SIP-18: A Solr Kubernetes Module for native integration

Status

Current state: Implementation

Discussion thread: here https://lists.apache.org/thread/nlqgk3cb8z9v7gnrw3kkqxpp3v4y80j5

JIRA: SOLR-16739 - Introduce Solr Kubernetes Module

Released: <Solr Version>

Please keep the discussion on the mailing list rather than commenting on the wiki (wiki discussions get unwieldy fast). Confluence supports inline comments that can also be used.

Motivation

Currently all integration between Solr and Kubernetes is managed via the Solr Operator.

There are many Solr features that the Solr Operator uses Kubernetes constructs to manage:

- Solr Authorization/Authentication (security.json and BasicAuth)
- SSL
- ZK ACLs

And there are other features that the Solr Operator does not manage, but users have often requested:

- ConfigSets

Public Interfaces

- Location of the security.json may be found in a Kubernetes ConfigMap instead of Zookeeper
  - There should be no reason to need a Kubernetes Secret, since secrets will be loaded via a plugin described below.

Proposed Changes

A new "kubernetes" Solr Module, that provides multiple plugins that a user might want, such as:

- KubernetesConfigSetService - Managing configSets that live in Kubernetes ConfigMaps
- KubernetesSSLCredentialsProvider - An SSLCredentialsProvider that reads from Kubernetes Secrets
- KubernetesAuthenticationPlugin - An Authentication Plugin that uses Kubernetes Secrets
  - This plugin could just support BasicAuth, but it would be great to support ServiceAccount tokens as well.

Compatibility, Deprecation, and Migration Plan

- The Solr Operator supports many versions of Solr, so it would not be able to use these options by default until the minimum solr version matches the version that the new kubernetes Solr plugin is released in.
  - The SolrCloud CRD can have a flag that enables this new plugin if specified. That way we can begin testing and letting users use this new feature early-on.
- Once the minimum version is upgraded to a version that has the kubernetes solr module, we can begin removing the old logic for supplying Solr with information from Kubernetes.

Security considerations

Kubernetes Secrets are not the most secure way of storing secrets. However, that is a concern for end users, and Kubernetes has suggestions on mitigating, and alternatives that we can look at as well.

Test Plan

The testing plan will use the Solr Operator integration testing framework. From the Solr side we need to look at possible Java Kubernetes integration testing options.
Rejected Alternatives

- Adding various Solr related CRDs to Kubernetes that duplicate information stored in ZK (e.g. ConfigSet)
- Overwriting ZK information from changes in a Kubernetes resource (e.g. security.json)