Scaling Solr: From Desktop to Cloud Scale

A mental model

Why this talk?

*This talk will introduce a **specific mental model** for thinking about how to scale Solr from a single node developer centric desktop deploy through the small scale setups such as what the Drupal community would use, through to the largest scale deploys of Solr.*

Why this talk?

This talk will introduce a specific mental model for thinking about how to scale Solr from a single node developer centric desktop deploy through the small scale setups such as what the Drupal community would use, through to the largest scale deploys of Solr.

*This talk will seek to **spark discussion** about this mental model for scaling Solr with a goal to building **community buy in for a more opinionated story** around how to scale Solr.*

Structure

- Some perspectives on deploying Solr..
- Results from the 2024 Survey
- A direction we can take...
- What you can do!

Kubernetes Story

POC Phase

- It’s incredibly easy to spin up a Solr Cluster
  - Try modules
  - Try out custom settings
  - Auto-restarts with safe replica handling
- Make sure its the right technology for you

Make it Production Ready

- Once that you know that Solr is right for you, you need to make it run correctly
- Use the recommended best practices:
  - TLS
  - Authentication/Authorization
  - Pod Spreading
  - SIP-18 - Make all of the Security and config options Kube-native (Not merged)
- DR
  - Pods auto-restart on failure
  - Solr pods can be spread across machines and availability zones automatically

Scaling to meet demand

- The Solr Operator provides autoscaling for Kubernetes
  - SIP-17
- Scaling doesn’t always mean making bigger clouds
  - Clouds can be split so that collections live on separate clouds
  - Using the Solr Operator, Solr Clouds are easily created, managed and deleted
  - More Clouds scales better than bigger Clouds.

Tom’s Story

- Solr is not the quickest with adopting vector search features
- Lucene’s segmented index design could cause performance issues in some vector search use cases
- No user friendly way yet for hybrid search. Wanted: RRF on a distributed index
- Also wanted: internal vector quantization
- Dreaming is free: multi-valued vector fields, that would remove the need for chunking

Christopher’s Story

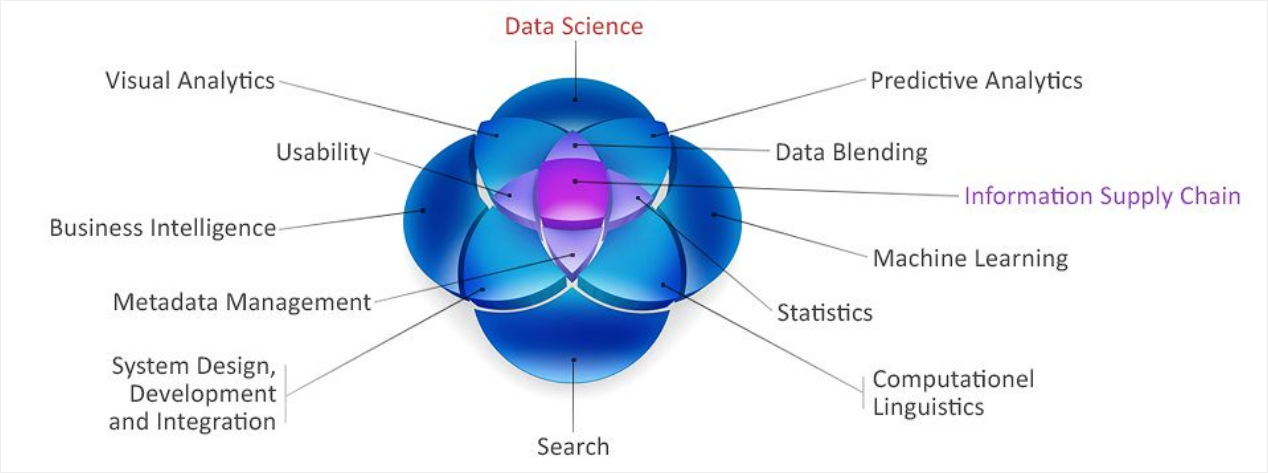
Run’s Solr on the desktop as part of his analytics tool....

“bin/solr start -c” is the default....

^^^^ On Windows! Via Window’s Subsystem for Linux. No “.cmd”!

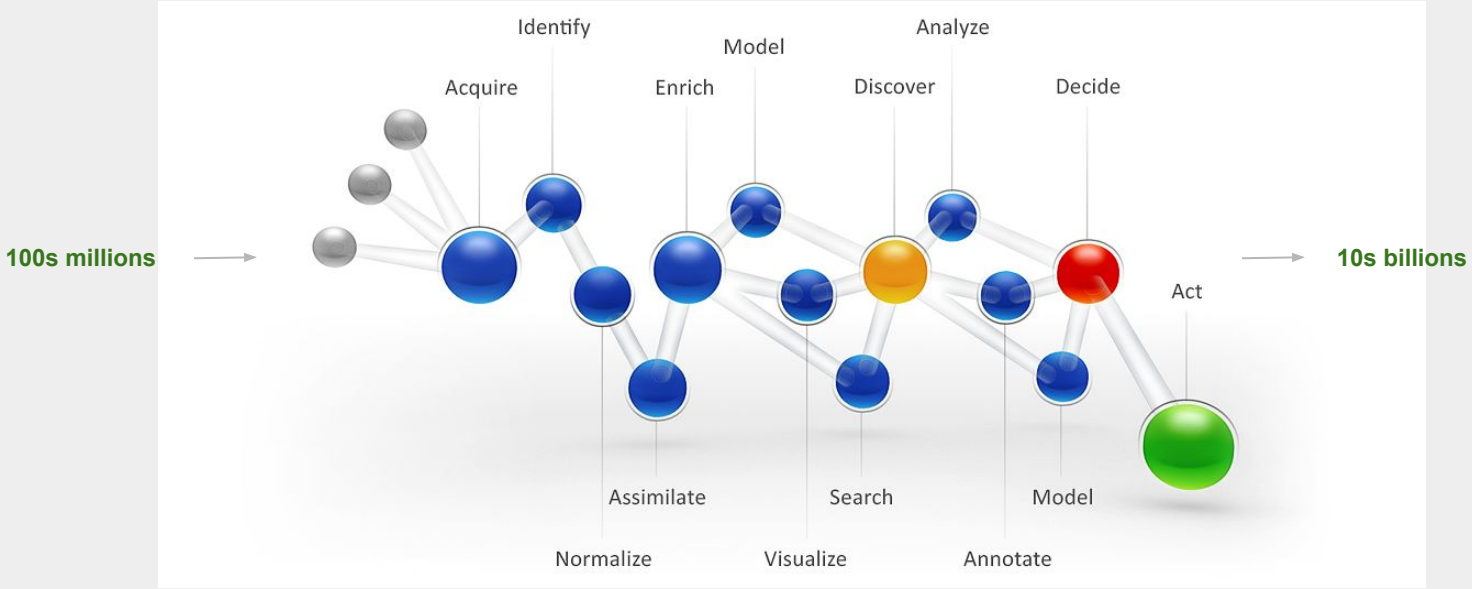
Interdisciplinary

Data Science is interdisciplinary, treating content as a raw material flowing horizontally between usages - blends advanced parsing technology with machine learning and semantic encoding to enable rich interactive visualizations to facilitate discovery.



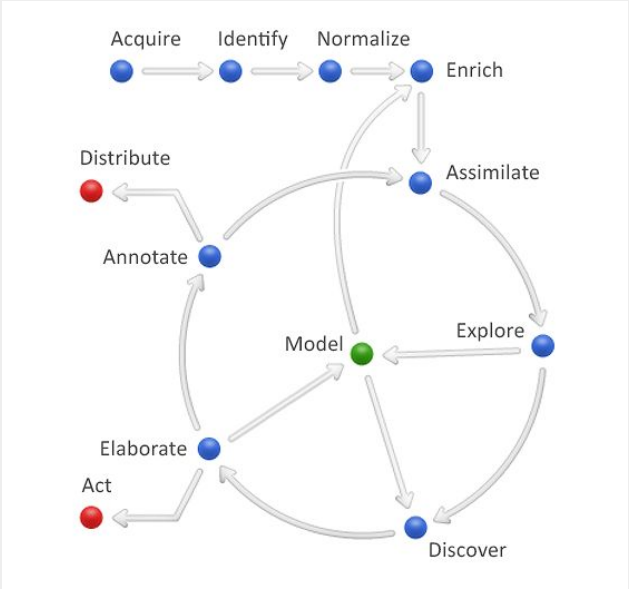
Information Supply Chains

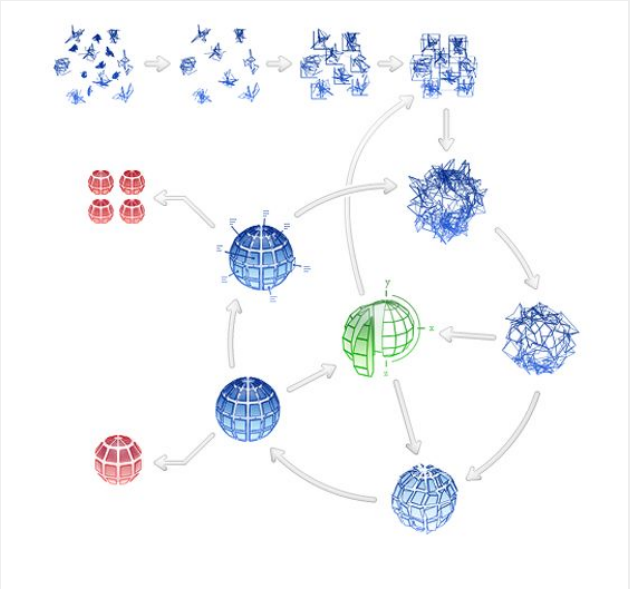
An **information supply chain** (ISC) orientation is based on recognizing the importance of **metadata** throughout the information lifecycle: starting with content at its point of capture, through enrichment and assimilation, all the way to the point of consumption with automated enrichment and rich interactive visual analytics.



Discovery Life Cycle

Capabilities that facilitate discovery in large unstructured and often unreliable data - getting content talking - answering questions that could not previously be asked. Harvesting unstructured content, making it **highly actionable** and **highly discoverable** to achieve **new insights**.





≈ Solr Collection

meta Heuristica

Bloomberg's Story: Major version upgrades

- Minor versions: backwards compatible! 🎉
- Major versions: only support current version + 1
- What if I already upgraded major version once?
- Current solution is to:
  - Create new cloud with latest version
  - Re-index from golden copy
  - Re-run any recent updates
  - Route queries to new cloud
- Repeat for every cloud… 😱

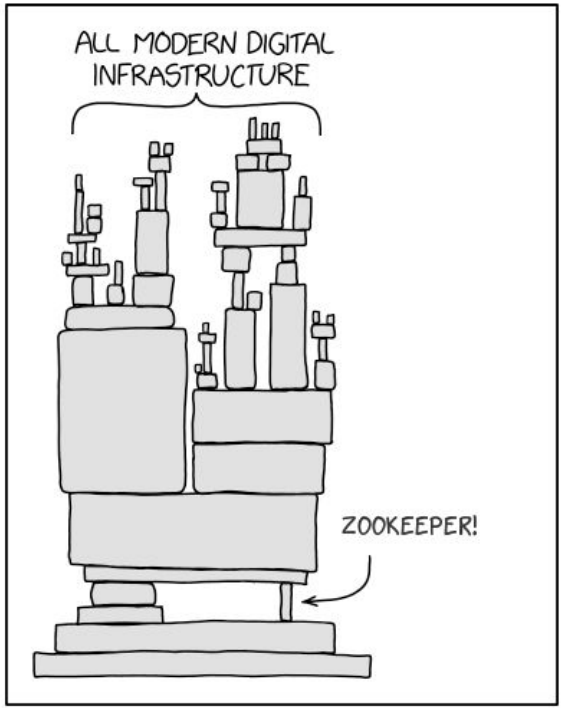
© 2024 Bloomberg Finance L.P. All rights reserved.

Bloomberg's Story: Apache ZooKeeper & SolrCloud

- **100,000s of Apache Solr** nodes across **~1,000 ZooKeeper** ensembles

- Benefits
  - Economies of scale
  - Limited isolation
  - ZooKeeper-as-a-Service

- Challenges
  - Infrastructure to manage ZooKeeper
  - En masse operations → load on ZooKeeper



based on https://xkcd.com/2347/

© 2024 Bloomberg Finance L.P. All rights reserved.

Bloomberg's Story: Keeping up with organic growth

- **Thousands of Apache Solr** clouds ranging from GBs to TBs in size
- Organic growth → larger shard size → performance degradation
- We found **SPLITSHARD** command to be unreliable & time consuming
- Current solution is to:
  - Delete the collection
  - Re-create the collection with updated number of shards
  - Re-index from golden copy
- Downtime for users 😞

© 2024 Bloomberg Finance L.P. All rights reserved.

(Salesforce)

Thousands of Collections per SolrCloud

But SolrCloud wasn't designed for thousands of collections

- Replica placement algorithm: use random. Others loop all collections
- Replica balancing: balance "actively used" replicas; custom
- Listing collections starts to not scale. ZK getChildren "jute maxbuffer" limit
  - And lookup state.json on each .. and every … one… doesn't scale either – SOLR-16909
- ZkStateReader watches /collections; working to stop this
  - Try CloudSolrClient v9.8 w/o ZK
- A ClusterState instance that's immutable and knows all collections; working on this

(Salesforce)

Thousands of Cores per Node

Thousands of cores/replicas on each node… how do we do that?

Memory: “Transient Cores” – don’t open cores until they are used to service a request. LRU cache. We enhanced this Solr feature to support (our) SolrCloud.

Memory: sharedSchema=true

Memory: Removed “version buckets” (2MB) per SolrCore (v9.8)

Metrics don’t scale per-core; aggregate to node level (thanks FullStory)

Startup perf: (this is an ugly hack!) We lie about the state of most replicas to avoid state updates. Hacks to deal with the consequence of lying

(Salesforce)

IDs, Shard Splits and Routing

Our docs have a “type” and all queries scoped to a type. Doc IDs are prefixed “`type!theRest`” and queries use `_route_=type`. Thus scope to a range of shards, never the entire collection.

Using `SHARDSPLIT` constantly, and don’t *yet* pre-size collections. `shard=XYZ&splitMethod=link&splitByPrefix=true` and `preferredLeader` changes in SOLR-16438. Blocking updates during splits; want to change that. A custom URP sees the shard is big and initiates the split. Inactive shards removed via a ClusterSingleton in v9.6 (SOLR-16403).

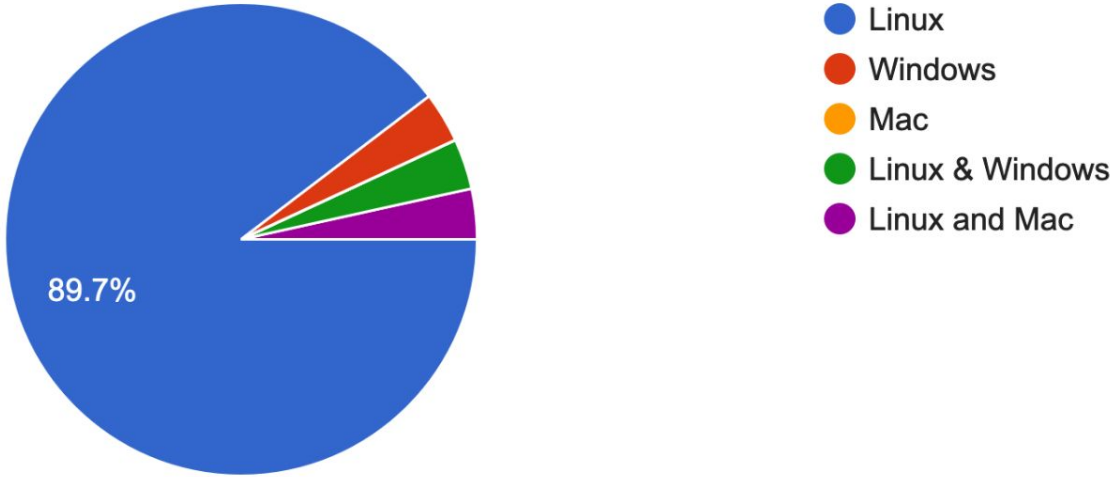
All shards have a replica with “preferredLeader” for stability & balancing.

Solr Community Survey 2024 Discussion Session

11:45 AM

What Operating System are you running Solr on?

29 responses

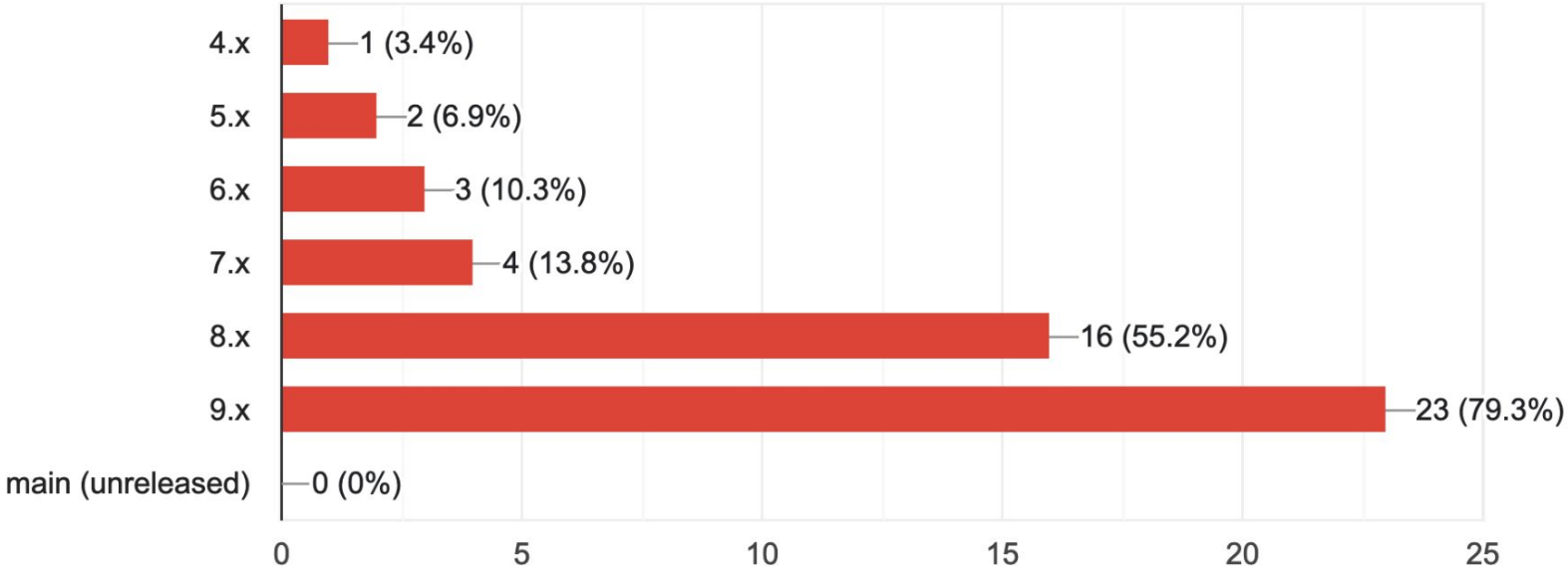


What did we learn from the 2024 Survey?

- 29 responses
- Not very scientific
- Any data better than no data?
- https://docs.google.com/forms/d/1KzKJqpd0XoDXa45EKx2HLOpBd-q4elVLE-BN_GD8zzE/viewanalytics

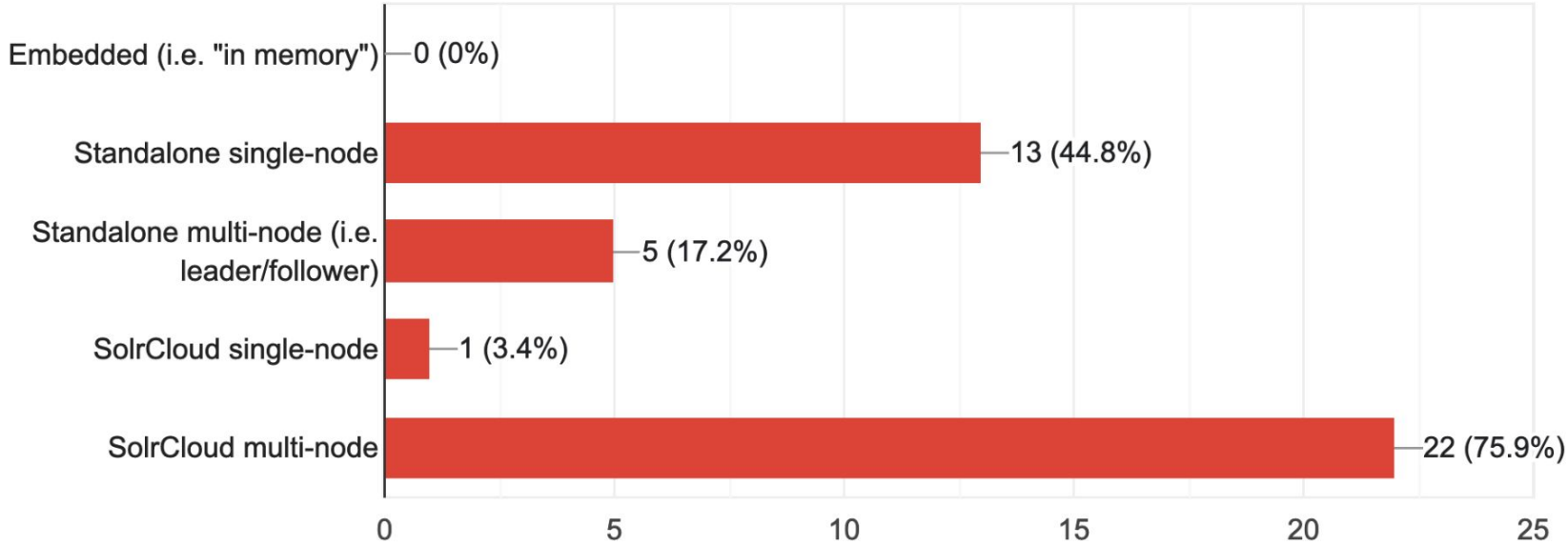
What Major Version(s) of Solr do you use?

29 responses



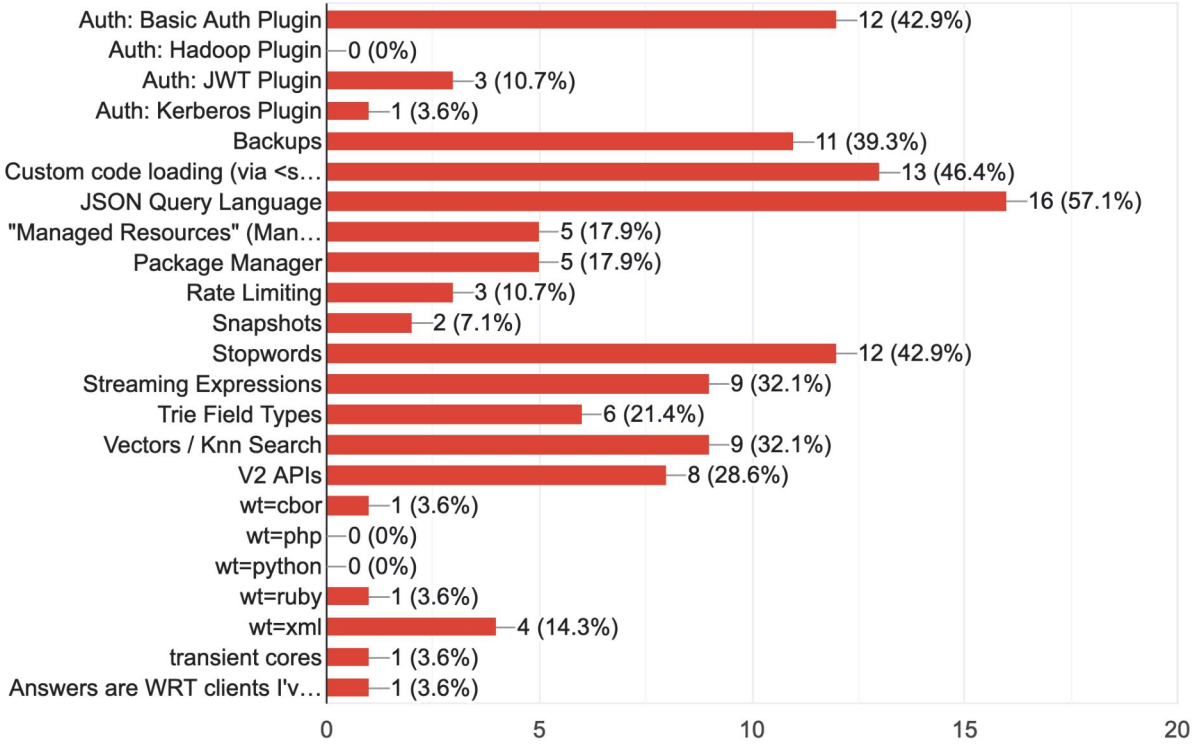
How do you deploy Solr?

29 responses



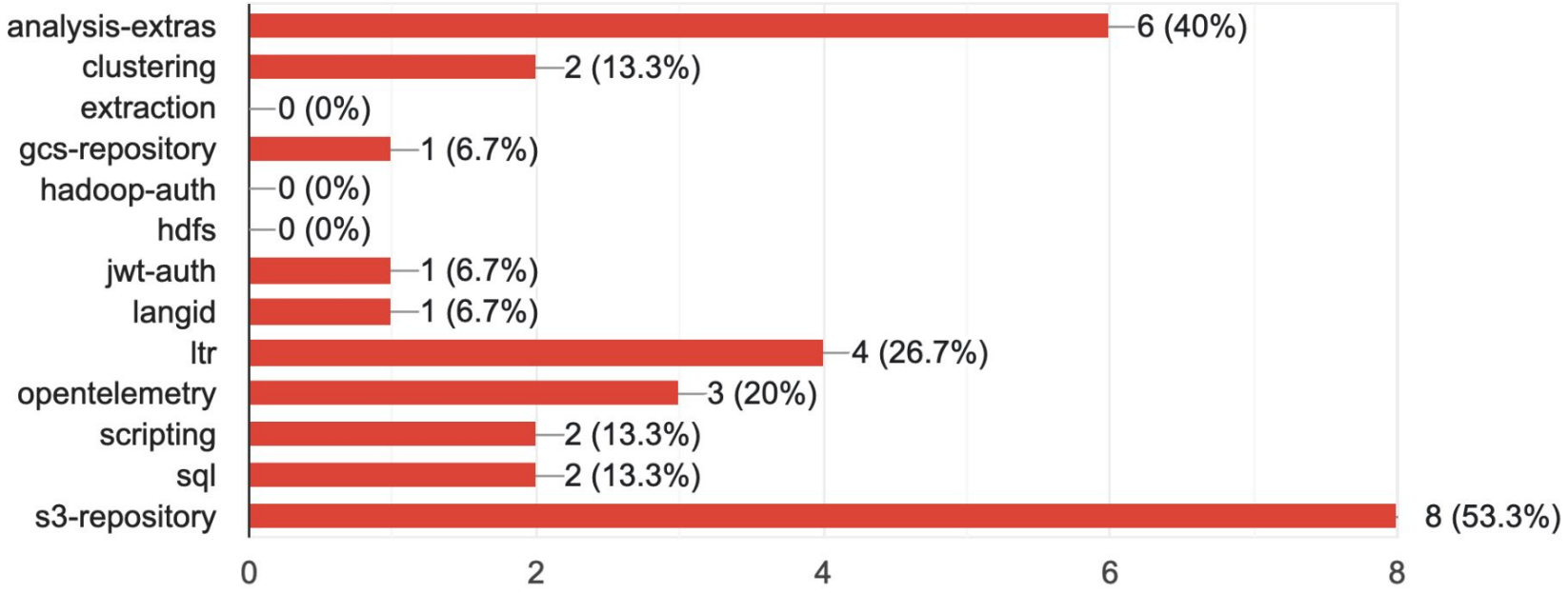
Which of the following less-common features of Solr do you use?

28 responses



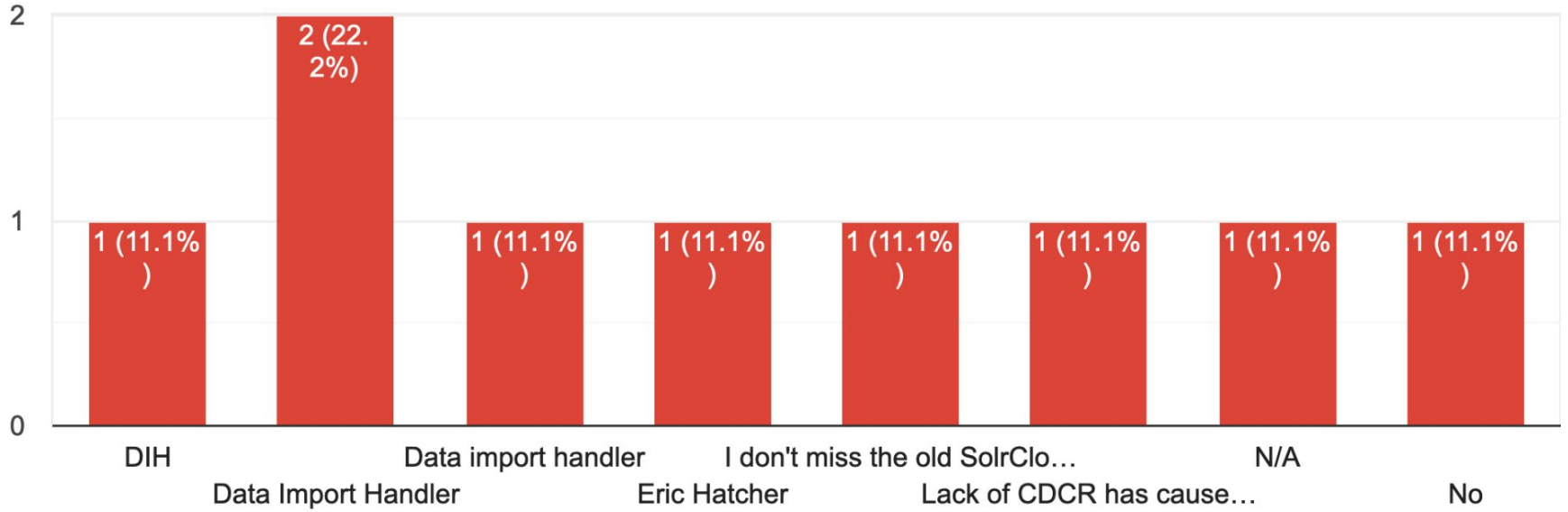
What Solr Modules do you use?

15 responses



Are there features of Solr that have been deprecated and you miss?

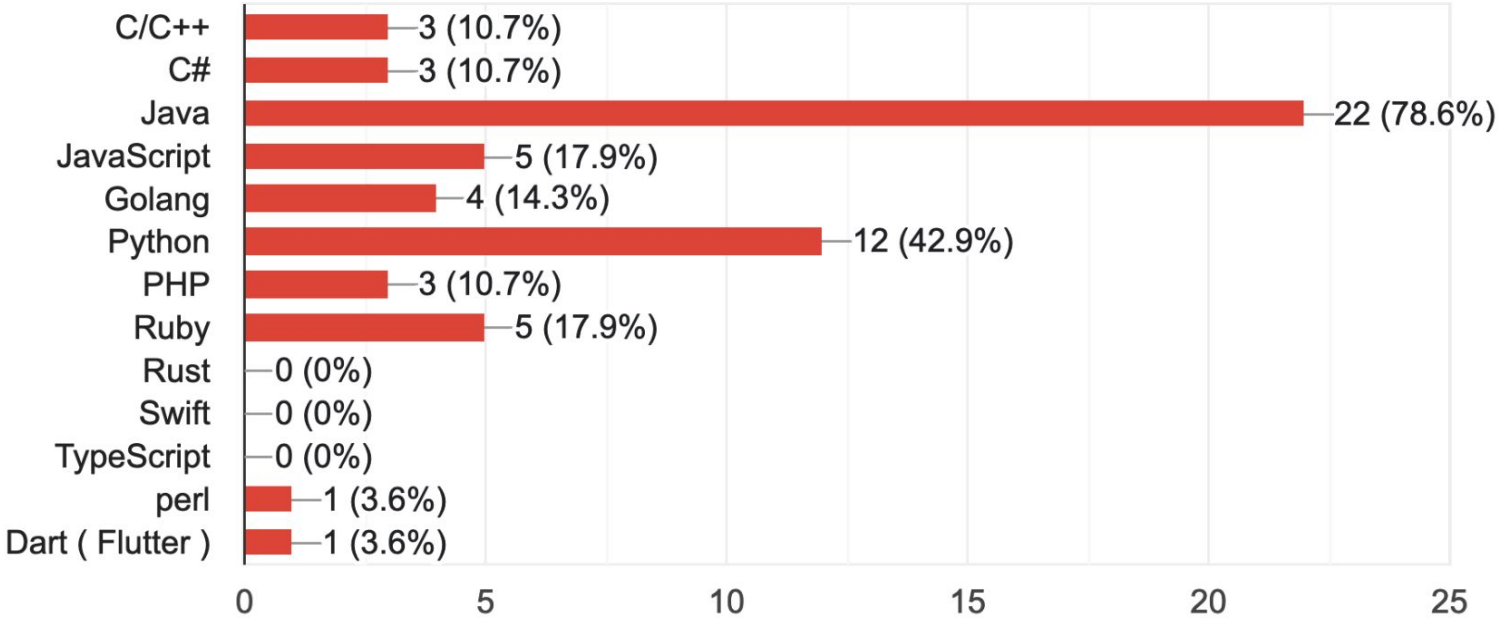
9 responses



4 votes for DIH
1 for Erik Hatcher
1 for CDCR
1 for “the buggy days of SolrCloud” ;-)

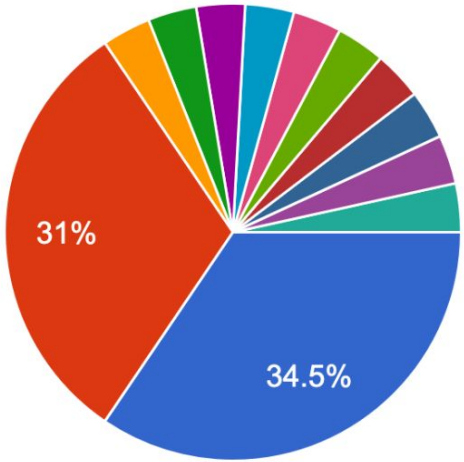
What languages do you use to interact with Solr?

28 responses



What tooling do you use to deploy Solr?

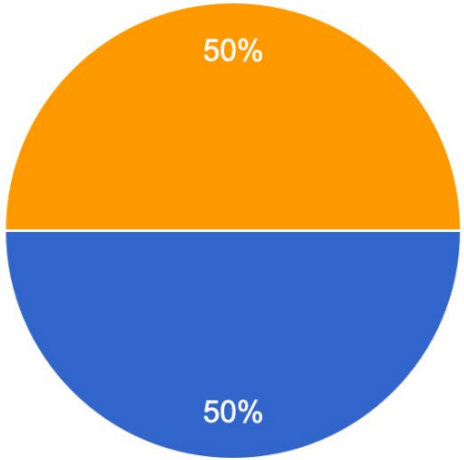
29 responses



- docker image
- install service script
- solr-operator
- Puppet
- custom install scripts in some cases, d…
- Custom build
- ansible
- (different projects:) ansible, docker, in…
- custom way of invoking the jetty Main class and passing start parameters
- tar xzf
- This should have been checkboxes not radios :) Client home grown kubernetes management system, install script and solr operator (diff clients)
- Manual

As a thank you for providing valuable information, and to follow up, I'd like to offer a beer or coffee, or a Solr Therapy Session via Zoom ;-). For the first two, just grab me at the conference!

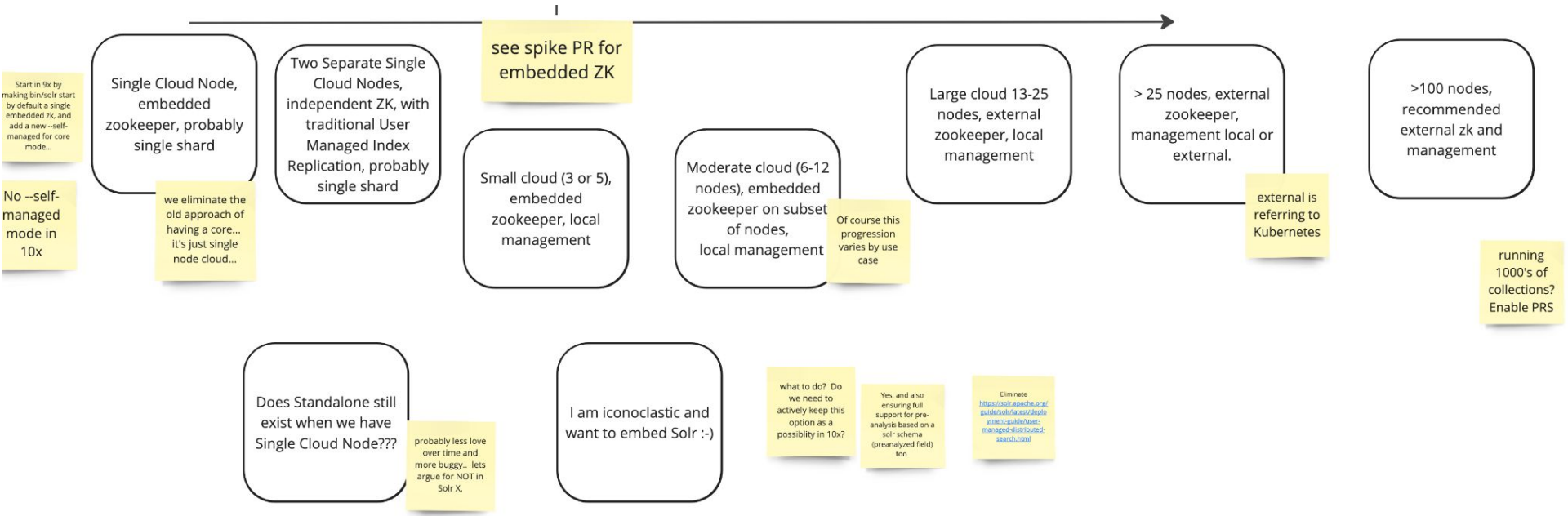
16 responses



- I'll take a beer at Community over Code from Eric!
- I'll take a coffee from Community over Code from Eric!
- I'll take a virtual coffee and thirty minutes answering Solr questions. Sign up on Calendly: https://calendly.com/epugh/solr-therapy-session

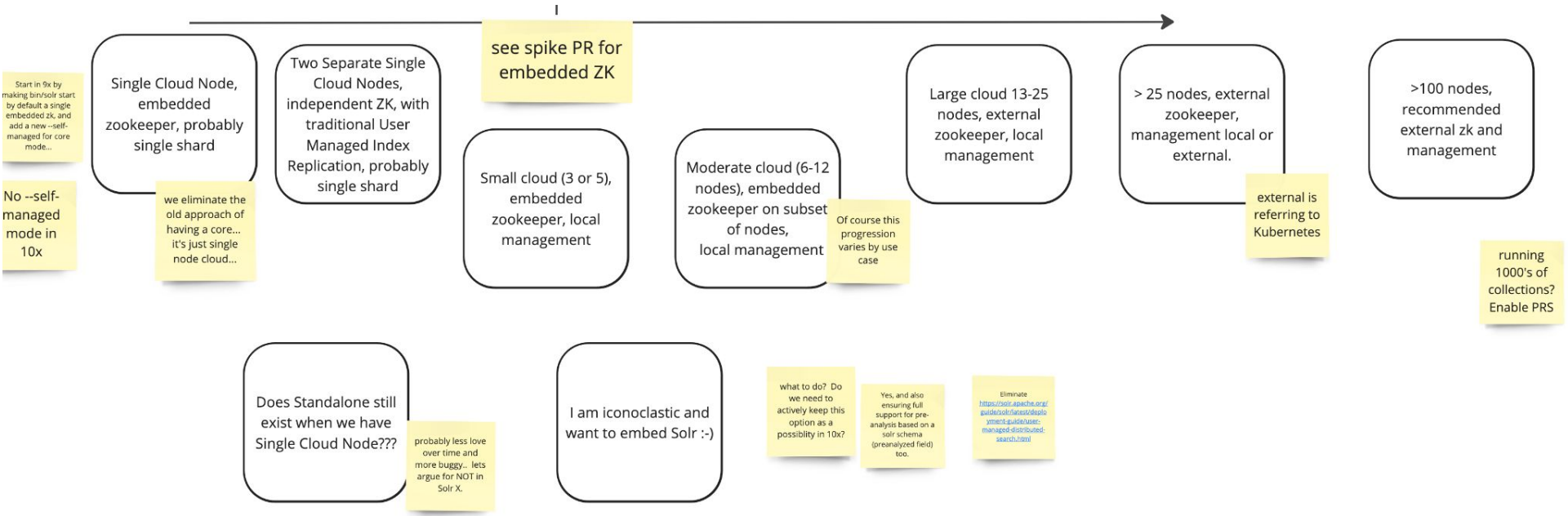
So....

Imagine if Solr could seamlessly scale from 1 to N?



Imagine if Solr could seamlessly scale from 1 to N?

Imagine if Solr could seamlessly scale from 1 to N?



Imagine if SolrCloud was just..... “Solr”

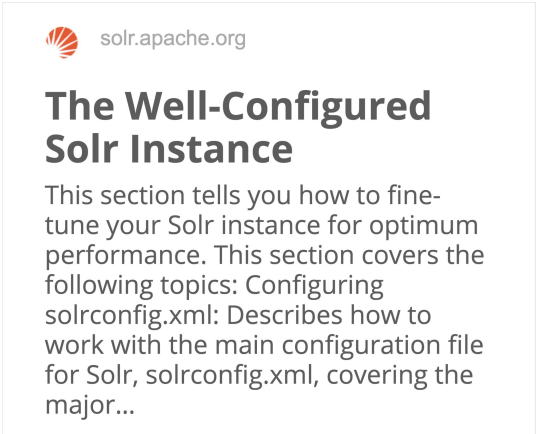
Some shibboleths we need to ditch

- 1 Node SolrCloud is somehow “bad”
- Leader/Follower is incompatible with SolrCloud and shouldn’t be used
- We can’t run ZooKeeper embedded in Solr
- We can’t hide ZooKeeper behind an interface `SolrState`
- ETL has nothing to do with Solr and we shouldn’t think about it
- ML has nothing to do with Solr and we shouldn’t think about it

Shibboleth: a custom, principle, or belief distinguishing a particular class or group of people, especially a long-standing one regarded as outmoded or no longer important.

What we would need to do

- Make some hard decisions on what we continue to include in Solr 10
- Take a much strong opinionated view of how to deploy Solr
  - Be clear in the Ref Guide about the different directions people can go
  - Establish some clear “Patterns of Deploying Solr”
  - Reboot the old “Well configured Solr”
- Eliminate the various rough edges in deployment (looking at you bin/solr start -c)
- Try some new things, and take some new risks



What you can do?

- Be active in the mailing lists (dev and user)
- Try out new features in your internal environments to provide real world feedback
- Contribute content and knowledge in scaling Solr to the Ref Guide
- Jump in on some of the modernization efforts (v2 APIs, Http2 effort) where many hands make light work
- Review JIRA tickets and plans
- Commit to continuing to use open source Solr over the next five to ten years

Thank you!