KIP-398: Support reading trust store from classpath

Status

Current state: Under discussion

Discussion thread: here

JIRA: Unable to render Jira issues macro, execution error.

Motivation

Certificate pinning as well as authenticating kafka brokers using a non-public CA certificate maintained inside an organisation is desirable to a lot of users. This can be accomplished today using the `ssl.truststore.location` configuration property. Unfortunately, this value is always interpreted as a filesystem path which makes distribution of such an alternative truststore a needlessly cumbersome process. Instead of simply providing a dependency declaration in your build configuration you need to distribute and manage a separate file.

If we had the ability to load a trust store from the classpath as well as from a file, the trust store could be shipped in a jar that could be declared as a dependency and piggyback on the distribution infrastructure already in place. Trusted certificate distribution would be a simple matter of declaring the right dependencies.

While this would mostly be useful for CA certificates, it seems reasonable to make this loading mechanism available for the `ssl.keystore.location` property as well.

Public Interfaces

The value of the `ssl.truststore.location` and `ssl.keystore.location`, currently being interpreted as the name argument to the FileInputStream constructor in all cases, would change such that a value with the prefix `classpath:` instead would be interpreted as a resource path to be loaded by the class loader from the classpath. Values without the `classpath:` prefix would be handled as they currently are, as a filesystem path.

Proposed Changes

To support this change, have the code that parses the properties `ssl.truststore.location` and `ssl.keystore.location` identify values that begin with the string `classpath:` and pass along the remainder of the value to the ClassLoader's getResource() method.

This means that a value such as `conf/truststore.jks` would still be interpreted as it is currently being interpreted, whereas `classpath:com/example/truststore.jks` would be resolved on the classpath at runtime.

There is a prior example of using `classpath:` as a prefix in a string to load the resource it refers to in the widely used Spring Framework's Resource concept, which in turn builds on the Uniform Resource Locator design.

To support users that currently store their truststore in a file or directory with a name that begins with `classpath:` this proposal suggests that another prefix, `file:`, is introduced for this purpose. Values prefixed with `file:` are always treated as files, regardless of the presence of an additional `classpath:` prefix.

Compatibility, Deprecation, and Migration Plan

This change is mostly backwards compatible.

There is a corner case for users of the `ssl.truststore.location` and `ssl.keystore.location`, where semantics would change from today with this proposal: if the current value happens to begin with the string `classpath:`. As filenames with the colon character has limited portability between platforms, notably it is not an allowed filename character in Windows, this corner case should be rare. Those users would be asked to prefix their value with a `file:` prefix to preserve the current Kafka behaviour.

Rejected Alternatives

To make the change completely backwards compatible even for users with surprising filenames, it would be possible to introduce two new configuration keys `ssl.truststore.location.resolution.method` and `ssl.keystore.location.resolution.method` that would default to `file` but could optionally be set to `classpath`. This suggestion seems undesirable as every new configuration property adds to the cognitive overhead of configuring Apache Kafka. It seems to me like the cost of that additional complexity to avoid a very straight forward migration process of prefixing their path with `file:` for the extremely rare case of a user with current value that begins with `classpath:` is not worth the benefit.