KIP-373: Allow users to create delegation tokens for other users

This page is meant as a template for writing a KIP. To create a KIP choose Tools->Copy on this page and modify with your content and replace the heading with the next KIP number and a description of your issue. Replace anything in italics with your own description.

Status

Current state: *Accepted*

Discussion thread: *here*

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

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

Motivation

In KIP-48, we added support for delegation token based authentication mechanism. Currently, we only allow a user to create delegation token for that user only. This limits the usage of delegation token mechanism. This KIP proposes to allow users to create delegation tokens for other users. This allows the following use cases.

1. A designated superuser can create tokens without requiring individual user credentials.
2. A designated superuser can run kafka clients on behalf of another user.

In this use case, a superuser with username ‘superuser’ wants to run kafka clients on behalf of a user 'joe'. The 'superuser' has secure authentication credentials (kerberos, SSL, SCRAM) but user 'joe' doesn’t have any. The clients are required to run as user 'joe' and authorizations are required to be done as user 'joe.' In this case, 'superuser' can get a delegation token for user 'joe', and use the generated token to run the Kafka clients. This will mimic the impersonation functionality. This will help the stream processing frameworks/libs (Apache Spark, Storm, Kafka Streams) to run the jobs (Kafka clients) as submitted users.

In case of Storm, Storm’s master (nimbus) is the only node that needs a superuser keytab. Using this keytab Nimbus will authenticate with Kafka broker and acquire a delegation token for the submitted jobs. Nimbus can then distribute this delegation token to all of its worker hosts and all worker jobs (Kafka clients)) will be able to authenticate to kafka using tokens. Similar logic can be implemented on Spark jobs.

Public Interfaces

Protocol Changes

CreateDelegationTokenRequest

We will bump the version of the CreateTokenRequest API to include the owner details in CreateDelegationTokenRequest and "Token requester" in CreateDelegationTokenResponse. For old versions, we will skip the owner and owner will be same as token request principal.

```
{
  "apiKey": 38,
  "type": "request",
  "name": "CreateDelegationTokenRequest",
  // Version 1 is the same as version 0.
  //
  // Version 2 is the first flexible version.
  //
  // Version 3 adds the owner principal
  "validVersions": "0-3",
  "flexibleVersions": "2+",
  "fields": [
    { "name": "OwnerPrincipalType", "type": "string", "versions": "3+", "nullableVersions": "3+",
      "about": "The principal type of the owner of the token. If it's null it defaults to the token request 
principal." },
    { "name": "OwnerPrincipalName", "type": "string", "versions": "3+", "nullableVersions": "3+",
      "about": "The principal name of the owner of the token. If it's null it defaults to the token request 
principal." },
    { "name": "Renewers", "type": "[]CreatableRenewers", "versions": "0+",
      "about": "A list of those who are allowed to renew this token before it expires.", "fields": [
      { "name": "PrincipalType", "type": "string", "versions": "0+",
        "about": "The type of the Kafka principal." },
      { "name": "PrincipalName", "type": "string", "versions": "0+",
        "about": "The name of the Kafka principal." }
    ]},
    { "name": "MaxLifetimeMs", "type": "int64", "versions": "0+",
      "about": "The maximum lifetime of the token in milliseconds, or -1 to use the server side default." }
  ]
}
```

Field	Description
`Owner`	Owner is an Kafka PrincipalType+name string, who is the owner of the token. If owner string is null, then token request principal is treated as owner

CreateDelegationTokenResponse

```
{
  "apiKey": 38,
  "type": "response",
  "name": "CreateDelegationTokenResponse",
  // Starting in version 1, on quota violation, brokers send out responses before throttling.
  //
  // Version 2 is the first flexible version.
  //
  // Version 3 adds the token requester principal
  "validVersions": "0-3",
  "flexibleVersions": "2+",
  "fields": [
    { "name": "ErrorCode", "type": "int16", "versions": "0+",
      "about": "The top-level error, or zero if there was no error."},
    { "name": "PrincipalType", "type": "string", "versions": "0+",
      "about": "The principal type of the token owner." },
    { "name": "PrincipalName", "type": "string", "versions": "0+",
      "about": "The name of the token owner." },
    { "name": "TokenRequesterPrincipalType", "type": "string", "versions": "3+",
      "about": "The principal type of the requester of the token." },
    { "name": "TokenRequesterPrincipalName", "type": "string", "versions": "3+",
      "about": "The principal type of the requester of the token." },
    { "name": "IssueTimestampMs", "type": "int64", "versions": "0+",
      "about": "When this token was generated." },
    { "name": "ExpiryTimestampMs", "type": "int64", "versions": "0+",
      "about": "When this token expires." },
    { "name": "MaxTimestampMs", "type": "int64", "versions": "0+",
      "about": "The maximum lifetime of this token." },
    { "name": "TokenId", "type": "string", "versions": "0+",
      "about": "The token UUID." },
    { "name": "Hmac", "type": "bytes", "versions": "0+",
      "about": "HMAC of the delegation token." },
    { "name": "ThrottleTimeMs", "type": "int32", "versions": "0+",
      "about": "The duration in milliseconds for which the request was throttled due to a quota violation, or zero if the request did not violate any quota." }
  ]
}
```

Field	**Description**
`Token requester`	Token requester is an Kafka PrincipalType+name string, who requested this token.

DescribeDelegationTokenResponse

We will bump the version of the DescribeTokensRequest API to include the "Token requester" info in response details.

```
{
  "apiKey": 41,
  "type": "response",
  "name": "DescribeDelegationTokenResponse",
  // Starting in version 1, on quota violation, brokers send out responses before throttling.
  // Version 2 adds flexible version support
  // Version 3 adds the token requester details
  "validVersions": "0-3",
  "flexibleVersions": "2+",
  "fields": [
    { "name": "ErrorCode", "type": "int16", "versions": "0+",
      "about": "The error code, or 0 if there was no error." },
    { "name": "Tokens", "type": "[]DescribedDelegationToken", "versions": "0+",
      "about": "The tokens.", "fields": [
      { "name": "PrincipalType", "type": "string", "versions": "0+",
        "about": "The token principal type." },
      { "name": "PrincipalName", "type": "string", "versions": "0+",
        "about": "The token principal name." },
      { "name": "TokenRequesterPrincipalType", "type": "string", "versions": "3+",
        "about": "The principal type of the requester of the token." },
      { "name": "TokenRequesterPrincipalName", "type": "string", "versions": "3+",
        "about": "The principal type of the requester of the token." },
      { "name": "IssueTimestamp", "type": "int64", "versions": "0+",
        "about": "The token issue timestamp in milliseconds." },
      { "name": "ExpiryTimestamp", "type": "int64", "versions": "0+",
        "about": "The token expiry timestamp in milliseconds." },
      { "name": "MaxTimestamp", "type": "int64", "versions": "0+",
        "about": "The token maximum timestamp length in milliseconds." },
      { "name": "TokenId", "type": "string", "versions": "0+",
        "about": "The token ID." },
      { "name": "Hmac", "type": "bytes", "versions": "0+",
        "about": "The token HMAC." },
      { "name": "Renewers", "type": "[]DescribedDelegationTokenRenewer", "versions": "0+",
        "about": "Those who are able to renew this token before it expires.", "fields": [
        { "name": "PrincipalType", "type": "string", "versions": "0+",
          "about": "The renewer principal type" },
        { "name": "PrincipalName", "type": "string", "versions": "0+",
          "about": "The renewer principal name" }
      ]}
    ]},
    { "name": "ThrottleTimeMs", "type": "int32", "versions": "0+",
      "about": "The duration in milliseconds for which the request was throttled due to a quota violation, or
zero if the request did not violate any quota." }
  ]
}
```

Field	Description
`Token requester`	Token requester is an Kafka PrincipalType+name string, who requested this token.

AdminClient API Changes

CreateDelegationTokenOptions class will be updated to take "owner" principal as input.

```
AdminClient {
    //create delegation token with supplied options
        public abstract CreateDelegationTokenResult createDelegationToken(CreateDelegationTokenOptions options)
}

public class CreateDelegationTokenOptions extends AbstractOptions<CreateDelegationTokenOptions> {
    private long maxLifeTimeMs = -1;
    private List<KafkaPrincipal> renewers =  new LinkedList<>();
    private Optional<KafkaPrincipal> owner = Optional.empty(); // New

    public CreateDelegationTokenOptions owner(KafkaPrincipal owner) {  // New
        this.owner = Optional.of(owner);
        return this;
    }

    public CreateDelegationTokenOptions renewers(List<KafkaPrincipal> renewers) {
        this.renewers = renewers;
        return this;
    }

    public Optional<KafkaPrincipal> owner() {
        return owner;
    }

    public List<KafkaPrincipal> renewers() {
        return renewers;
    }

    public CreateDelegationTokenOptions maxlifeTimeMs(long maxLifeTimeMs) {
        this.maxLifeTimeMs = maxLifeTimeMs;
        return this;
    }

    public long maxlifeTimeMs() {
        return maxLifeTimeMs;
    }
}
```

Update TokenInformation class to include "token request" details.

```
public class TokenInformation {

    private KafkaPrincipal owner;
    private KafkaPrincipal tokenRequester; // New
    private Collection<KafkaPrincipal> renewers;
    private long issueTimestamp;
    private long maxTimestamp;
    private long expiryTimestamp;
    private String tokenId;
    ....
    ....
    ....
}
```

ACL Changes

Resources

A new resource called "User" will be added that represents a user which is available to the Authorizer as a Resource.

Operations

We like to add two new Operations "CreateTokens", "DescribeTokens" on User resource, to allow users create token for other users and describe others tokens.

Owners/renewers/token requester principals can always renew/expire/describe their own tokens.

Operation	Resource	API
CreateTokens	User	createTokens for other users **// New**
DescribeTokens	User	describeTokens for others tokens **// New**
Describe	DelegationToken	describeTokens for a given tokenId // Existing

DelegationTokenCommand Changes

We will allow kafka-delegation-tokens.sh script with "--create" option to take owner principal from "--owner-principal" option.

```
>> bin/kafka-delegation-token.sh --bootstrap-server broker1:9092 --create -owner-principal User:owner1 --
renewer-principal User:renewer1 --max-life-time 1486750745585
```

AclCommand Changes

To represent the new User resource type we have to modify the AclCommand slightly and add a new option called `--user-principal` . This represents a user principal of principal type "User". By specifying this parameter we would control (allow or deny) the token requester principal to create or describe tokens for the user-principal.

For instance:

```
>> bin/kafka-acls.sh --authorizer-properties zookeeper.connect=localhost:2181 --add --allow-principal User:
tokenRequester --allow-host * --operation CreateTokens --user-principal "owner1"
```

Protocol Changes

The version of CreateAcl, DescribeAcl and DeleteAcl will be increased to avoid serialization errors in case of older brokers which can't handle the newly added User resource type. This way the client can reject a request that the broker doesn't support.

Proposed Changes

Create/Renew Tokens:

Token requester users with 'CreateTokens' permission on 'User' Resource can create or renew tokens for other users which are authorized by the ACL. The token requester must be authenticated using any of the available secure channels (Kerberos, SCRAM, SSL) to create or renew tokens for other users. The token requester can not use delegation token based authentication for creating or renewing tokens.

Describe Tokens:

Users with 'DescribeTokens' permission on User resource can describe others tokens which are authorized by the ACL. A token can also be described if the user has a Describe permission on the DelegationToken resource.

Token Details in Zookeeper

Token details properties storage format version will be updated to 2.

Delegation Token Details

```
//Delegation Token Details for tokenID token123: Zookeeper persistence path /tokenauth/tokens/token123
{
   "version":2,  // New version
   "owner" : "owner",
   "tokenRequester": "tokenRequester" // New
   "renewer" : "renewer",
   "issueTimestamp" : "issueTimestamp",
   "maxTimestamp" : "maxTimestamp",
   "expiryTimestamp" : "expiryTimestamp",
   "tokenID" : "UUID",
};
```

Compatibility, Deprecation, and Migration Plan

Older version of the the modified CreateTokenRequest APIs will continue to work as expected. When using older API, owner and token requester principals will be set to same.