KIP-208: Add SSL support to Kafka Connect REST interface

- Status
- Motivation
- Proposed Change
- Public Interfaces
- Migration Plan and Compatibility
- Rejected Alternatives

Status

Current state: Approved

Discussion thread: here

JIRA: KAFKA-4029

Pull Request: https://github.com/apache/kafka/pull/4429

Released: 1.1.0

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

Motivation

Kafka Connect has a REST interface for managing and monitoring connectors. Currently this REST interface is using only plain HTTP without any encryption and authentication. This is not ideal because:

- The interface might be used to transfer sensitive information (e.g. passwords in Kafka Connect connector configurations)
- Anyone with access to the interface can add connectors (which can forward sensitive messages from Kafka brokers somewhere else)

Extending the REST interface with support for SSL / TLS encryption and SSL / TLS client authentication could address these issues.

Proposed Change

This KIP proposes enabling SSL support in the Jetty HTTP server. Jetty already supports SSL / TLS. So the main work in this KIP will be around enabling and configuring SSL / TLS.

Users will be able to configure REST listeners using a new configuration option `listeners`. It will allow to configure the protocol (which will be either HTTP or HTTPS), host and port. Users will be allowed to configure either HTTP listener or HTTPS listener or both. For example:

```
listeners=http://myhost:8080,https://myhost:8443
```

or

```
listeners=https://myhost:8443
```

When the `listeners` parameter is configured, the existing options `rest.host.name` and `rest.port` will be ignored. The fields `rest.host.name` and `rest.port` will be marked as deprecated.

The HTTPS listener (when configured in `listeners`) will by default use the SSL configuration from the `ssl.*` options. In case the user needs to use different SSL configuration for connecting to Kafka brokers and for the REST interface, the default settings can be overridden by using the prefix `listeners.https.` - for example:

```
listeners.https.ssl.keystore.location=/my/path/keystore.jks
```

The `rest.advertised.host.name` and `rest.advertised.port` options will continue to be used as today to specify the connection address which should be used by other workers. In addition a new option `rest.advertised.listener` will define whether other workers should connect using HTTP or HTTPS protocols. In case HTTPS is selected, the connecting worker will use the SSL configuration from the existing `ssl.*` options. Even in case when `rest.advertised.host.name` and `rest.advertised.port` options are not specified this field will be used to define which protocol should be advertised to other workers in combination with the appropriate hostname and port from the listener field.

This proposal doesn't include any authorization / ACL features. Only encryption and authentication. Authorization / ACLs should be subject of separate KIP in order to keep the scope of this KIP under control. It also doesn't add any other authentication options than SSL/TLS client authentication.

Public Interfaces

Configuration of SSL / TLS for the Kafka Connect REST interface tries to follow the configuration for other SSL / TLS enabled server interfaces. It will be done through the properties configuration file for the distributed Kafka Connect workers.

Following **new** options will be added:

Parameter	Default value	Note
`listeners`	null	List of REST listeners in the format `protocol://host:port,protocol2://host2:port2` where the protocol is one of `HTTP` and `HTTPS`.
`rest.advertised.listener`	null	Configures the listener used for communication between workers. Valid values are either `HTTP` or `HTTPS`. When the listeners configuration is not defined or when it contains `HTTP` listener, the default value for this field will be `HTTP`. When the listeners option is configured and contains only `HTTPS` listener, the default value will be `HTTPS`.
`ssl.client.auth`	none	Valid values are `none`, `requested` and `required`. It will controls whether: • the connecting client is required to do SSL/TLS client authentication (`required`) • it can decide to skip the SSL/TLS client authentication (`requested`) • the SSL/TLS authentication will be completely disabled (`none`) This is the only authentication option suggested as part of this KIP.
`listeners.https.ssl.*`		The `listeners.https.` prefix can be used with any SSL configuration option mentioned below to override the default SSL configuration which is shared with the connections to Kafka broker. In case at least one option with this prefix exists, the implementation will use only SSL options with this prefix and will ignore all options without prefix. In case no option with prefix `listeners.https.` exists, the options without prefix will be used.

Following existing options will be **affected** by this KIP:

Parameter	Default value	Note
`rest.host.name`	null	When `listeners` option is defined, this field will be ignored.
`rest.port`	8083	When `listeners` option is defined, this field will be ignored.

The `rest.host.name` and `rest.port` will be marked as deprecated. The `listeners` field would be the one preferred for the long term future.

Following existing options will be reused by this KIP without any changes:

Parameter	Default value	Note
`rest.advertised.host.name`	null	
`rest.advertised.port`	null	This field will be reused without any changes.
`ssl.keystore.location`	null	
`ssl.keystore.password`	null	
`ssl.keystore.type`	JKS	
`ssl.key.password`	null	
`ssl.truststore.location`	null	
`ssl.truststore.password`	null	
`ssl.truststore.type`	JKS	
`ssl.enabled.protocols`	TLSv1.2,TLSv1.1,TLSv1	
`ssl.provider`	null	
`ssl.protocol`	TLS	
`ssl.cipher.suites`	null	
`ssl.keymanager.algorithm`	SunX509	
`ssl.secure.random.implementation`	null	
`ssl.trustmanager.algorithm`	PKIX	
`ssl.endpoint.identification.algorithm`	null	

Migration Plan and Compatibility

This KIP is a new implementation and doesn't have any backwards compatibility issues or special requirements on migration from older versions. Existing Kafka Connect installation would work in the same way as before this change. Without the SSL configuration, the REST interface will continue to be configured as today - i.e. without SSL / TLS.

Rejected Alternatives

- The first version of this KIP suggested only single REST listener - either HTTP or HTTPS. This was changed based on the feedback from the discussion