



salesforce

Istio Advanced Usecases

Rama Chavali & Devesh Kandpal

rama.rao@salesforce.com

devesh.kandpal@salesforce.com

Agenda - Salesforce Service Mesh Adoption



Our Istio Adoption

- Salesforce Istio Adoption
- Current phase
 - Multicluster Adoption
 - Advanced Usecases
 - HBase - Multicluster (Datastore in Remote cluster, Clients in Primary)
 - Cassandra (Clients in one cluster, Cassandra nodes in different clusters)
 - AWS Managed Services like Elastic Cache, Postgres Database etc.
- Demo
- Next Efforts



Istio Adoption Phases

Early Adoption

- Http/gRPC services
- TCP Services
- Focus on Ease of Adoption

Advanced Usecases

- Multicluster
- Several complex OSS stacks like HBase, Elastic Search, Cassandra....
- AWS Managed Services like Elastic Cache, Postgres ...

Scale and HA

- High Availability
- Scale
- Larger Meshes

Multicluster Adoption



- Driving usecases
 - Datastores in Isolated clusters
 - Handle service growth
 - High Availability
- Primary - Remote Model
- Inhouse built software to generate config in Primary Cluster
 - Multicluster aware
 - Have support for remote cluster initialization
 - Proper resource cleanup
- Crosscluster discovery via Istio DNS
- Automated remote cluster deployments via spinnaker pipelines



Advanced Usecases

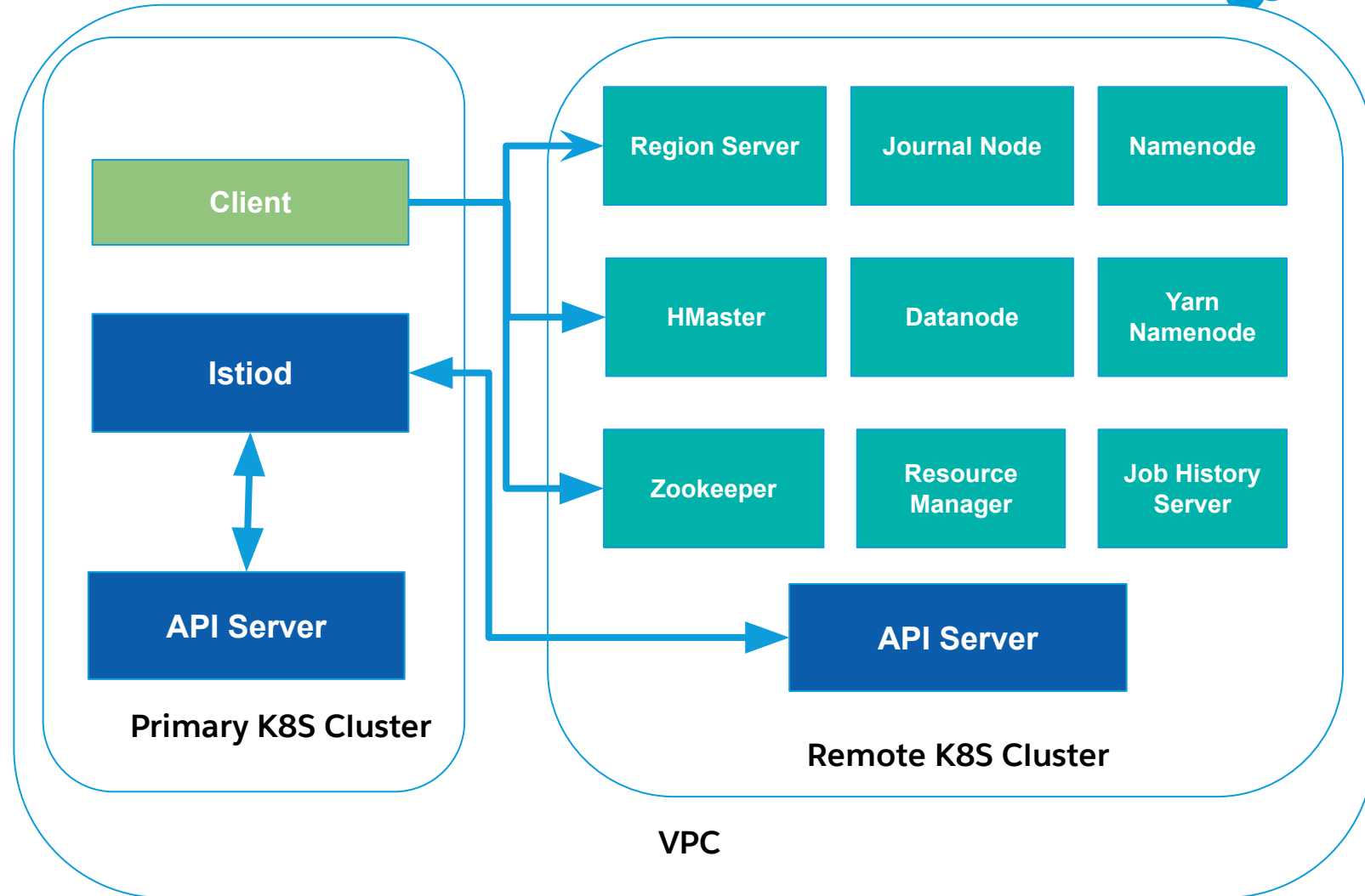


- HBase
 - Uses Multicluster deployment
 - Clients in one clusters and datastore is in another cluster
 - Uses advanced features like TPROXY
- Cassandra
 - Uses Multicluster deployment
 - Clients in one cluster, datanodes replicated across clusters
 - Support for migrating non-mesh Cassandra instances to mesh Cassandra instances
- AWS Managed Services
 - Uses Service Entry model created by Service Owners
 - Inhouse application to generate opinionated Istio configuration
 - Uses [DNS_AUTO_ALLOCATE](#) to support multiple services using same TCP port.

HBase



- Hadoop ecosystem runs in remote k8s cluster while clients may run in primary k8s cluster.
- The hadoop components are all TCP statefulsets with headless service.
- With Istio DNS, clients in primary cluster use *.svc.cluster.local to talk to statefulsets in remote k8s cluster.
- Resolved outgoing calls are matched to podIP_Port outbound envoy listeners, which then points to an originalDST cluster that is tied to a tcp_proxy network filter
- TPROXY interception mode is used by Zookeeper and Namenode for maintaining original source IP.
- Demo



Cassandra



- Multi-node Cassandra deployed as k8s statefulset + headless k8s service
- Clients may reside in different k8s cluster from where Cassandra runs but within same network boundary
- Individual nodes of cassandra are accessed via `<sts-node>.<k8s-service-name>.<namespace>.svc.cluster.local`
- Istio DNS resolves above address on basis of name table entries populated by Istiod
- Datanode sync between Mesh and Non Mesh nodes during migration
- Mesh Cassandra should allow both Mesh and Non Mesh clients on same port
- Demo
 - Cassandra on Mesh
 - Data Migration usecase

AWS Managed Services



- For AWS managed services such as ElastiCache, service owners create a Service Entry with additional configuration as Annotations.
- Our tooling generates Istio configuration like Virtual Service, Destination Rule etc..
- They mostly use Simple TLS
- This is completely self served with out any Mesh Admin involvement
- Demo

Next Phase



- High Availability
 - Withstand primary cluster failures
- Scale - Better support for larger meshes
 - Reduced proxy initialization times
 - Optimized config delivery (Delta Xds)
 - Support more number of proxies per controlplane instance
- Adoption, Adoption & More Adoption





Thank You