

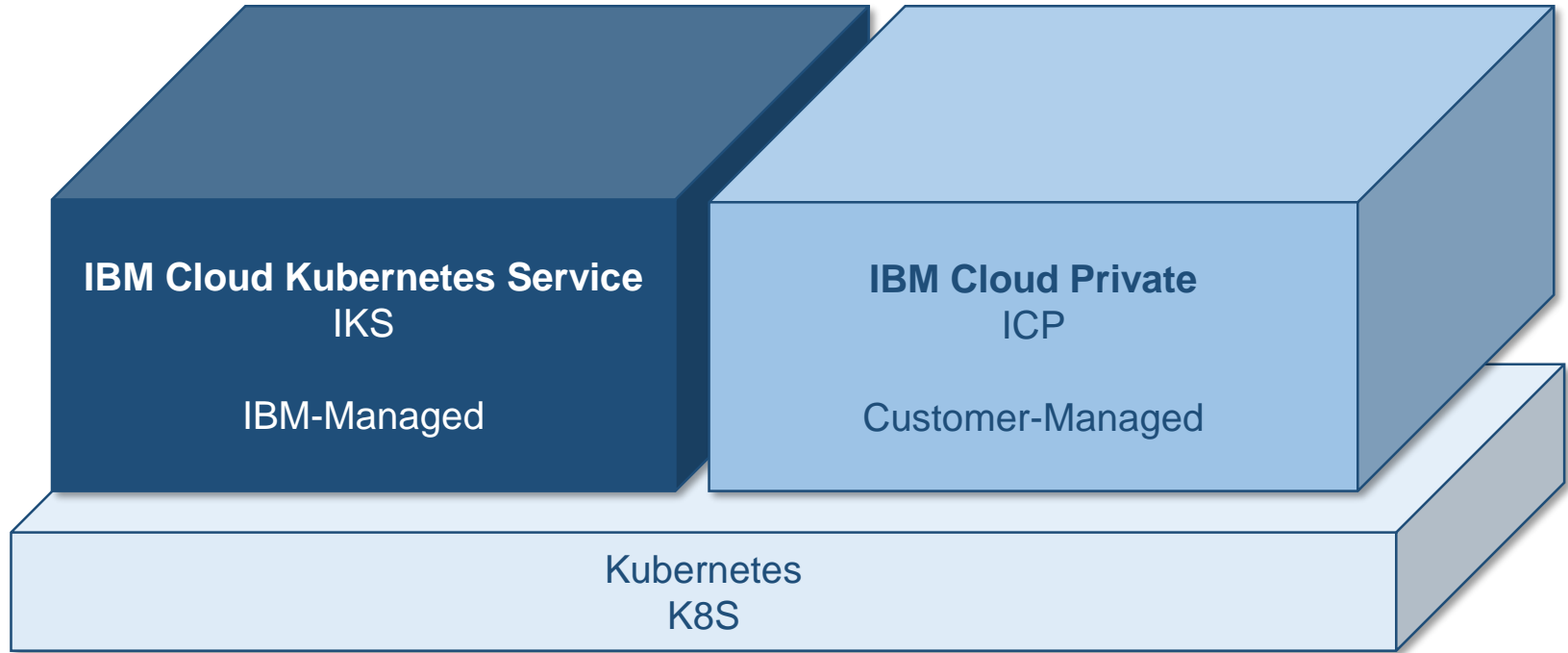


IBM Cloud Containers Workshop

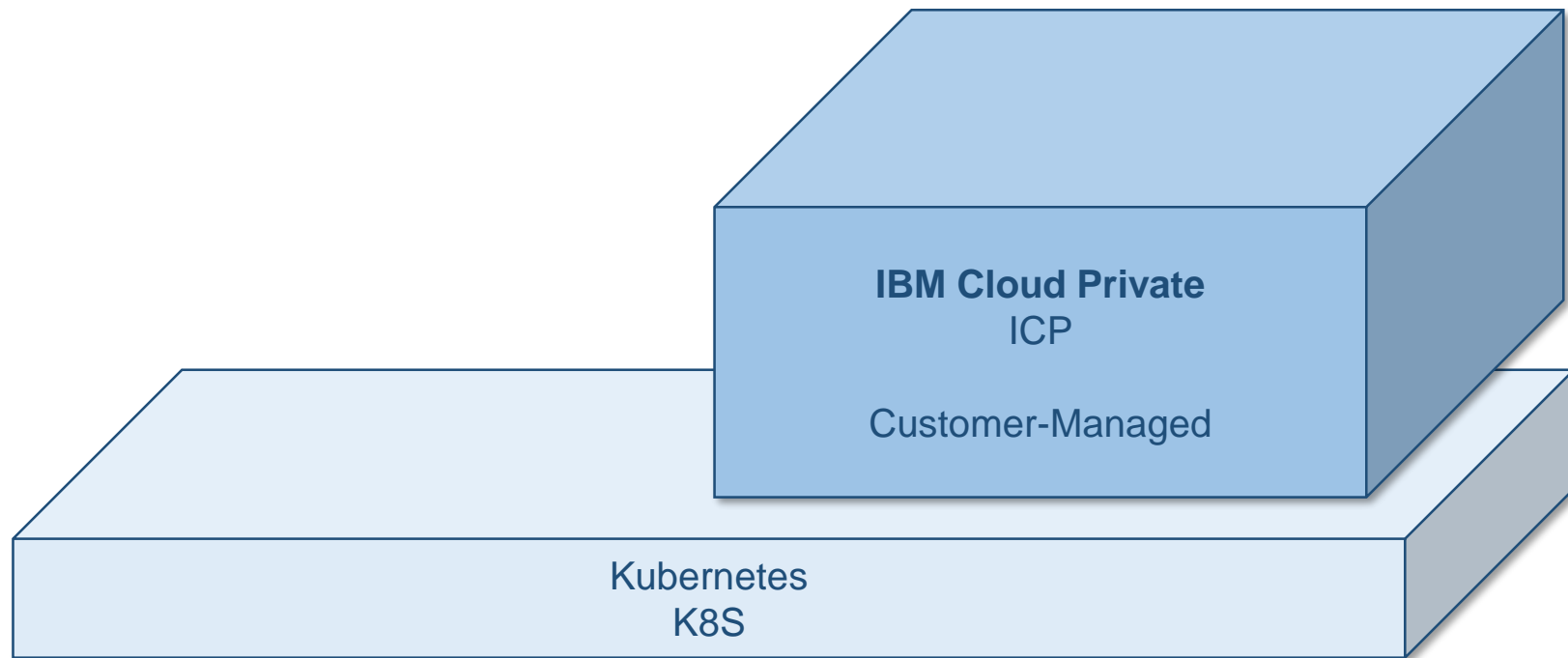
IBM Cloud Private Overview



- IBM Solutions based on Kubernetes



- IBM Solutions based on Kubernetes



What is IBM Cloud Private?

- Full stack private cloud SW
- Built using industry standard open source projects
- Sits behind the client's firewall (on-prem **OR** off-cloud)
- Software development, delivery & distribution platform
- Manages multi-cloud deployments



Rapid Innovation



Differentiated
Integration

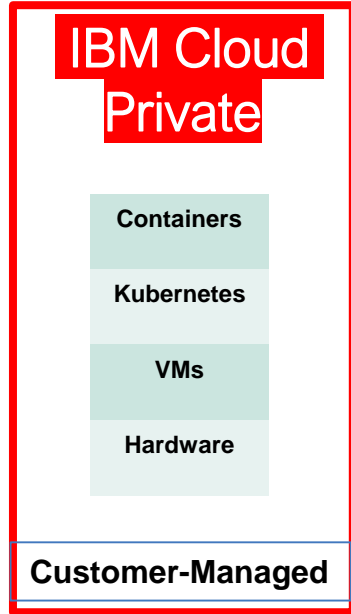


Investment Leverage

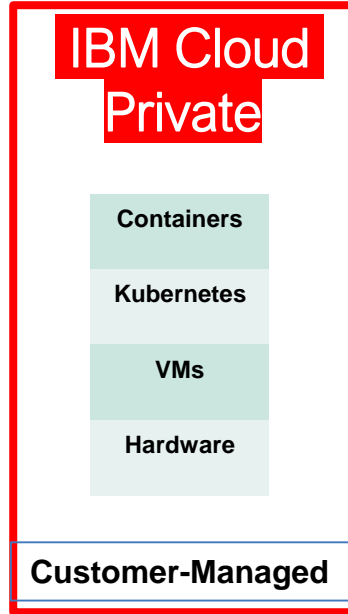


Management and
Compliance

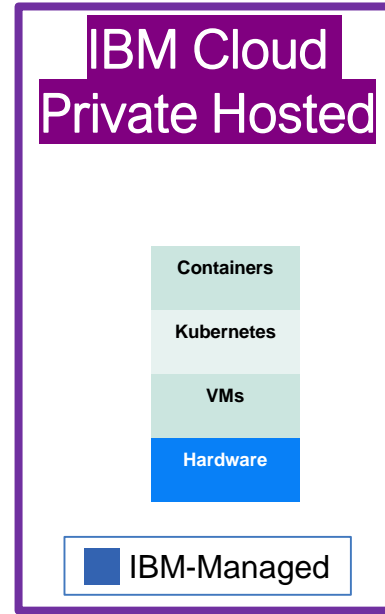
IBM solutions based on Kubernetes



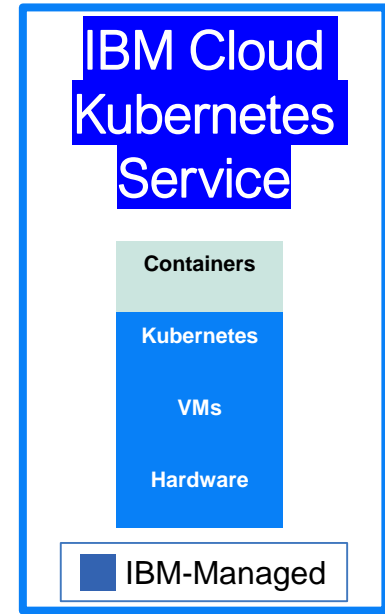
On Premises



IBM Cloud
Azure
AWS
Google Cloud Platform
(...)



IBM Cloud



IBM Cloud Private Solution Overview



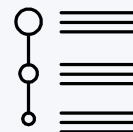
IBM Middleware & Open Source – e.g. Data, Analytics and Developer Services

Cloud-enabled middleware, application runtimes, messaging, databases & analytics to optimize current investments and rapidly innovate



Core Operational Services

To simplify Operations Management, Security, DevOps, and hybrid integration



Kubernetes-based Container Platform

Industry leading container orchestration platform



Cloud Foundry

For prescribed application development & deployment



Terraform (CAM)

Infrastructure as Code for multi-cloud provisioning to public and on-prem private clouds

Runs on existing IaaS: **vmware**



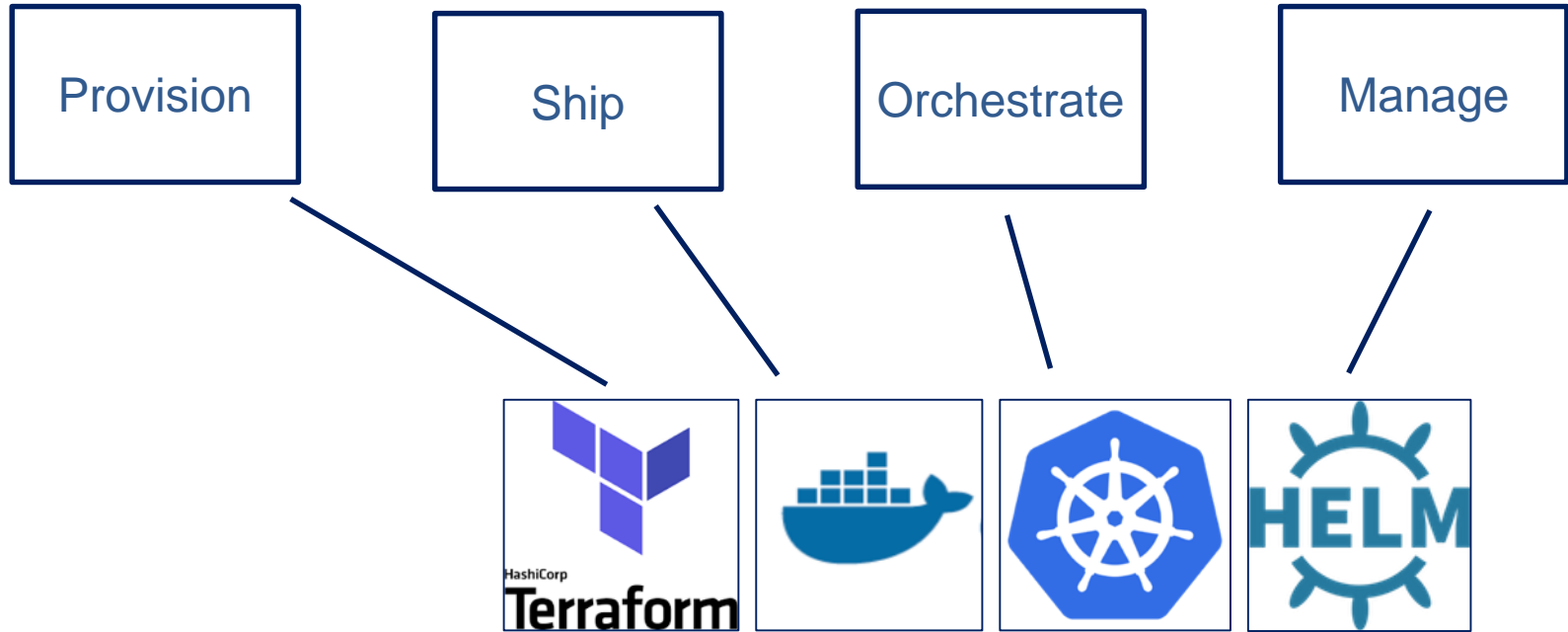
System Z



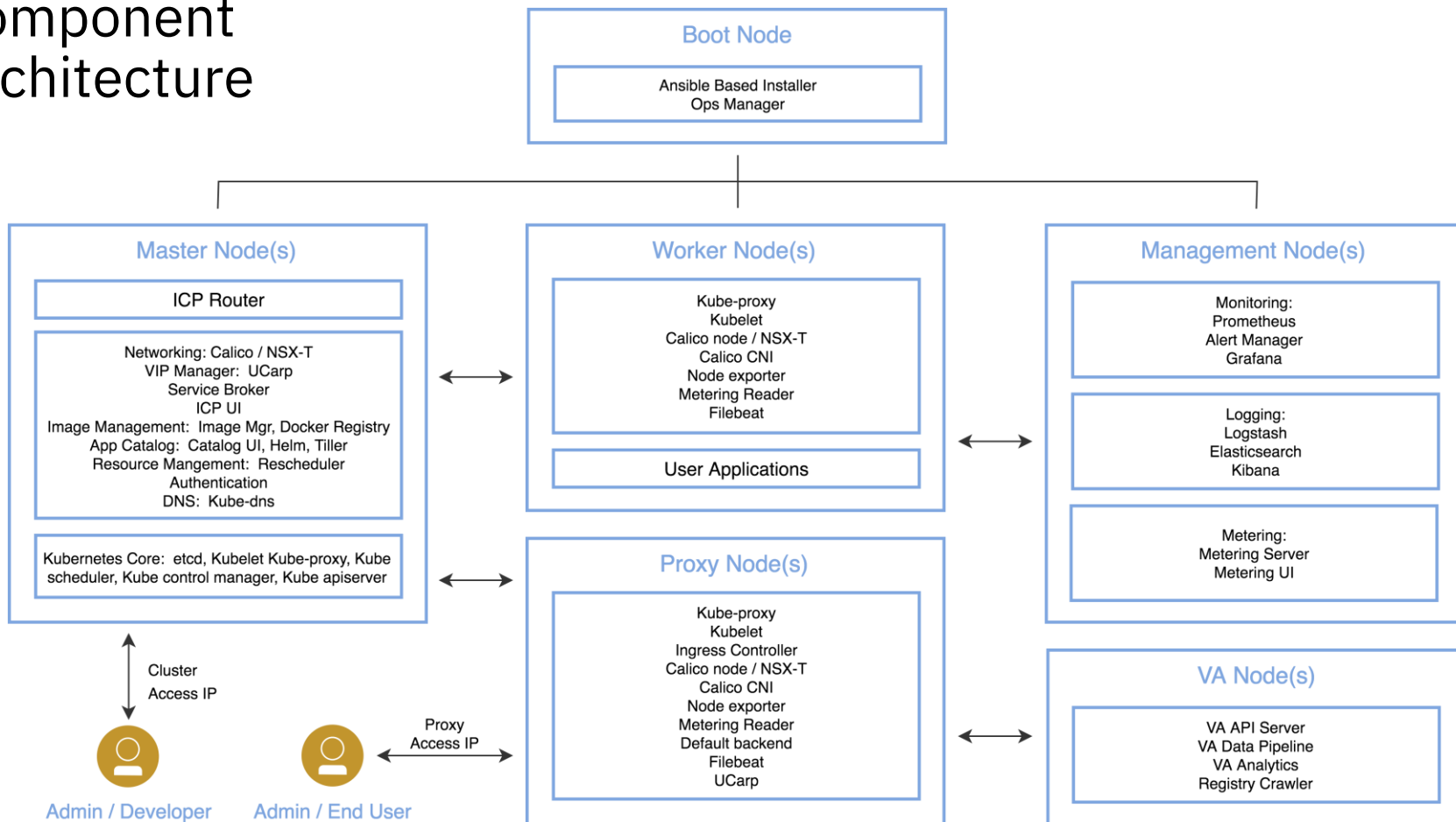
IBM Spectrum

Dell, Cisco, NetApp, Lenovo, ...

Technologies used with IBM Clouds



Component Architecture

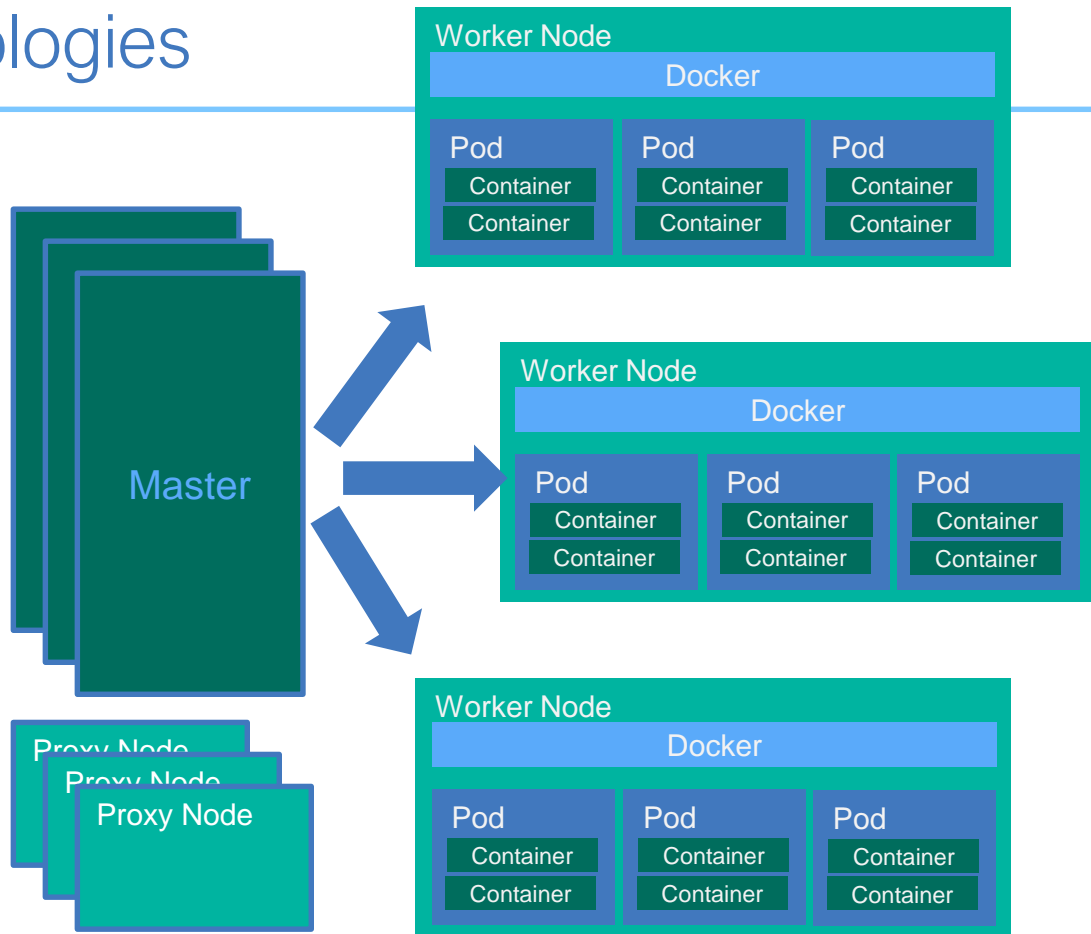


IBM Cloud Private – Types of Nodes

- **Boot node** : A boot or bootstrap node is used for running installation, configuration, node scaling, and cluster updates. Only one boot node is required for any cluster. You can use a single node for both master and boot.
- **Master node** : A master node provides management services and controls the worker nodes in a cluster. Master nodes host processes that are responsible for resource allocation, state maintenance, scheduling, and monitoring. Because a high availability (HA) environment contains multiple master nodes, if the leading master node fails, failover logic automatically promotes a different node to the master role. Hosts that can act as the master are called master candidates.
- **Worker node** : A worker node is a node that provides a containerized environment for running tasks. As demands increase, more worker nodes can easily be added to your cluster to improve performance and efficiency. A cluster can contain any number of worker nodes, but a minimum of one worker node is required.
- **Proxy node** : A proxy node is a node that transmits external request to the services created inside your cluster. Because a high availability (HA) environment contains multiple proxy nodes, if the leading proxy node fails, failover logic automatically promotes a different node to the proxy role. While you can use a single node as both master and proxy, it is best to use dedicated proxy nodes to reduce the load on the master node. A cluster must contain at least one proxy node if load balancing is required inside the cluster.
- **Management node** : A management node is an optional node that only hosts management services such as monitoring, metering, and logging. By configuring dedicated management nodes, you can prevent the master node from becoming overloaded. You can enable the management node only during IBM Cloud Private installation.
- **Vulnerability Advisor node**: node is an optional node that is used for running the Vulnerability Advisor services. Vulnerability Advisor services are resource intensive. If you use the Vulnerability Advisor service, specify a dedicated VA node.

IBM Cloud Private Topologies

- **Simple**
 - Single machine install (master is a worker)
 - Great for testing and learning about the platform
- **Standard**
 - Single master (single master, 3 workers, 1 proxy)
 - Great for non-production testing environment
- **High Availability**
 - Multiple masters (3 masters, 3+ workers, 3 proxy)
 - Production installation



IBM Cloud Private ~ 50 Components

Components

Table 1. IBM Cloud Private node components

Component	Version	Location	Role
Alert manager	0.13.0	Single management node	Handles alerts sent by the Prometheus
Ansible based installer and ops manager	2.5.0	Boot node	Deploys IBM Cloud Private on master and worker nodes
Authentication manager	2.1.0.3	Each master node	Provides an HTTP API for managing users and roles
calico/node	3.0.4	All nodes, except the boot node.	Sets the Calico network configurations
calicoctl	2.0.2	Each master node	A client tool that runs as a Kubernetes job
calico/cni	2.0.3	All nodes, except the boot node.	Sets the network CNI plug-ins on each node
calico/kube-policy-controller	2.0.2	Each master node	A controller center that sets the network policy
Docker Registry	2	Each master node	Private image registry that is used to store images
Default backend	1.2	Single master node	Minor component of the ingress controller
Elasticsearch	5.5.1	Single management node	Stores the system and application logs
etcd	3.2.14	Each master node	Distributed key-value store that maintains the cluster state
Filebeat	5.5.1	All nodes, except the boot node.	Collects the logs for all system components
Federation components	<ul style="list-style-type: none"> coredns (1.0.3) Kubefed (1.8.3) opa (0.5.13) opa_kube_mgmt (0.4) 	Single management node	Facilitates cluster discovery and management
GlusterFS	3.12.1	Selected worker nodes	A storage file system.
Grafana	4.6.3	Single management node	Data visualization & Monitoring with support for Prometheus
Heapster	1.4.0	Single master node	Connects to the kubelet that is running on each node
Heketi	5.0.0	Runs as a pod on any worker node.	CLI to manage GlusterFS.
Helm (Tiller)	2.7.2	Single master node	Manages Kubernetes charts (packages).
IBM Cloud Private management console	2.1.0.3	Each master node	A web portal that is based on the OpenShift console
Image manager	2.1.0.3	Each master node	Manages images by providing extended cataloging of image libraries.
Indices-cleaner	0.2	Single management node	Cleans up Elasticsearch data.
Kibana	5.5.1	Single management node	A UI providing easy access to data stored in Elasticsearch
Kubelet	1.10.0	All nodes, except the boot node.	Supervises the system components of the node
<ul style="list-style-type: none"> Kube-dns kubedns_dnsmasq kubedns_sidecar 	1.14.4	All master nodes	Manages DNS for the cluster

Kubernetes apiserver	1.10.0	Each master node	Provides a REST API for validating and configuring objects
Kubernetes control manager	1.10.0	Each master node	Maintains the shared state of the Kubernetes cluster
Kubernetes pause	3.0	All nodes, except the boot node.	Stores the IP address for pods, and sets up the network
Kubernetes proxy	1.10.0	All nodes, except the boot node.	Takes traffic that is directed at Kubernetes services
Kubernetes scheduler	1.10.0	Each master node	Assigns pods to worker nodes based on scheduling policies
kube_state_metrics	1.2.0	Single management node	Communicates with the Kubernetes API server to collect metrics
Logstash	5.5.1	Single management node	Transforms and forwards the logs that are collected
mariaDB	10.1.16	Each master node	Database that is used by OIDC.
Metering components	2.1.0.3	<ul style="list-style-type: none"> Metering server (Single management node) Metering reader (All nodes, except the boot node.) 	Collects usage metrics for your applications and services
MongoDB	3.6	Each master node	Database that is used by metering service (IBM®)
OpenID Connect (OIDC)	1.0	Each master node	Identity protocol over OAuth 2.0. Websphere Liberty
Prometheus components	<ul style="list-style-type: none"> Prometheus (2.0.0) collectd_exporter (0.3.1) node_exporter (0.15.2) configmap_reload (0.1) elasticsearch-exporter(1.0.2) kube-state-metrics-exporter(1.2.0) 	Single management node	Collects metrics from configured targets at given intervals
Rescheduler	0.5.2	Each master node	Used for pod management in a cluster. A rescheduler, see https://github.com/kubernetes/kubernetes/blob/master/pkg/scheduler/rescheduler.go
Router	2.1.0.3	Each master node	Hosts the management console and acts as the management console
Service Catalog	0.1.2	Each master node	Implements the Open Service Broker API to provide service discovery
UCarp	1.5.2	Each master and proxy node	Used to manage virtual IP (VIP) on the master node
Unified router	2.1.0.3	Single master node	Used to support backend functioning of the IBM Cloud Private

Vulnerability Advisor (VA) components (optional)

Table 2. Vulnerability Advisor node components

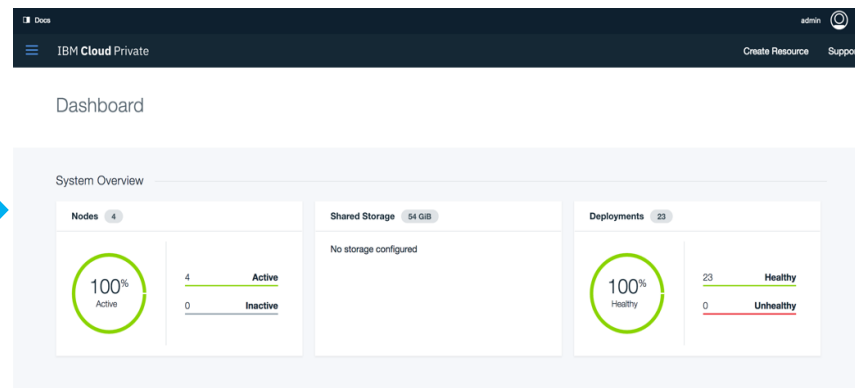
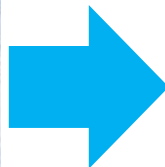
Component	Version	Location	Role
Kafka	0.10.0.1	VA node	Data pipeline
Security Analytics Service (SAS) components	1.2.1	VA node	Vulnerability analysis, and remediation
<ul style="list-style-type: none"> SAS API server SAS Management server 			The crawler, The Vulnerability Advisor
Statsd	0.7.2	VA node	Used by the crawler
VA Elasticsearch	5.5.1	VA node	Data pipeline
VA Elasticsearch curator	5.4.1	VA node	Elasticsearch
VA Annotators	1.2.1	VA node	Vulnerability analysis, and remediation
<ul style="list-style-type: none"> VA Compliance annotator VA Config parser VA Password annotator VA Rootkit annotator VA Vulnerability annotator 			
VA Indexers	1.2.1	VA node	Data pipeline
<ul style="list-style-type: none"> VA Config indexer VA Generic indexer 			
<ul style="list-style-type: none"> VA Live scan proxy VA Notification dispatcher 	1.2.1	VA node	Data pipeline
VA Unscrawler	1.2.1	VA node	Data pipeline
VA Crawlers	1.2.1	VA node	Vulnerability analysis, and remediation
			Live and managed
			The registry
		VA node	Used by the crawler

https://www.ibm.com/support/knowledgecenter/en/SSBS6K_3.1.0/getting_started/components.html

Experience from clients drove IBM Cloud private design



Lessons Learnt:



Time to Value Matters: Clients want to Operate an Orchestrated Container Cloud, NOT Assemble One



+

IBM Value-add
for Enterprises



**IBM
Cloud
private**

Supported system configurations (ICP 3.2.0)

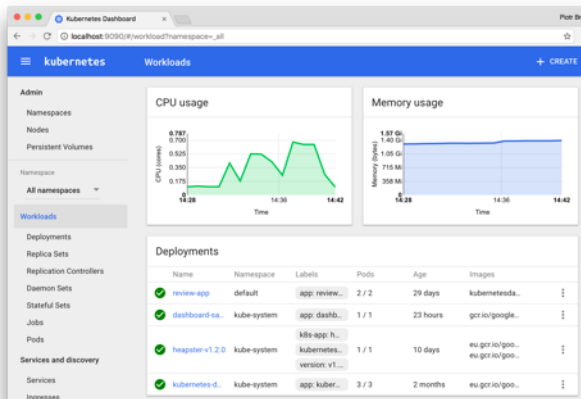
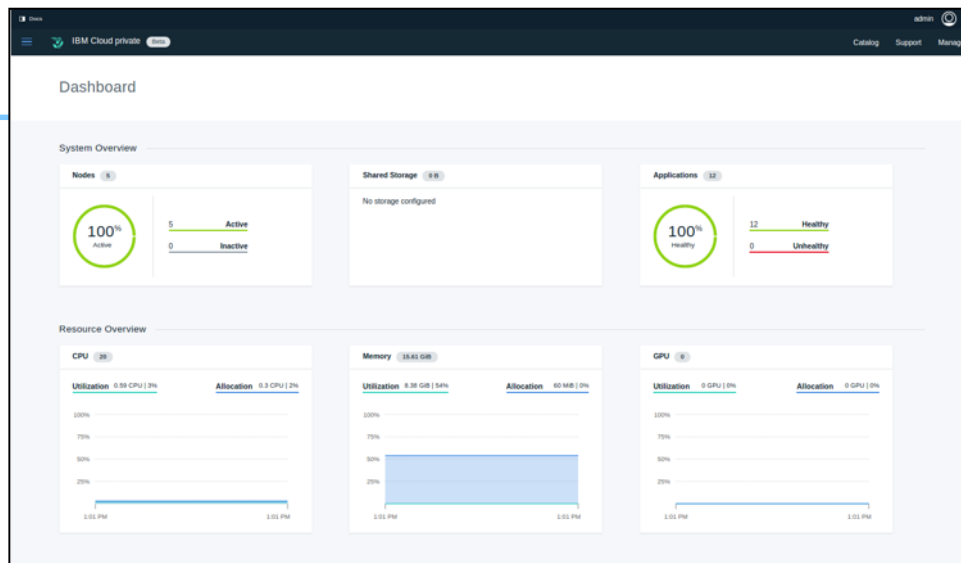
Specs		Support Statement	
OS	x86	RHEL 7.4, 7.5, 7.6, Ubuntu 18.04 LTS and 16.04 LTS, SUSE 12 SP3	
	Power	RHEL 7.4, 7.5, 7.5-alt, 7.6, Ubuntu 18.04 LTS and 16.04 LTS, SUSE 12 SP3	
	IBM Z	RHEL 7.4, 7.5, 7.6, Ubuntu 18.04 LTS and 16.04 LTS, SUSE 12 SP3; VA node not supported on zOS.	
Browsers	Windows	Edge, Firefox and Chrome : latest version	
	Linux	Firefox and Chrome : latest version	
	MacOS	Safari, Firefox and Chrome : latest version	
Docker	x86	18.06.2	Or older versions
	Power	18.06.2	Or older versions
	Z (workers)	18.06.2	Or older versions
Storage		Persistence storage options: GlusterFS 4.1.5+ Heketi, vSphere vVol, Hostpath, NFS 4, GlusterFS 4.0.2, Minio (S3 Object Storage), Spectrum Scale + all Kubernetes supported storage types	
Networking		Calico 3.5.2 (default), NSX-T 2.4 (optional)	

Minimum Requirements (ICP 3.1.x)

Requirement	Boot node	Master node	Proxy node	Worker node	Management node	VA node	etcd node
Number of hosts	1	1, 3, or 5	1 or more	1 or more	1 or more	1, 3, or 5	1 or more odd number of nodes
Cores	1 or more	8 or more	2 or more	1 or more	4 or more	<ul style="list-style-type: none">• 4 or more• 8 or more (production environment)	1 or more
CPU	>= 2.4 GHz	>= 2.4 GHz	>= 2.4 GHz	>= 2.4 GHz	>= 2.4 GHz	>= 2.4 GHz	>= 2.4 GHz
RAM	>=4 GB	>=16 GB	>=4 GB	>=4 GB	>=16 GB	<ul style="list-style-type: none">• >=8 GB• >=16 GB (production environment)	>=4 GB
Disk space to install	>=100 GB	>=200 GB	>=150 GB	>=150 GB	>=150 GB	<ul style="list-style-type: none">• >=100 GB• >=400 GB (production environment)	>=100 GB

User Interfaces

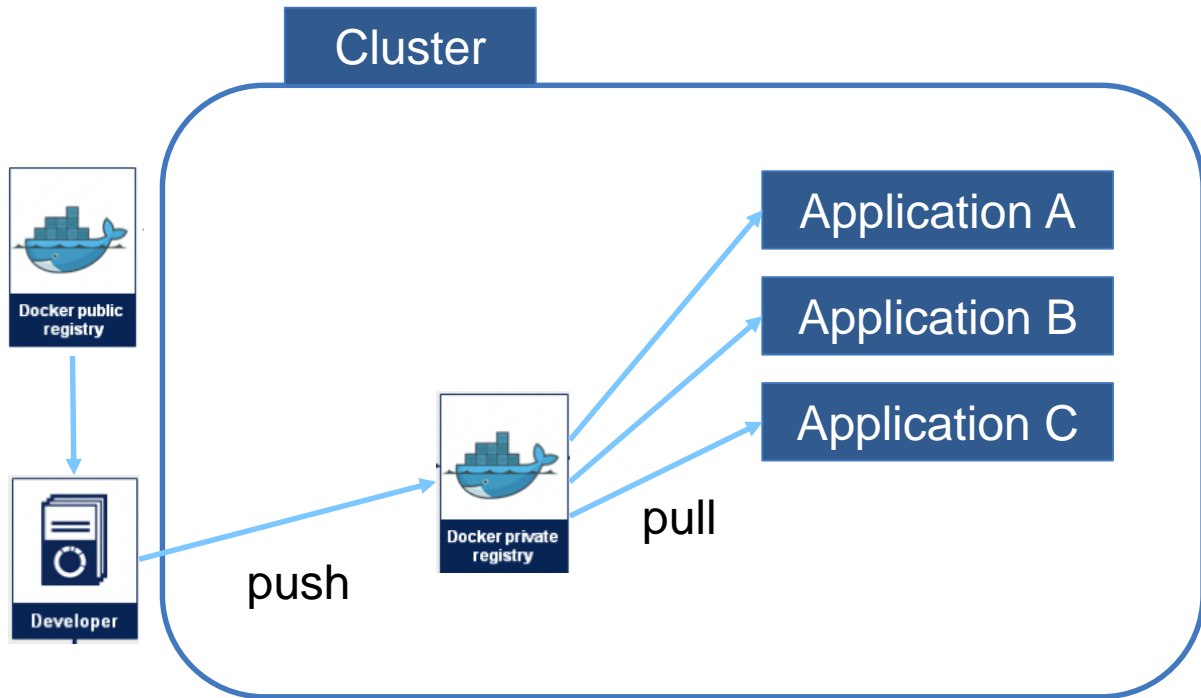
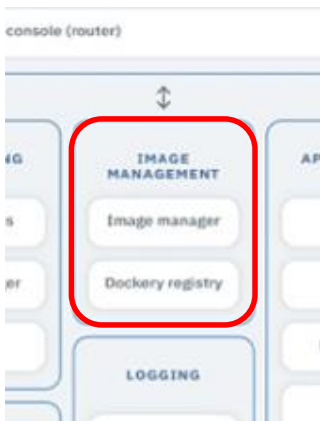
- **Cluster Management Console:** (ICP component) Use to manage, monitor, and troubleshoot your applications and cluster from a single, centralized, and secure management console.
- **K8S Web UI:** Can use to deploy containerized applications to a Kubernetes cluster, troubleshoot your containerized application, and manage the cluster itself along with its attendant resources.
- **Web CLI / cloudctl / kubectl:** A command-line interface for running commands against Kubernetes clusters.



```
root@nicetest1:~/helloworld# kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
abraca-deployment-7bf5d487d-btbf6   1/1     Running   0           2d
bluecompute-auth-6f77cf979-tnqlr     1/1     Running   0           1d
bluecompute-catalog-668749466-x4pvt  1/1     Running   0           1d
bluecompute-catalogdb-elasticsearch-79f9c4ddf-b-cwhjk  1/1     Running   0           1d
bluecompute-customer-58ffb8f955-vg9cn  1/1     Running   1           1d
bluecompute-customer-create-user-sl98e-4551b  0/1     Error     0           1d
bluecompute-customer-create-user-sl98e-91176  0/1     Error     0           1d
bluecompute-customer-create-user-sl98e-55ct6  0/1     Error     0           1d
bluecompute-customer-create-user-sl98e-6w89z  0/1     Error     0           1d
bluecompute-customer-create-user-sl98e-8x6hr  0/1     Error     0           1d
bluecompute-customer-create-user-sl98e-kazrz  0/2     Completed 0           1d
bluecompute-customer-create-user-sl98e-l65qv  0/1     Error     0           1d
bluecompute-customer-create-user-sl98e-vtjmq  0/1     Error     0           1d
bluecompute-customerdb-couchdb-695b569998-2zvqd  1/1     Running   0           1d
bluecompute-inventory-84846c50c-5fpg6  1/1     Running   0           1d
bluecompute-inventory-mysql-97680c6-r9b5f  1/1     Running   0           1d
bluecompute-inventory-populate-mysql-qvqv9-knt8z  0/1     Completed 0           1d
bluecompute-orders-7785cb4dc8-n6fs3  1/1     Running   0           1d
bluecompute-orders-mysql-5754894bb-hqmdn  1/1     Running   0           1d
bluecompute-web-5797ab0fe-6tqw6  1/1     Running   0           1d
node-app-bdcfffd8b-lhtts             1/1     Running   0           11s
root@nicetest1:~/helloworld#
root@nicetest1:~/helloworld#
root@nicetest1:~/helloworld#
root@nicetest1:~/helloworld# nano ingress.yaml
root@nicetest1:~/helloworld#
root@nicetest1:~/helloworld# kubectl apply -f ingress.yaml
Ingress.extensions "test-ingress" created
```

Images and Registries

- You create a Docker image and push it to a registry before referring to it in a Kubernetes pod
- There will likely be many registries used in your deployment



Vulnerability Advisor

▼ Tools

Vulnerability Advis...

Mutation Advisor

Vulnerability Advisor for IBM Cloud Private Vulnerability Advisor (List Containers) Vulnerability Advisor (List Images) Manage Policies Go to Mutation Advisor

Vulnerability Advisor (List Containers)

The Vulnerability Advisor has scanned all of your containers looking for known security vulnerabilities. Click on a row to see the details for that container.

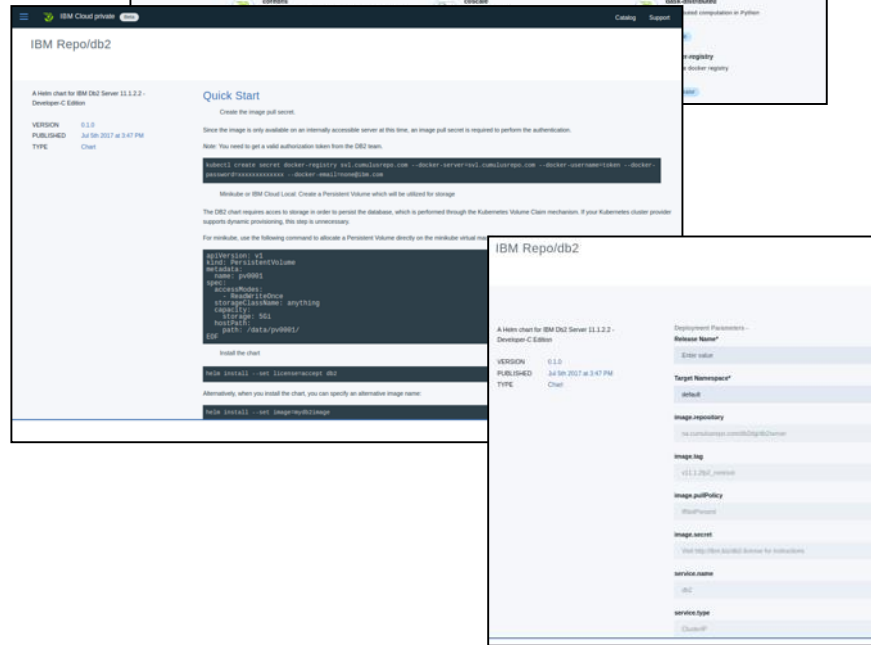
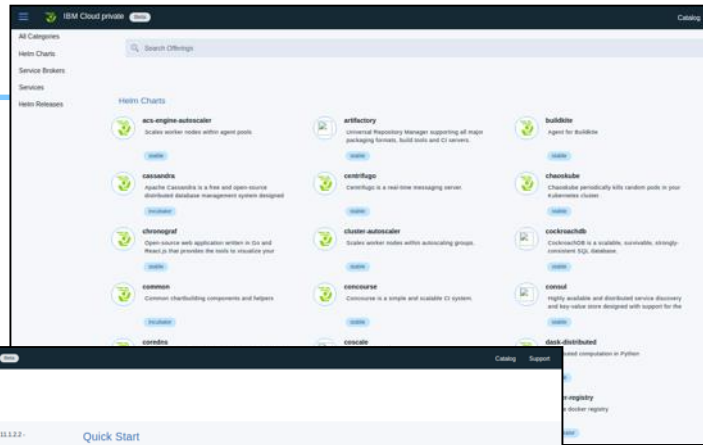
First Previous 1 Next Last

Filter

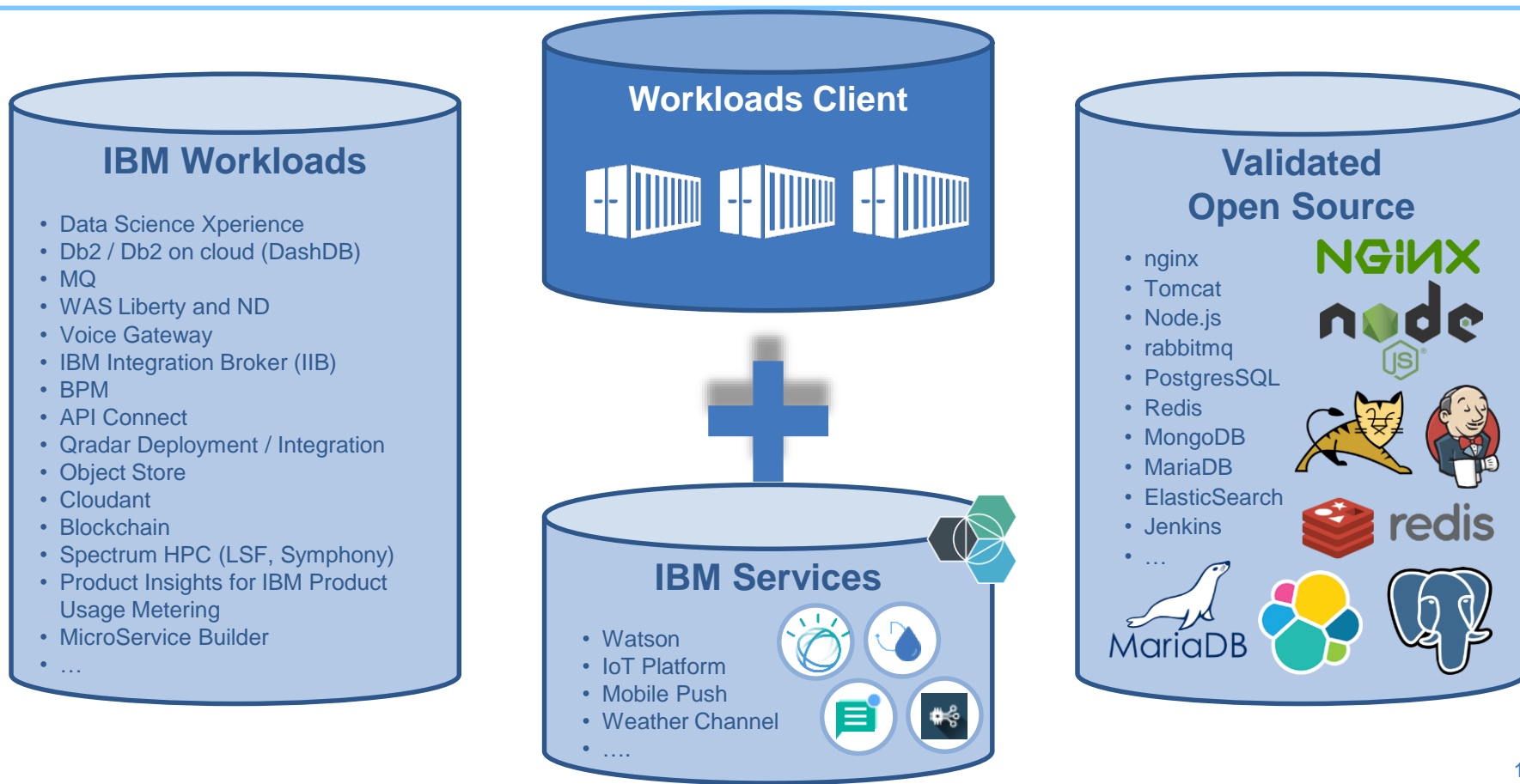
Name	Owner	Latest Scan	Type	Organizational Policies	Vulnerable Packages	Container Settings
default/openldap-97bd8f564-dfck4/POD/f07a64f2c5b814638dc490c7037070db309bb7c68c48a9588ba3efe01b89a2fe	default	2019/03/25 11:12:20	Container	Passed	OS Unsupported	4 / 27
default/openldap-admin-7f6485b648-bx22/POD/225a54fa15d6556e99377d6da6b176d99267aaf68e0e9c335f3fe06f6dffa5e	default	2019/03/25 11:12:17	Container	Passed	OS Unsupported	4 / 27
default/openldap-97bd8f564-dfck4/openldap/91be0ec2e83c8a4d73eece127d793203ef6940c1820780a033112f69be0a2264	default	2019/03/25 11:12:16	Container	Passed	7 / 156	3 / 27

Application Center components

- Application center or **Catalog** provides a centralized location from which you can browse, and install packages in your cluster.
- **Helm**: A tool for managing Kubernetes charts. Charts are packages of pre-configured Kubernetes resources.
- **Helm Repository**: A Helm chart repository is a location where packaged charts can be stored and shared.
- **Tiller**: Runs inside of the cluster, and manages releases (installations) of your charts.



ICP: Multi-sources catalog

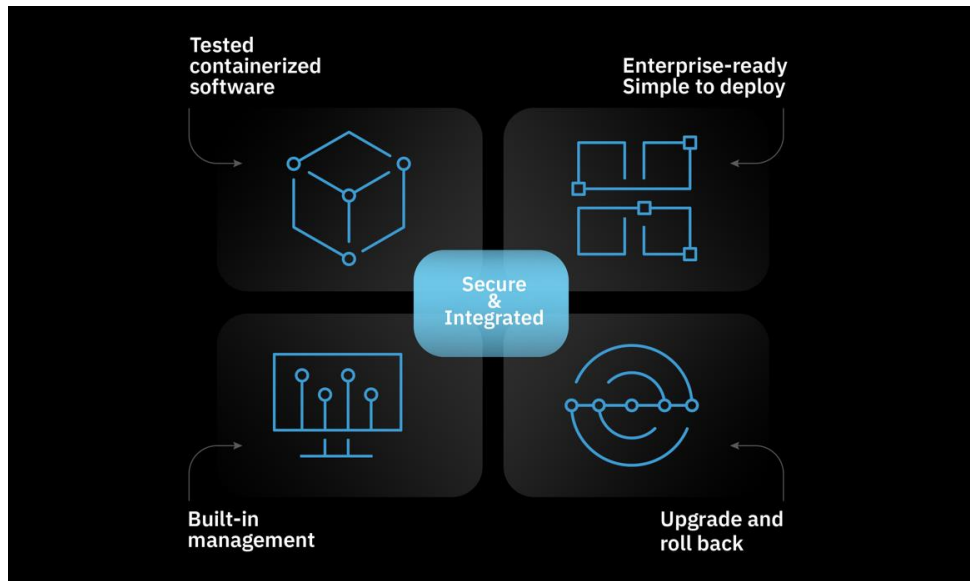


Containers are not enough for the Enterprise

Providing containers is not enough for the Enterprise.

IBM Cloud Paks on ICP:




- ✓ Provide enterprise capabilities for deployment, lifecycle management, and production use cases
- ✓ Unlock the value of IBM Cloud Private, out-of-the-box integration with core operational services
- ✓ Accelerate Time to Production for Enterprise client use cases



Certified IBM Cloud Paks are the next step in our container content strategy

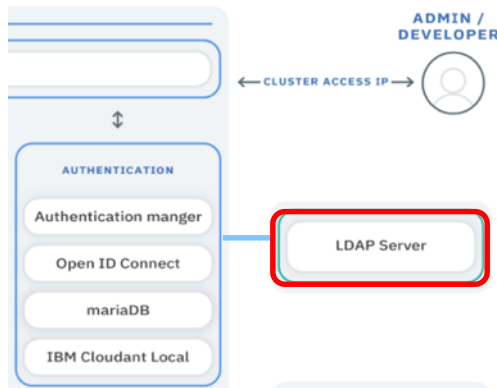
IBM Cloud Paks on IBM Cloud Private

New IBM Cloud Paks are enterprise ready out of the box

Container delivery models	 Ad hoc containers			 IBM provided containers			 Certified IBM Cloud Paks on IBM Cloud Private		
	Client takes software binaries, Creates their own containers			Client receives IBM Software in the form of container(s)			Easy, Enterprise grade, Fully supported		
Capabilities & Value									
IBM Software supported	Depends on product			Yes			Yes		
Full stack support by IBM (Base OS, software, deployment on cloud platform)	No			No			Yes		
Vulnerability Scanned (Manages image vulnerabilities)	Scan yourself			Yes			Yes		
Orchestrated for Production (Built for Kubernetes by product experts)	None			None			Yes		
Management and Operations	Roll your own			Roll your own			Built-in		
License Metering Integration	Do it yourself			Do it yourself			Yes		
Lifecycle Management	Manage it yourself			Manage it yourself			Tested upgrade & rollback		

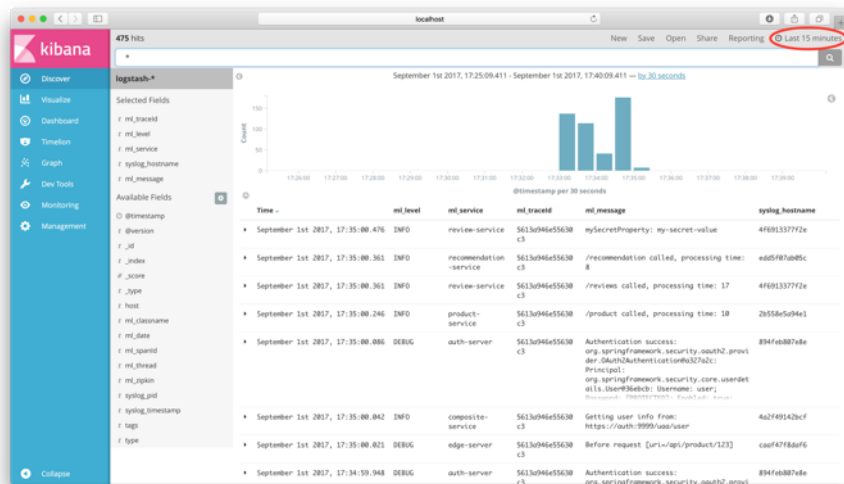
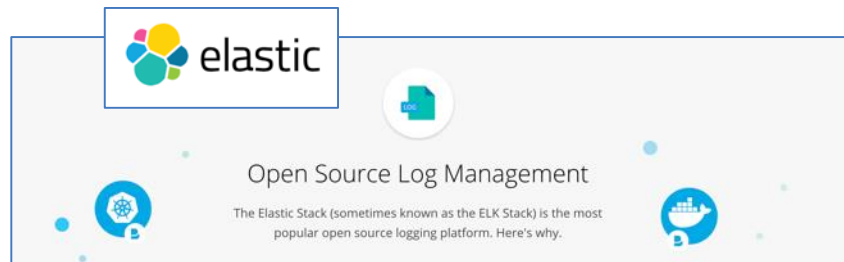
Authentication components

- **Authentication Manager (IAM):** Provides an HTTP API for managing users. Protocols are implemented in a RESTful manner. Keystone is used for authentication. Pass-through is used for external LDAP integration.
- **Keystone:** The OpenStack provided identity service currently supporting token-based authN and user-service authorization.
- **MariaDB:** An open source relational database made by the original developers of MySQL. In this case it is used to back-end Keystone.
- **OIDC:** **OpenID Connect** is an authentication layer on top of [OAuth 2.0](#), an authorization framework
- **Mongo DB:** (instead of Cloudant) Database that is used by metering service.
- **RBAC – Role Based Access Control**



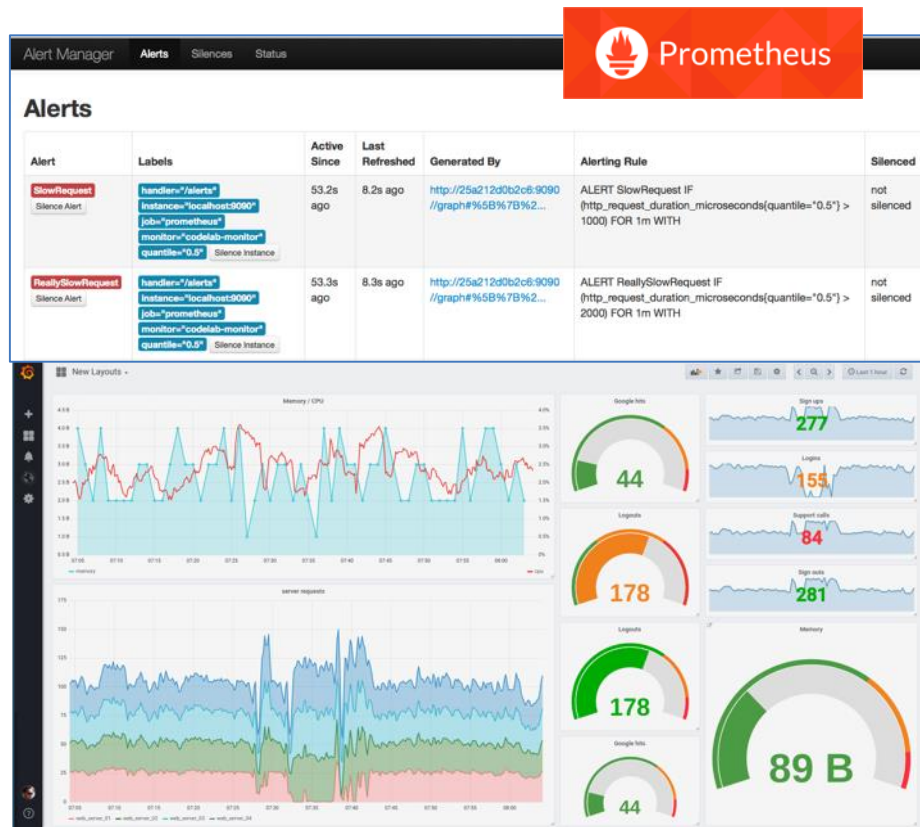
Logging Components : Elastic Stack

- The easiest and most embraced **logging** method for containerized applications is to write to standard out and standard error
- **Elastic Stack** (also known as ELK stack)
- **Filebeat**: A log data shipper for local files. Filebeat monitors the log directories or specific log files, tails the files, and forwards them either to Elasticsearch and/or Logstash for indexing.
- **Elasticsearch**: An open source full-text search engine based on Lucene. It provides HTTP web interface and schema-free JSON documents.
- **Logstash**: A open source tool for collecting, parsing, and storing logs for future use.
- **Heapster**: The Kubernetes network proxy runs on each node.
- **Kibana**: An open source data visualization plugin for Elasticsearch. Users can create bar, line and scatter plots, or pie charts and maps on top of large volumes of data.



Monitoring Components: Prometheus and Grafana

- **Prometheus:** An open-source systems monitoring and alerting toolkit originally built at SoundCloud. Since its inception in 2012, many companies and organizations have adopted **Prometheus**, and the project has a very active developer and user community. It is now a standalone open source project and maintained independently of any company.
- **Grafana:** An open-source, general purpose dashboard and graph composer, which runs as a web application.



Persistent storage components

- Traditionally Containers: **stateless, ephemeral** in nature
 - Storage exists within the container
 - The container goes away and so goes the storage
- Some applications desire state and thus **persistent storage**:
 - Specific aspects of configuration
 - Database (structured and unstructured)
 - Application data (website definitions, etc.)
- Storage must be **universally accessible** across the K8s environment
- ICP Persistent Storage Support: HostPath, **NFS, GlusterFS, vSphereVolume**
- Access Modes:
 - ReadWriteOnce – the volume can be mounted as read-write by a single node
 - ReadOnlyMany – the volume can be mounted read-only by many nodes
 - ReadWriteMany – the volume can be mounted as read-write by many nodes



What's new in version 3.1

- Kubernetes upgraded to 1.11.1
- Helm upgraded to 2.9.1
- Scalability improvements to **5,000 nodes**
- Full **Istio** 1.0 support
- New OS support: SLES and Ubuntu
- Operational dashboards now include storage and networking views

What's new in version 3.1

- **IBM Multi-Cloud Manager** provides true multi-cloud, multi-cluster management by working across clouds and providing integrated views, application-centric management, and governance across all of your organization's clusters and environments.
- Clusters can span **geographic locations, technology or business purpose** (such as development, test, production), and **cloud type**. With the Multi-cloud Manager toolset, your organization can gain visibility, improve governance, and integrate automation into its entire multi-cluster environment.

What's new in version 3.1.2

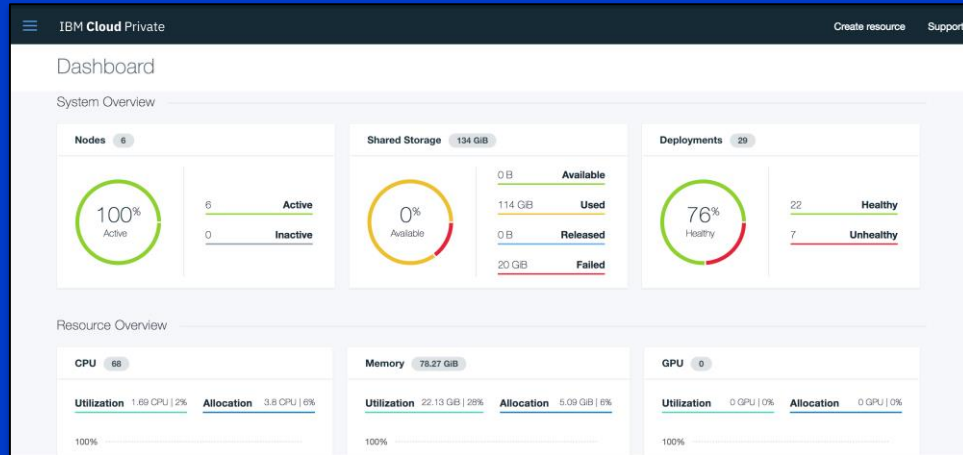
- **Kubernetes version 1.12.4, Helm version 2.9.1**
- **Multi-release Upgrade** - Upgrade in one step from releases with the same major. For example, move from 3.1.0 to 3.1.2 directly. **Note** does not apply to 2.1.x.x to 3.1.x
- **Zero Downtime** for kubernetes applications during ICP upgrade
- **Microsoft Azure** as a cloud provider for ICP deployment
- **Manage from Z** [Linux® on IBM® Z and LinuxONE (s390x)] cluster install HA config
- **MCM public cloud** support for IKS, EKS, AKS, GKE clusters
- **MCM new dashboard** and query interface to quickly find resources
- **MCM Community Edition** (with ICP CE) for testing puproses
- **Cloud Foundry deployment moves to 6.2** and gains support for NFS Volume Service
- **Cloud Foundry zero downtime** app updates via ``cf push``
- Tech Preview: **Windows worker node** support

What's new in ICP version 3.2.0

- Kubernetes version 1.13.5, Helm version 2.12.3
- Enabling IBM Multicloud Manager
- Key Management Service Hardware Security Module package
- Modularized installation
- Audit and certificate manager
- VMware NSX-T is upgraded to version 2.4.
- RBAC for Grafana (multi-tenancy)
- Performance Improvements
- ICP UI console change
- Multicloud manager installation and dashboard
- Technology preview : System health service, Mutation policy controller, Knative,

Try IBM Cloud Private

Free Community Edition



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



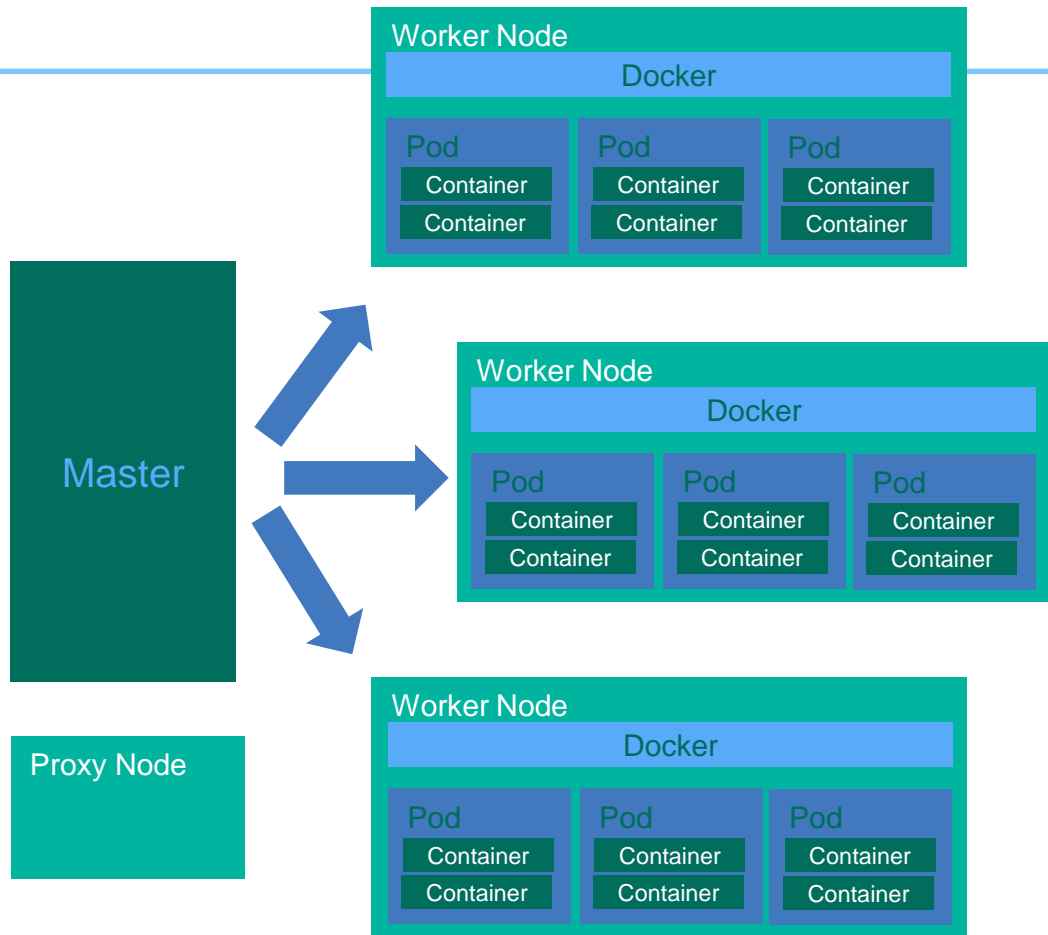
Demonstrations

IBM Cloud Private Console



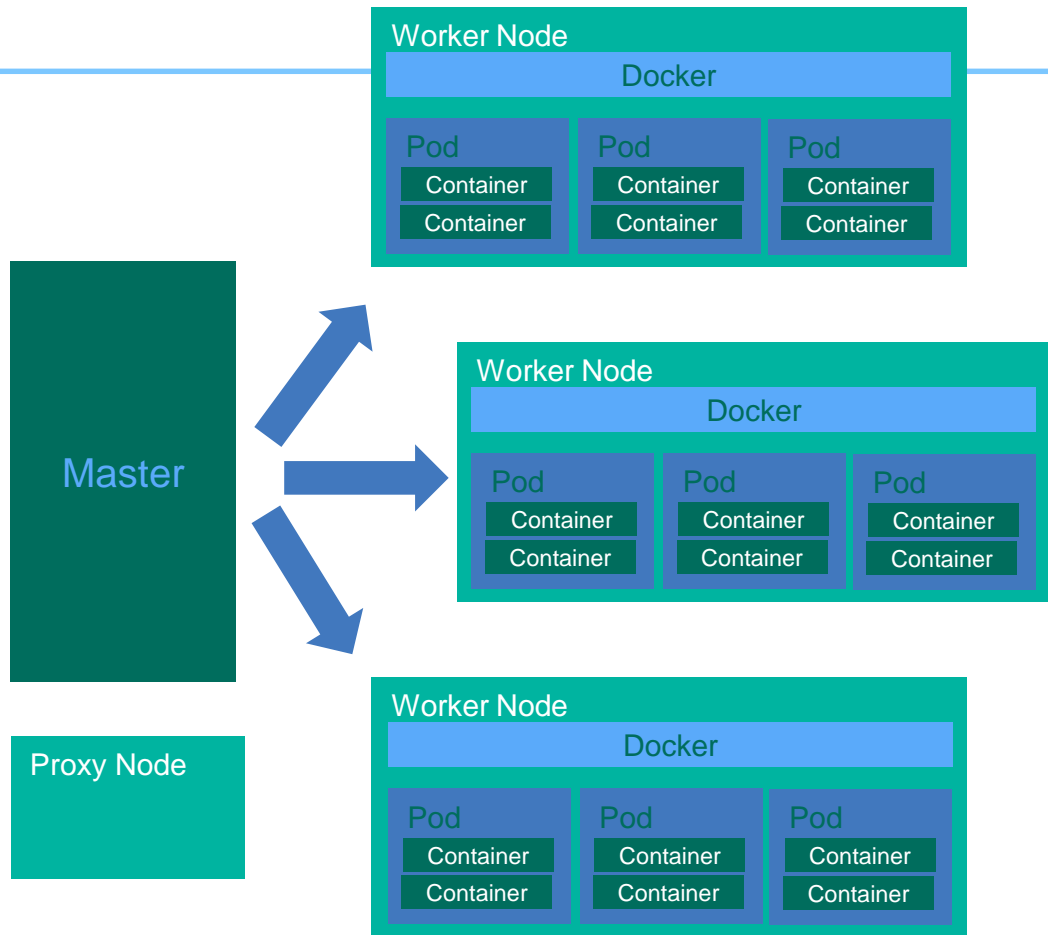
ICP Topologies

- Simple
 - Single machine install (master is a worker)
 - Great for testing and learning about the platform
- Standard
 - Single master (single master, 3 workers, 1 proxy)
 - Great for non-production testing environment
- High Availability
 - Multiple masters (3 masters, 3+ workers, 3 proxy)
 - Production installation



ICP Topologies

- Simple
 - Single machine install (master is a worker)
 - Great for testing and learning about the platform
- Standard
 - Single master (single master, 3 workers, 1 proxy)
 - Great for non-production testing environment
- High Availability
 - Multiple masters (3 masters, 3+ workers, 3 proxy)
 - Production installation



FYI - Installing ICP yourself


- **Purpose** : installation of a **single node** IBM Cloud Private **from scratch**
- Prerequisites checks for installing **ICP-ce** (community edition)
- Possible installations
 - on big laptops,
 - on VMs in IBM locations
 - on IBM Cloud Infra using VSI (Virtual Server Instance)
 - on others VMs with other cloud providers
- possibilities :
 - **Ubuntu 16.04.04** (with or without Vmware)
 - Or Vagrant (prereq VirtualBox)

Prerequisites if you want to run your own ICP

- Use VMWare or Virtual Box
- Test your connection with a terminal console (SSH or Putty)
- Some knowledge of Linux & Network commands is required

- **Hardware and OS prerequisites at minima**

- [] one host (physical or virtual)
- [] CPU = 8 cores
- [] RAM = 16 GB or more
- [] Storage = 40 GB or more for some deployments
- [] Ubuntu 16.04.04 - 64 bits + packages



```
[*] Manual package selection
[ ] DNS server
[ ] LAMP server
[ ] Mail server
[ ] PostgreSQL database
[ ] Samba file server
[*] standard system utilities
[*] Virtual Machine host
[*] OpenSSH server
```

<Continue>

Verifications done during the installation

Installation checks

- Master
 - `/var` \geq 240 GB
 - Cores \geq 2
 - Mem \geq 4
- Management
 - `/var` \geq 220 GB
 - Cores \geq 4
 - Mem \geq 8
- Worker
 - `/var` \geq 110 GB
 - Cores \geq 2
 - Mem \geq 4
- Proxy
 - `/var` \geq 110 GB
 - Cores \geq 2
 - Mem \geq 4

`root` - recommended size of 10 GiB

This is where `/`, or the root directory, is located. The root directory is the top-level of the directory structure. By default, all files are written to this file system unless a different file system is mounted in the path being written to (for example, `/boot` or `/home`).

While a 5 GiB root file system allows you to install a minimal installation, it is recommended to allocate at least 10 GiB so that you can install as many package groups as you want.



Important

Do not confuse the `/` directory with the `/root` directory. The `/root` directory is the home directory of the root user. The `/root` directory is sometimes referred to as *slash root* to distinguish it from the root directory.

`/home` - recommended size at least 1 GiB

To store user data separately from system data, create a dedicated file system for the `/home` directory. This file system should be sized based on the amount of data that will be stored locally, number of users, and so on. This will enable you to upgrade or reinstall Red Hat Enterprise Linux without erasing user data files. If you select automatic partitioning, it is recommended to have at least 55GiB of disk space available for the installation, to ensure that the `/home` file system is created.

`swap` partition - recommended size at least 1 GB

Installation Overview

- Container-based installer:
 - Download the installation container for CE or EE and execute the install
 - The installer pulls down additional containers from Docker Hub for CE, local repo for EE
- Supported for RHEL and Ubuntu on X, POWER and Z (workers)
- Basic installation steps:
 - 1. Configure OS
 - 2. Modify installation configuration files and run the installer
- Overall installation should take < 4 hours depending on scenario
 - (90% System Config, 10% Installation)