KIP-55: Secure Quotas for Authenticated Users

Status

Current state: *Accepted*

Discussion thread: *here*

JIRA: *KAFKA-3492*

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

Motivation

KIP-13 introduced client quotas in Kafka 0.9.0.0. Rate limits on producers and consumers are enforced to prevent clients saturating the network or monopolizing broker resources. The current implementation allocates quotas to client-ids. This works well in single user clusters or clusters that use PLAINTEXT where all users have the same identity. But since client-id is unauthenticated and can be set to any value by the client, multi-tenant secure installations require secure quotas to be enforced to guarantee fair allocation of resources and prevent denial-of-service.

With the introduction of security in Kafka 0.9, the identity of Kafka clients is the user principal. User principal is an authenticated user or a grouping of unauthenticated users chosen by the broker using a configurable `PrincipalBuilder` and is currently used for ACLs. Client-id is a logical grouping of clients with a meaningful name that is used in client metrics and logs. Multi-user systems have a hierarchy - user owns zero or more clients. `<user-principal, client-id>` defines a safe group of clients. The shorter unsafe client-id is sufficient in client metrics and logs, but quotas should be allocated to safe groups to avoid clients of one user throttling clients of another user with the same client-id.

This KIP proposes the following changes to the existing implementation:

1. Quota configuration for user principal. This prevents users generating heavy traffic from monopolizing resources and impacting the performance of other users in a multi-tenant cluster.
2. Quota overrides for clients of an authenticated user. Like the current client-id implementation, this enables a user to rate-limit some producers or consumers to ensure that they don't impact other more critical clients. For instance, users may be able to rate-limit an auditing client running in the background, leaving resources always available for a critical event processing client.
3. Client-id quotas for users without a user quota override. Existing quota configuration for client-ids will continue to be applied to users with unlimited quota, if no user-specific defaults are configured. Defaults may also be specified for client-ids that do not span multiple users.

The goal of this KIP is to provide a single unified quota implementation with a simple configuration for client-id based quotas, a similar simple configuration for user quotas and the flexibility to define more complex quota configurations.

Public Interfaces

Overview

Quotas can be set at `<user, client-id>`, `user` or `client-id` levels. For a given client connection, the most specific quota matching the connection will be applied. For example, if both a `<user, client-id>` and a `user` quota match a connection, the `<user, client-id>` quota will be used. Otherwise, `user` quota takes precedence over `client-id` quota. The order of precedence is:

1. `/config/users/<user>/clients/<client-id>`
2. `/config/users/<user>`
3. `/config/users/<default>/clients/<client-id>`

4. /config/users/<default>/clients/<default>
5. /config/users/<default>
6. /config/clients/<client-id>
7. /config/clients/<default>

Quota Entity

Quotas are currently configured for client-ids. All clients with the same client-id are currently grouped together as a quota entity, enforcing one quota for all clients with the same client-id. This KIP proposes to define quotas for safe client groups which share the same user-principal and client-id and user-level quotas that limit the total quota of users in addition to the existing client-id quotas.

Configuration Options

Default quotas for users and client-ids will be added as dynamic properties. The existing default configuration options for client-id quotas will be deprecated and these properties will be applied only if default user quota is unlimited and the dynamic client-id defaults are not specified.

Deprecate static properties for client-id default quotas

`quota.producer.default, quota.consumer.default`: Default client-id producer/consumer quota is currently applied to each unique client-id across all users. This `client-id` default will be applied only if default user quota is unlimited and no user-specific defaults are configured. These properties will be deprecated and will be applied only if dynamic default quotas are not configured for clients.

Dynamic configuration for default quotas

- Default quota for <user, client-id> will be stored in Zookeeper at /config/users/<default>/clients/<default>.
- Default user quota will be stored in Zookeeper at /config/users/<default>.
- Default client-id quota will be stored in Zookeeper at /config/clients/<default>. If not specified, the broker properties `quota.producer.default, quota.consumer.default` will be used as the default client-id quota for clients of users with unlimited quota.

Metrics

Quota related metrics are currently generated for client-ids and use the tag "`client-id`". URL-encoded user principal will be added with the tag "`user`". Both user and client-id tags will be specified when <user, client-id> quotas are applied, only `user` tag will be specified if user quota is applied and the existing `client-id` tag will be retained if client-id quotas are applied.

Sensor names are currently a sensor type concatenated with the client id value, eg. *FetchThrottleTime-clientA*. This will be modified to use `quota-id` instead of `client-id`. This is not a public interface change since sensor names are not reflected in JMX metrics.

Tools

`kafka-configs.sh` will be extended to support authenticated user quotas and specific quotas for <user, client-id>. A new entity type "`users`" will be added with the same key-value pairs as the existing "`clients`" entity type:

- `producer_byte_rate` : The total rate limit for the user's producers without a client-id quota override
- `consumer_byte_rate` : The total rate limit for the user's consumers without a client-id quota override

For example:

```
bin/kafka-configs  --zookeeper localhost:2181 --alter --add-config 'producer_byte_rate=1024,
consumer_byte_rate=2048' --entity-name user1 --entity-type users
```

Default quotas for users or clients or <user, client-id> can be configured by omitting entity name. For example:

```
bin/kafka-configs  --zookeeper localhost:2181 --alter --add-config 'producer_byte_rate=10000,
consumer_byte_rate=20000' --entity-type users
```

Quotas for clients of a user can be configured by specifying entity types "`users`" and "`clients`" in the same command line . For example, the following command sets quotas for <user2, clientA>:

```
bin/kafka-configs  --zookeeper localhost:2181 --alter --add-config 'producer_byte_rate=10,
consumer_byte_rate=20' --entity-name clientA --entity-type clients --entity-name user2 --entity-type users
```

The existing entity type "`clients`" will be retained to set `client-id` quotas which are used when user quotas are not overridden.

Proposed Changes

User Principal

Authenticated user principal will be obtained from the `Session` object. URL-encoded string version of the Principal will be used so that it can be used as a node name in Zookeeper and in metrics without placing any restrictions on the characters allowed in the principal. Characters that cannot be used for Zookeeper node names or metrics (eg. *) will be percent-encoded. Encoded user principal will be cached in `Session.` For PLAINTEXT, the principal is "ANONYMOUS" by default and quotas will be applied for that principal. But principal can be overridden using a custom principal builder even for PLAINTEXT, enabling different user quotas, for example, for connections from different IP addresses.

Quota Configuration

Quotas are currently configured as rate limits for producers and consumers based on their client-id. Default rate limits can be configured for clients without a config override. The same set of limits will be configurable each quota-id.

Quota configuration for a client with client-id *clientX* and user principal *userN* is determined by the following sequence (this example is for producer, similar sequence is applied to consumer):

1. If quota override is defined for <*userN*, clientX> in `/config/users/userN/clients/clientX`, this quota is allocated for the sole use of <*userN, clientX*>.
2. If user quota override is defined for *userN*, *clientX* in `/config/users/userN` shares this quota with other clients of *userN*
3. If <user, client-id> quota is defined *in `/config/users/<default>clients/clientX`,* this quota is allocated for the sole use of <*userN, clientX*>
4. If default <user, client-id> quota is defined *in `/config/users/<default>clients/<default>`,* this quota is allocated for the sole use of <*userN, clientX*>
5. If default user quota is defined in `/config/users/<default>`, *clientX* shares this default quota with other clients of *userN*
6. If client-id quota override is defined for *clientX in `/config/clients/clientX`,* this quota is shared across client-id <*clientX*> of all users
7. If dynamic client-id default is configured in *`/config/clients/<default>`*, this default quota is shared across client-id <*clientX*> of all users
8. If `quota.producer.default is` configured for the broker in server.properties, this *is shared across client-id <clientX> of all users*
9. Client is not throttled

Use cases:

- Simple client-id based quotas are configured using client-id quota override, dynamic client-id default and static `quota.producer.default :` (steps 6, 7, 8, 9)
- Simple user-principal based quotas are configured using user quota override and user quota default `:` (steps 2, 5, 9)
- More specific <user, client-id> quotas and defaults for users and client-ids can be configured if required: (steps 1 - 9)

Sample configuration: Default user quota

```
// Default user quota
// Zookeeper persistence path /config/users/<default>
{
    "version":1,
    "config": {
        "producer_byte_rate":"10000",
        "consumer_byte_rate":"20000"
    }
}
```

Sample configuration: User quota without client-id overrides

```
// Quotas for user1 (without client-id overrides).
// Zookeeper persistence path /config/users/<encoded-user1>
{
    "version":1,
    "config": {
        "producer_byte_rate":"1024",
        "consumer_byte_rate":"2048"
    }
}
```

Sample configuration: User quota with client-id overrides

```
// User-level quotas for user2
// Zookeeper persistence path /config/users/<encoded-user2>
{
    "version":1,
    "config": {
        "producer_byte_rate":"4096",
        "consumer_byte_rate":"8192"
    }
}
// Quota override for <user2, clientA>
// Zookeeper persistence path /config/users/<encoded-user2>/clients/clientA
{
    "version":1,
    "config": {
        "producer_byte_rate":"10",
        "consumer_byte_rate":"30"
    }
}
// Quota override for <user2, clientB>
// Zookeeper persistence path /config/users/<encoded-user2>/clients/clientB
{
    "version":1,
    "config": {
        "producer_byte_rate":"20",
        "consumer_byte_rate":"40"
    }
}
```

Sample configuration: Client-id quota

```
// Quotas for client-id clientA of users without user quota override.
// Zookeeper persistence path /config/clients/clientA
{
    "version":1,
    "config": {
        "producer_byte_rate":"100",
        "consumer_byte_rate":"200"
    }
}
```

In the sample configuration above:

1. Total rate limits for all clients with user principal *user1* is (1024, 2048).
2. Total rate limits for all clients with user principal *user2* without additional client-id quota is (4096, 8192).
    - The rate limits for clients with user principal *user2* AND client-id *clientA* is (10, 20).
    - Clients of *user2* with client-id other than *clientA* and *clientB* share the quota (4096, 8192).
3. Total rate limits for all clients of *user3* is the default (`10000, 20000`), since no config override is specified.
4. If default user quota is not specified or is unlimited, clients of *user3* use client-id quota configuration. For example quota for client-id *clientA* of *user3* is (100, 200) and this is shared with clients of other users with client-id *clientA*. And quota for client-id *clientB* of *user3* without a client-id override is (`quota.producer.default, quota.consumer.default`)
    - In a multi-user cluster, defaults can also be specified for `<user, client-id>` quotas which treat *clientA* of *user4* as a different group from *clientA* of *user2*.

Quota Identifier

Quota configuration and sensors currently use `client-id` as the unique key, enforcing one quota for all clients with the same `client-id`. This will be replaced with a new `quota-id` that includes user principal. Each `quota-id` is associated with a pair of producer and consumer rate limits which may be config overrides or the default quota.

- `quota-id` is the concatenation of url-encoded user principal and client-id for `<user, client-id>` quota, just the client-id for `client-id` quotas and just the encoded user principal for `user` quotas. Similarly, quota-related metrics names will include the tags `client-id` and/or `user`.
- In the example:
    - All clients of *user1* share the quota-id *user1:*

- *clientA* of *user2* uses the quota-id *user2:clientA*
- *clientC* of *user2* uses the quota-id *user2:* since it does not have a client quota override, sharing a quota with other clients of *user2*.
- *clientA* of *user3* uses the quota-id *:clientA*

Quota Persistence in Zookeeper

Client-id based quota configuration overrides will continue be stored under `/config/clients`, but these will be applied only to clients of users without a quota override and only if default user quota is unlimited. Quota configuration overrides for user principals will be stored under `/config/users/<user>`. Note that url-encoded version of the user principal will be used as node name under `/config/users` to cope with Zookeeper naming restrictions. Default user quotas will be stored under `/config/users/<default>`. Quota overrides for clients of a user will be stored under `/config/users/<user>/clients`.

Configuration change notifications will be generated for changes to quota configuration similar to the current notifications for `client-id` quotas. JSON for change notification will be modified to provide entity path instead of specifying entity type and name separately.

Sample configuration change notification

```
// Change notification for default user quota
{
    "version":2,
    "entity_path": "users/<default>"
}
// Change notification for user quota of user1
{
    "version":2,
    "entity_path": "users/user1"
}
// Change notification for quota of <user2, clientA>
{
    "version":2,
    "entity_path": "users/user2/clients/clientA"
}
// Change notification for default client-id quota
{
    "version":2,
    "entity_path": "clients/<default>"
}
// Change notification for client-id quota of clientA
{
    "version":2,
    "entity_path": "clients/clientA"
}
```

Tools

`kafka-configs.sh` will be extended to support a new entity type `"users"`. Quota configuration for users will be provided as key-value pairs to be consistent with other configuration options. Hence no new command line arguments will be added to the tool. The tool will parse the key-value pairs specifying rate limits, validate these and convert them to the equivalent JSON for persistence in Zookeeper. The existing entity "`clients`" will continue to be supported to set client-id quotas for users with unlimited quota. The tool will be extended to accept multiple entity types to configure <user, client-id> quotas. The tool will also be updated to configure default quotas (`/config/users/<default>`, `/config/clients/<default>`, `/config/users/<default>/clients/<default>`).

Compatibility, Deprecation, and Migration Plan

No client API changes are necessary to work with the new implementation. Existing client-id quota configuration will be processed from Zookeeper and since users have unlimited quota as default, existing client-id quotas will be applied to all clients with the client-id.

Rejected Alternatives

Broker configuration property to choose either client-id or user quota

Single user clusters require only client-id based quotas and some secure clusters may require only overall user quotas. A simple switch between these two provides a simpler configuration and implementation. But a server option that changes the semantics of quota scan be confusing and it is hard to configure a cluster without actually knowing which quota mode is being used. Since hierarchical quotas are useful in multi-user clusters, it is better to handle both client-id quotas and user quotas with semantics suitable to both secure and plaintext clusters.

Add quota-id to Kafka protocol to define a new grouping

Both client-id and user principal are groupings of clients currently used for other purposes. User-principals are used for ACLs and client-ids are used for client metrics and logs. It may be useful to define a quota group that doesn't have to align with existing uses of user principal or client-id. But this require addition of a new concept of quota-id into the Kafka protocol that is authenticated. Use of existing ids simplifies configuration and keeps the definitions consistent.