KIP-567: Kafka Cluster Audit - Original Version

Status

Current state: *Under Discussion*

Discussion thread: *here*

JIRA: *KAFKA-9413*

Motivation

It is highly demanded in most businesses to have the ability of obtaining audit information in case someone changes cluster configuration (like creation /deletion/modify/description of any topic or ACLs).
We may add this ability. Since audit requirements are so broad, it's impractical to support all of them.
Hence we have to provide ability for users to plug resources helping to achieve required capabilities.

Public Interfaces

AuditExtension

Developers will be required to implement only the **AuditExtension** interface to provide an extension. We provide default implementation of AuditExtension that will send audit events to log file specified in logging properties.
We will provide several implementations of **AuditEvent** interface likewise there are a lot of audit events like create ACL, delete ACL, create topic, etc. Moreover, it might be necessary to know which result the operation finishes with. Therefore, it is convenient to have separate event implementation for every audit event.
Every kind of audit event will be implemented by default.

```
package org.apache.kafka.common.audit;
public interface AuditExtension extends Configurable, Closeable {
    /**
     * AuditExtension implementations received an audit event via the {@link
     * #onEvent(AuditEvent)} method. The extension implementation will invoke this method for every audit event
occurrence.
     *
     * @param event The audit event we want to describe{@link org.apache.kafka.common.audit.AuditEvent}.
     */
    void onEvent(AuditEvent event);
}
```

As mentioned above, even though the developers are required to only implement the AuditExtension, they will be using new public interface implemented by default.

AuditEvent

```
package org.apache.kafka.common.audit;
import java.util.UUID;
public interface AuditEvent {
    /**
     * Get the unique event id.
     *
     * @return the uuid
     */
    UUID uuid();

    /**
     * Get the request context.
     *
     * @return {@link org.apache.kafka.common.requests.RequestContext}
     */
    RequestContext requestContext();
}
```

This also introduces a new configuration that audit.extension.classes that allows to configure a comma separated list of Audit extension implementations. E.g.:

Property settings example
`audit.extension.classes=org.apache.kafka.common.audit.LoggingAuditExtension,org.apache.kafka.common.audit.KafkaAuditExtension`

Proposed Changes

Users will be able to create a plugin by implementing the **AuditExtension** interface that has a single method that takes an AuditEvent's children instance as the only parameter.
This allows us to change the interface easily in the future to add new parameters.
Audit runtime will also provide a default implementation for **AuditEvent** interface.
One or more of **AuditExtension**'s implementations can be configured via the configuration audit.extension.classes as a comma separated list of class names.

These implementations will get access to broker configuration via the **configure(Map<String, ?> configs)** method in the **AuditExtension** implementation (through org.apache.kafka.common.Configurable).

Reference Implementation

This KIP proposes to include reference implementation that allows users to send audit events to separate logger.

Compatibility, Deprecation, and Migration Plan

This is entirely new functionality so there are no compatibility, deprecation, or migration concerns.

Rejected Alternatives

None