KIP-632: Add DirectoryConfigProvider

Status

Current state: *Accepted*

Discussion thread: *here [Change the link from the KIP proposal email archive to your own email thread]*

JIRA: Original JIRA KAFKA-7370 - Getting issue details... STATUS , *New JIRA* *KAFKA-10211 - Getting issue details... STATUS*

Please keep the discussion on the mailing list rather than commenting on the wiki (wiki discussions get unwieldy fast).

Motivation

KIP-297 added the `ConfigProvider` interface for connectors within Kafka Connect, and KIP-421 extended support for `ConfigProviders` to all other Kafka configs. The `FileConfigProvider` added by KIP-297 provides values for keys found in a properties file. While this works fine for many use cases it is not ergonomic on Kubernetes.

In Kubernetes a user creates a named `Secret` object which contains pairs of name (sometimes termed key, though since the value is often an encryption key that can be confusing) and security-sensitive value. For example the Secret `my-secret` might contain the keys `keystore-password` and `truststore-password`. Such Secrets can be made visible in the container filesystem as a directory containing a file for each key, the contents of the file being the security-sensitive value.

At one point during the discussion of KIP-297 it was proposed to support a `DirectoryConfigProvider` to cater to this use case. Although the original JIRA was closed with "Won't Do" there doesn't seem to be any record of the rationale.

This KIP proposes adding `DirectoryConfigProvider`.

Public Interfaces

A new class will be added to the `org.apache.kafka.common.config.provider` package:

```
public class DirectoryConfigProvider implements ConfigProvider {
    public void configure(Map<String, ?> configs) {
        // no configs supported
    }

    /**
     * Retrieves the data at the given directory path.
     *
     * @param path the directory containing the secret files.
     * @return the configuration data
     */
    public ConfigData get(String path) {
        // list the regular files in the directory to construct the result
        // ...
    }

    /**
     * Retrieves the data with the given keys at the directory.
     *
     * @param path the directory where the data resides
     * @param keys the keys whose values will be retrieved
     * @return the configuration data
     */
    public ConfigData get(String path, Set<String> keys) {
        // Construct the result from ${path}/${key}
        // ...
    }

    public void close() {
    }
}
```

Proposed Changes

An example of use in a broker configuration:

```
config.providers=directory
config.providers.directory.class=org.apache.kafka.connect.configs.DirectoryConfigProvider
# ...
ssl.keystore.password=${directory:/var/run/my-secret-mount-point:keystore-password}
ssl.truststore.password=${directory:/var/run/my-secret-mount-point:truststore-password}
# ...
```

Compatibility, Deprecation, and Migration Plan

This proposal is completely backwards compatible.

Rejected Alternatives

If there are alternative ways of accomplishing the same thing, what were they? The purpose of this section is to motivate why the design is the way it is and not some other way.