IBM Cloud Private Architecture

IBM Cloud

IBM Cloud Private Major Components

Topologies
Nodes
Services

IBM Cloud Private Architecture

Components
Interfaces
Services

IBM Cloud Private Architecture

Major Components

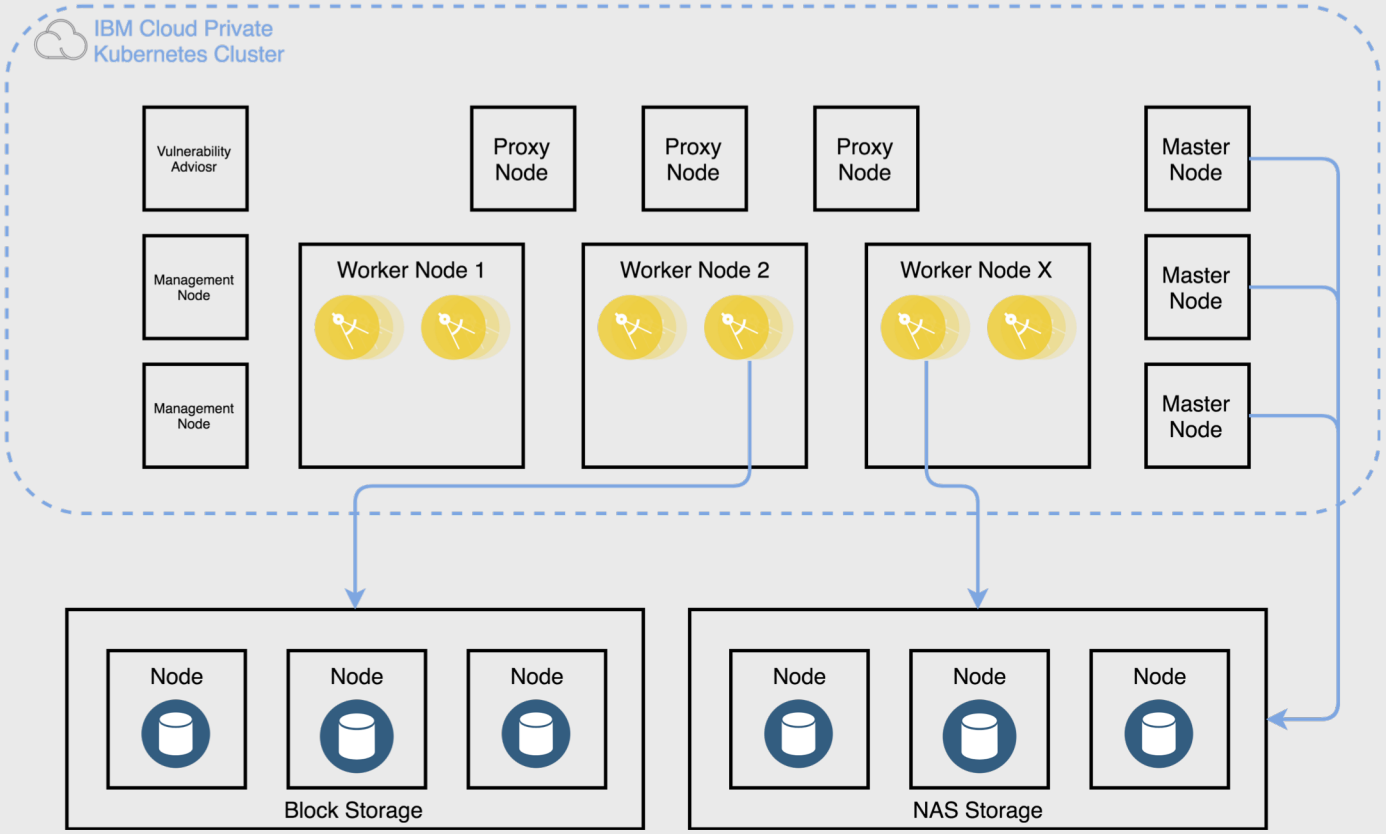
Topologies

The ICP Kubernetes cluster consists of both mandatory and optional components. Consider the following typical topologies:

Simple
- Single machine install with the master node performing worker node duties
- Works well for learning the platform and testing basic workloads

Standard
- 1 master, 1 proxy, 1 management, 3+ workers and optionally a Vulnerability Advisor node
- Great for non-production workloads and testing

High Availability (Enterprise)
- 3, 5 or 7 masters, 10+ worker nodes, 3 proxies, 2+ management nodes, 1 or 3 Vulnerability Advisors
- Ideal for production and enterprise deployments



Boot Node

A boot or bootstrap node is used for running IBM Cloud Private 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 and sharing this node is common

For convenience purposes and if you plan on performing multiple installations, it is advisable to create a standalone boot node

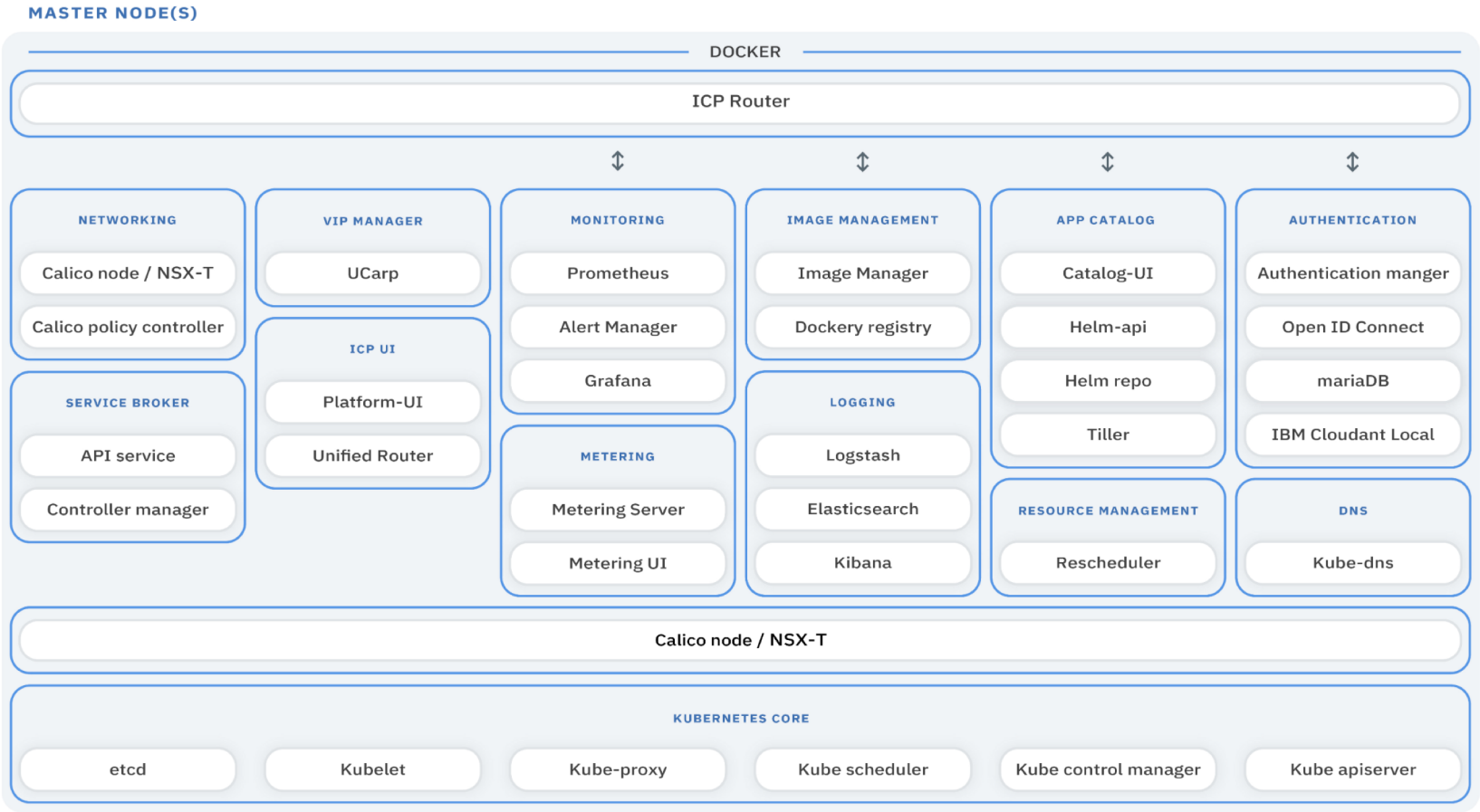
Master Node

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

Multiple master nodes are in a high availability (HA) environment to allow for failover if the leading master host fails

It is possible to load balance across multiple master nodes



etcd Node

An etcd node is an optional node that is used for running
the etcd distributed key value store that typically runs on the master nodes

Calico shares etcd with cluster management

Configuring an etcd node in an IBM Cloud Private cluster that has many nodes, such as 100 or more, helps to improve the etcd performance by separating workload from the master node(s)

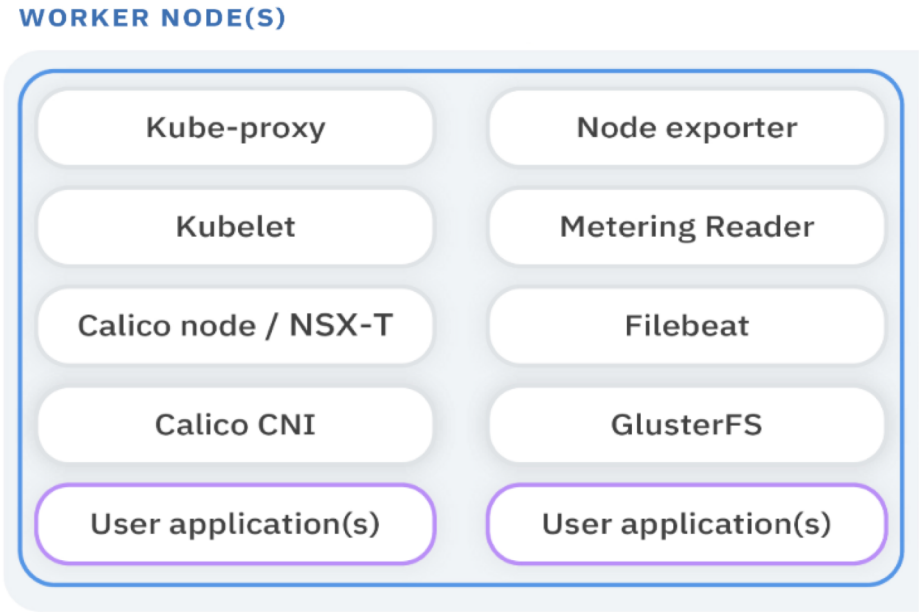
Worker Node

Provides a containerized environment for running workload and tasks

Worker nodes can be added at any time to accommodate additional workload

Worker nodes can be drained and removed to scale down the capacity of the cluster

A cluster can contain any number of worker nodes, but a minimum of one worker node is required



Proxy Node

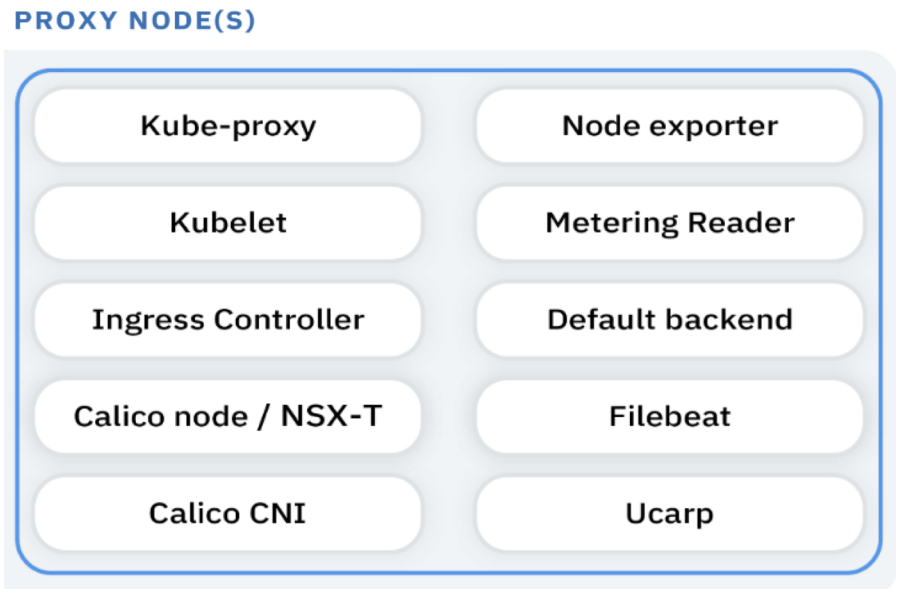
Transmits external request to the services created inside your cluster

Multiple proxy nodes are typically deployed for resiliency and scaling

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

Configure traffic to use the proxy via the Ingress object

For ICP the proxy nodes run NGINX and can be configured / tuned via ConfigMaps



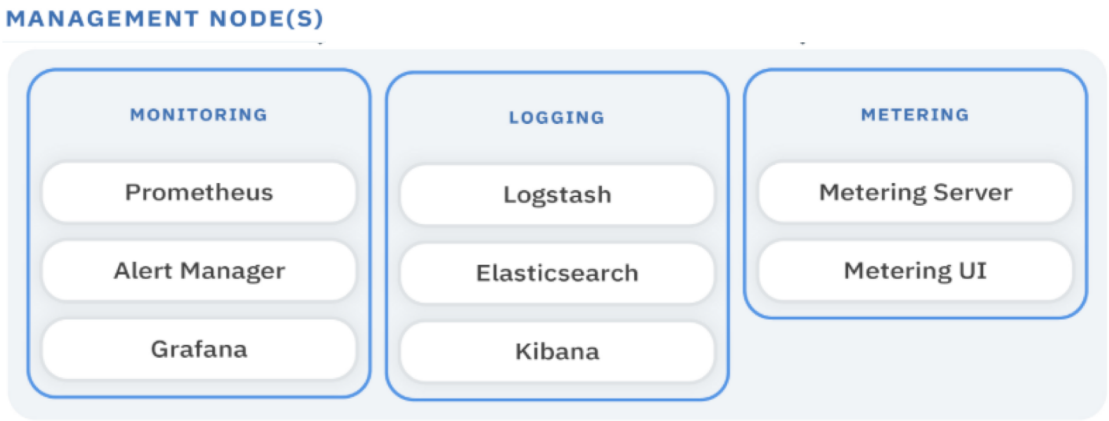
Management Node

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

Additional management nodes can be added post installation

Multiple management nodes do not natively provide for high availability



Vulnerability Advisor (VA) Node

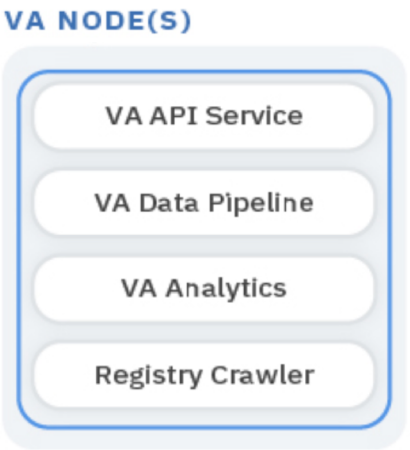
Optional node that is used for running the Vulnerability Advisor services

VA provides security management for Kubernetes environments

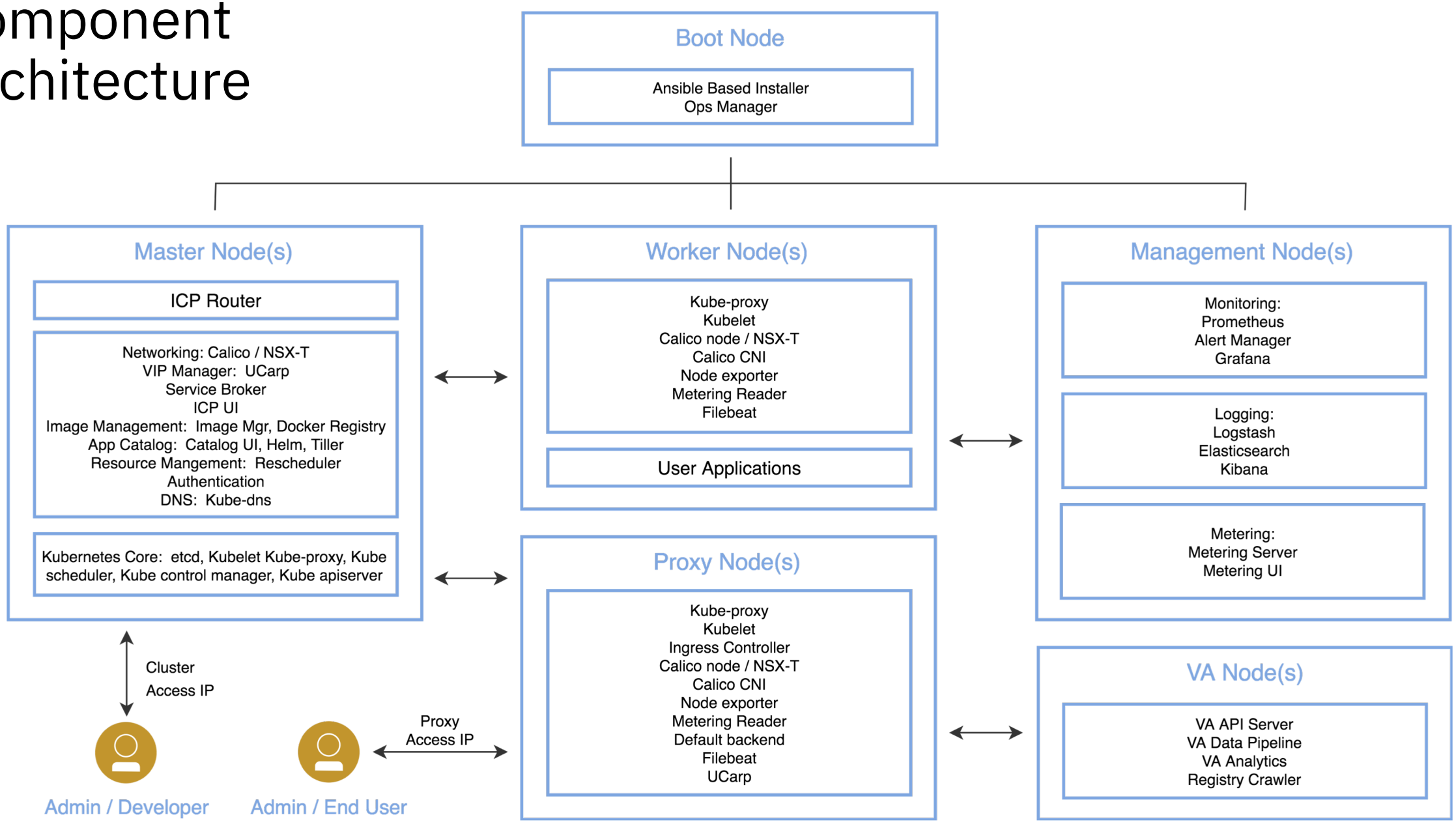
It generates a security status report suggesting fixes and best practices also providing management to restrict non-secure images from running

Can be added post cluster deployment

Can deploy multiple VA nodes to accommodate failover



Component Architecture



IBM Cloud Private Architecture

Components, Interfaces, Services

Kubernetes Core Components

These primary components provide the cluster's control plane and global decisions about the cluster and detecting and responding to cluster events

Etcd: A strong, consistent, and highly-available key value store which Kubernetes uses for persistent storage of all of its API objects and also shares with Calico

Kubelet: The primary "node agent" that runs as a service on each of the cluster nodes

K8s Proxy: The Kubernetes network proxy runs on each node implementing a VIP for each of the services running in K8s (except the externalName service)

K8s Scheduler: A policy-rich, topology-aware, workload-specific function that significantly impacts availability, performance, and capacity of the cluster taking into consideration resource requirements, hardware / software / policy constraints, affinity specifications, data locality, inter-workload interference, etc.

Kubernetes Core Components
(continued)

K8s Control Manager: A controller is a control loop that watches the shared state of the cluster through the API Server and makes changes attempting to move the current state towards the desired state with the Control Manager being a daemon that embeds the core control loops shipped with K8s

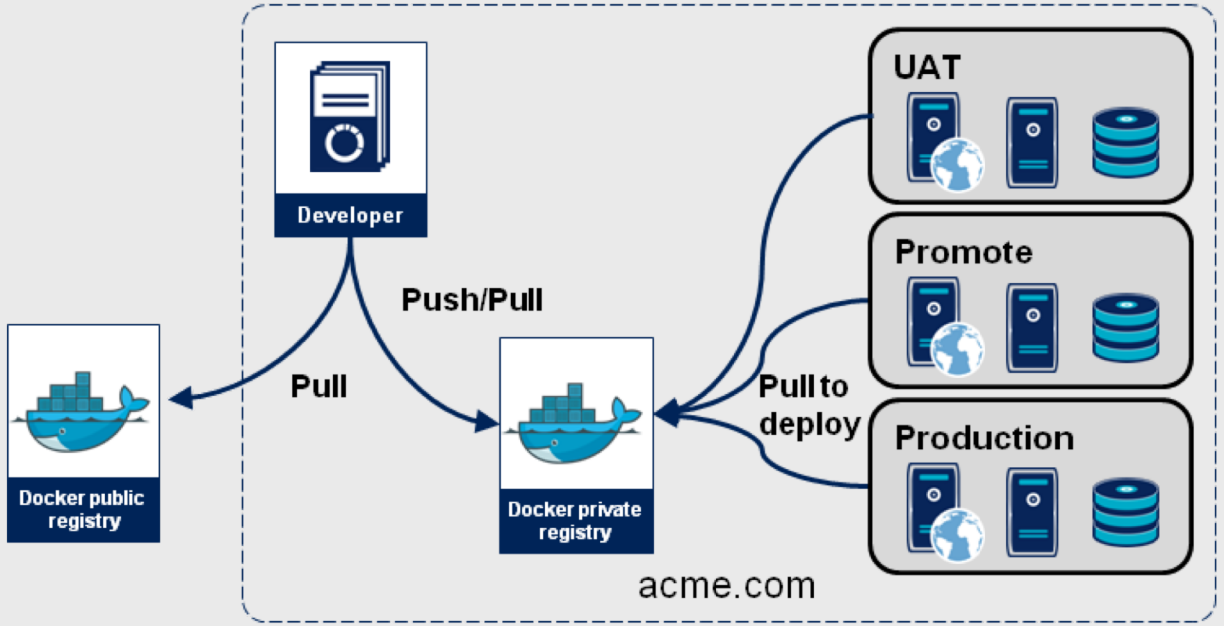
K8s API Server: Validates and configures data for the API objects which include pods, services, replication controllers, and others via REST operations providing the frontend to the cluster's shared state through which all other components interact

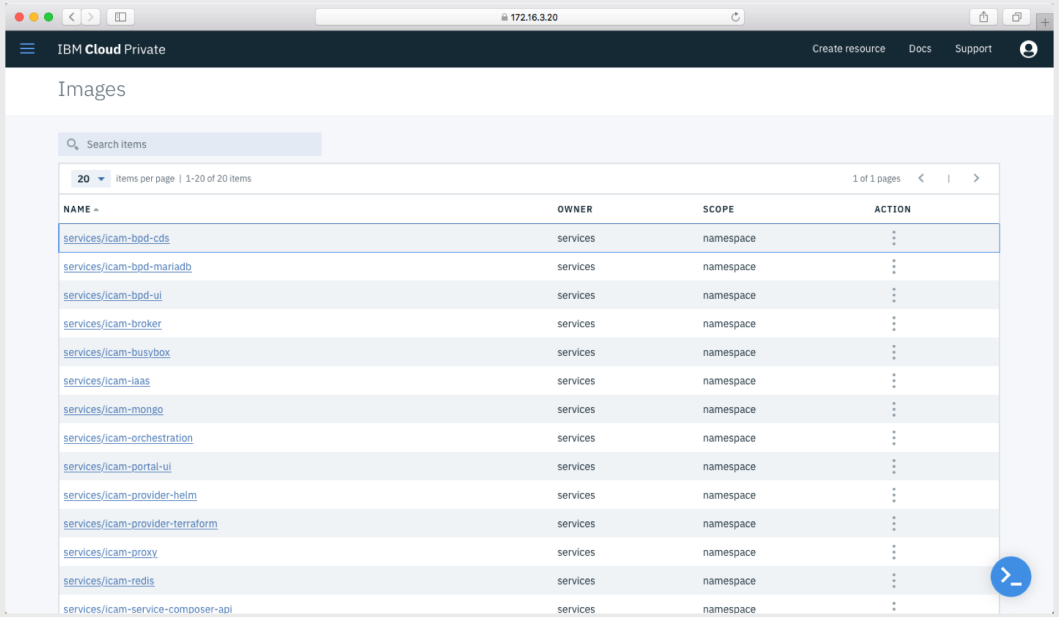
Image Management

Runs as a layer over the Docker registry V2 API

Provider of management functions and authorization for image repositories that the Docker registry stores

Similar to the Docker Hub but with restrictions

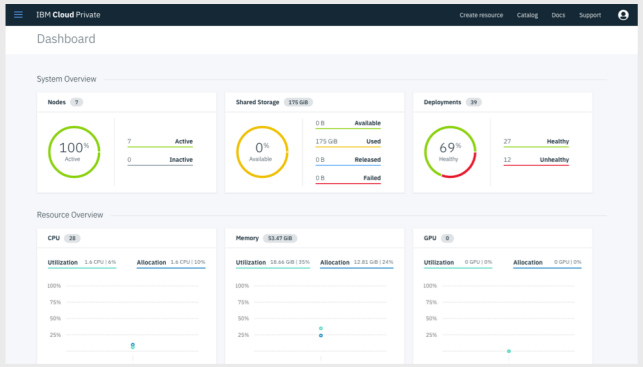


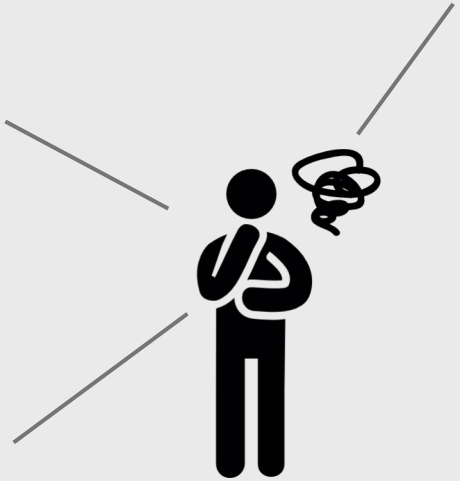


User Interfaces

Several convenient was to access the private cloud

Cluster Management Console: Manage, monitor, and troubleshoot your applications and cluster from a single, centralized, and secure management console

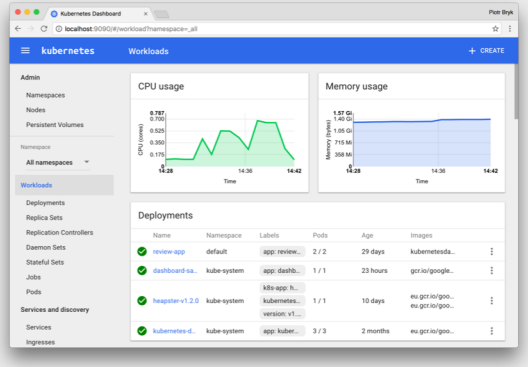




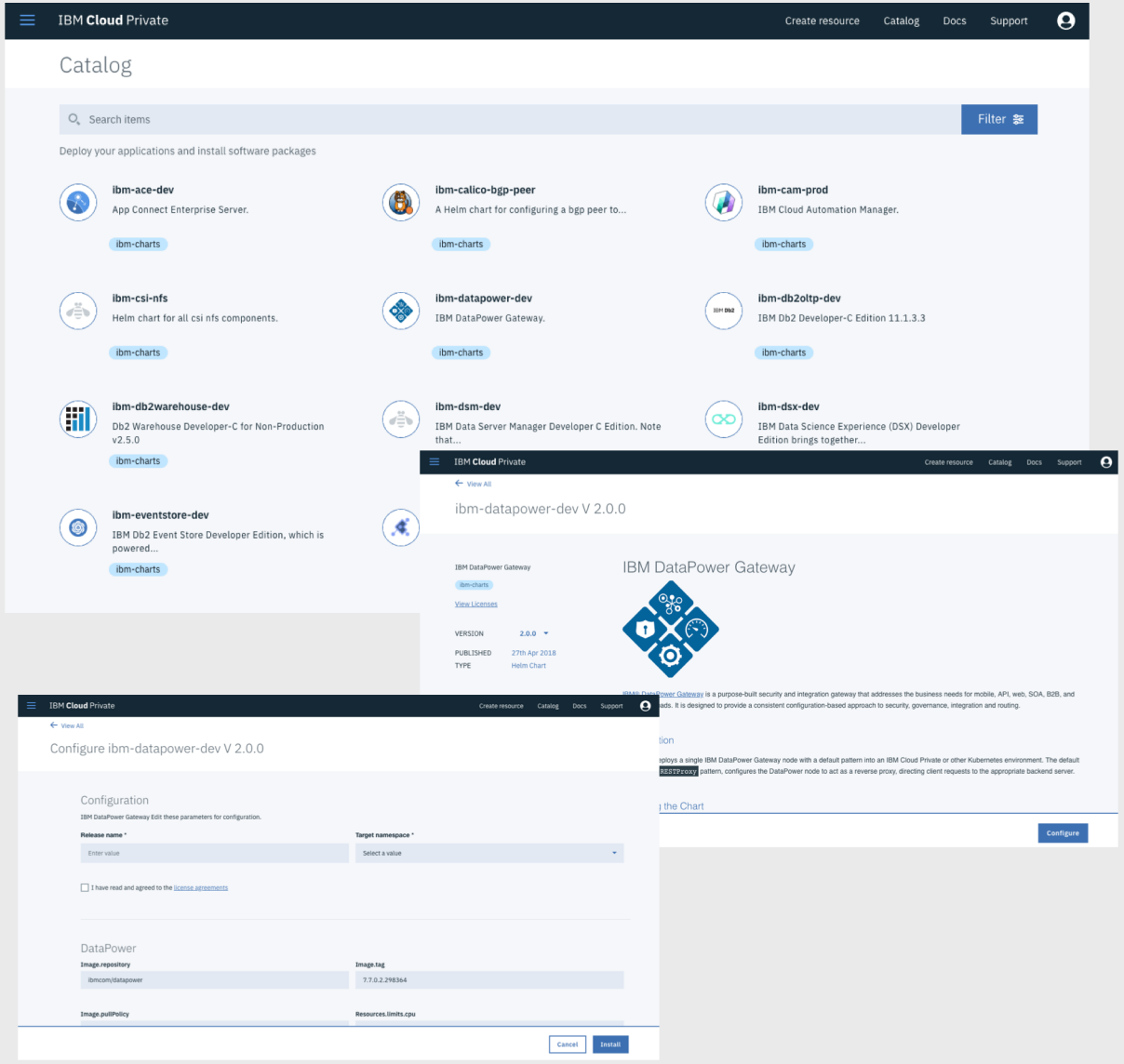
CLIs: Manage any / all aspects of your clusters, Kubernetes, charts, deployments etc. using your favorite command line tools such as the ICP CLI, K8s CLI, Helm and Calico interfaces / APIs

```
[root@ip-172-31-56-194 kubernetes]# export PATH=/home/ec2-user/kubernetes/platforms/linux/amd64:$PATH
[root@ip-172-31-56-194 kubernetes]# kubectl get nodes
NAME                          LABELS                                                    STATUS    AGE
ip-172-20-0-138.ec2.internal   kubernetes.io/hostname=ip-172-20-0-138.ec2.internal   Ready     3m
ip-172-20-0-25.ec2.internal    kubernetes.io/hostname=ip-172-20-0-25.ec2.internal    Ready     20m
[root@ip-172-31-56-194 kubernetes]# kubectl run ttnd-nginx --image=nginx
replicationcontroller "ttnd-nginx" created
[root@ip-172-31-56-194 kubernetes]# kubectl get pods
NAME              READY     STATUS    RESTARTS   AGE
ttnd-nginx-wp2y9   0/1       Pending   0          8s
[root@ip-172-31-56-194 kubernetes]# kubectl get pods --namespace=kube-system
NAME                                                  READY     STATUS    RESTARTS   AGE
elasticsearch-logging-v1-mcd2q                        1/1       Running   0          23m
elasticsearch-logging-v1-mmqqm                        1/1       Running   0          23m
fluentd-elasticsearch-ip-172-20-0-138.ec2.internal    1/1       Running   0          3m
fluentd-elasticsearch-ip-172-20-0-25.ec2.internal     1/1       Running   0          20m
heapster-v10-eco91                                    1/1       Running   0          23m
kibana-logging-v1-slamn                               1/1       Running   0          23m
kube-dns-v9-7pe5g                                     4/4       Running   0          23m
kube-ui-v2-0573n                                      1/1       Running   0          23m
monitoring-influxdb-grafana-v2-mgzvo                  2/2       Running   0          23m
[root@ip-172-31-56-194 kubernetes]#
[root@ip-172-31-56-194 kubernetes]#
[root@ip-172-31-56-194 kubernetes]# kubectl get pods
NAME              READY     STATUS    RESTARTS   AGE
ttnd-nginx-wp2y9   1/1       Running   0          36s
```

K8s Web UI: You can deploy and use the traditional K8s UI



Catalog



The catalog provides a centralized location from which you can browse, and install applications and software packages in your cluster

Helm: A tool for managing Kubernetes packages of pre-configured Kubernetes resources called “charts”

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

Identity and Access Management (IAM)

Login success → OIDC tokens

Role-based access on resources represented via Cloud Resource Name(CRN)s

Identity and Access Management

User Interfaces, APIs, CLIs

Identity Management

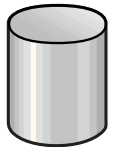
- Manage Users
- Manage Service IDs
- Manage Teams
- Manage Groups
- Manage Credentials
- Manage LDAP

Authentication Management

- Token Management
- User Authentication

Access Management

- Policy Administration
- Policy Decision
- Policy Enforcement
- CRN Management
- K8 Integration



Enterprise Directory

Key

- Client provides this security control
- IBM provides this security control capability that client needs to enable

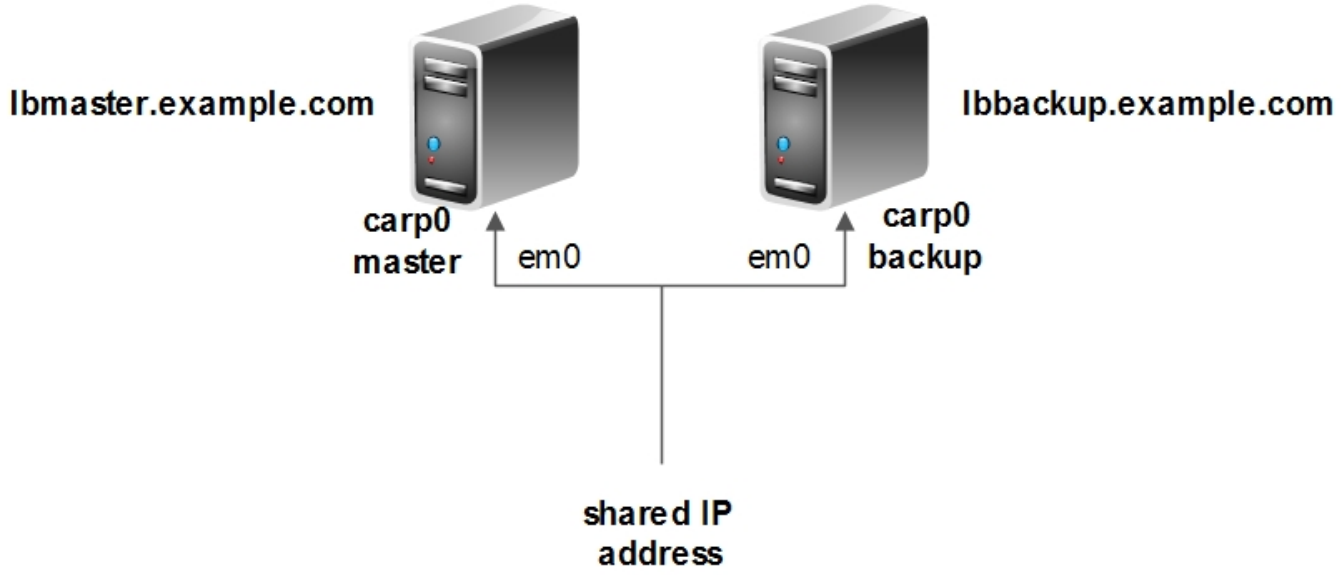
DNS

Every Service defined in the cluster gets a CNAME record in the cluster DNS
Only Services can be resolved by the DNS service
Name resolution is based upon namespace and the service name can also be resolved from outside of the name space by appending .<namespace>

```
web-terminal@webbterm-ibm-webterminal-6bf989c956-nrtfv:~$ kubectl get svc
NAME                        TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
csi-attacher-nfsplugin      ClusterIP   10.0.0.104   <none>        12345/TCP        1d
details                     ClusterIP   10.0.0.144   <none>        9080/TCP         5d
dx9-digitalexperience       ClusterIP   10.0.0.218   <none>        30015/TCP        14d
kubernetes                  ClusterIP   10.0.0.1     <none>        443/TCP          54d
my-nginx-service            ClusterIP   10.0.0.248   <none>        80/TCP           32d
my-other-service            ClusterIP   10.0.0.252   <none>        80/TCP           32d
nginx-np-service            NodePort    10.0.0.225   <none>        80:31357/TCP     32d
productpage                 ClusterIP   10.0.0.45    <none>        9080/TCP         5d
ratings                     ClusterIP   10.0.0.126   <none>        9080/TCP         5d
reviews                     ClusterIP   10.0.0.157   <none>        9080/TCP         5d
webbterm-ibm-webterminal    NodePort    10.0.0.204   <none>        3000:31864/TCP   6d
web-terminal@webbterm-ibm-webterminal-6bf989c956-nrtfv:~$ ping reviews
PING reviews.default.svc.cluster.local (10.0.0.157) 56(84) bytes of data.
```

VIP and UCarp / etcd
kube-dns / Cluster DNS

UCarp / etcd allows multiple hosts to share common virtual IP (or floating IP) addresses in order to provide automatic failover

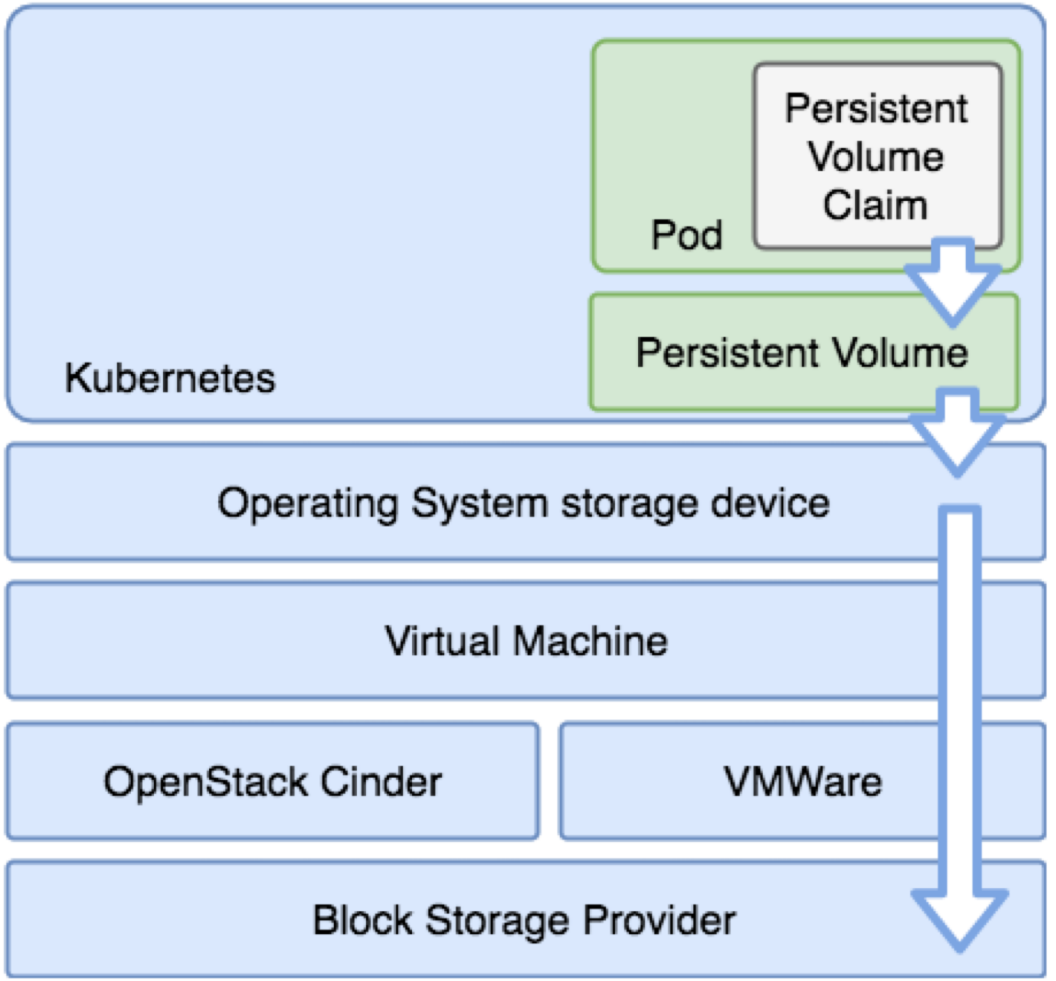


Persistent Storage

Persistent Volume is a storage resource within the cluster. PVs have a lifecycle independent of any individual pod that uses it. This API object encapsulates the details of the storage implementation or cloud-provider-specific storage system.

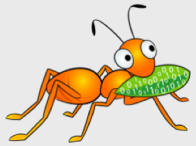
A **Persistent Volume Claim** is a storage request, or claim, made by the developer. Claims request specific sizes of storage, as well as other aspects such as access modes.

A **StorageClass** describes an offering of storage and allow for the dynamically provisioning of PVs and PVCs based upon these controlled definitions.



IBM Cloud Private Storage Providers

Kubernetes and IBM Cloud Private offer many options for managing persistent storage within the cluster. ICP features the following:



GlusterFS enterprise grade of storage to K8s pods offering ease of configuration, scaling, encryption support, replication, striping and dynamic provisioning.

vSphere Cloud Provider (vSphereVolume Plugin) gives access to enterprise grade storage (vSAN, VMFS, Vvol) that is native to and already supported by the VMware infrastructure.

IBM Spectrum Scale for solutions not hosted in VMware provides direct access to IBM block storage via dynamic provisioning.

NFS provides a versatile and easy to use method of getting persistent storage to pods that is already available in most customer environments.

HostPath is ideal for testing persistence in non-production environments.

Ceph (Rook) is an industry proven option that can provide several storage options along with persistent volumes for Kubernetes

Logging
ELK / Elastic Stack

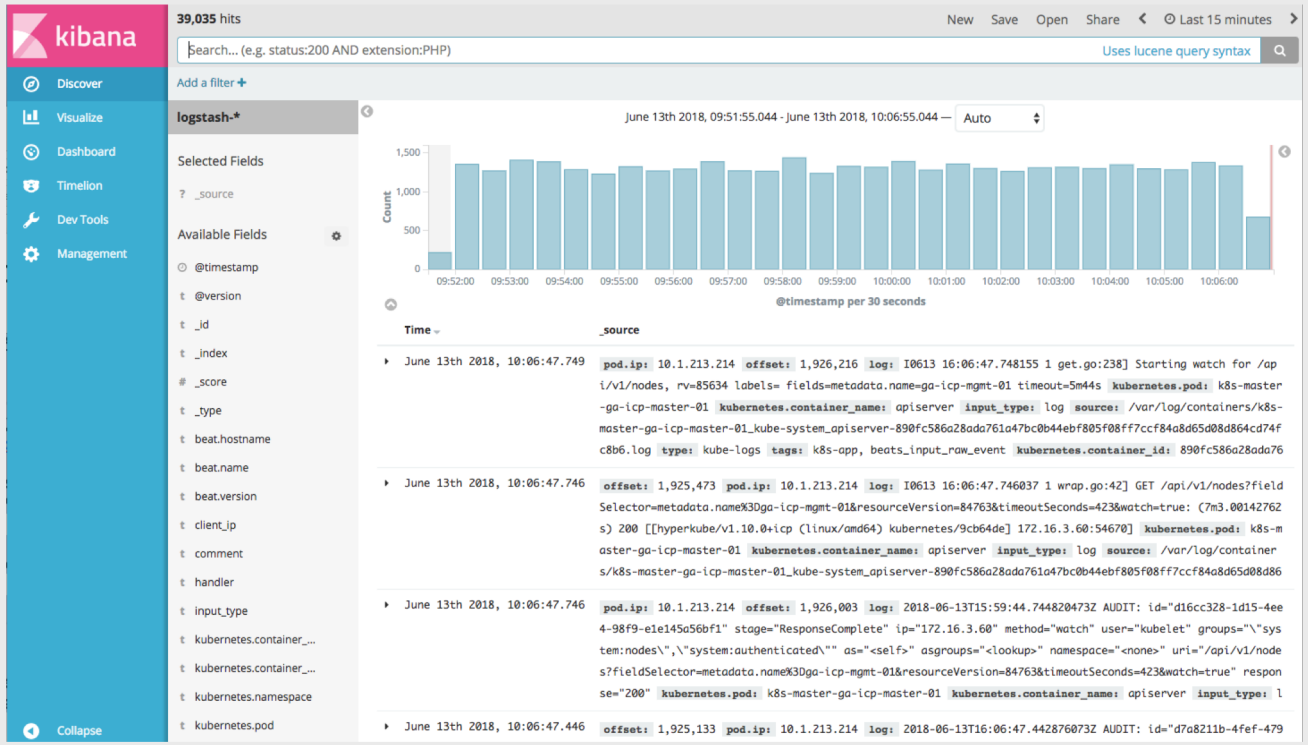
Filebeat: A log data shipper for local files. Filebeat monitors the log directories or specific log files, tails the files, and forwards them either to Elasticsearch and/or Logstash for indexing

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

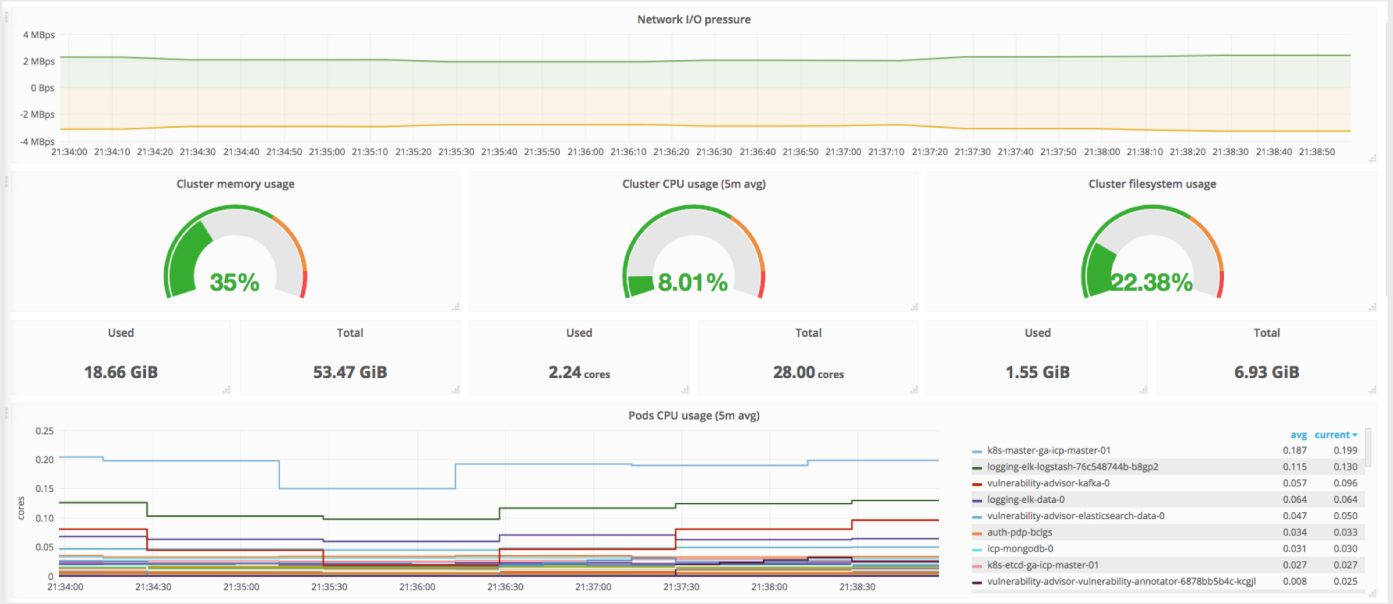
Kibana: An open source data visualization plugin for Elasticsearch. Users can create bar, line and scatter plots, or pie charts and maps on top of large volumes of data



Monitoring
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.



Network Overlay

PROJECT CALICO

A new approach to virtual networking and network security for containers, VMs, and bare metal services, that provides a rich set of security enforcement capabilities running on top of a highly scalable and efficient virtual network

- The calico/node Docker container runs on the Kubernetes master and each Kubernetes node in the cluster
- The calico-cni plugin integrates directly with the Kubernetes kubelet process on each node to discover which pods have been created, and adds them to Calico networking
- The calico/kube-policy-controller container runs as a pod on top of Kubernetes and implements the NetworkPolicy API
- Calico makes use of Layer 3

Calico network policy enforcement ensures that the only packets that flow to and / or from a workload are the ones the developer expects

Ingress Resources
Ingress, Ingress Controller, Proxy

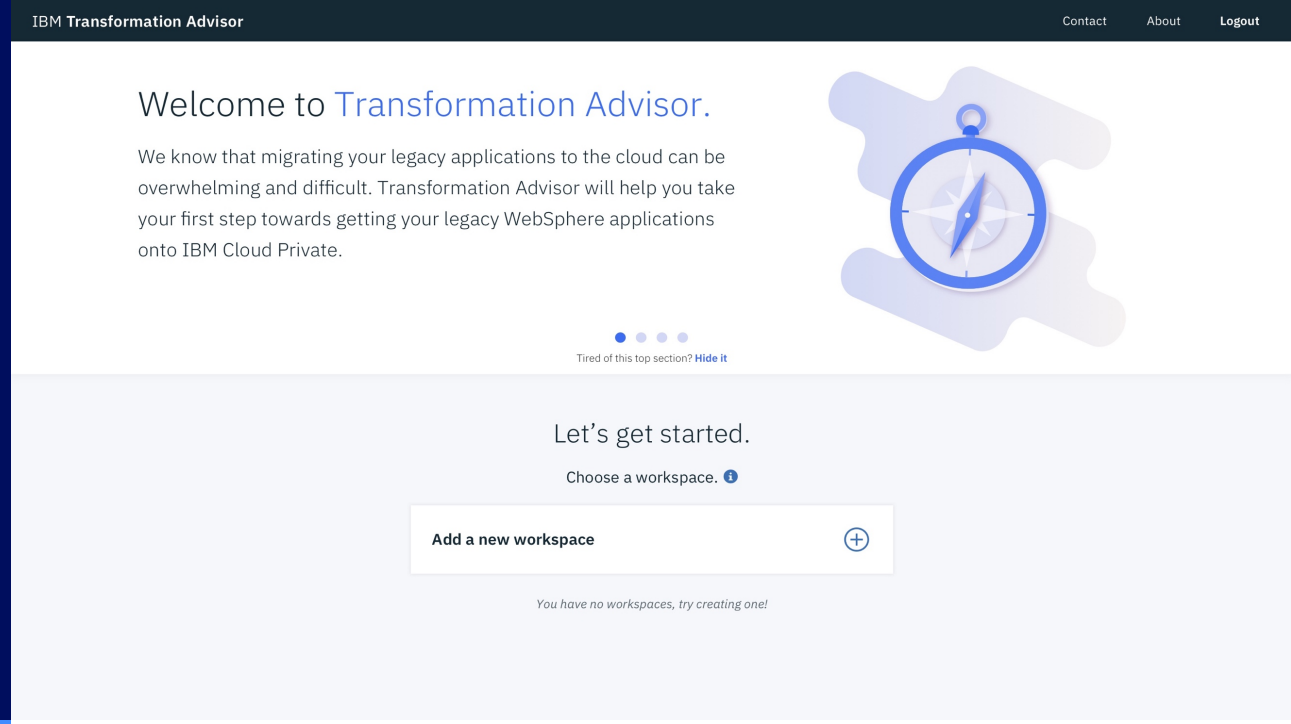
Ingress: Typically, services and pods have IPs only routable by the cluster network. All traffic that ends up at an edge router is either dropped or forwarded elsewhere.

- An Ingress is a collection of rules that allow inbound connections to reach the cluster services
- It can be configured to give services externally-reachable URLs, load balance traffic, terminate SSL, offer name based virtual hosting etc.
- Users request ingress by POSTing the Ingress resource to the API server

Ingress Controller: Responsible for fulfilling the Ingress, usually with a load balancer, though it may also configure your edge router or additional frontends to help handle the traffic in an HA manner

IBM Transformation Advisor

Transformation Advisor is a tool that **consumes information about your WebSphere Environment and Applications.** These inputs are combined with rules and insights gained from years of working with WebSphere and WebSphere applications to provide recommendations for your cloud journey.



CHALLENGES

Leverage existing application logic

Need to accelerate application development and maintenance

Monolithic applications that are complex and brittle

BENEFITS

Included and deployed on IBM Cloud Private

Introspects existing WebSphere Deployments

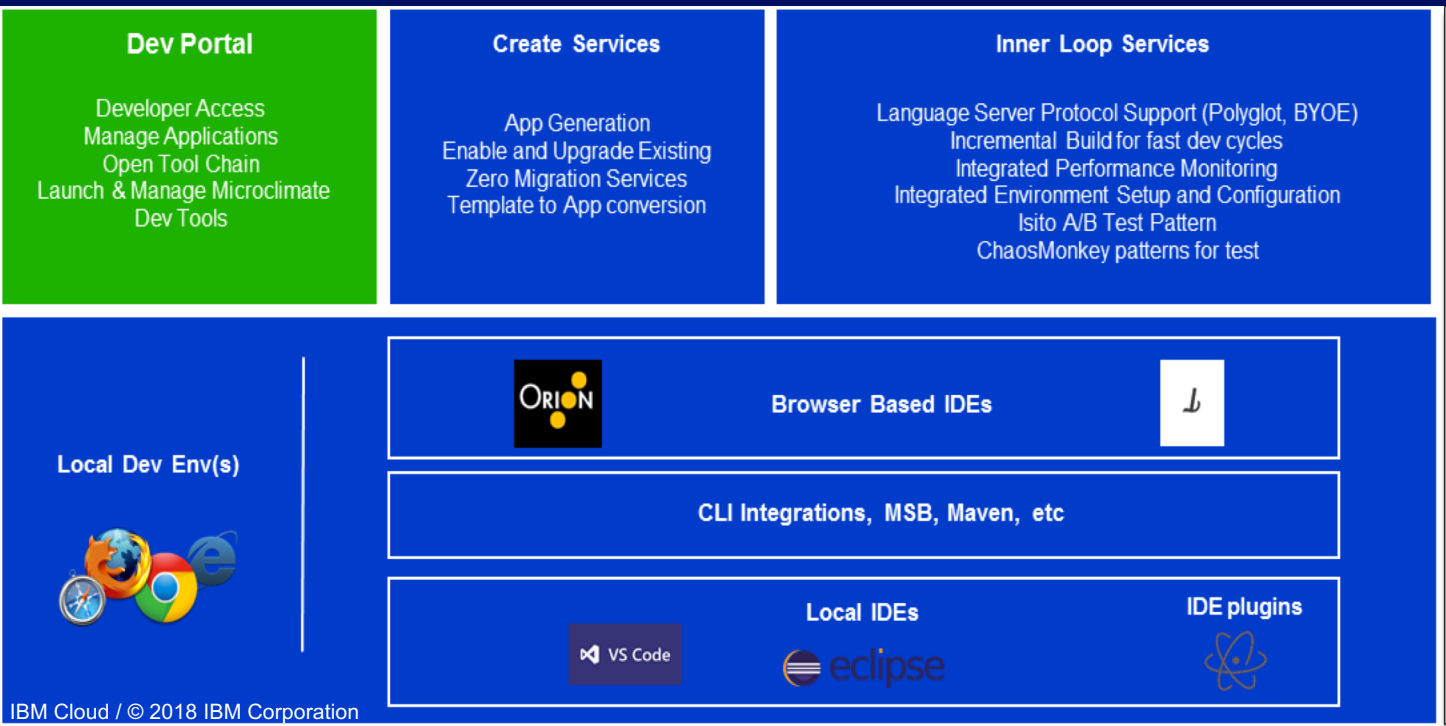
THE source of truth

Provides recommendations for Application Modernization

Enabling app modernization for our existing estates is part of IBM Cloud Private

IBM Microclimate

Microclimate is an **end to end development environment that lets you rapidly create, edit, and deploy applications.** Applications are run in **containers** from day one and can be delivered into production on **Kubernetes** through an automated DevOps pipeline using **Jenkins**. Microclimate can be installed locally or on **IBM Cloud Private.**



Providing an integrated DevOps experience for all apps is part of *IBM Cloud Private*

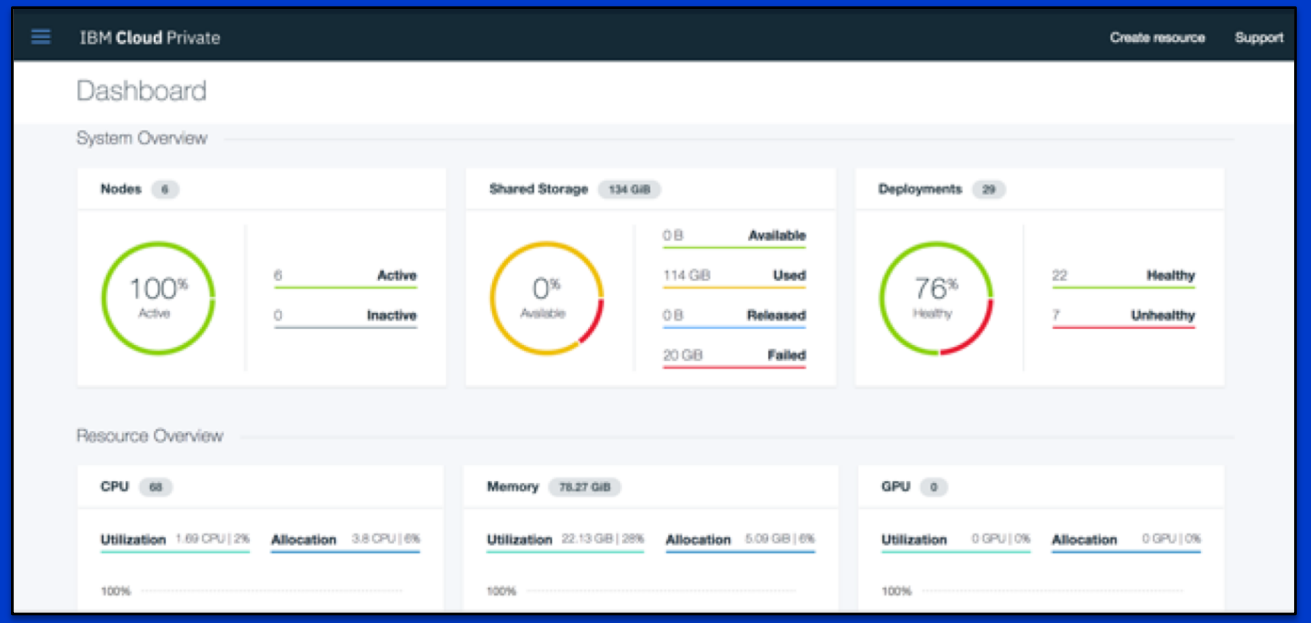
LEGEND:

ICP | Microclimate Local or ICP | Option for 3rd Party

Try IBM Cloud Private today!

Guided and Proof of Technology demos

Free Community Edition!



http://ibm.biz/ICP-DTE