Integration Technical Conference 2019

E03: High Availability for Cloud Integration Platform

Rob Nicholson

rob_nicholson@uk.ibm.com

IBM **Cloud**

IBM

Cloud Integration Platform High Availability

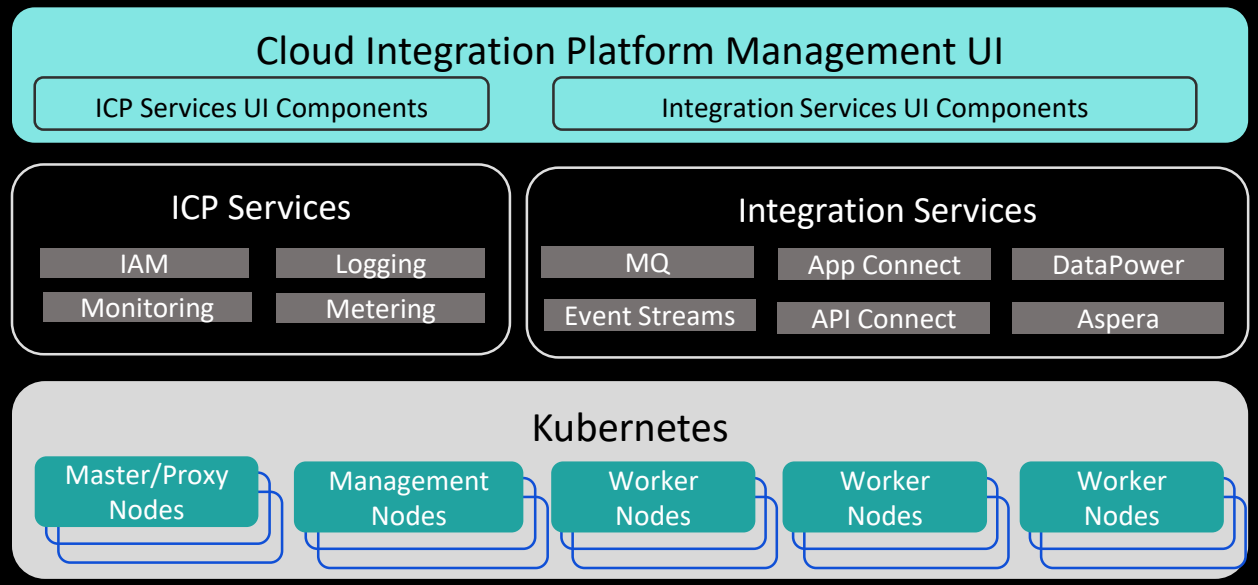
Understanding High Availability for the Cloud Integration Platform involves first understanding the overall architecture and the High availability considerations for the container platform. Based on this we then explore HA for each Integration service.

Agenda:

- Overview of Cloud Integration Platform Architecture.
- Kubernetes and IBM Cloud Private High Availability Concerns.
- Deployment considerations
- MQ High Availability
- Event Streams High Availability
- API Connect High Availability
- App Connect High Availability
- Aspera High Availability

Cloud Integration Platform Architecture

- Cloud Integration Platform (CIP) deploys and manages instances of **Integration Services** running on a Kubernetes Infrastructure
- Instances of Integration services are deployed individually as needed to satisfy each use-case.
- Services can be deployed in **Highly Available** topologies across multiple Kubernetes worker nodes or non HA on a single worker.
- Deployment and management is via the UI or CLI allowing integration with **CI/CD pipelines**.
- CIP Leverages the IBM Cloud Private **(ICP) services** which run on dedicated master/proxy and Management nodes in HA or non HA configurations.
- The Management UI unifies the management UIs of the Integration Services and the ICP services.



Container Platform Availability

- Cloud Integration Platform is based on ICP Foundation.
- ICP Foundation is functionally identical to ICP Cloud Native edition. The only difference is licensing.
- All HA, sizing and deployment information provided for ICP Cloud Native edition applies to Cloud Integration Platform.
- This presentation provides a summary. For more detail consult ICP Cloud Native edition documentation, articles and education. See the links page at the end of this presentation.

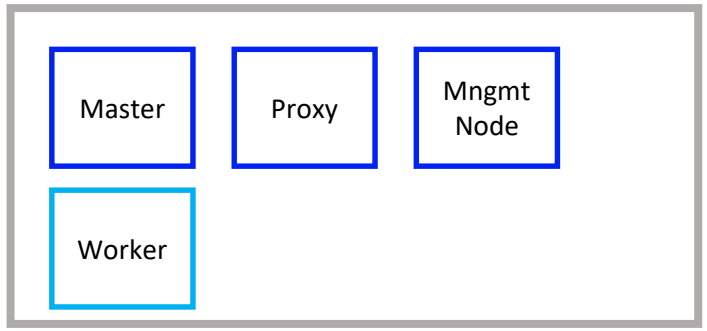
Node Types

- A kubernetes **Node** is a VM or bare metal machine which is part of a Kubernetes cluster.
- **Master Nodes** run the services that control the cluster including the *etcd* database that stores the current state of the cluster.
- **Proxy Nodes** transmit external requests to the services created inside your cluster.
- **Management Nodes** host management services such as monitoring, metering, and logging.
- **Worker Nodes** run Integration Services. If Cloud Integration Platform is running in a cluster with other workloads then these also run on the worker nodes.

- **Master, Proxy and Management** node types can be combined together but it is recommended in production clusters to keep them separate so that excessive workload does not impact stability.

- The minimum *theoretical* configuration is therefore one Master/Proxy/Management node and one Worker node. This would not be Highly available and is unlikely to have enough CPU to be usable for more than limited demos.

Note: There are other more obscure node types such as dedicated etcd nodes and vulnerability advisor nodes but these are beyond the scope of this presentation

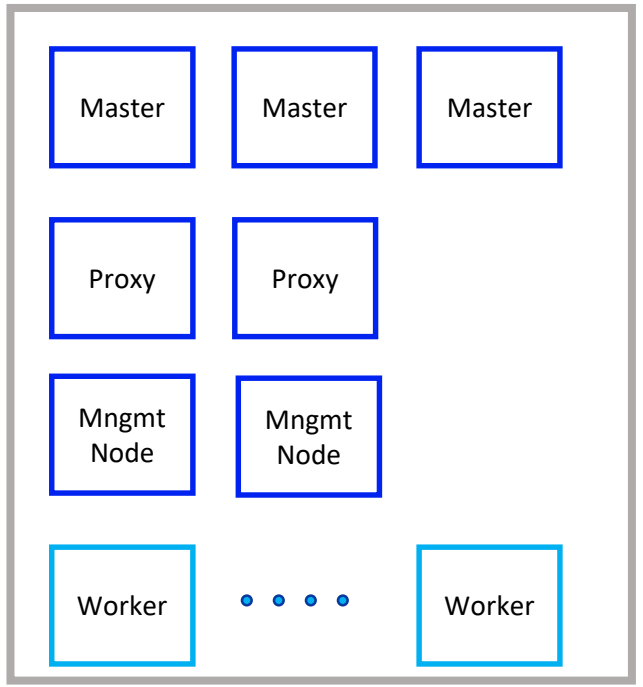


Cloud Integration Platform Cluster
Node types

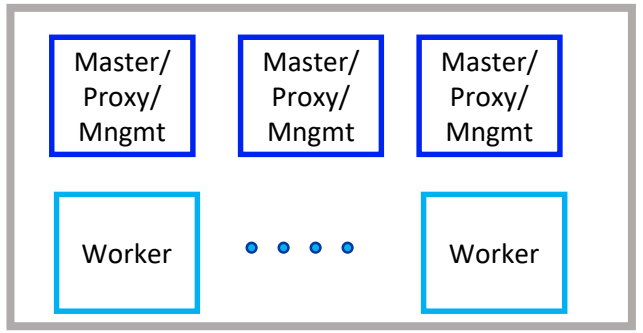
Cloud Integration Platform
Theoretical minimum cluster

Container Platform Availability

- To make the solution fully Highly Available, each component must be deployed in HA topology.
- **Master Nodes** contain software that uses a **quorum*** paradigm for high availability and so these must be deployed as an odd number of nodes. Typically either **3** or **5** masters are used in a HA cluster depending on the size of the cluster and the type of load.
- **Proxy Nodes** do not require a quorum so 2 or more Proxy nodes are needed for HA
- **Management Nodes** do not require a quorum so 2 or more Proxy nodes are needed for HA
- **Worker Nodes** run Integration Services. Depending on the integration services required 2 or more worker nodes may be needed for HA. (More detail in subsequent slides)
- As previously noted of course Master/Proxy/Management nodes can be combined so a minimal configuration could be 3 Master/Proxy/Management nodes.
- A topology that is often used is: 3 Master/Proxies + 2 Management + 3 Workers.



Cloud Integration Platform HA Cluster



Cloud Integration Platform minimal HA Cluster

Taken from ICP Think 2019 session.

A typical production environment

Machine Role	Number	vCPU (>= 2.4 GHz)	Memory	Disk Space	Comment
Master	3	16	32GB	500GB	3 for HA
Management	2	16	32GB	500GB	2 for HA
Proxy	2	4	16GB	400GB	2 for HA
Vulnerability Advisor	1	8	32GB	500GB	Optional (none-HA)
Worker Nodes	2-50	8	32GB	400GB	

http://ibm.biz/icpcapacityplan

Deployment Considerations

- **How many independent clusters do you need?**
  - Enterprises deploy multiple clusters for any/all of the following reasons:
    - Geographic availability – Ability to survive a regional outage
    - To separate organizations
    - To separate development, test and production.
    - To guard against errors by individual operators.

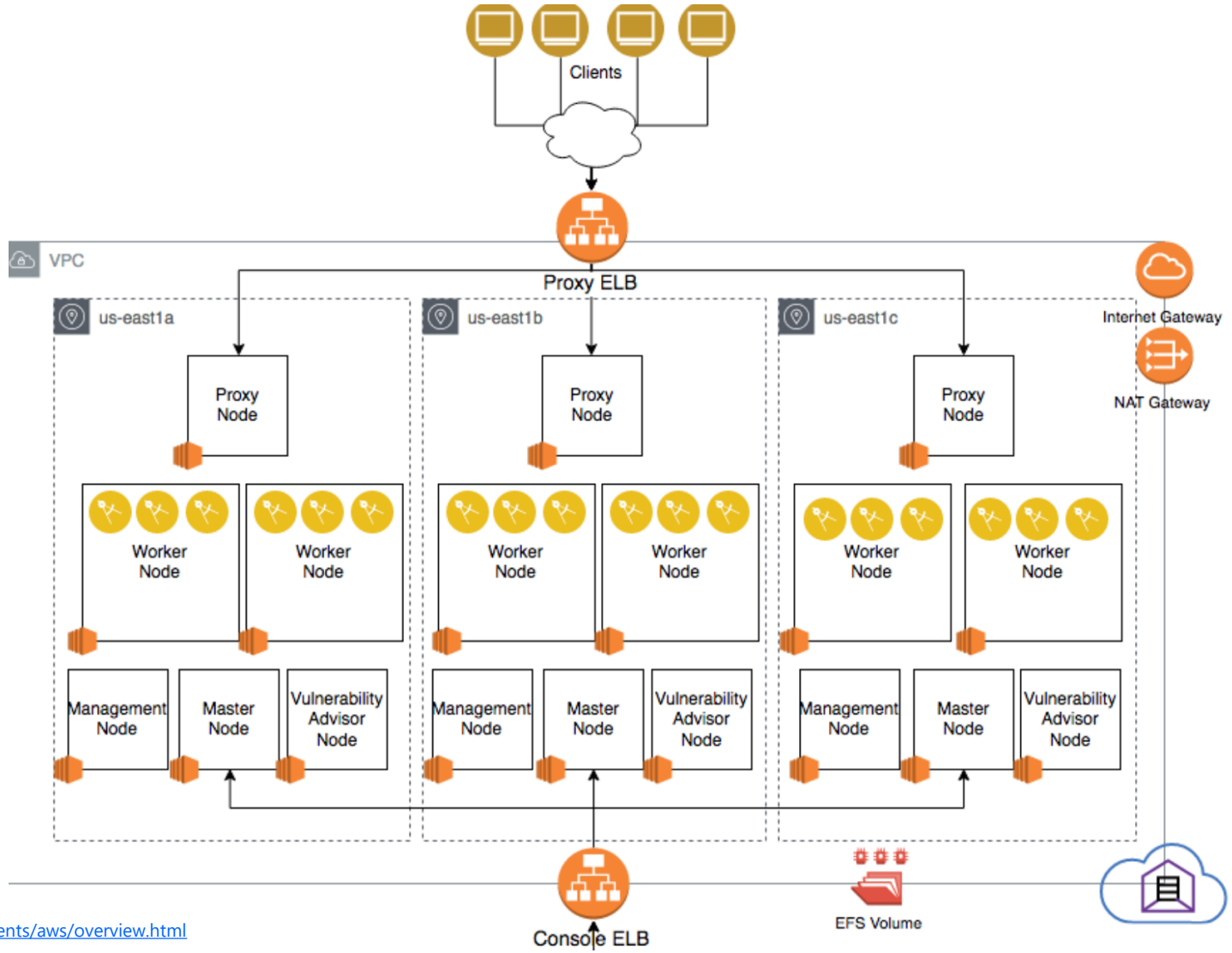
- **Are the Kubernetes nodes deployed into separate failure domains?**
  - Separate Availability zones in a public cloud.
  - Separate physical servers/racks in a data centre
    - What are the common points of failure?
  - See https://kubernetes.io/docs/reference/kubernetes-api/labels-annotations-taints/#failure-domainbetakubernetesiozon

Example – HA Cluster in AWS

Leverage available zone
- Master/mgmt/va across available zone
- User application across available zone

AWS ALB/NLB
- Load balancer for management control plane
- Load balancer for user application
- Security group to control network access

EBS as persistent storage



Article: http://ibm.biz/icponaws
Documentation: https://www.ibm.com/support/knowledgecenter/en/SSBS6K_3.1.2/supported_environments/aws/overview.html
links to a quickstart: https://aws.amazon.com/quickstart/architecture/ibm-cloud-private/

Public Cloud Deployment

- ICP 3.1.1 is supported on IBM Cloud and Amazon.
  - Current CIP is based on ICP 3.1.1
- ICP 3.1.2 adds support for azure Azure
- On public clouds it is recommended to use the cloud storage solution.
  - IBM Block Storage
  - Amazon EBS
- Distribute cluster across 3 Availability zones with less than 30ms latency.
  - Multiple AZs in single region. Not multiple regions.

High Availability for Integration services.

Integration Services run on the **worker nodes**.

The Cloud Integration **SolutionPak** is composed from the **CloudPaks** making from the component products.

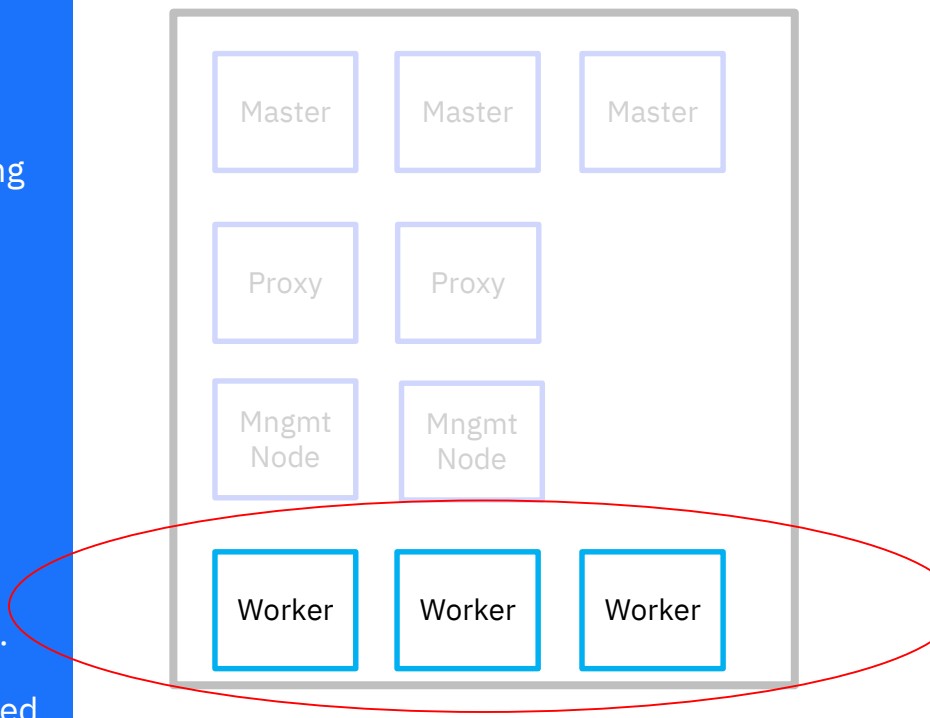
IBM cloudpaks are developed by the product development teams to embody best practice for deploying the product onto kubernetes as a secure, scalable **Highly Available** deployment.

Thus, at a high level, all that has to be ensured to provide a highly available deployment is:

- There are sufficient nodes for the solution*
- The nodes are deployed into separate failure domains so that a single failure will not take out multiple nodes. **

Kubernetes takes care of:

- Running appropriate numbers of instances of the solutionpak, spread across the available workers.
- Mixing together the workloads on the available worker nodes taking account the resources they need.
- Restarting workloads that fail and recovering from failed nodes by scheduling workloads onto alternative nodes.



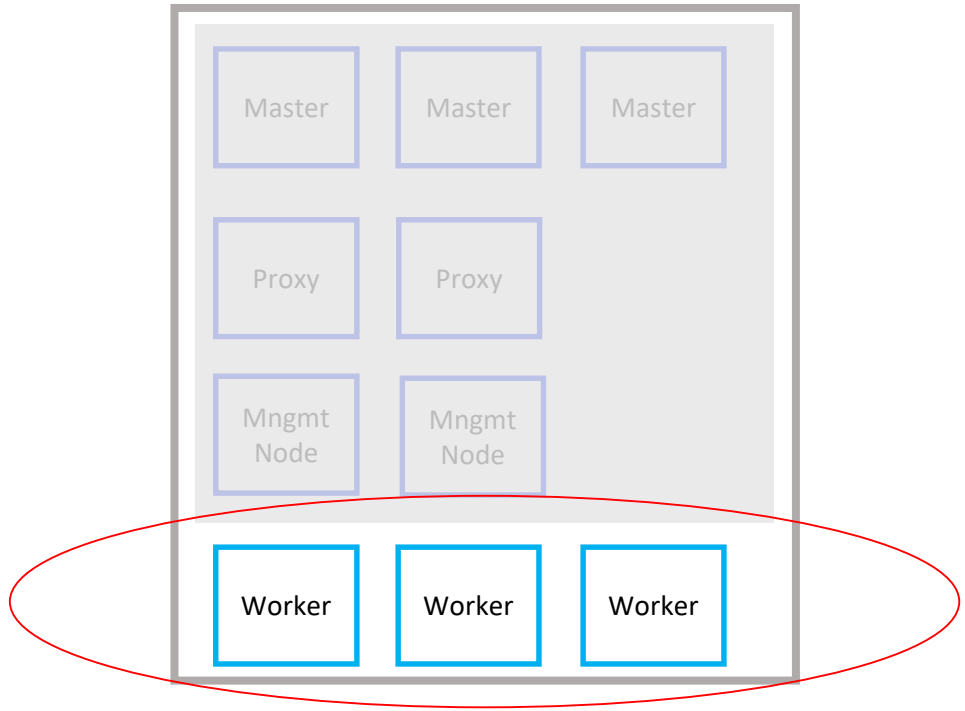
Cloud Integration Platform HA Cluster

* Some of the component products require 2 worker nodes for active/standby syle HA and others require 3 or more workers for quorum style deployment.
** In larger clusters the nodes should be spread between 3 ore more availability zones.

Myth Busting

- **MYTH 1:** I need separate worker nodes for each Integration Service
- **MYTH 2:** I cannot run multiple Solutionpaks on the same nodes as ‘CloudNative’ workloads.

- **REALITY:** Kubernetes will schedule pods across the available nodes based on its scheduling policies and the declared resource requirements. This will almost certainly mix workloads together on nodes unless you explicitly prevent it.
  - Its possible to use taints and annotations to control which nodes a workload is scheduled to. (Advanced topic not covered here)



Cloud Integration Platform HA Cluster

Summary - High Availability for Integration.

At a high level. All you need to do is deploy the cluster in a HA configuration with at least 3 workers.

Deploy the MQ services from the helm charts and they will be highly available by default.

Product	Approach(es) to HA	Minimum number of worker nodes in the cluster
MQ	Data Availability – Failover Service availability – Active/Active	2
Event Streams	Quorum	3
APIC	Quorum	3
App Connect	Stateless – Active/active Stateful – Failover	2
Aspera	Quorum	3
Datapower	Quorum	3

License and Cores required for minimal and HA

Component	Cores provisioned – Non HA	Cores provisioned - HA	Min CIP licenses non HA	Min CIP licenses HA	Nodes
ICP Foundation. (Master, Proxy, Management nodes)	16	40	0	0	Typically: 2 dedicated Nodes non HA 7 dedicated Nodes HA
CIP Platform, - Navigator,	0.5	1.5	0	0	At least 3 shared Worker Nodes
MQ	1	1 or 2	0.5	0.5 or 1	At least 2 shared Worker Nodes
APIC	16	48	16	48	At least 3 shared Worker Nodes
App Connect	1.1	1.1 or 2.1	3	3 or 6	At least 2 shared Worker Nodes
Aspera	4	12	4	12	At least 3 shared Worker Nodes
Event Streams	20.8	20.8	12	12	At least 3 shared Worker Nodes

Cloud Integration Platform High Availability

Understanding High Availability for the Cloud Integration Platform involves first understanding the overall architecture and the High availability considerations for the container platform. Based on this we then explore HA for each Integration service.

Agenda:

- Overview of Cloud Integration Platform Architecture.
- Kubernetes and IBM Cloud Private High Availability Concerns.
- Deployment considerations
- MQ High Availability
- Event Streams High Availability
- API Connect High Availability
- App Connect High Availability
- Aspera High Availability

IBM MQ High Availability Concepts

Availability of the **'MQ Service'**.

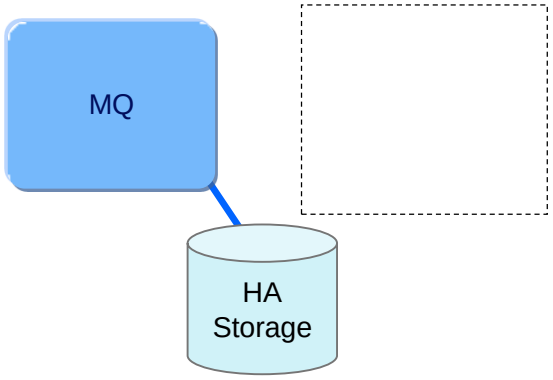
- The "MQ Service" is available when applications can connect to an MQ queue and send and receive messages.
- The MQ service can be made highly available by running multiple queue managers in active/active configurations.

Availability of **individual messages**.

- An individual message that has been sent to an MQ Queue is Available when a client can receive that particular message.
- The data can be made highly available by minimizing the time when the actual queue manager holding the message is not available.

IBM MQ High Availability Options on Distributed Servers

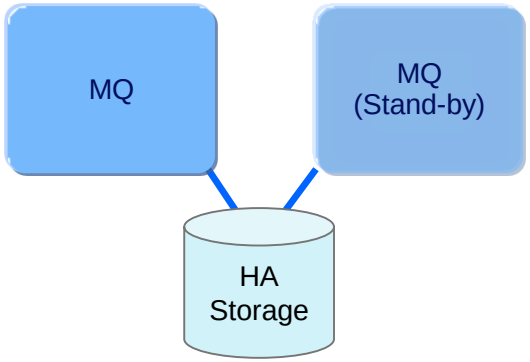
Single resilient queue manager



Platform Load Balancing

Platform HA

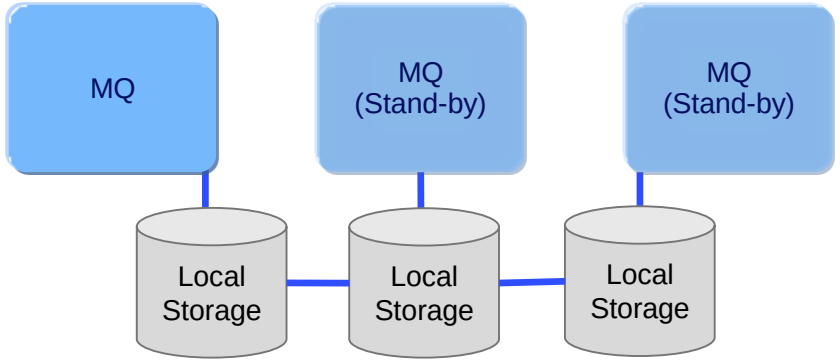
Multi-instance queue manager



Client load balancing

IBM MQ Product HA

Replicated data queue manager

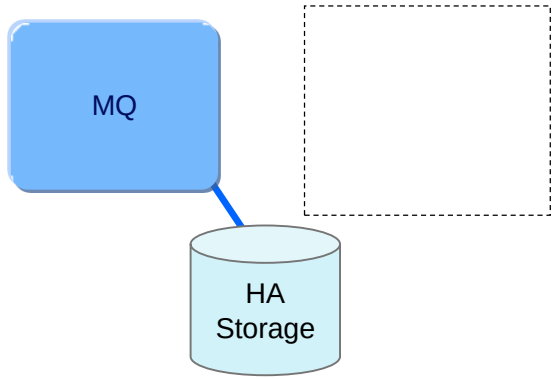


Client or Server load balancing

IBM MQ Product HA

IBM MQ High Availability in Cloud Pak / CIP

Single resilient queue manager

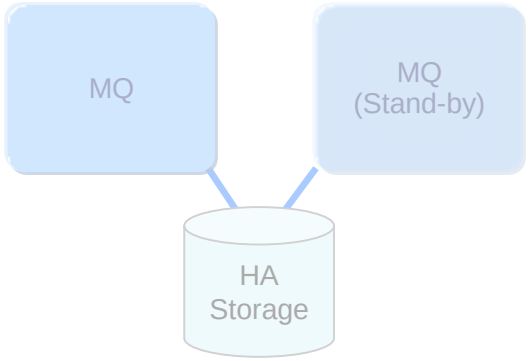


Platform Load Balancing

Platform HA

Current Approach

Multi-instance queue manager

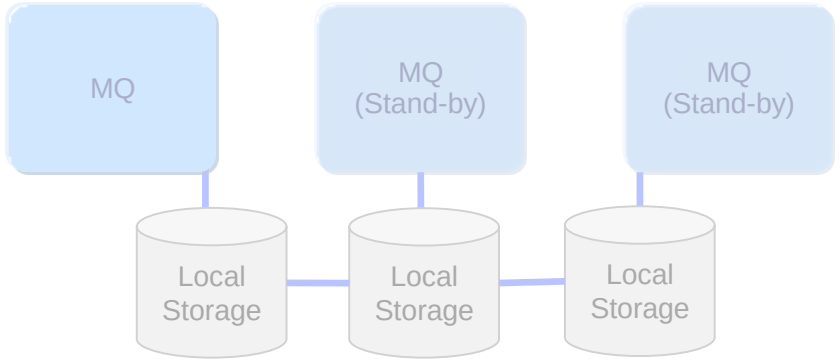


Client load balancing

IBM MQ Product HA

Investigating use of multi-instance within Kubernetes

Replicated data queue manager



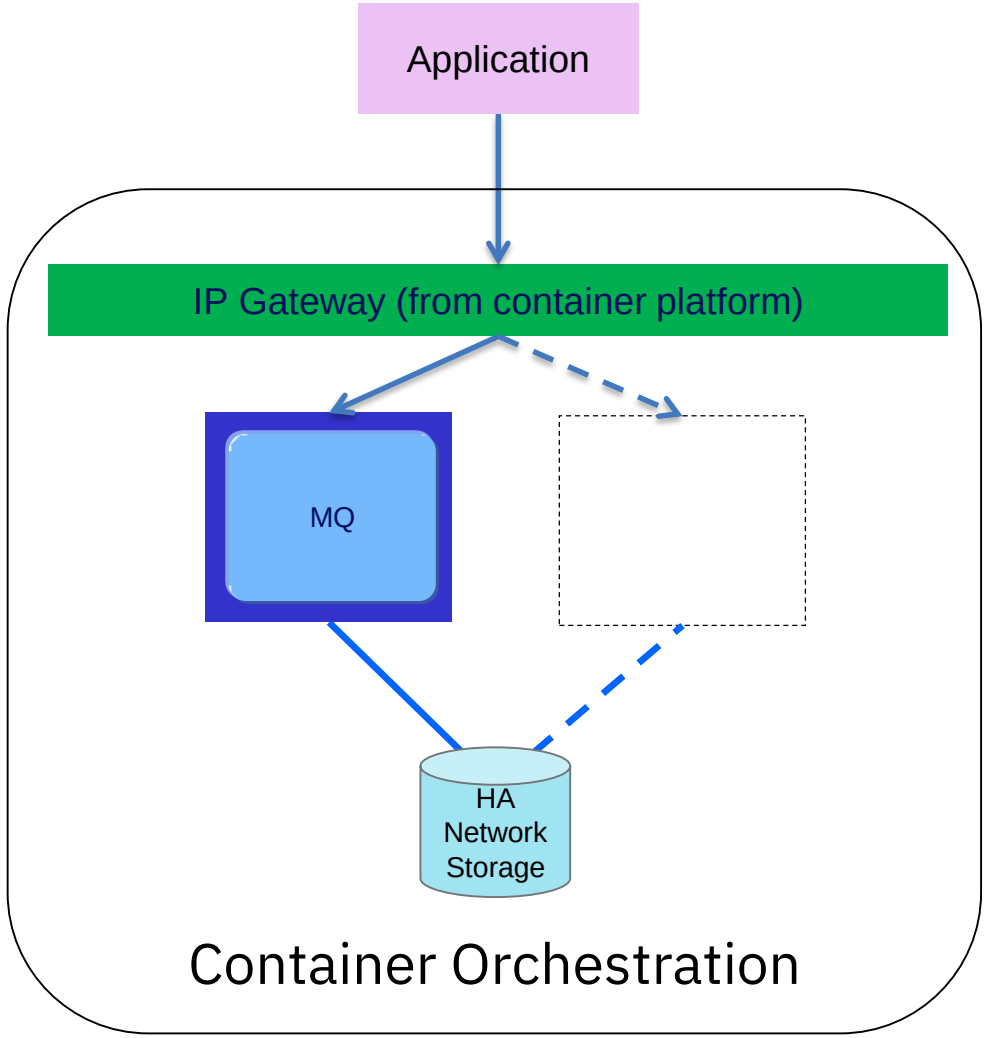
Client or Server load balancing

IBM MQ Product HA

Technology not supported in Kubernetes

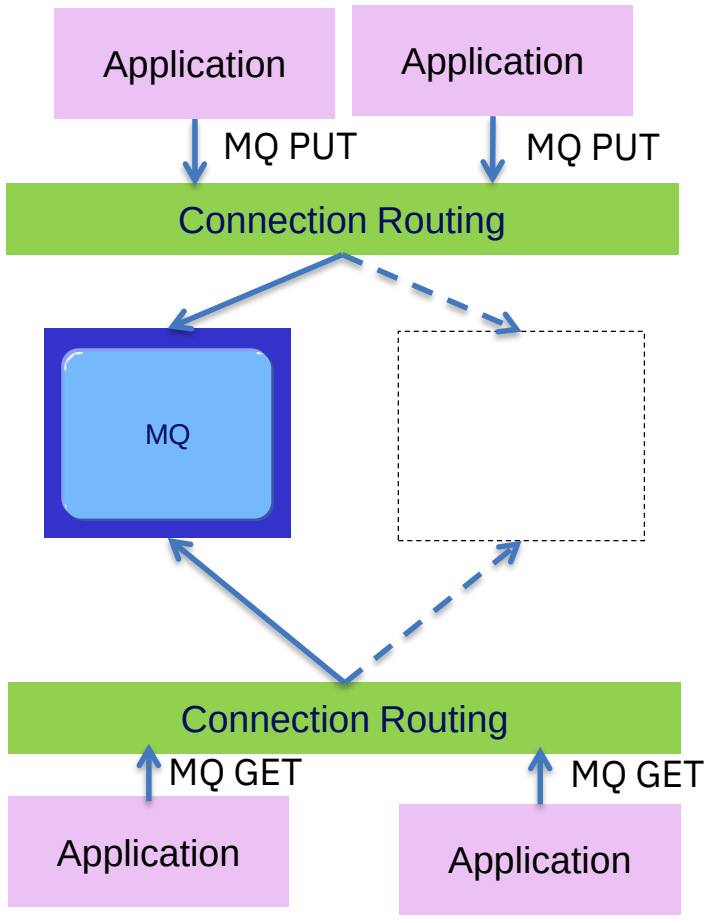
Single Resilient Queue Manager on Kubernetes

- Container restarted by Kubernetes*
- Data persisted to external storage
- Restarted container connects to existing storage
- IP Gateway routes traffic to the active instance
- Implemented as Kubernetes Stateful set of 1
- **Implications:** Both the service and the messages stored in the single queue manager are unavailable during failover

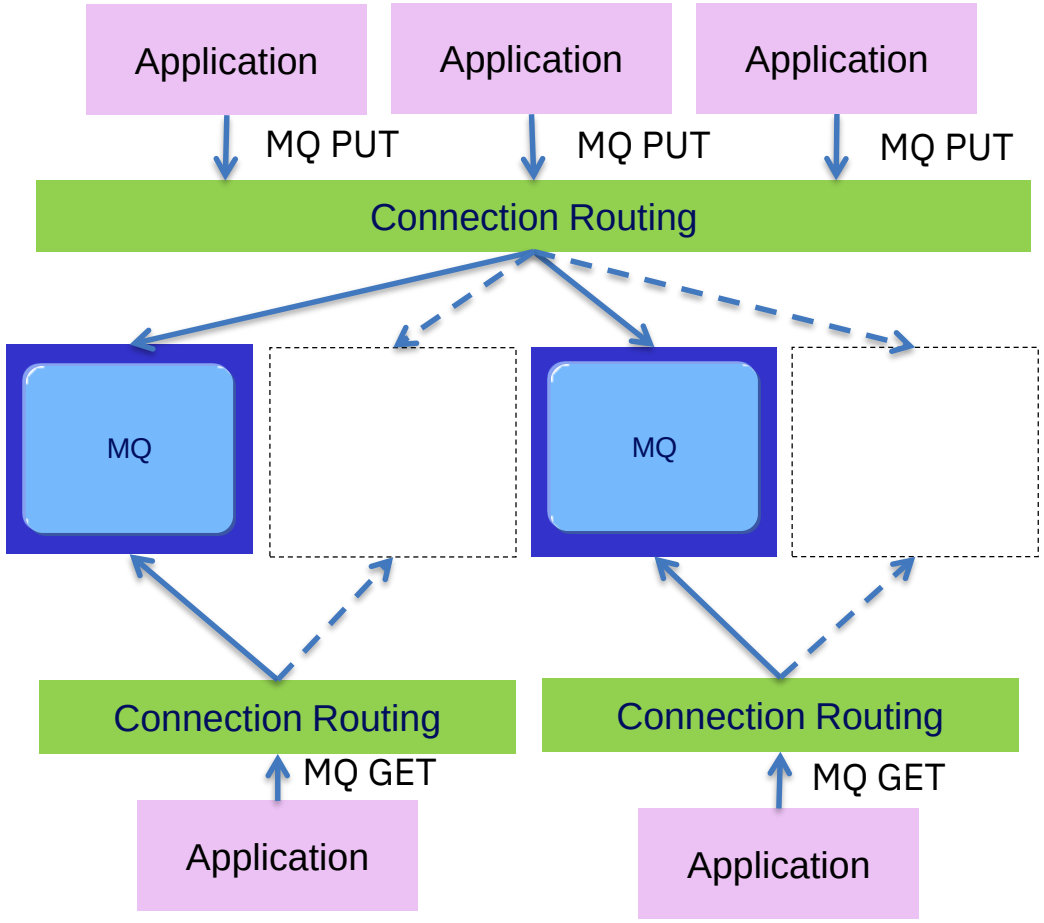


* Total Kubernetes node failure considerations described in https://developer.ibm.com/messaging/2018/05/02/availability-scalability-ibm-mq-containers/

Improving the MQ Service availability



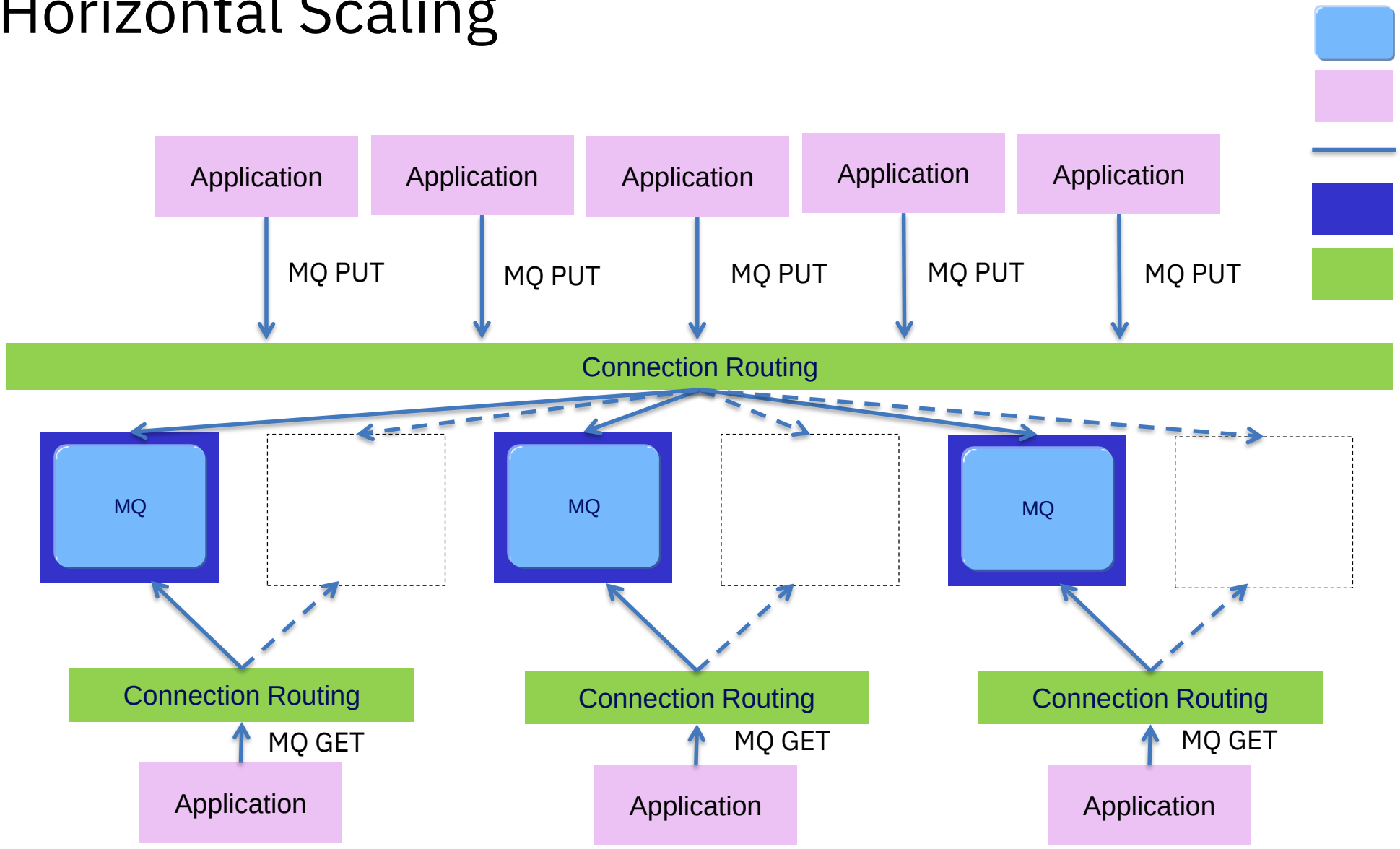
Single resilient container



Multiple resilient containers

- ✓ Provide access to store messages, even in failover situations. Connection Routing provided to distribute the traffic across the available instances.
- ✓ Applications retrieving messages attach to the individual Queue Manager. Connection Routing provided to route traffic to container location.

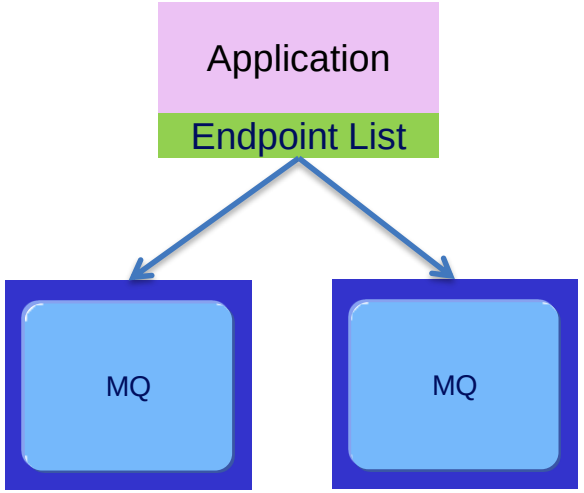
Horizontal Scaling



- ✓ Scale the MQ instances based on workload requirements

Connection Routing

Static Routing



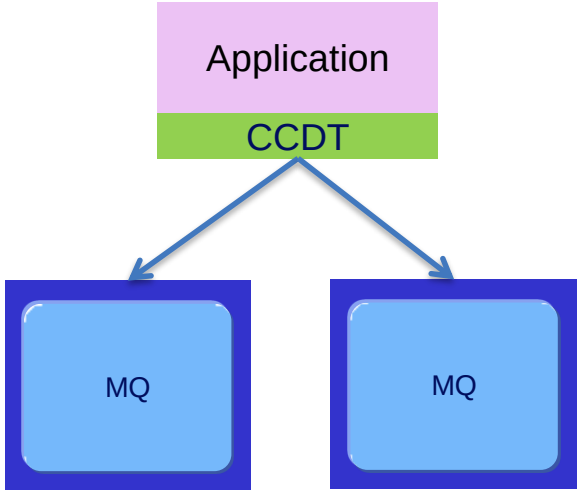
Client embeds endpoints

Performance impact when primary unavailable

Brittle configuration

No load balancing

Client Connection Definition Table

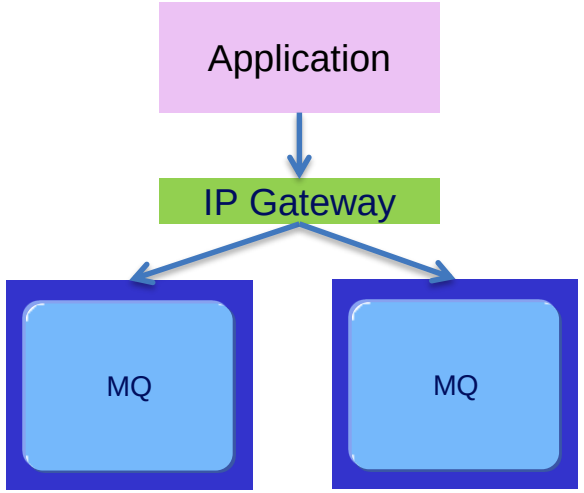


Client references endpoints

Enhanced Workload Management Strategies

Central configuration management

Load Balancer



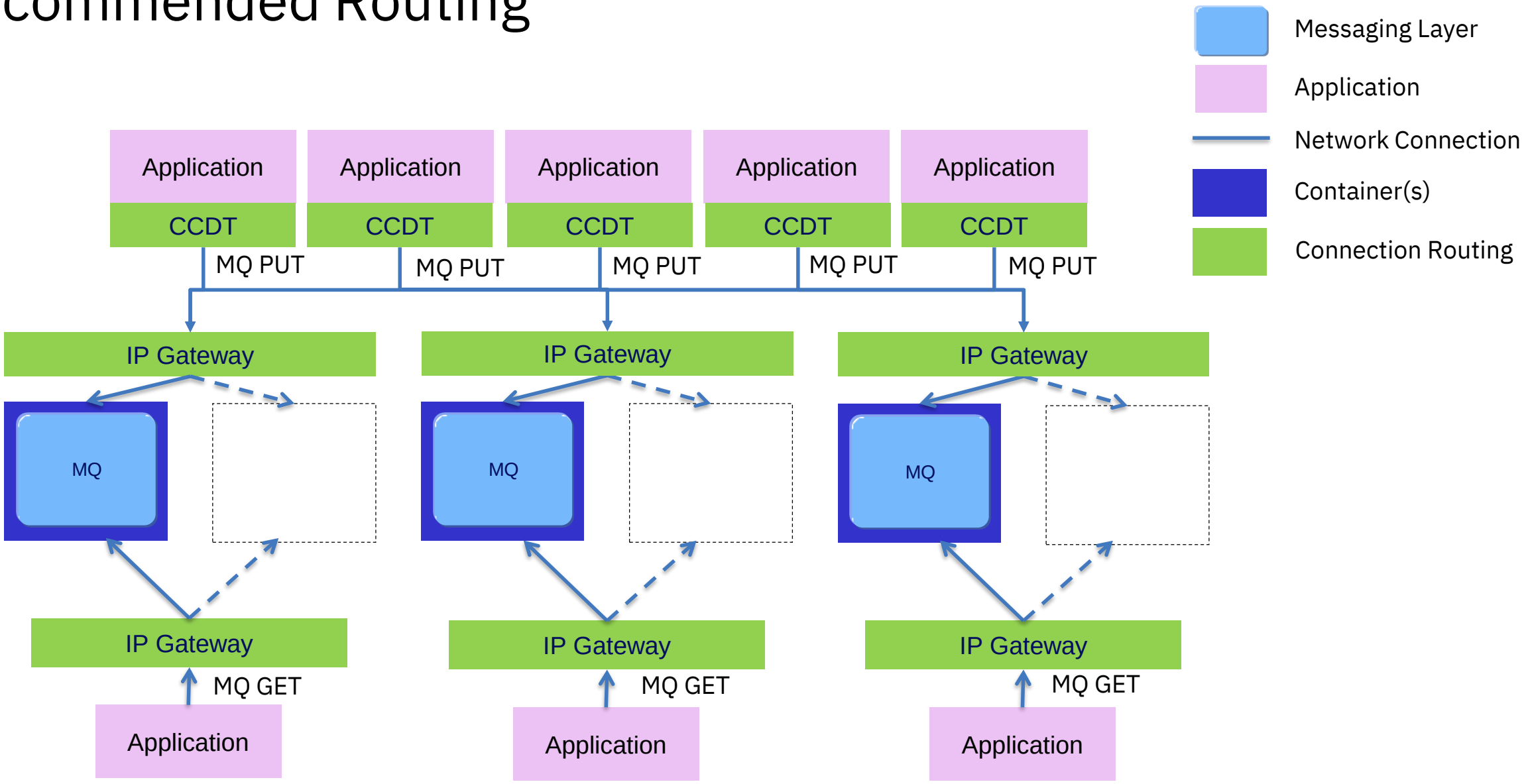
Client references endpoints

Enhanced Workload Management Strategies

Central configuration management

Not recommended for JMS

Recommended Routing



Cloud Integration Platform High Availability

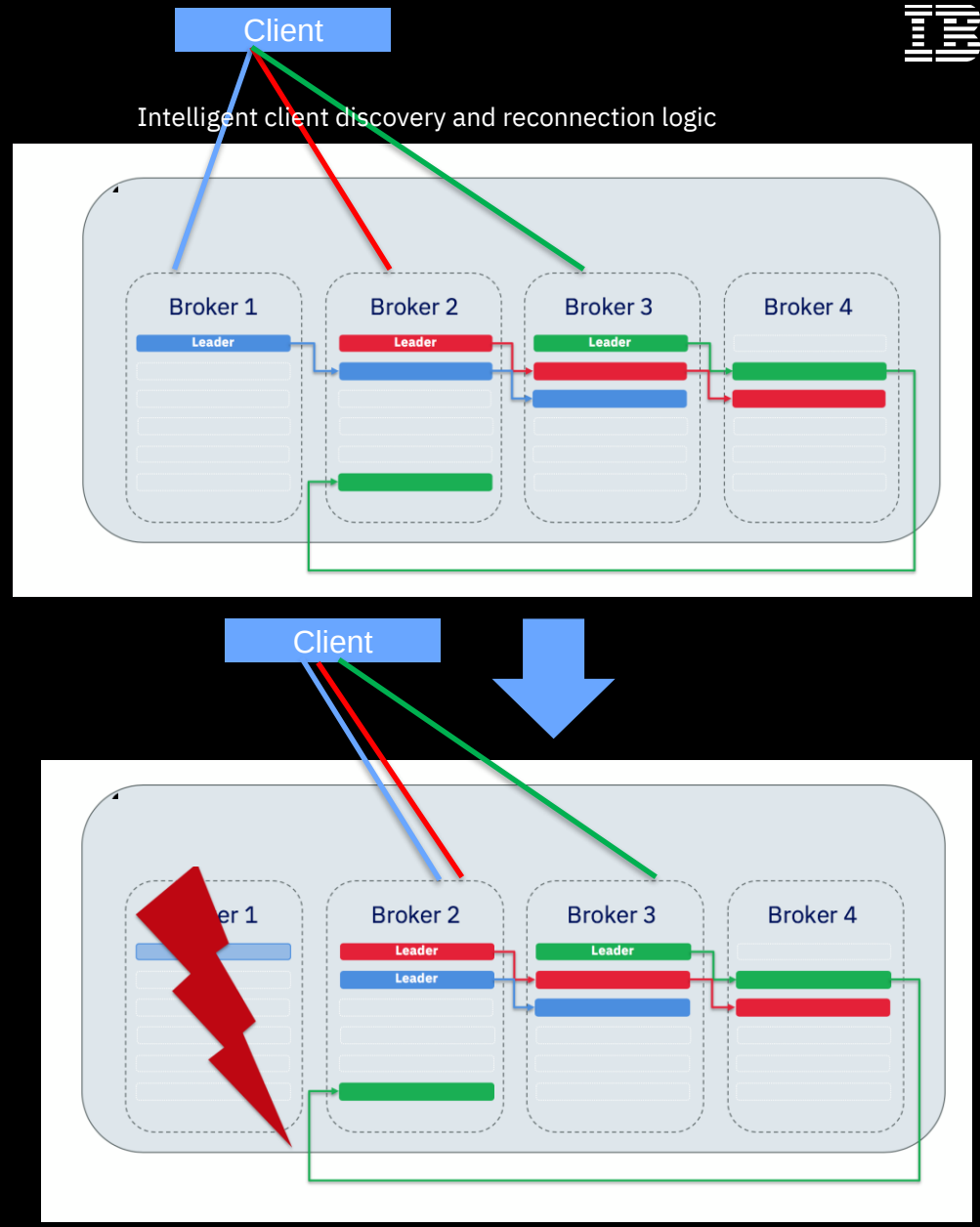
Understanding High Availability for the Cloud Integration Platform involves first understanding the overall architecture and the High availability considerations for the container platform. Based on this we then explore HA for each Integration service.

Agenda:

- Overview of Cloud Integration Platform Architecture.
- Kubernetes and IBM Cloud Private High Availability Concerns.
- Deployment considerations
- MQ High Availability
- Event Streams High Availability
- API Connect High Availability
- App Connect High Availability
- Aspera High Availability

Event Streams High Availability

- Event streams deploys Apache Kafka in a HA topology by default.
- Chart deploys brokers as a stateful set with 3 members by default.
- Kubernetes will schedule the pods to different nodes.
- Kafka's architecture is inherently HA. Client connection protocol handles discovery and failure.
- Default for Topics creation is 3 replicas with min in-sync copies=2
- For Multi-AZ deployments number of brokers must match number of AZs today.



Cloud Integration Platform High Availability

Understanding High Availability for the Cloud Integration Platform involves first understanding the overall architecture and the High availability considerations for the container platform. Based on this we then explore HA for each Integration service.

Agenda:

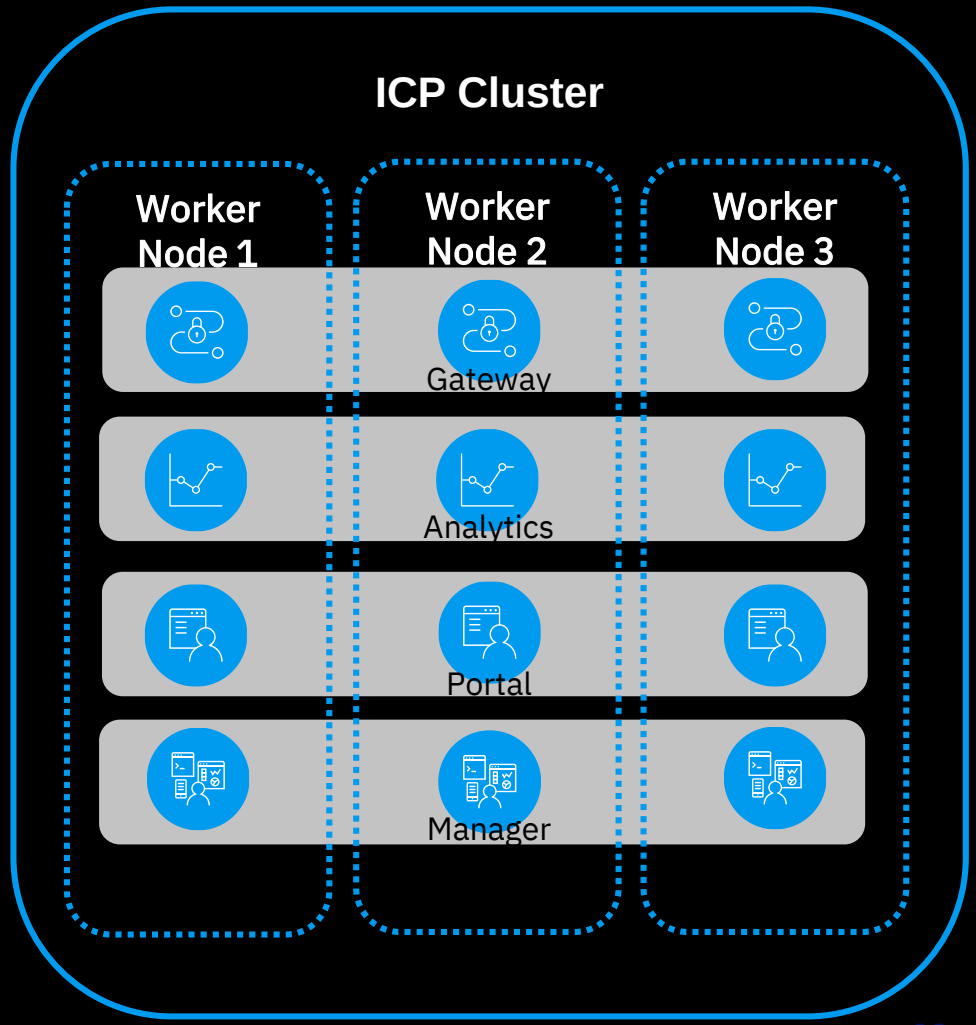
- Overview of Cloud Integration Platform Architecture.
- Kubernetes and IBM Cloud Private High Availability Concerns.
- Deployment considerations
- MQ High Availability
- Event Streams High Availability
- API Connect High Availability
- App Connect High Availability
- Aspera High Availability

API Connect High Availability

- API Connect requires 3 Worker nodes for HA.
- Chart deploys as HA by default:
  - global setting: mode = standard (dev mode is non HA)
  - Cassandra cluster size = 3
  - Gateway replica count = 3
- Kubernetes distributes the resulting resources across 3 (or more) nodes on the cluster for high availability.

API Connect whitepaper discusses multi-cluster, multi data centre and DR topics.

http://ibm.biz/APIC2018paper



Cloud Integration Platform High Availability

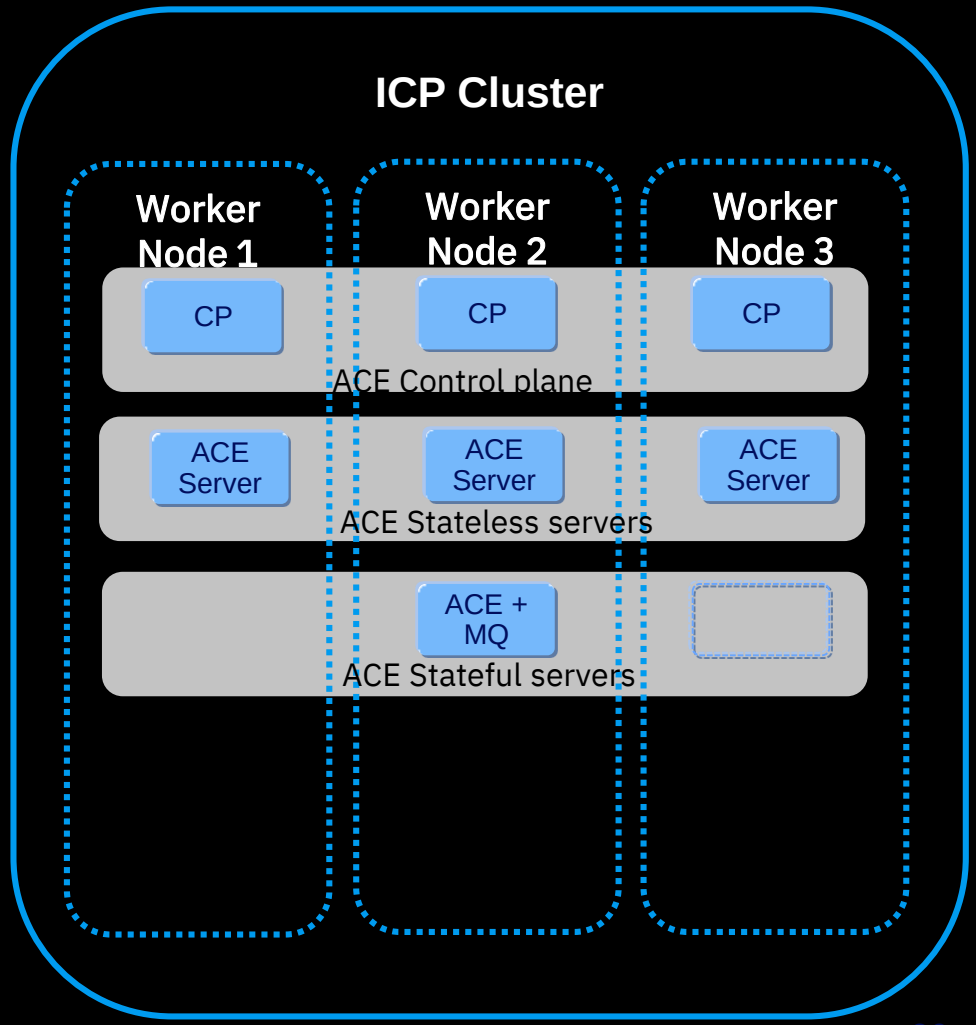
Understanding High Availability for the Cloud Integration Platform involves first understanding the overall architecture and the High availability considerations for the container platform. Based on this we then explore HA for each Integration service.

Agenda:

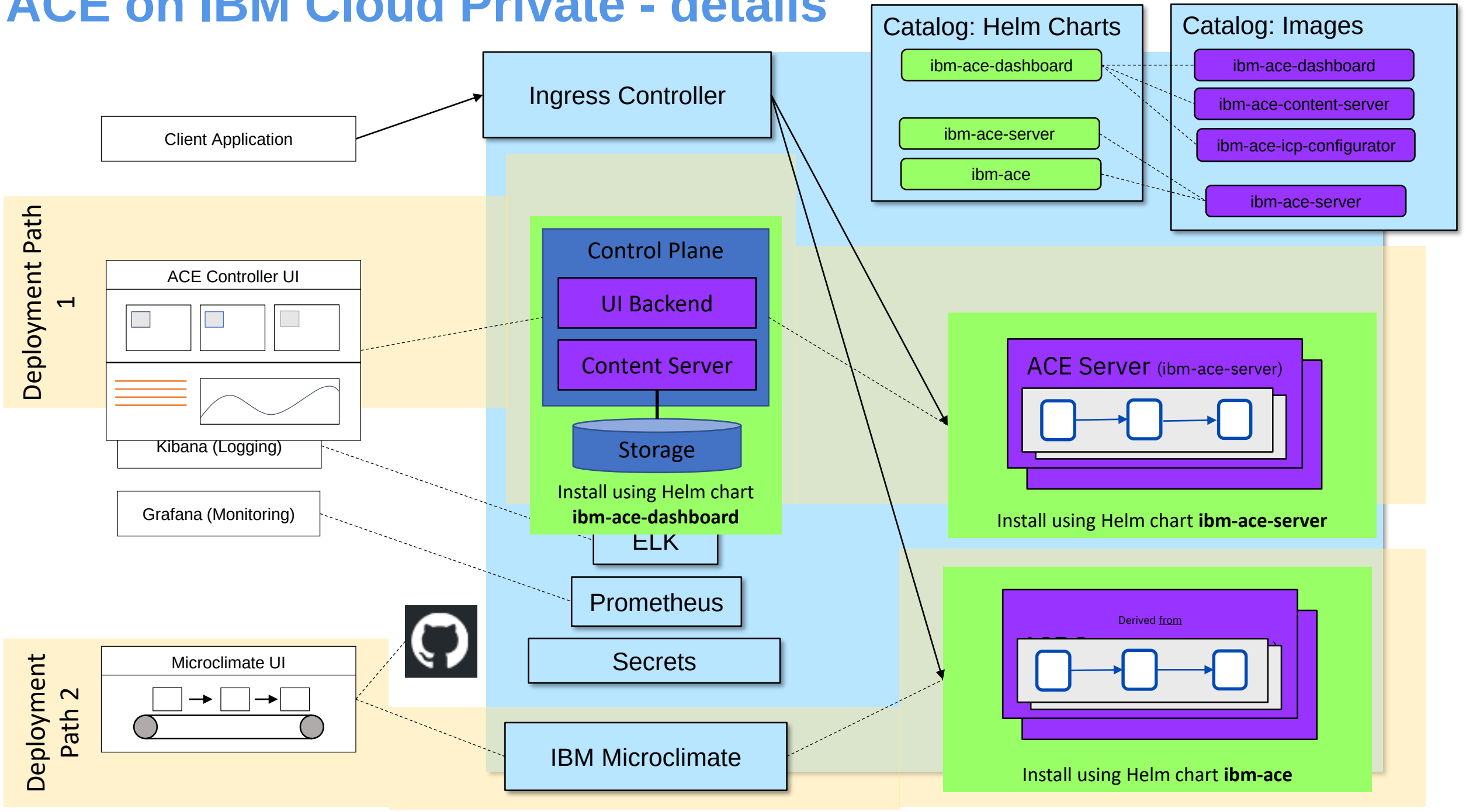
- Overview of Cloud Integration Platform Architecture.
- Kubernetes and IBM Cloud Private High Availability Concerns.
- Deployment considerations
- MQ High Availability
- Event Streams High Availability
- API Connect High Availability
- App Connect High Availability
- Aspera High Availability

App Connect High Availability

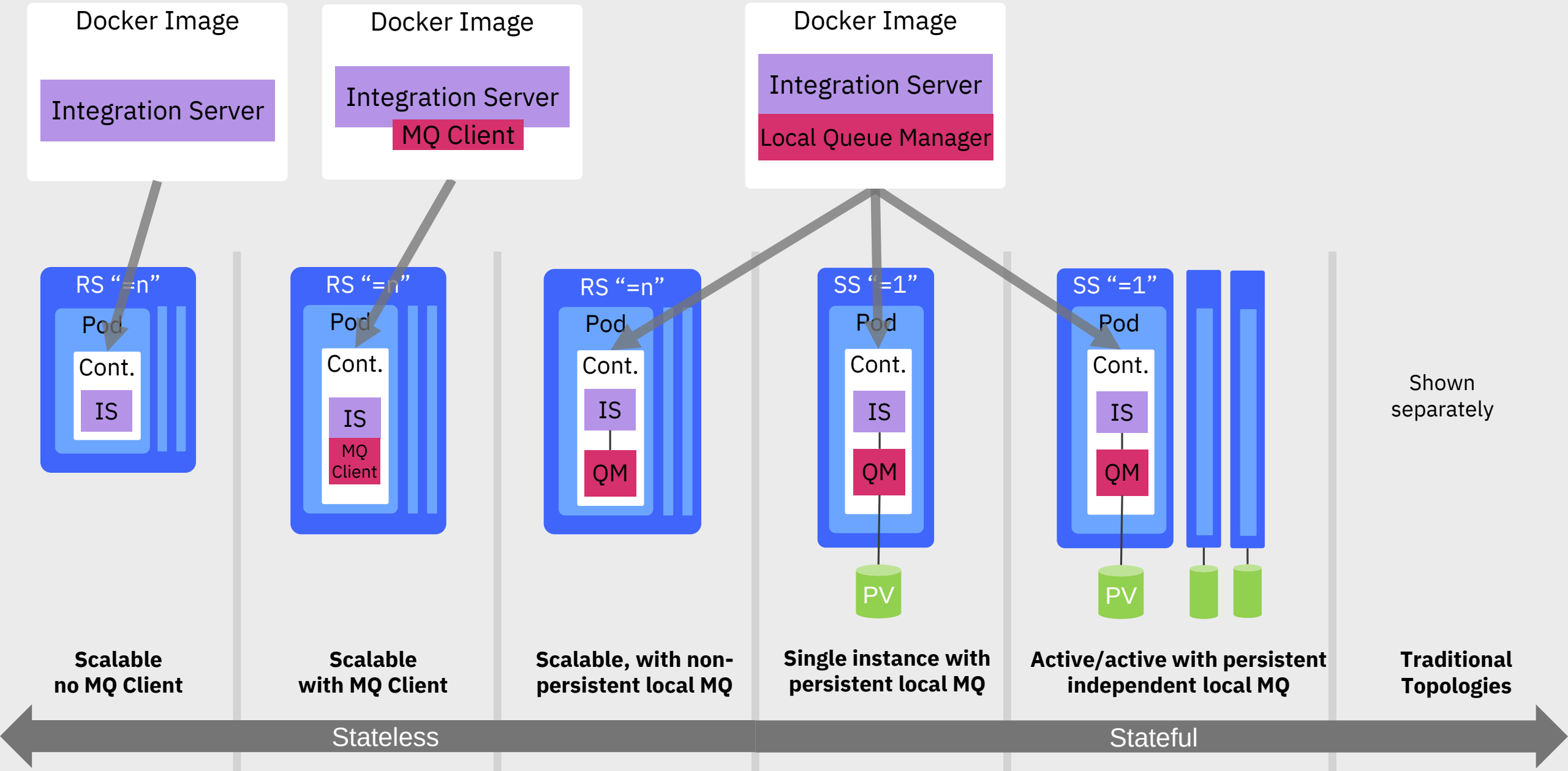
- ACE Control plane is HA (replicaset of 3) by default.
- Integration servers deployed without local MQ Queue Managers (QM) are stateless.
  - Deployed HA (replicaset of 3) by default.
  - BAR file is retrieved from control plane at startup.
- Integration servers deployed with a local QM are deployed like MQ.
  - Single Resilient Queue Manager (stateful set of 1).



ACE on IBM Cloud Private - details



Primary images mapped to core topologies



Cloud Integration Platform High Availability

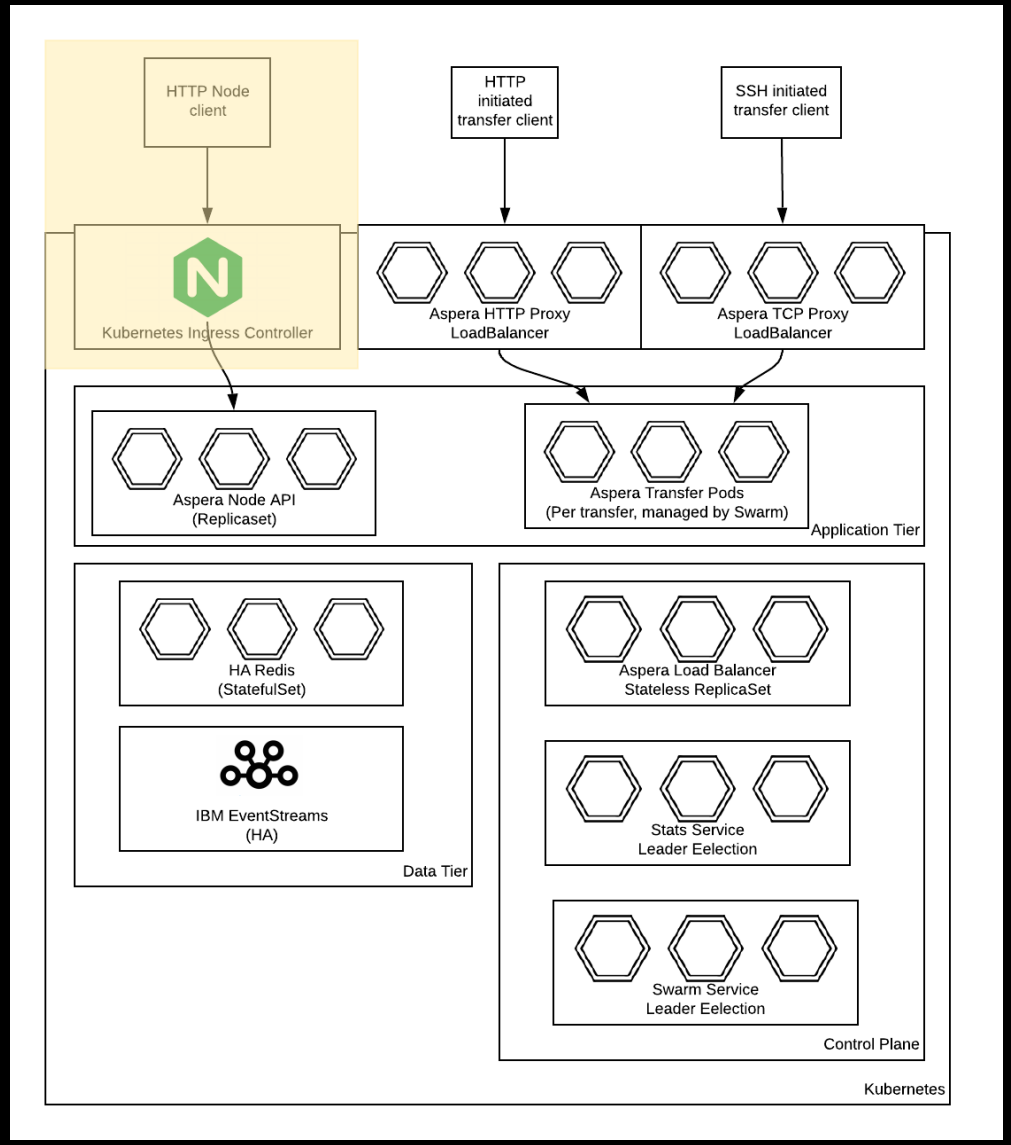
Understanding High Availability for the Cloud Integration Platform involves first understanding the overall architecture and the High availability considerations for the container platform. Based on this we then explore HA for each Integration service.

Agenda:

- Overview of Cloud Integration Platform Architecture.
- Kubernetes and IBM Cloud Private High Availability Concerns.
- Deployment considerations
- MQ High Availability
- Event Streams High Availability
- API Connect High Availability
- App Connect High Availability
- Aspera High Availability

Aspera HSTS

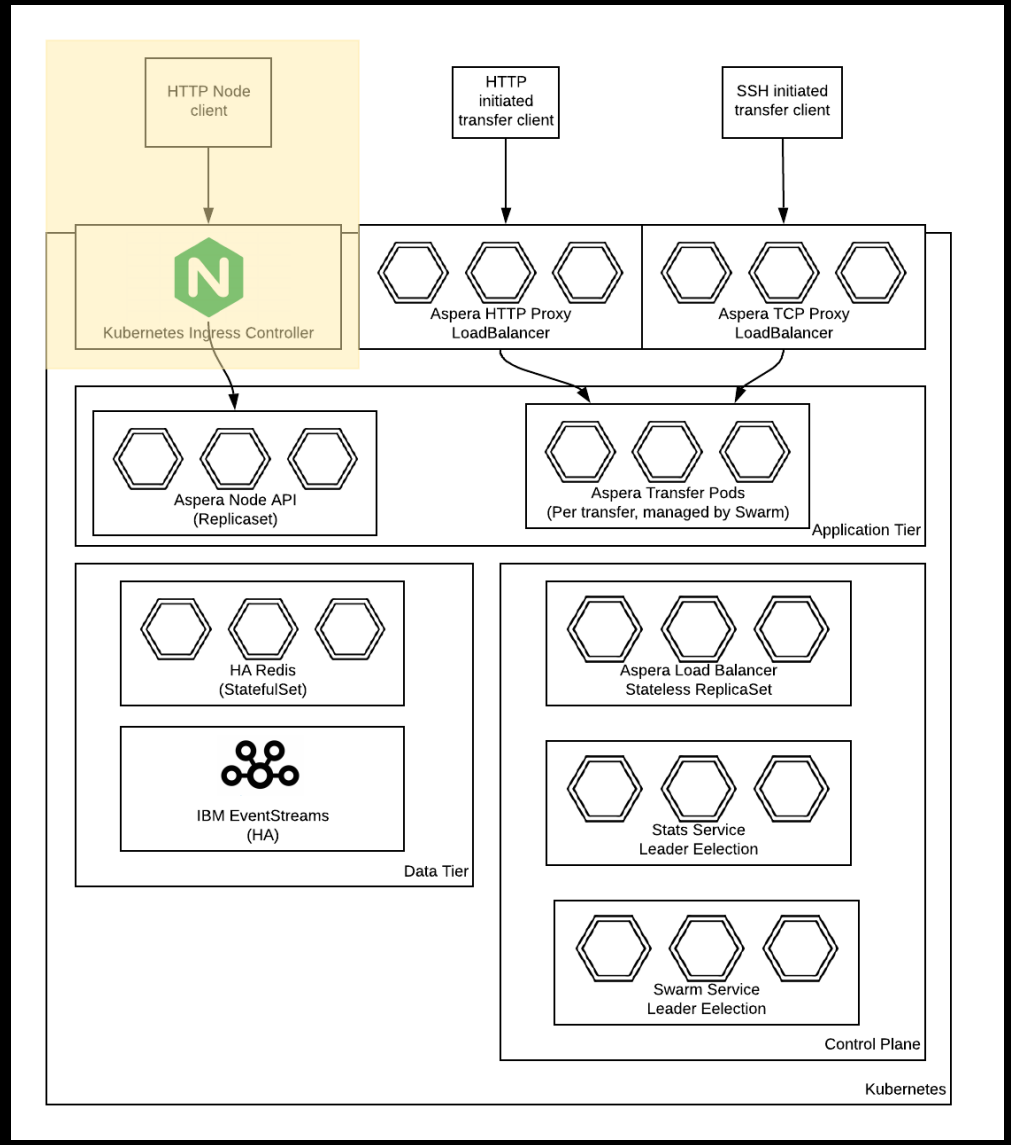
- Aspera HSTS has an architecture that can be deployed Highly Available.
- Aspera HSTS in CIP GA code cannot be deployed as Highly available directly from the helm chart.
- The deployment can be modified to make it highly available.



Aspera HSTS – Routing HA

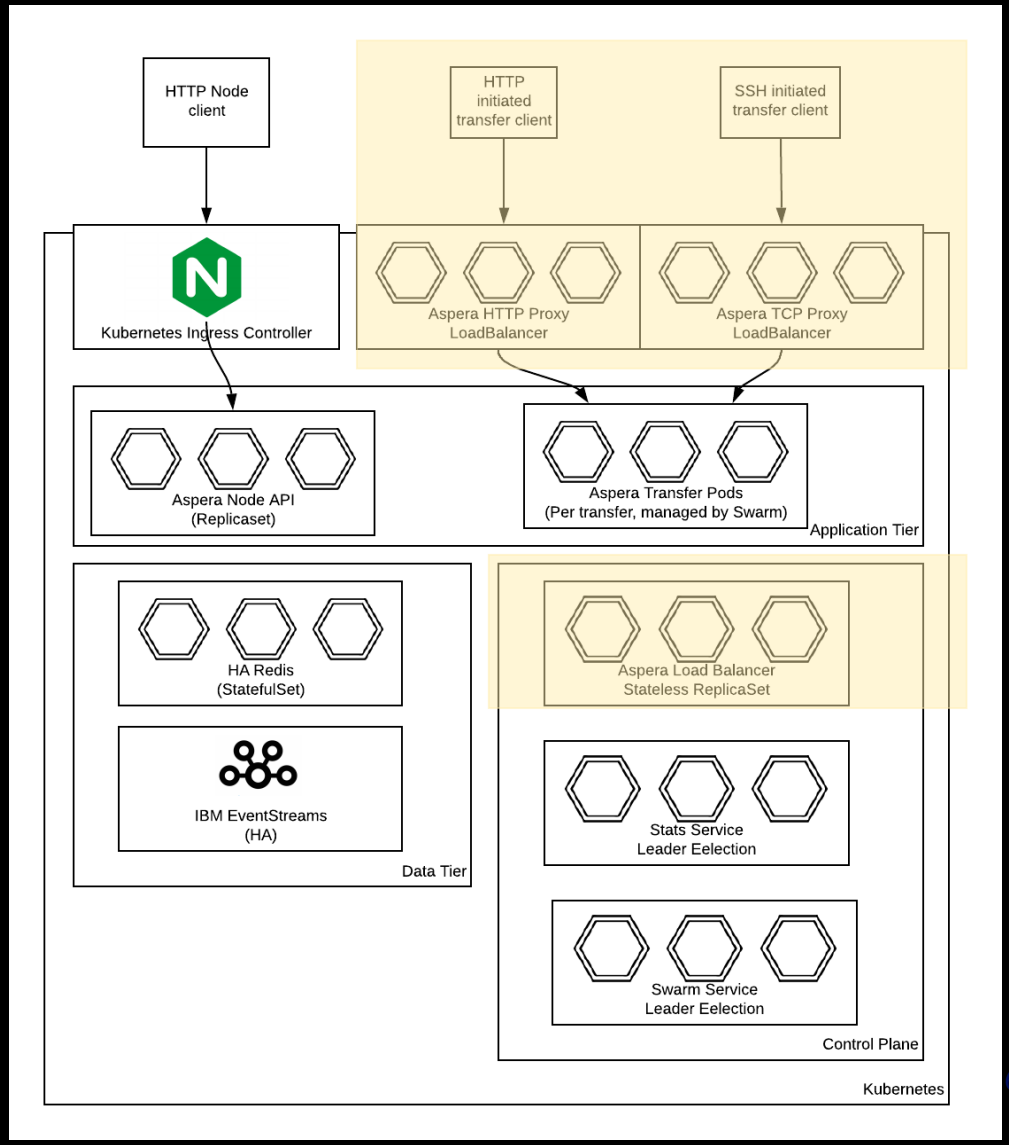
Aspera HSTS serves two types of traffic

- HTTPS API traffic
  - This traffic is routed via ICP's Ingress Controller, with High Availability



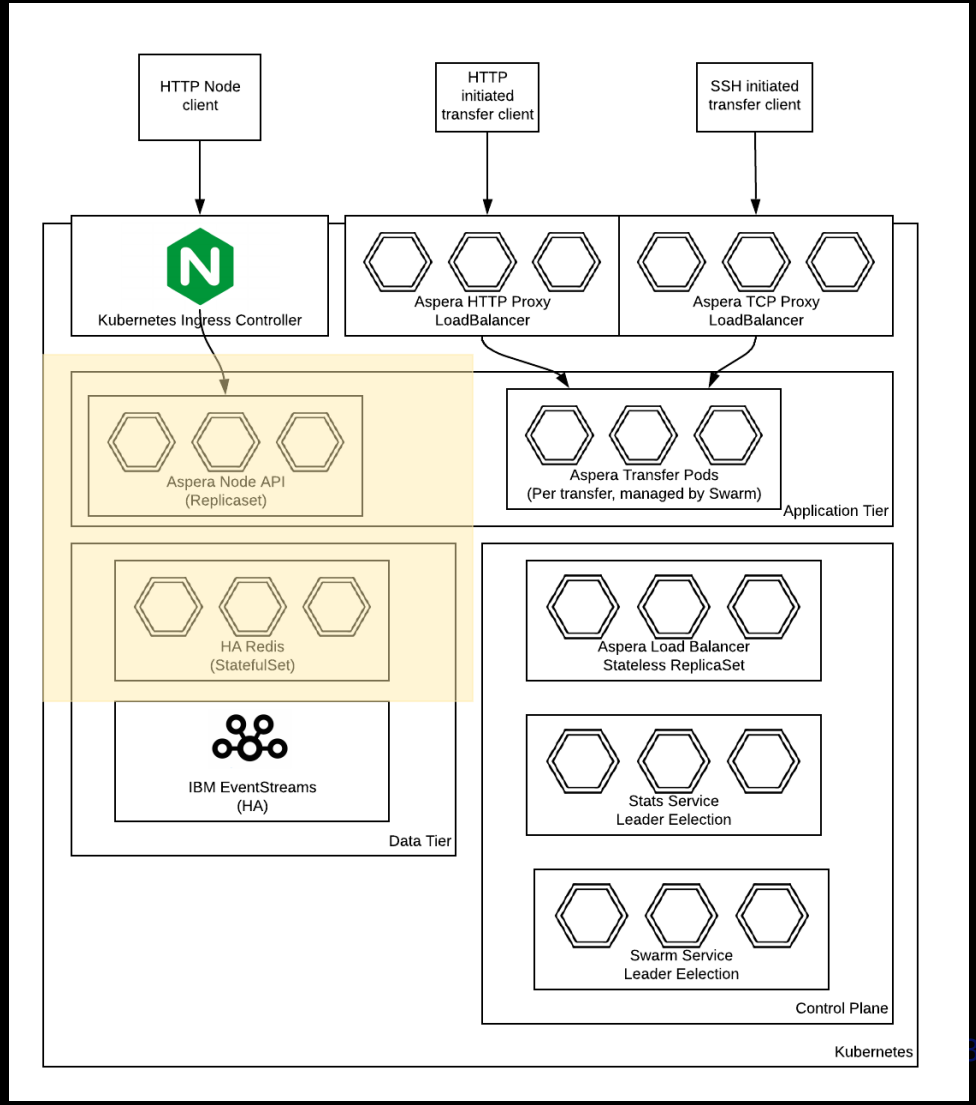
Aspera HSTS – Routing HA

- FASP Transfer traffic
  - This traffic is routed by custom proxies built by Aspera
  - These can be scaled horizontally as a ReplicaSet and are redundantly available
  - Proxies co-ordinate with Load Balancer to find available transfer pods.
  - Load Balancer is stateless and can be as ReplicaSet



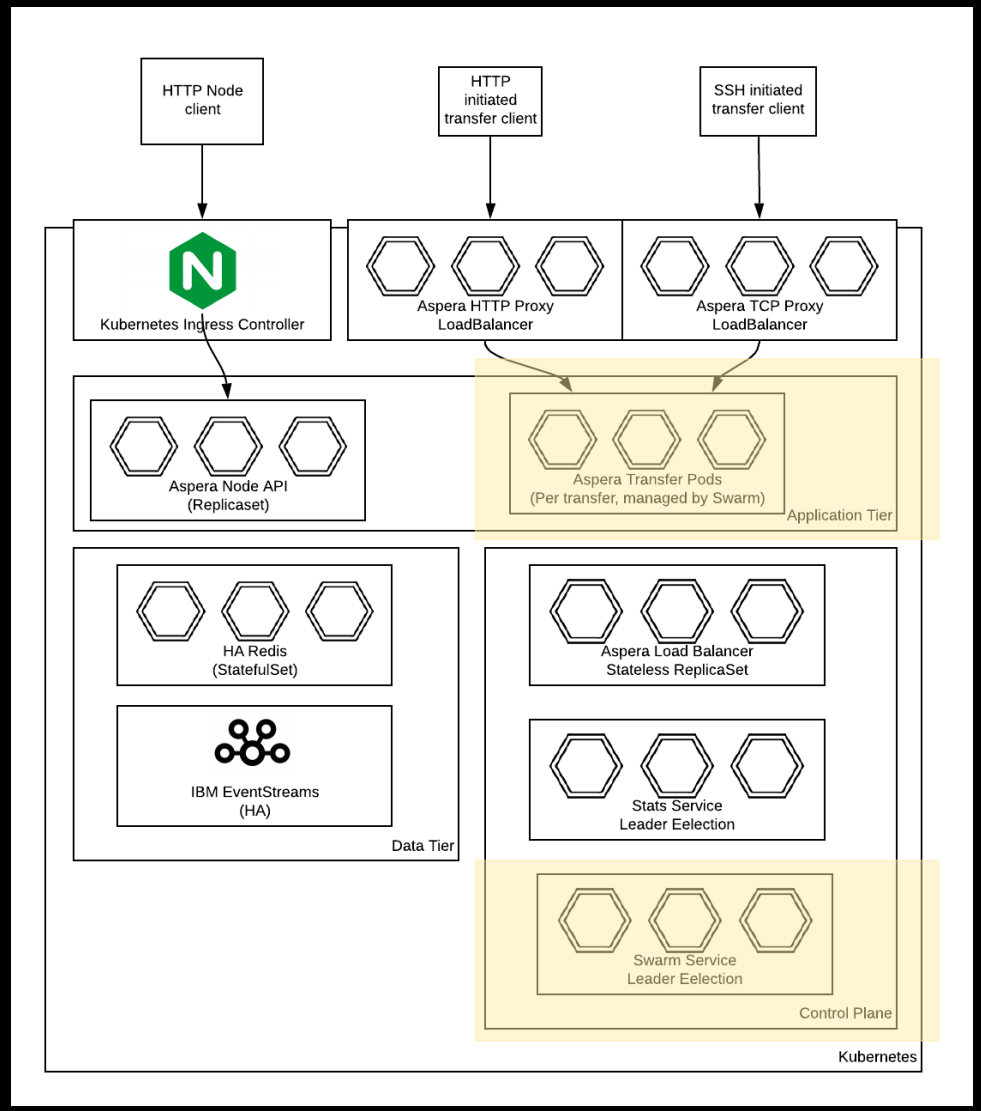
Aspera HSTS – Application HA

Aspera Node API is a stateless HTTP server backed by an HA Redis Stateful Set



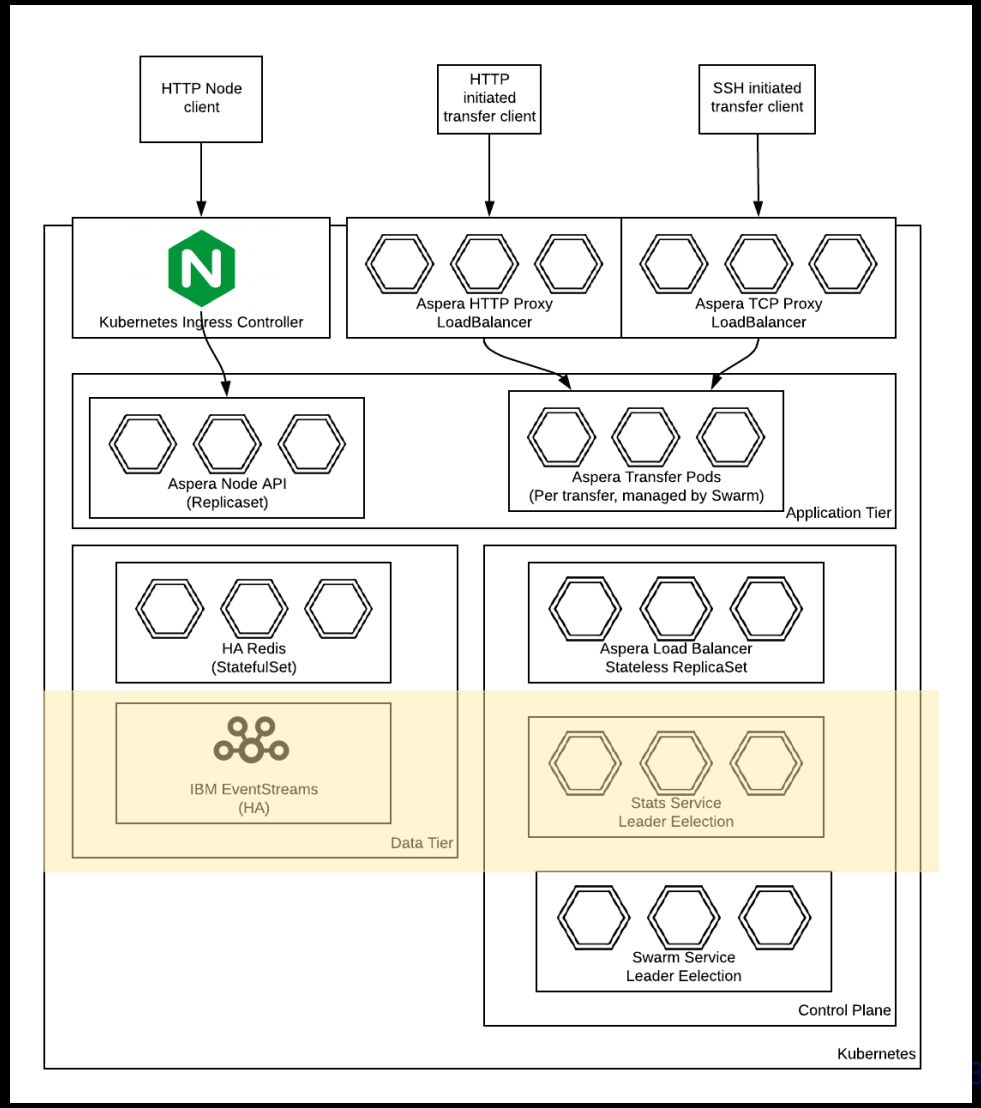
Aspera HSTS – Application HA

- Aspera Transfer Pods are each dedicated to a single transfer session
- Swarm Service manages Transfer Pods– auto-scaling as needed and ensuring free pods are available at all times
- If the Swarm Service dies, no existing traffic will be impacted, but no new transfer pods will be scheduled until it is restarted
- Aspera Swarm Service is currently deployed as a single instance
- Swarm Service will be adding leader election as part of L2 Certification



Aspera HSTS – Control Plane

- Stats Service consumes transfer events from Kafka and publishes pod availability for consumption by Aspera Load Balancer
- Stats Service is currently a single instance, but will be updated to use Leader Election as part of L2 Certification
- If Stats Service fails, load-balancing will be based on stale information until Stats Service restarts
- Kafka is provided by IBM Event Streams, which is an HA service



Thank You

Backup

Rule Of 3 and Quorum

- "Rule of 3"
  - HA Discussion Typically Is in Context of Replicas/Instances of Size 2
  - With *Only* 2 of "Everything"
    - An Outage (Planned or Unplanned) Reduces Capacity by 50 %
    - *Is No Longer Fault Tolerant*
  - Also a consideration for Quorum
- Quorum
  - In distributed computing minimum number of participants (servers) to insure consistency
  - Requires Majority of Voting Members (n/2 + 1)
  - Used for Management Actions, transaction processing, etc.
  - Peers monitor each other to see if one has crashed
    - Broken communications appears to be the same as a crash
    - All you really know is that you can't hear anything from them.
  - 3 (or more) required to avoid "split brain" in the event of a network outage

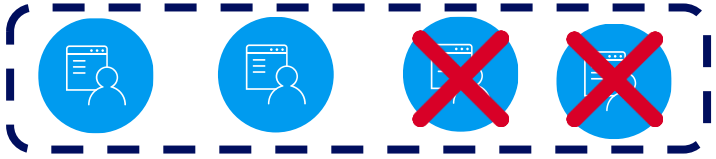
Slide shamelessly stolen from "IBM Cloud Private High Availability and Disaster Recovery" by Tom Alcott STSM alcott@us.ibm.com

High Availability: Quorum

Cluster has Quorum

Cluster does not have Quorum

Cluster has Quorum



Cluster does not have Quorum

Cluster has Quorum

Cluster has Quorum

Cluster does not have Quorum

Odd numbers are better for Quorum!

- An odd number of voting members can reduce the chance of tied elections where half the members vote for a different (but equally eligible) candidate to become primary, or where a network split creates several partitions without a majority of voting members.

- An even number of voting replica set members does **not** improve fault tolerance over a similar replica set with one less voting member.

Slide shamelessly stolen from “IBM Cloud Private High Availability and Disaster Recovery” by Tom Alcott STSM alcott@us.ibm.com

Quorum Implications- IBM Cloud Private

- **ICP quorum implementation**
- **etcd**
  - etcd is a distributed key-value store designed to reliably and quickly preserve and provide access to critical data. It enables reliable distributed coordination through distributed locking, leader elections, and write barriers. An etcd cluster is intended for high availability and permanent data storage and retrieval.

    https://github.com/coreos/etcd/blob/master/Documentation/docs.md

- **ICP Technology Behavior if Quorum Lost**
  - When you lose quorum etcd goes into a read only state where it can respond with data, but no new actions can take place since it will be unable to decide if the action is allowed.

- **HA and DR for etcd**
  - etcd data stored in /var/lib/etcd/ directory on Master node

https://kubernetes.io/docs/tasks/administer-cluster/configure-upgrade-etcd/

https://blog.containership.io/etcd

https://www.ibm.com/support/knowledgecenter/en/SSBS6K_2.1.0.3/troubleshoot/replace_master_node.html

Slide shamelessly stolen from “IBM Cloud Private High Availability and Disaster Recovery” by Tom Alcott STSM alcott@us.ibm.com