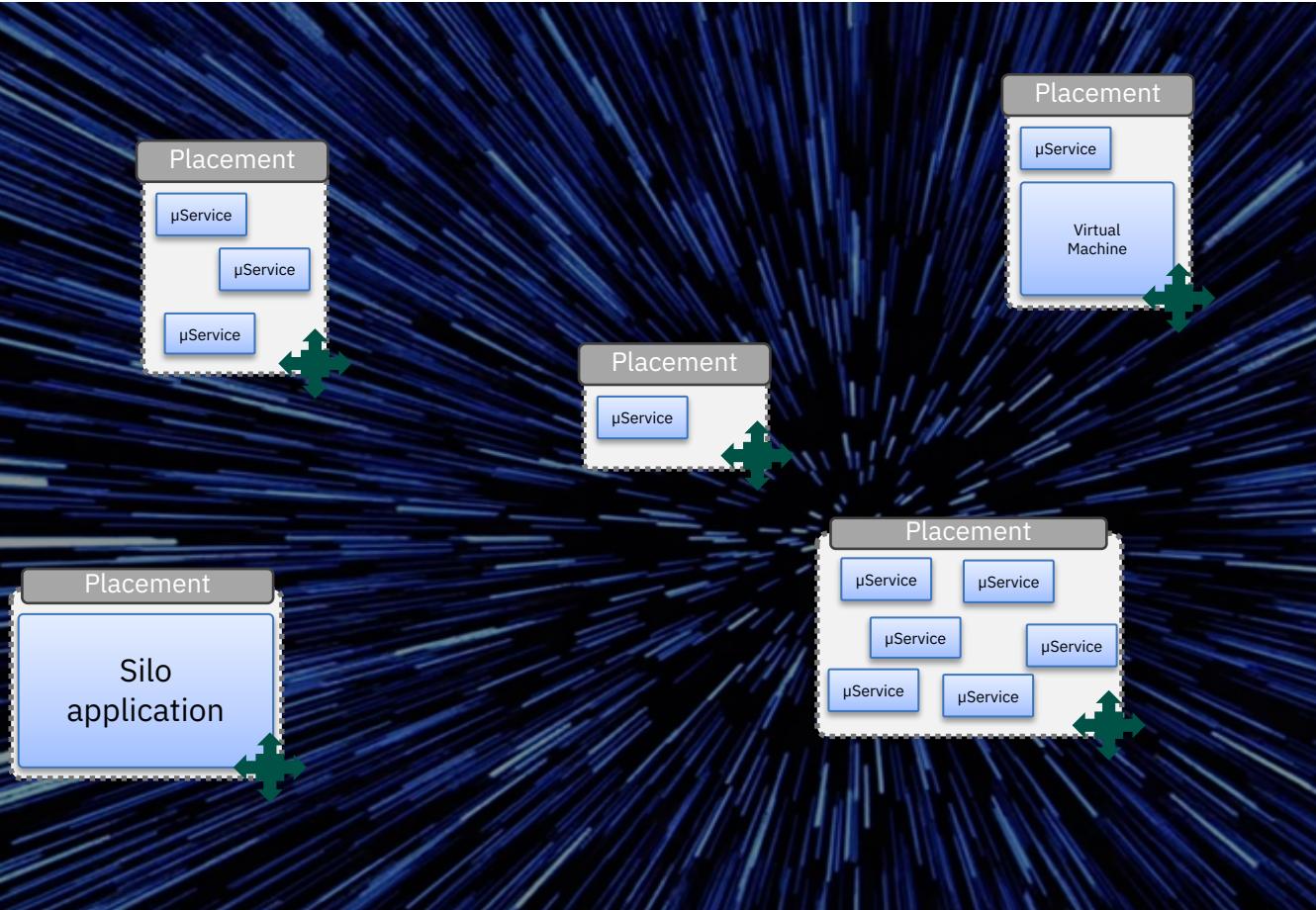
Multi cloud management need to look at all the different life cycles



Across Clouds, Public and Private

A new way of defining applications for Multi Cloud

<https://github.com/kubernetes-sigs/application>



Deployable component

Application Boundary

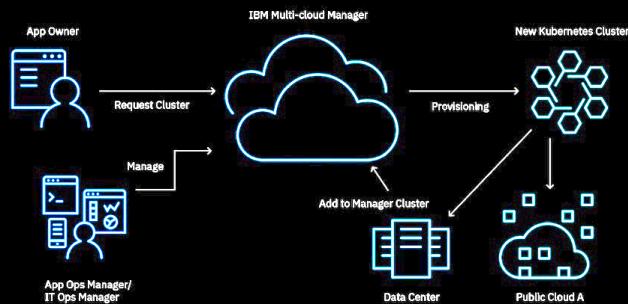
Placement Policy

Deployment technology

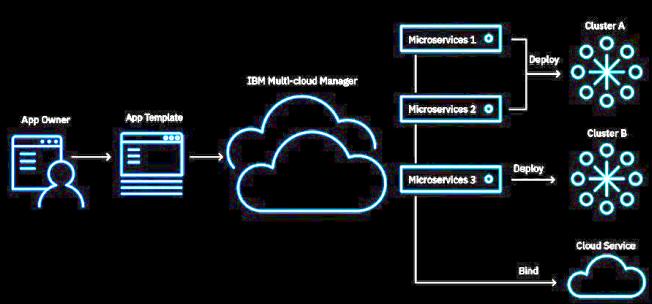
IBM Multi Cloud Manager

Across Kubernetes Clusters - Public and Private

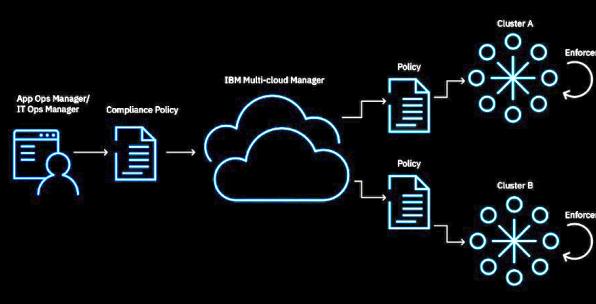
Cluster Lifecycle Management



Application-Centric Management



Policy-Based Governance



*As organizations modernize and deploy **containerized** clusters, new challenges are introduced....*

I need
broad
Visibility

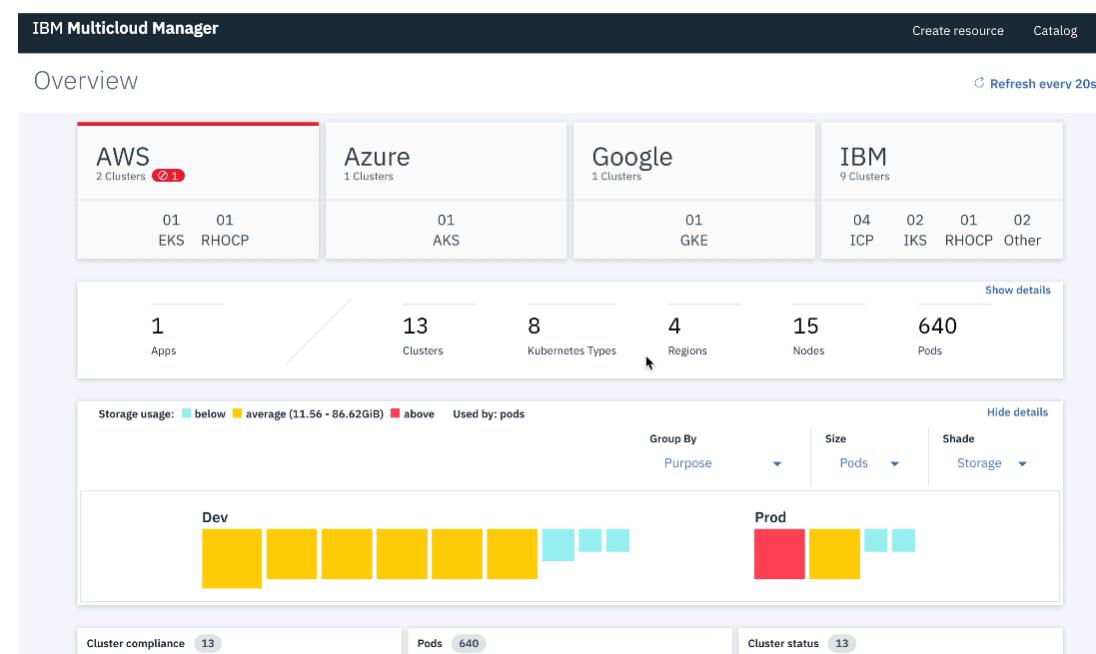
I need
automated
Governance

I need
seamless
Application
Management

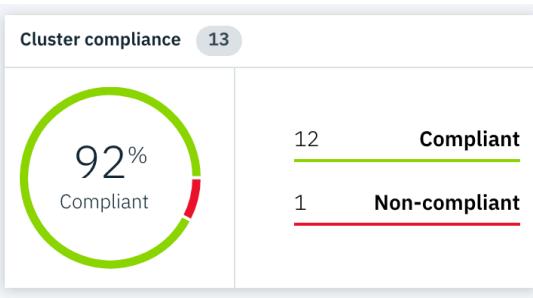
Visibility: Clear insight into any environment, any application, any cluster

One Dashboard: See health, usage, policy adherence **on any cluster, any environment**

- Where are the failed components?
- Where are my services running?
- How can I monitor applications across clusters and clouds?
- How can I manage clusters as if they were one environment?
- How do I monitor usage across clouds?



Governance: Maintain controls across applications & clusters with policies



Overview

Azure	Google	AWS	IBM
1 Clusters	1 Clusters	2 Clusters (0:1)	9 Clusters
01 AKS	01 GKE	01 EKS 01 RHOC	04 ICP 02 IKS 01 RHOC 02 Other

I need automated Governance

- How do I set consistent security policies across environments?
- Which clusters are in compliance with our policies?
- How can I manage configuration across this large environment?
- How can I place workloads based on capacity, policy?

⟳ Refresh every 20s

Create Placement & Security policies directly from MCM console and push to all clusters with a click



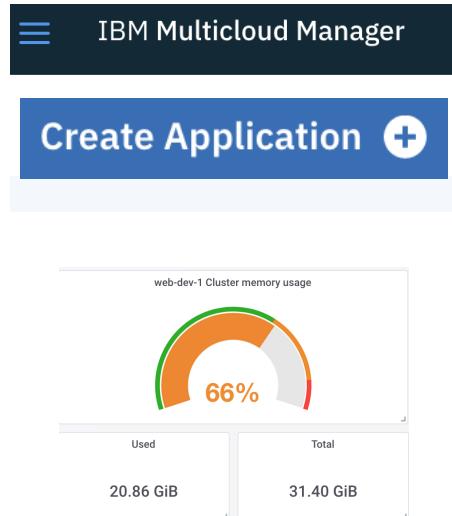
Create Compliance

Paste content from your compliance YAML file in the editor.

```
1 51 -   - from:
2 52 -     podSelector: {}
3 53 -       matchLabels: null
4 54 -       matchNamespaces: null
5 55 -       matchAnnotations: null
6 56 -       matchTaints: null
7 57 -       namespacesMustHave:
8 58 -         objectDefinition:
9 59 -           kind: LimitRange
10 60 -             metadata:
11 61 -               name: mem-limit-range
12 62 -             spec:
13 63 -               default:
14 64 -                 memory: 512Mi
15 65 -                 hard:
16 66 -                   memory: 256Mi
17 67 -             type: Container
18 68 -       remediationAction: enforce
```

Cancel Create Compliance

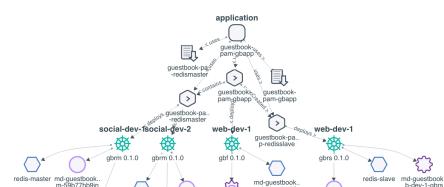
Application Management: Create, Monitor, Manage and Backup



Create an Application across environments all in **one place**



View all relevant information, deployments and placement policies for your applications

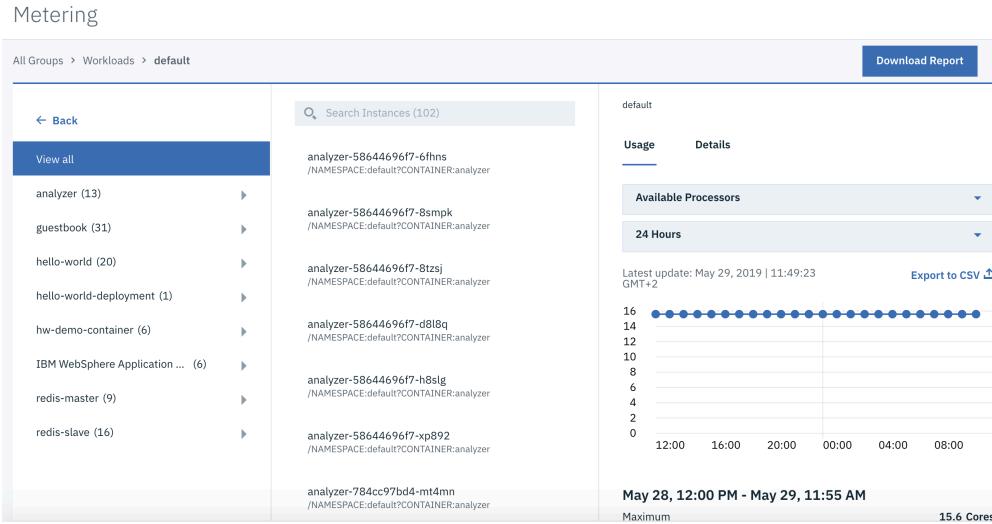


I need help with Application Management

- How do I deploy applications across these environments?
 - How do I move workloads across environments?
 - How can I backup my applications?
 - How do I do Business Continuity?

IBM Cloud Private metering service

- Metering reports at product and namespace level
 - Workloads
 - Storage
 - Platform
- Use the metering service to manage chargebacks
- Track usage of IBM products that are running outside of your IBM Cloud Private cluster



If MCM is installed, the metering service on the MCM hub cluster includes more MCM metrics

Levels of CSMO

Level 0 – Out of the box

Built-in Dashboard for metrics

Built-in Log console

Level 1 – Default monitoring workloads

Deploy Prometheus for metrics

Deploy ELK for Logging

Level 2 – External Monitoring and Logging

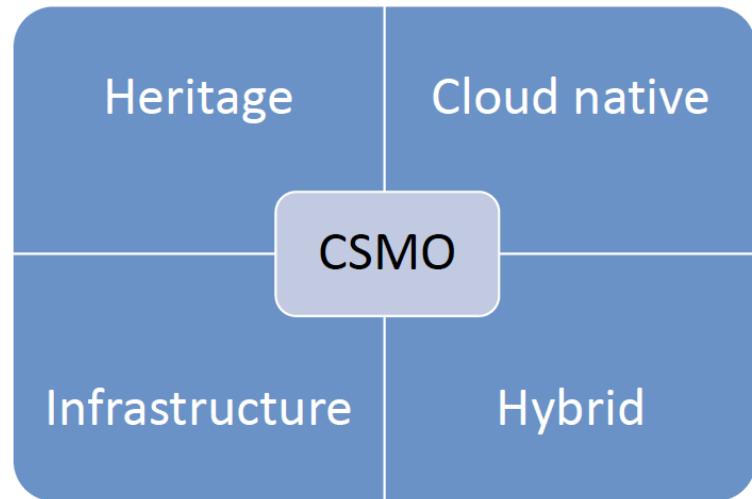
External APM monitors metrics and thresholds

ELK stacks forwards to external ELK stack

Level 3 – Integration with CSMO toolchain

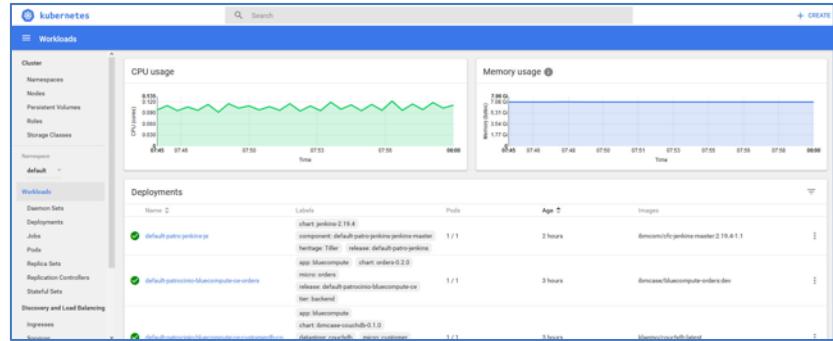
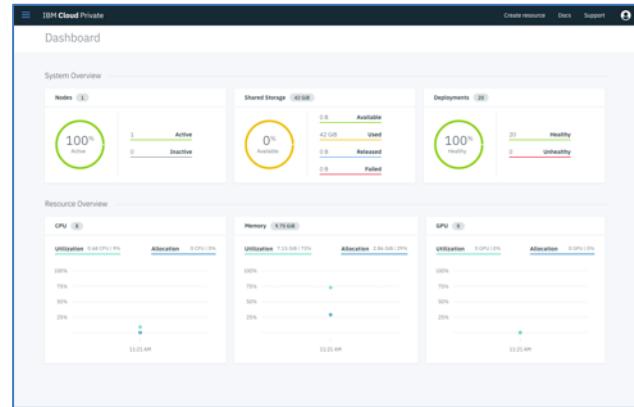
Forward events to NOI

Forward Log events to NOI



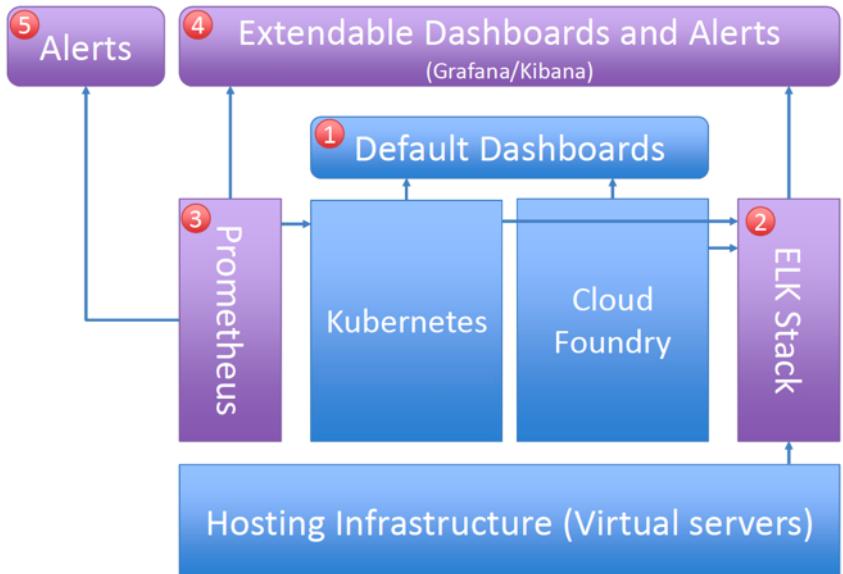
Level 0 - Monitoring an IBM Cloud Private Deployment

- High level Key Performance Indicators
 - 1. The **ICP dashboard** shows a high level overview of resource usage and status.
 - 2. The **Kubernetes Dashboard** shows CPU and memory consumption of the full cluster and deployments
 - 3. Within an application you can view **performance of the pod**.
- All well and good but you can only look and “not touch”.
- There are APIs available.
- We need to be able monitor and alert on Out of Bounds conditions for the applications and their infrastructure.



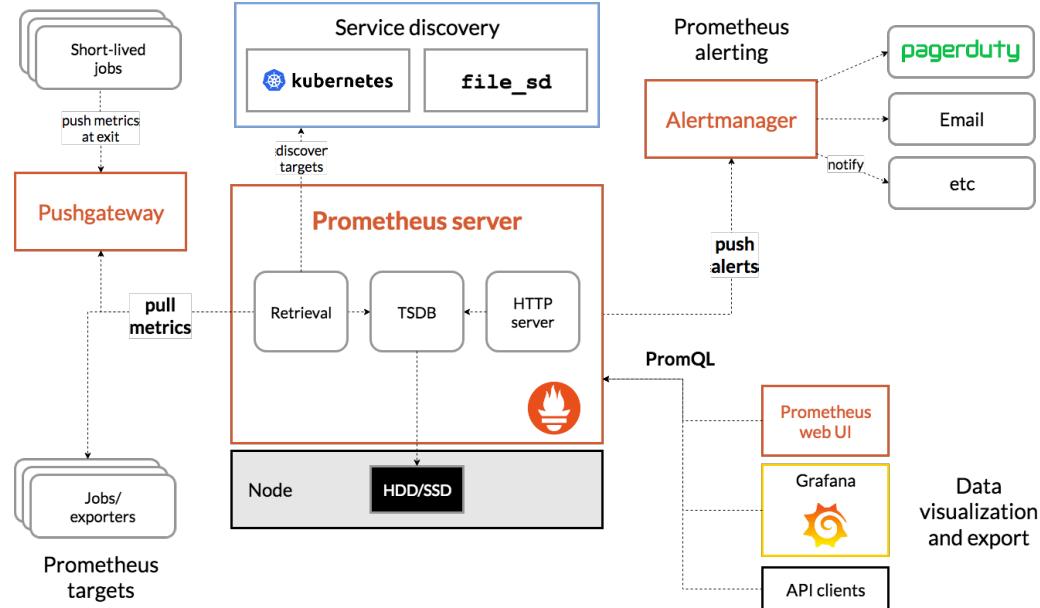
Level 1 – Default monitoring workloads

1. The ICP dashboard shows the status, metrics and log entries for all the workloads & Infrastructure.
2. An **ELK stack** collects logs from all the workloads & Infrastructure. Multiple ELK stacks are possible (but each log can only reside in one stack)
3. A Prometheus stack collects logs from the Kubernetes workload
4. Customizable Grafana and Kibana dashboards are available for use
5. Prometheus can send alerts to external system (email, webhooks..)



Prometheus

- Prometheus is a Cloud Native Computing Foundation (CNCF) project, is a systems and service monitoring system. It collects metrics from configured targets at given intervals, evaluates rule expressions, displays the results, and can trigger alerts if some condition is observed to be true.
- <https://prometheus.io/>
- Prometheus has several components for Time Series Data Collection, an Alert Manager and the Prometheus Server which scrapes and stores the data.
- Prometheus is installed via a Helm Chart into ICP



Grafana

- Grafana is an open source metric analytics & visualization suite. It is most commonly used for visualizing time series data for infrastructure and application analytics but many use it in other domains including industrial sensors, home automation, weather, and process control.
- <https://grafana.com/grafana>
- While Prometheus has limited dashboarding capabilities of its own, regular display of metrics will be done using Grafana dashboards.
- Prometheus and the Grafana Dashboards are installed together as applications into ICP using Helm charts.

Grafana – Dashboards for ICP

Not Secure | <https://169.50.200.93:8443/grafana/?orgId=1>

Find dashboards by name

General

- Cluster Network Health (Calico)
- ElasticSearch
- Etc by Prometheus
- Helm Release Metrics
- ICP Namespaces Performance IBM Provided 2.5
- ICP Performance IBM Provided 2.5
- Kubernetes: Cluster Monitoring
- Kubernetes: POD Overview
- NGINX Ingress controller
- Node Performance Summary
- Prometheus Stats
- Rook-Ceph
- Storage GlusterFS Health
- Storage Minio Health

elasticsearch

release

kubernetes

pods

nginx

nodes

prometheus

ICP Performance IBM Provided 2.5

interval 5m

Youngest Node Uptime	Total memory	Available Memory	Memory Free	ICP Total CPU 15 Minute Av...
4.93 day	31.40 GiB	16.43 GiB	52%	9%
Oldest Node Uptime	Total Disk Space	Disk Space Available	Disk Space Used	Active Containers Last 5 min
4.93 day	916.90 GiB	807.11 GiB	12.4%	183

Memory by node

chart	heritage	inst	name	namespace	utilization
ibm-icpmonitoring-1.4.0	Tiller	1	k8s_apiserver_k8s-master-169.50.200.93_kube-system_20bc0128b51679cbadffda288061cda2_2	kube-system	12.71
ibm-icpmonitoring-1.4.0	Tiller	1	k8s_custom-metrics-adapter_k8s-master-dabcbf54fb-11e9-9ead-06e5b28cb17b_1	kube-system	6.95

Top 5 Containers by CPU

name	namespace	utilization
k8s_microclimate-jenkins_k8s-master-169.50.200.93_kube-system_3ee51bbe-53a7-11e9-89f1-06e5b28cb17b_0	kube-system	12.71
k8s_es-master_logging-elk-master-76dcbbc-wqdrw_kube-system_851e0258-4fbc-11e9-9ead-06e5b28cb17b_1	kube-system	6.95

Top 5 Container by Memory

name
k8s_microclimate-jenkins_k8s-master-169.50.200.93_kube-system_3ee51bbe-53a7-11e9-89f1-06e5b28cb17b_0
k8s_es-master_logging-elk-master-76dcbbc-wqdrw_kube-system_851e0258-4fbc-11e9-9ead-06e5b28cb17b_1

Container CPU Utilization

Legend:

- k8s_apiserve
- k8s_custom
- k8s_auth-pd
- k8s_ec-client
- k8s_es-data
- k8s_controller
- k8s_logstash
- k8s_calico-node
- k8s_prometheus
- k8s_ip-allocator

23



Deploy a multi-cluster application using Multi Cloud Manager

Demo

