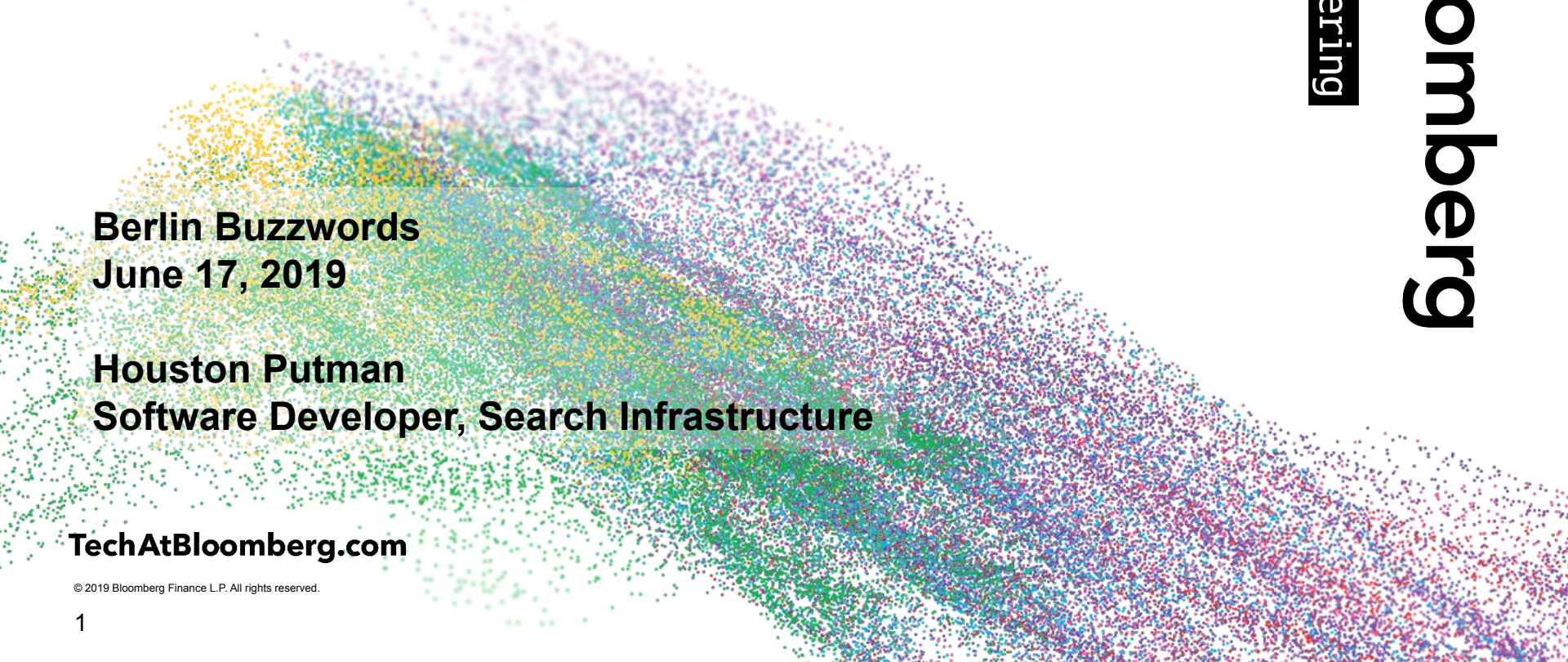
Solr on Kubernetes

Bloomberg

Engineering



Berlin Buzzwords
June 17, 2019

Houston Putman
Software Developer, Search Infrastructure

TechAtBloomberg.com

Bloomberg

- Largest provider of financial news and information
- Our strength is quickly and accurately delivering data, news and analytics
- Creating high performance and accurate information retrieval systems is core to our strength

TechAtBloomberg.com

Bloomberg

Engineering

Search Infrastructure at Bloomberg

- Hundreds of search applications & ZK Ensembles
  - Diverse use cases and scale
  - Displaced other technologies
- 10s of billions documents
- 100s of millions new documents daily
- 1,000s of servers
- 10,000s of Solr instances
- 10,000s of queries per second
- Critical to Bloomberg and the global financial markets

TechAtBloomberg.com

Bloomberg

Engineering

Agenda

- Managing Solr Cloud
- Kubernetes Intro
- Stateful Services
- Beyond a Single Cluster
- Solr Cloud Operator

TechAtBloomberg.com

Bloomberg

Engineering

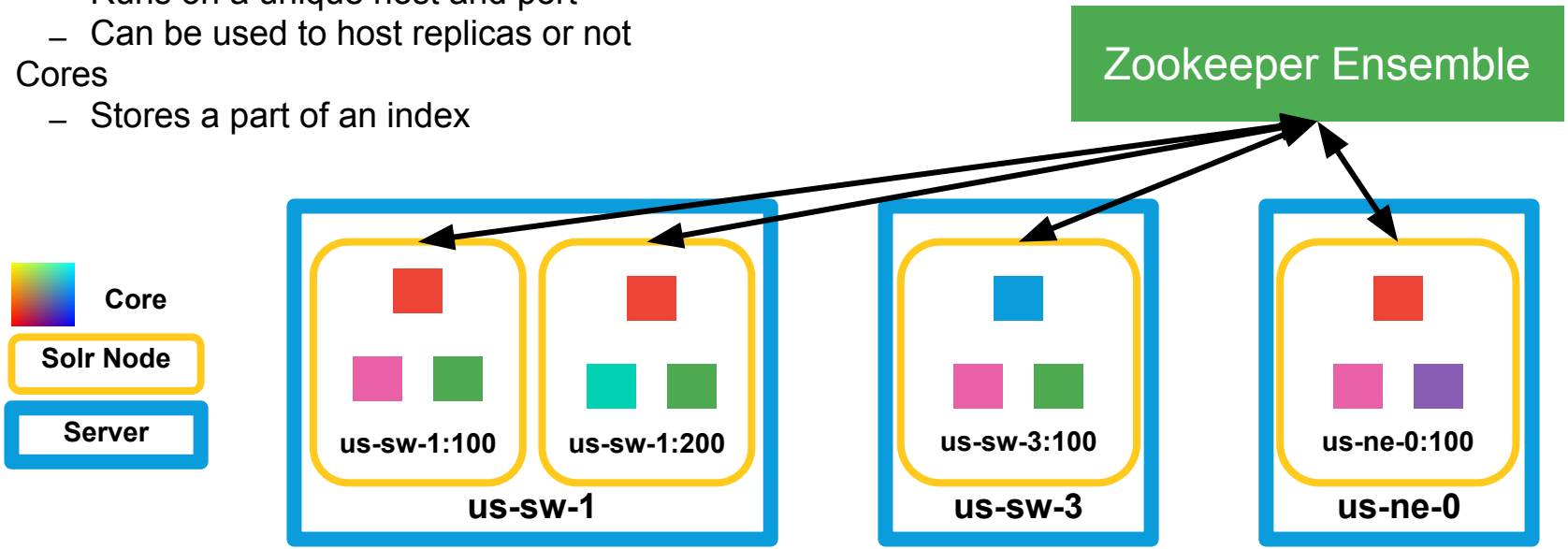
Managing Solr Cloud

Solr Cloud manages data through two separate topologies

- Physical
  - Where data physically lives
  - The processes that are running on servers
- Logical
  - How data is divided and grouped
  - The schema that defines a grouping of data

Physical Topology

- Nodes
  - A process that runs Solr Cloud, connected to Zookeeper at a certain path
  - Runs on a unique host and port
  - Can be used to host replicas or not
- Cores
  - Stores a part of an index

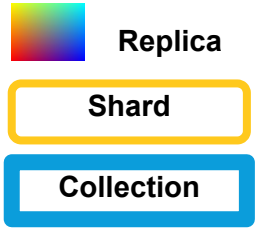


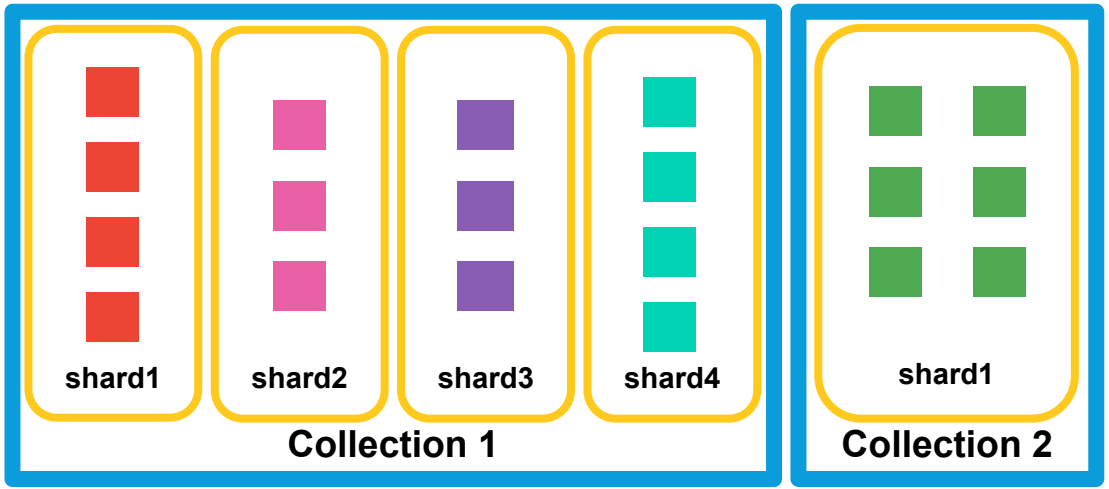
Bloomberg

Engineering

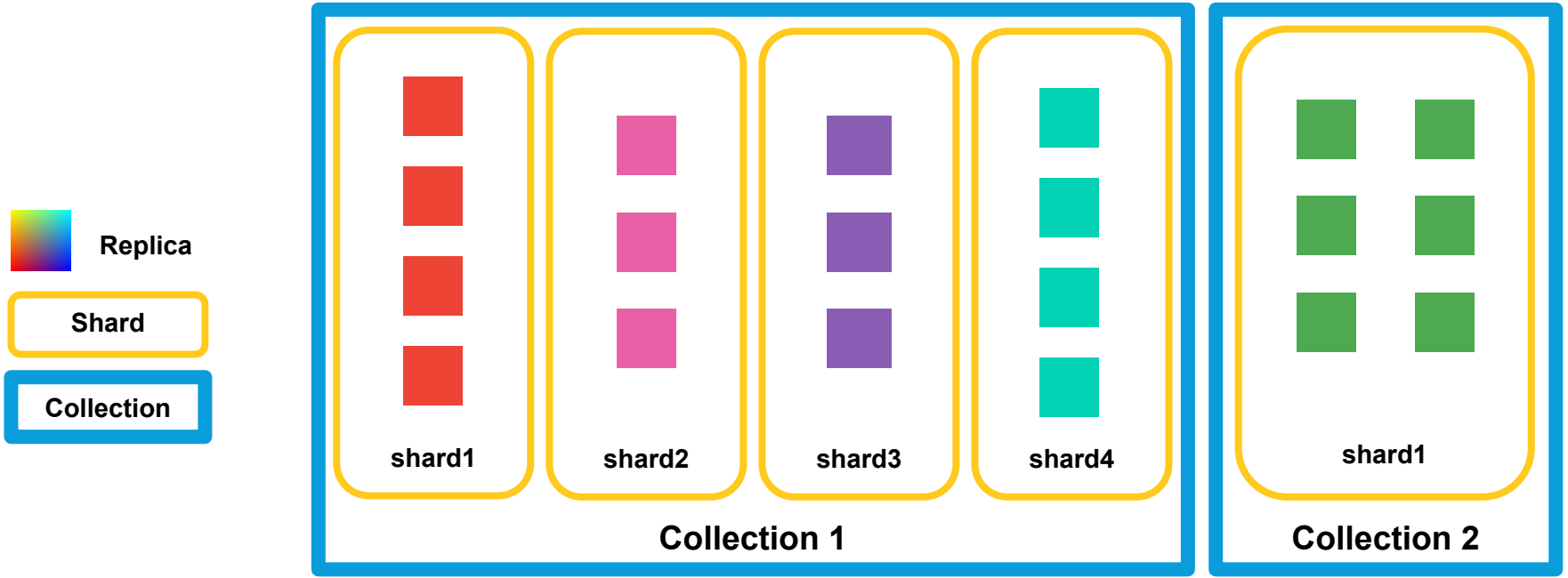
Logical Topology

- Collections
  - A grouping of data, following a common schema
- Shards
  - Solr collections are made up of 1 or more shards
  - Shards are logical splits of data
- Replicas
  - Stores the data of a shard
  - Houses a core (the index)





Logical Topology

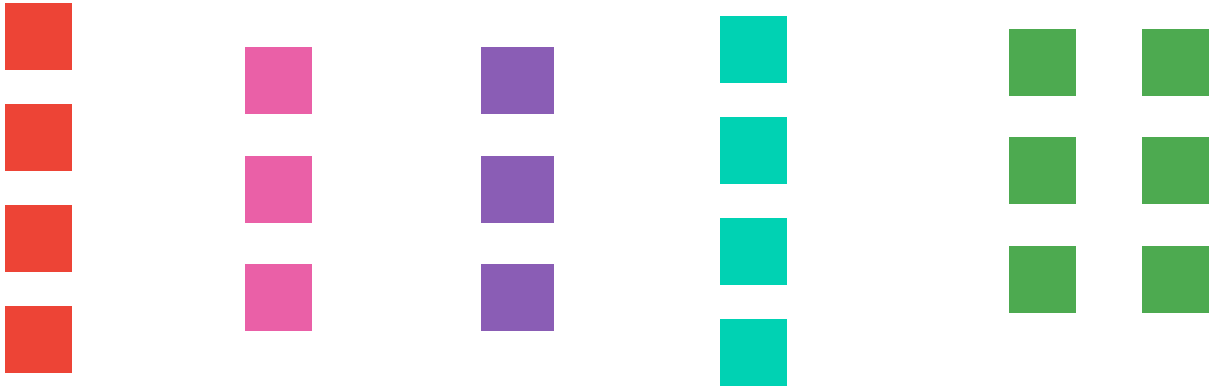


TechAtBloomberg.com

Bloomberg

Engineering

Replicas & Cores

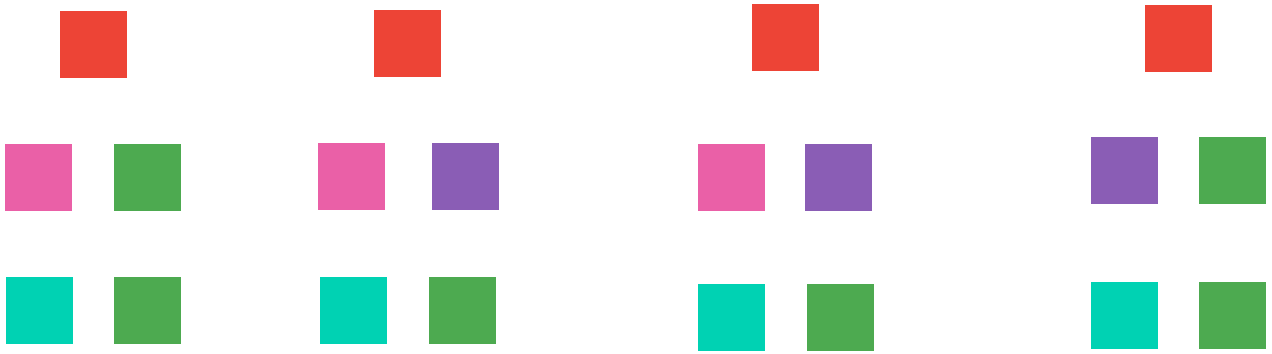


TechAtBloomberg.com

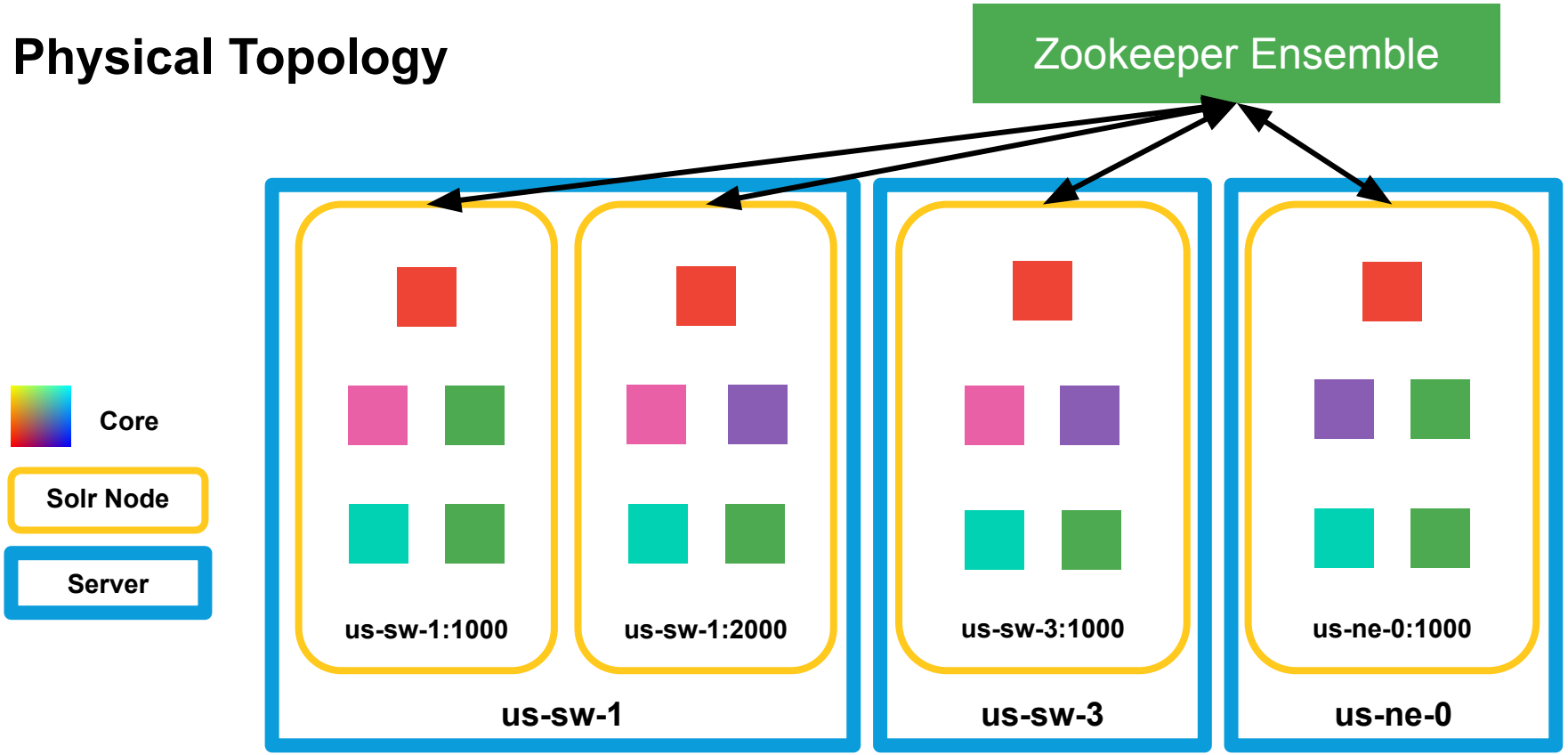
Bloomberg

Engineering

Replicas & Cores



Physical Topology



Bloomberg

Engineering

Managing Topologies

	Logical	Physical

Implicit	• Auto Scaling
	  ○ Shards
	  ○ Replicas
	• Time Routed Aliases
	  ○ Collections
		• Auto Scaling
		  ○ Shuffle replicas
		?

Explicit	• Collections API
	  ○ CRUD
	  ■ Collections
	  ■ Replicas
	  ○ Split/Add Shards
		• Collections API
		  ○ Migrate node
		?

What is Kubernetes?

- Kubernetes is an open source platform for managing containerized services
- Its ecosystem is large and rapidly growing
  - All major cloud providers support it
- Applications are run via declarative configuration
  - Automated processes are also supported

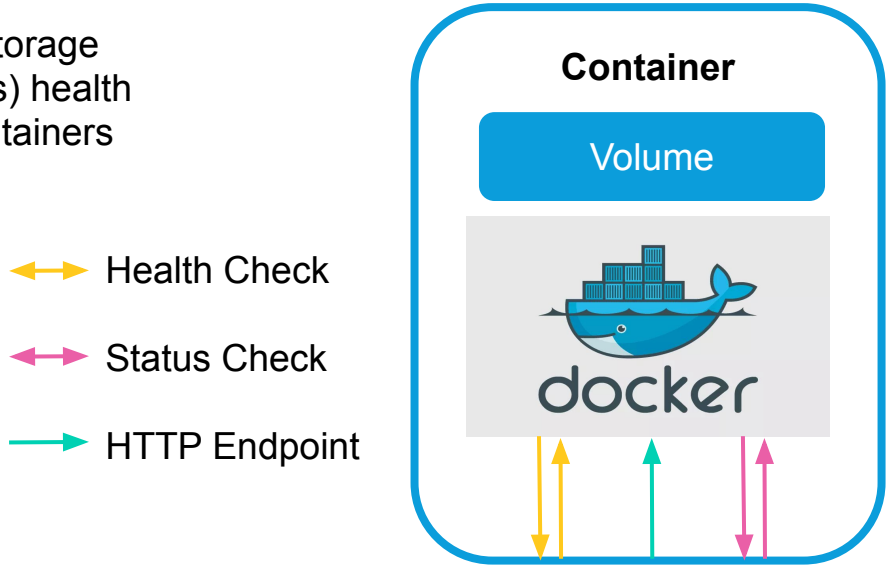
Establishing Terminology

	Solr	Kubernetes
Node	Solr Cloud Process running, connected to Zookeeper	A server or virtual machine running within the Kube cluster
Replica	One copy of a shard's data	One instantiation of a pod specification

Brief Kubernetes Intro

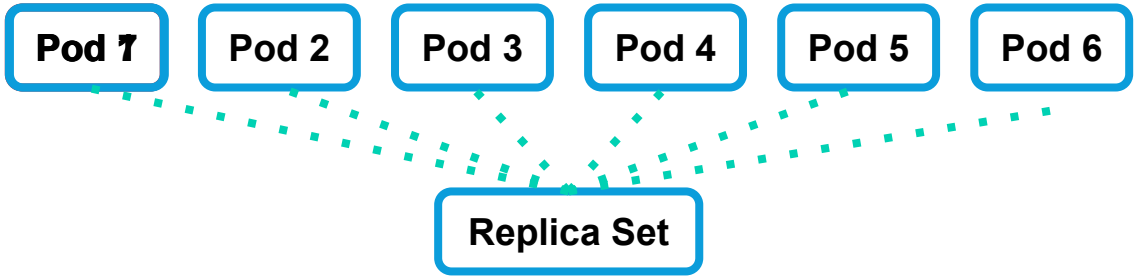
Running Processes

- Containers
  - Way of isolating external variables
  - Can protect functionality and restrict what is run
- Pods
  - Consistent network and storage
  - Keeps track of container(s) health
  - Manages volumes for containers



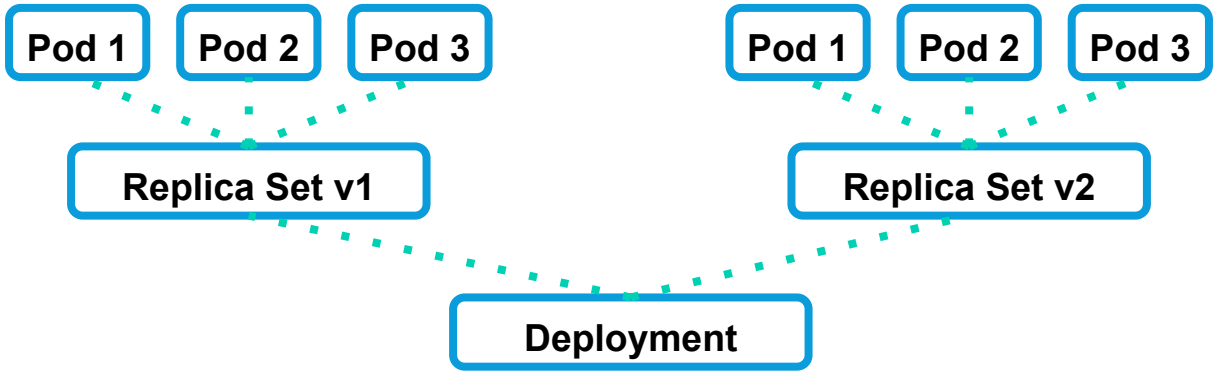
Resiliency

- Replica Sets
  - Makes sure that a set of *n* instances of a pod spec are running
  - If a pod dies, the replica set controller will schedule another pod to run



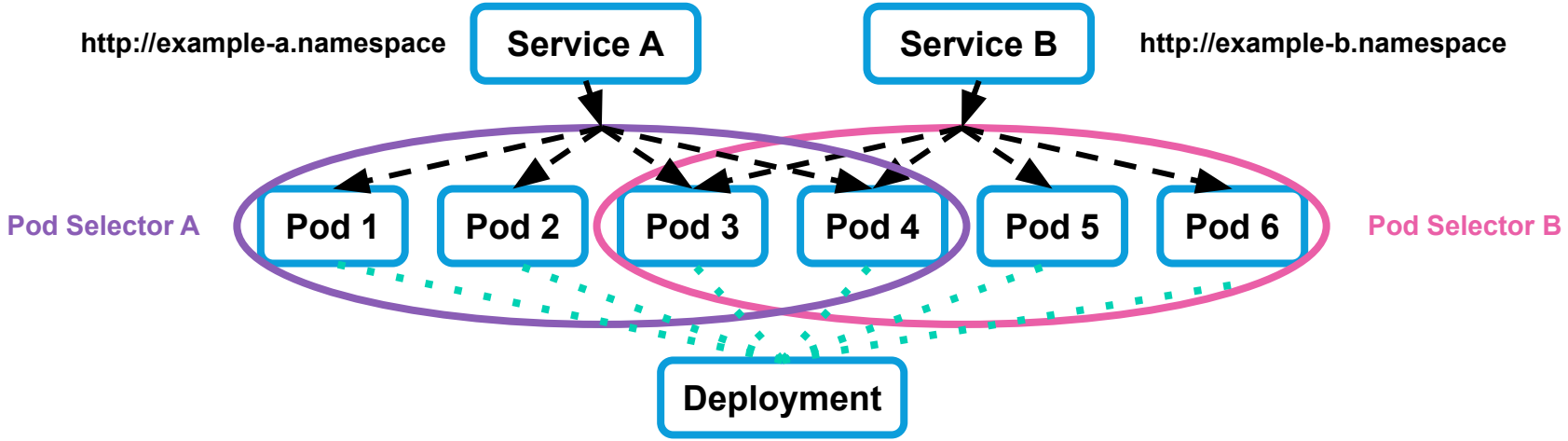
Safe Upgrades

- Deployments
  - Manages Replica Sets
  - If the Pod Spec changes (e.g., A container image version change)
    - A new Replica Set will be created
    - Pods will be started in the new Replica Set as pods are removed from the old Replica Set
  - Allows for seamless updates, without downtime in your service



Networking

- Services
  - Pods are not addressable by themselves
  - Services enable routing of requests to sets of pods

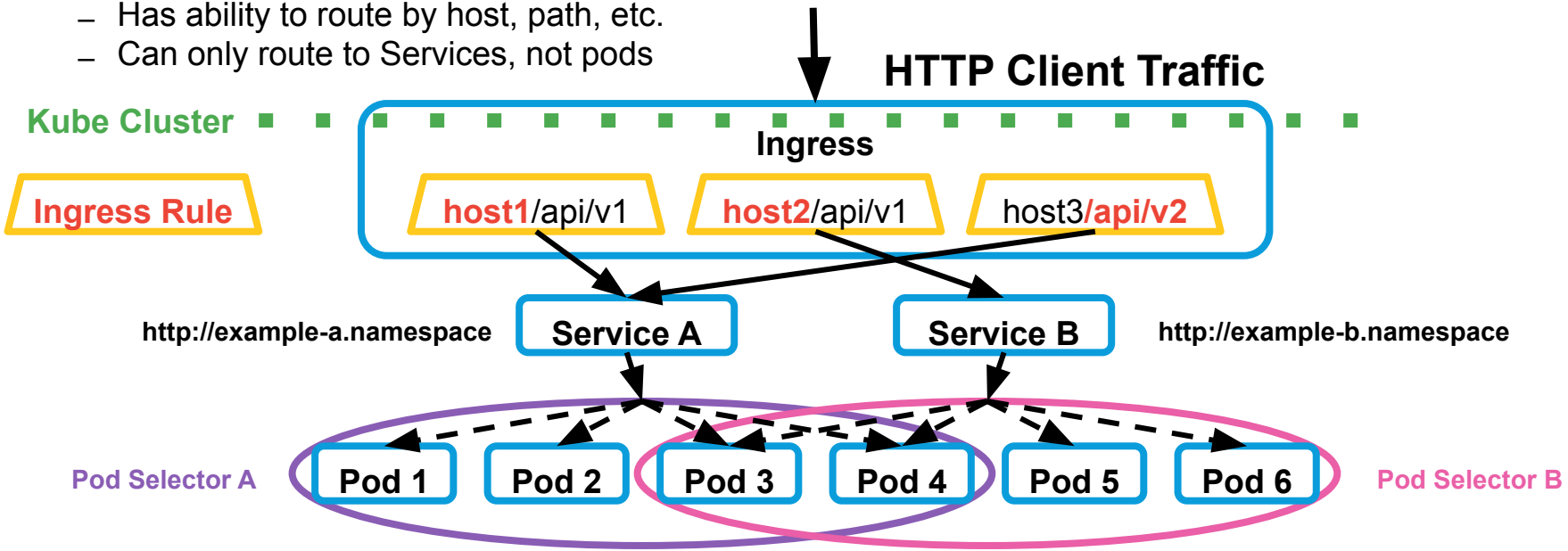


Bloomberg

Engineering

External Addressability

- Ingresses
  - Can route HTTP(s) traffic from outside the Kube cluster inside
  - Has ability to route by host, path, etc.
  - Can only route to Services, not pods



TechAtBloomberg.com

Bloomberg

Engineering

Can we build Solr with these pieces?

- Solr Nodes (Pods) have data unique to them
  - A name & address
  - Solr cores
- The following could break the state of a Solr cloud
  - Solr node renaming
  - Pod data loss
  - Removing Solr nodes

Room for Stateful Improvements

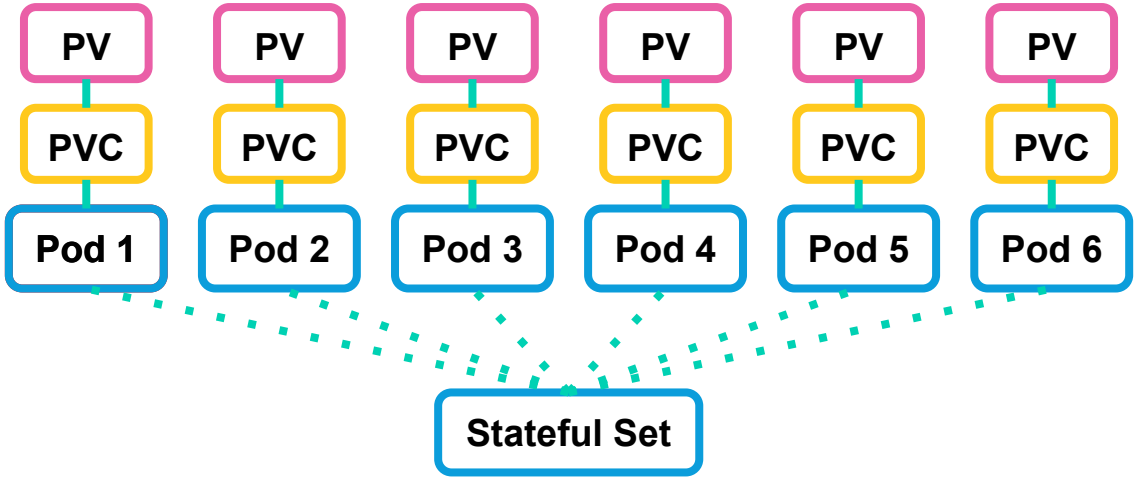
- Standard workflow was designed for stateless applications
  - Solr Prometheus Exporter
- Deployments/Replica Sets have issues with:
  - Data Persistence/Locality
  - Pod identity
  - Pod addressability

Persistent Data

- Persistent Volume
  - A way of storing state in Kubernetes
  - Are disconnected from Pods
  - Can have node affinity
- Persistent Volume Claim
  - A set of requirements that a Pod has for the volume it receives
- Types
  - Local Storage, Azure, AWS, GCE, NFS, etc.

Stateful Sets

- Stateful Sets
- Pods are now iteratively named
  - (name)-0, (name)-1, …
- Persistent Volume Claims can be set up to work natively



Bloomberg

Engineering

Headless Services

- Gives a hostname for each pod in a Stateful Set
  - <pod-name>.<service-name>.<namespace>
- Important for Solr
  - Each Solr node is defined by its unique URL
- Limitations
  - Does not work with Ingresses or load balancing
  - Can only be used within a Kube cluster

Why run beyond a single cluster?

It’s unlikely that you will be able to run all services within a single Kube cluster

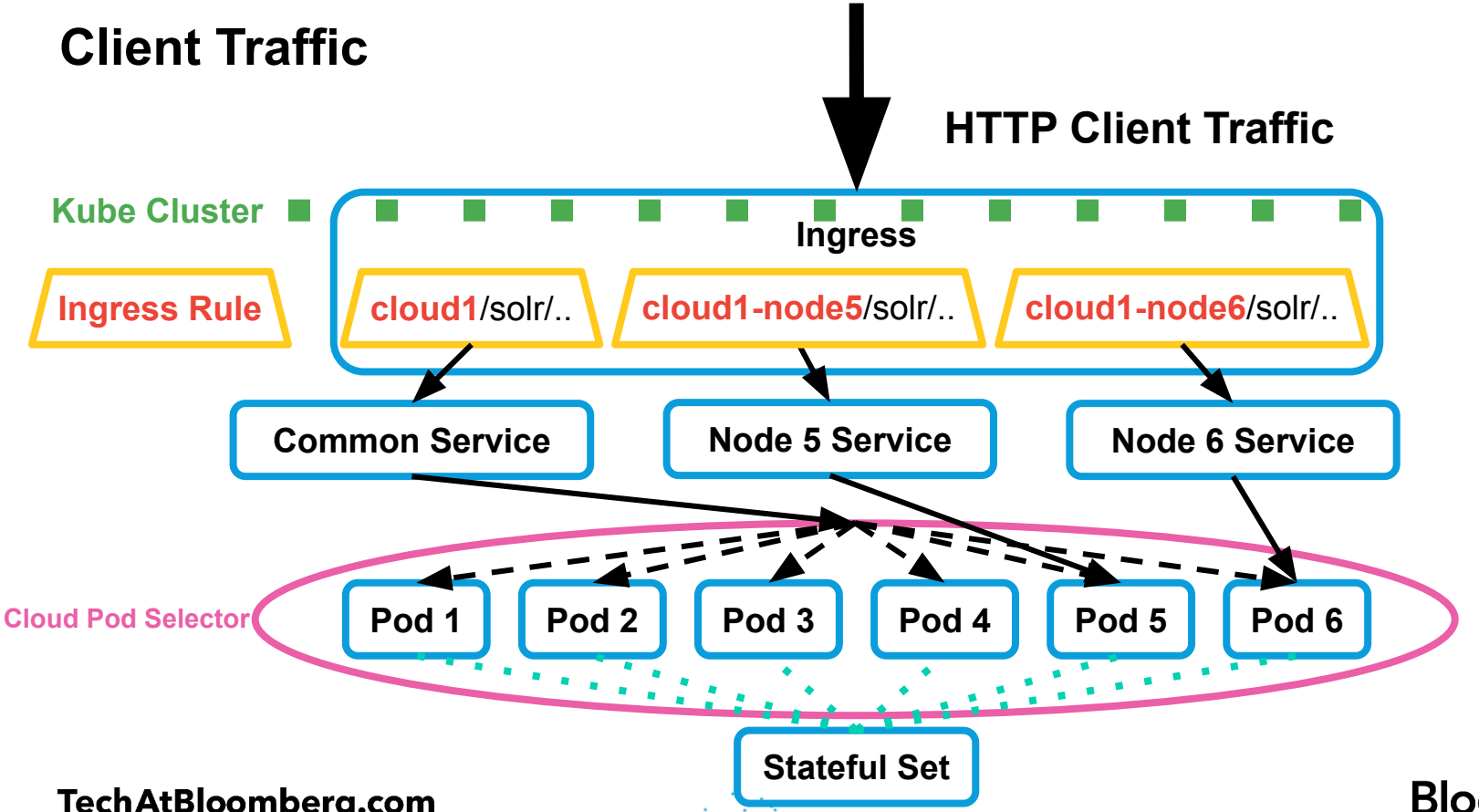
- Running one Solr Cloud in multiple Kube clusters
  - Staged rollout of Kubernetes Upgrades
  - Staged rollout of infrastructure pieces
  - Resilience to outages
- Running your applications outside of Kube, or in a different cluster

Node Addressing Solutions

Create a Service for every Solr node

- LoadBalancer
  - Creates an external IP address to route to the Solr Node
  - Requires as many IP addresses as solr nodes
- Ingress
  - Allows for custom path/hostname routing to services
  - The ingress and pod can listen on/advertise the same hostname

Client Traffic

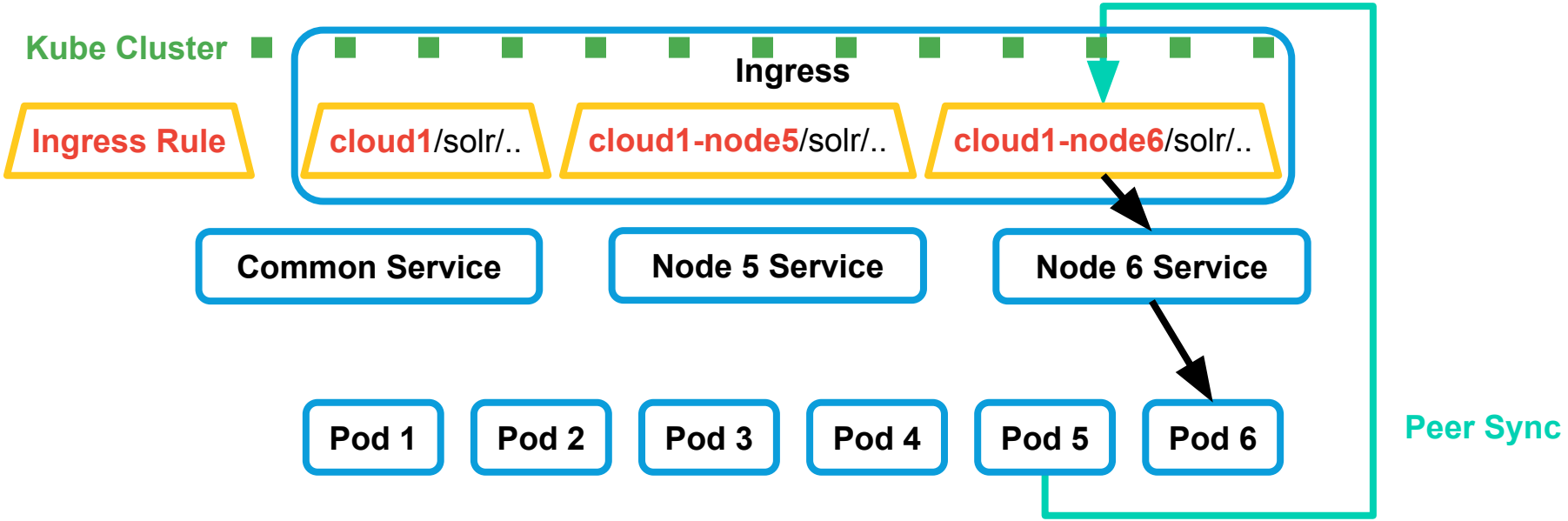


TechAtBloomberg.com

Bloomberg

Engineering

“Internal” Traffic



How we have built a Solr Cloud

```
$ kubectl get all

NAME                                        READY   STATUS             RESTARTS   AGE
pod/example-solrcloud-0                     1/1     Running            7          47h
pod/example-solrcloud-1                     1/1     Running            6          47h
pod/example-solrcloud-2                     1/1     Running            0          47h
pod/example-solrcloud-3                     1/1     Running            6          47h

NAME                                        TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)              AGE
service/example-solrcloud-0                 ClusterIP   ##.###.###.##    <none>        80/TCP               47h
service/example-solrcloud-1                 ClusterIP   ##.###.##.#      <none>        80/TCP               47h
service/example-solrcloud-2                 ClusterIP   ##.###.###.##    <none>        80/TCP               47h
service/example-solrcloud-3                 ClusterIP   ##.###.##.###    <none>        80/TCP               47h
service/example-solrcloud-common            ClusterIP   ##.###.###.###   <none>        80/TCP               47h
service/example-solrcloud-headless          ClusterIP   None             <none>        80/TCP               47h

NAME                                        READY   AGE
statefulset.apps/example-solrcloud          4/4     47h

NAME                                      HOSTS                                                                  PORTS   AGE
ingress.extensions/example-solrcloud-common   default-example-solrcloud.test.domain,default-example-solrcloud-0.test.domain + 3 more...   80      2d2h
```


Can we automate it?

- Custom Resource Definitions (CRDs)
  - Custom objects in Kubernetes
  - Solr, Zookeeper, Kafka, etc.
- Controllers
  - Listens for new/deleted/modified resources of a specific CRD
  - Manipulate other Kubernetes objects
    - Pods, Deployments, Services, Ingresses
- Operators
  - A grouping of controllers
  - E.g., Solr Operator
    - Solr Cloud, Backup, Restore

Solr Cloud Specification

```
Spec:
  Replicas:       4
  Solr Image:
    Repository:   library/solr
    Tag:          8.1.1
  Zookeeper Ref:
    Connection Info:
      Chroot:                     /test/example
      External Connection String: external1.test:2122,external2.test:2122
```

Solr Cloud Status

```
Status:
  External Common Address:  http://default-example-solrcloud.test.domain
  Internal Common Address:  http://example-solrcloud-common.default
  Ready Replicas:           2
  Replicas:                 2
  Solr Nodes:
    External Address:  http://default-example-solrcloud-0.test.domain
    Internal Address:  http://example-solrcloud-0.default.svc.cluster.local
    Name:              example-solrcloud-0
    Ready:             true
    Version:           8.1.0

    External Address:  http://default-example-solrcloud-1.test.domain
    Internal Address:  http://example-solrcloud-1.default.svc.cluster.local
    Name:              example-solrcloud-1
    Ready:             true
    Version:           8.1.1

  Target Version:      8.1.1
  Version:             8.1.0
  Zookeeper Connection Info:
    Chroot:                       /test/example
    External Connection String:  external1.test:2122,external2.test:2122
    Internal Connection String:  external1.test:2122,external2.test:2122
```

Solr Cloud Operator!

- Recently published as open source!
  - https://github.com/bloomberg/solr-operator
- We would love contributions!

Future of Project

- Data Persistence
  - Local Persistent Volumes are still in infancy
  - Remote Storage might not be performant enough for some clients
- Additional Operator Functionality
  - Backup
  - Restore
- Add a deployment of the Prometheus Exporter alongside each cloud

Bloomberg

Engineering

Thank You!

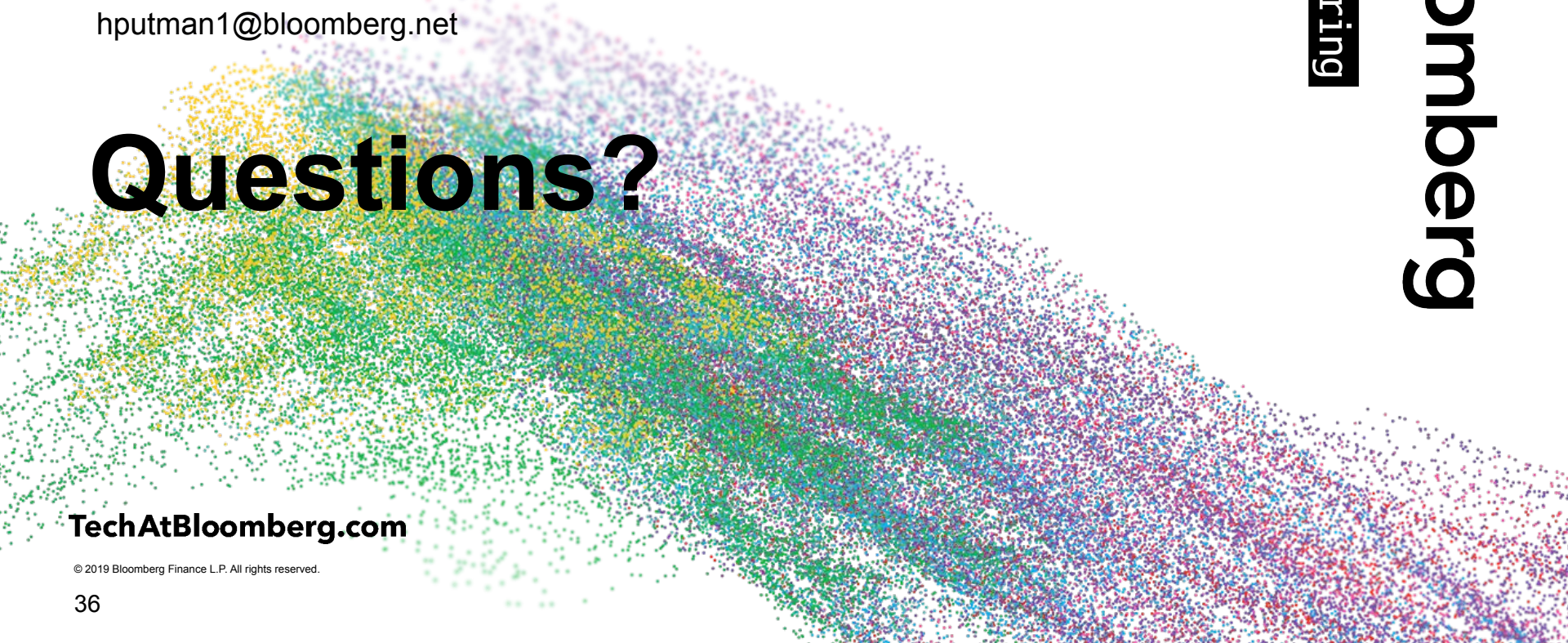
https://www.bloomberg.com/careers

hputman1@bloomberg.net

Questions?

Bloomberg

Engineering



TechAtBloomberg.com