KIP-902: Upgrade Zookeeper to 3.8.2

Status

Current state: *"Accepted"*

Discussion thread: *here*

Voting thread: here

JIRA: *here*

PR: *here*

Motivation

Kafka has a dependency on Zookeeper 3.6.3, which reached its end of life in December 2022. We would like to upgrade Zookeeper to version 3.8.2 which is the latest release of the 3.8.x versions.

Zookeeper 3.8.2 server supports clients no older than 3.5.x (i.e. Kafka version 2.4.0) and Zookeeper 3.8.2. clients support server versions no older than 3.5.x (i.e. Kafka version 2.4.0).

```
ZooKeeper clients from 3.5.x onwards are fully compatible with 3.8.x servers.
The upgrade from 3.6.x and 3.7.x can be executed as usual, no particular additional upgrade procedure is needed.
ZooKeeper 3.8.x clients are compatible with 3.5.x, 3.6.x and 3.7.x servers as long as you are not using new
APIs not present these versions.
```

In comparison, Zookeeper 3.6.3 server supports clients no older than 3.4.x and Zookeeper 3.6.3 clients support server versions no older than 3.5.x

```
ZooKeeper clients from 3.4 and 3.5 branch are fully compatible with 3.6 servers.
The upgrade from 3.5.7 to 3.6.0 can be executed as usual, no particular additional upgrade procedure is needed.
ZooKeeper 3.6.0 clients are compatible with 3.5 servers as long as you are not using new APIs not present in
3.5.
```

Public Interfaces

No public interfaces are being changed.

Proposed Changes

Similarly to https://github.com/apache/kafka/pull/12620/files we would like to upgrade to 3.8.2.

Compatibility, Deprecation, and Migration Plan

- *What impact (if any) will there be on existing users?*

Users of Kafka clusters with Zookeeper clients older than 3.5.x won't be able to communicate with a 3.8.2 Zookeeper cluster. As mentioned in the accompanying JIRA ticket Kafka has been using Zookeeper clients 3.5.x since version 2.4 so versions above and including it should be safe for this upgrade. It is acceptable to break compatibility with Kafka versions prior to 2.4 as they are considered beyond their end of life and are not maintained. (source: Time Based Release Plan#WhatIsOurEOLPolicy).

A Zookeeper cluster is separate from a Kafka cluster.

It is worth mentioning that Kafka is still tested for cluster upgrades between versions 0.8.2 and 3.4.

The table below makes the assumption you want to carry out the upgrades using a rolling restart

	Zookeeper cluster (< 3.5)	**Zookeeper cluster (>= 3.5)**
Kafka cluster (< 2.4)	1. Upgrade the Kafka cluster to version >= 2.4 2. According to Upgrading to 3.5.0 you will need to carry out a rolling upgrade to Zookeeper 3.4.6 before upgrading the Zookeeper cluster to version 3.8.2	1. Upgrade the Kafka cluster to version >= 2.4 2. Upgrade the Zookeeper cluster to version 3.8.2
Kafka client (< 2.4)	1. Upgrade the Kafka clients to >= 2.4. 2. According to Upgrading to 3.5.0 you will need to carry out a rolling upgrade to Zookeeper 3.4.6 before upgrading the Zookeeper cluster to version 3.8.2	1. Upgrade the Kafka clients to >= 2.4. 2. Upgrade the Zookeeper cluster to version 3.8.2
Kafka cluster (>= 2.4)	1. According to Upgrading to 3.5.0 you will need to carry out a rolling upgrade to Zookeeper 3.4.6 before upgrading the Zookeeper cluster to version 3.8.2	1. Upgrade the Zookeeper cluster to version 3.8.2
Kafka client (>= 2.4)	1. According to Upgrading to 3.5.0 you will need to carry out a rolling upgrade to Zookeeper 3.4.6 before upgrading the Zookeeper cluster to version 3.8.2	1. Upgrade the Zookeeper cluster to version 3.8.2

This is a compatibility matrix between Kafka and Zookeeper versions.

The two coloured rectangles indicate versions of Kafka and Zookeeper which are compatible with one another. When a new version of Kafka having a dependency on Zookeeper 3.8.2 is released the bridge release will be any of the version where the two squares intersect.

		Zookeeper				
		3.3.x	3.4.x	3.5.x	3.6.x	3.8.x
Kafka	3.5.x					x
	3.4.x				x	
	3.3.x				x	
	3.2.x				x	
	3.1.x				x	
	3.0.x				x	
	2.8.x			x		
	2.7.x			x		
	2.6.x			x		
	2.5.x			x		
	2.4.x			x		
	2.3.x		x			
	2.2.x		x			
	2.1.x		x			
	2.0.x		x			
	1.1.x		x			
	1.0.x		x			
	0.11.0.x		x			
	0.10.2.x		x			
	0.9.0.x		x			
	0.8.x	x				

Below is an analysis of changes introduced in 3.7.0 and 3.8.0 and whether they will introduce breaking changes in Kafka or not.

For reference, these are the configurations that Kafka passes onto Zookeeper clients (https://github.com/apache/kafka/blob/trunk/core/src/main/scala/kafka/server/KafkaConfig.scala#L285):

"zookeeper.connect"
"zookeeper.session.timeout.ms"
"zookeeper.connection.timeout.ms"
"zookeeper.set.acl"
"zookeeper.max.in.flight.requests"
"zookeeper.ssl.client.enable"
"zookeeper.clientCnxnSocket"
"zookeeper.ssl.keystore.location"
"zookeeper.ssl.keystore.password"
"zookeeper.ssl.keystore.type"
"zookeeper.ssl.truststore.location"
"zookeeper.ssl.truststore.password"
"zookeeper.ssl.truststore.type"
"zookeeper.ssl.protocol"
"zookeeper.ssl.enabled.protocols"
"zookeeper.ssl.cipher.suites"
"zookeeper.ssl.endpoint.identification.algorithm"
"zookeeper.ssl.crl.enable"
"zookeeper.ssl.ocsp.enable"

Notable Kafka-related changes in Zookeeper 3.7.0

- ZOOKEEPER-3959 - Getting issue details... STATUS - Zookeeper now allows multiple super users. Kafka does not pass on the value of zookeeper.superUser to its Zookeeper client so this change should not affect the proposed upgrade.
- ZOOKEEPER-3301 - Getting issue details... STATUS - Quotas which were previously logged but not enforced are now enforced. These quotas have to do with create/update/delete etc. operations. This will affect Kafka users who put quotas in their Zookeeper clusters as they will become enforceable.
- ZOOKEEPER-3482 - Getting issue details... STATUS - Kerberos authentication over SSL is now supported. Kafka does not support Kerberos authentication with Zookeeper so this change should not affect the proposed upgrade.
- ZOOKEEPER-3561 - Getting issue details... STATUS - User enforced authentication was only available for SASL before this change. Now user enforced authentication extends to all other types of authentication supported by Zookeeper. The point of this change is that no additional ACLs are needed to prevent unauthenticated access if one authentication method is enabled. This will affect users who used unauthenticated users while having an authentication mechanism.

Notable Kafka-related changes in Zookeeper 3.8.0

- ZOOKEEPER-4396 - Getting issue details... STATUS - Zookeeper used plaintext passwords for its trust and key stores. This change makes files which store those passwords take precedence, but they don't remove the already working logic. This change should not affect the proposed upgrade.

No other Zookeeper APIs which Kafka uses appear to have been changed.

- *If we are changing behavior how will we phase out the older behavior?* It should gradually be phased out as users update their Kafka versions
- *If we need special migration tools, describe them here.* N/A
- *When will we remove the existing behavior?* N/A

Test Plan

We ran the following test on the latest trunk of Kafka with Zookeeper 3.6.3 and Zookeeper 3.8.2:

1) Start 1 Zookeeper node on an m5.4xlarge machine

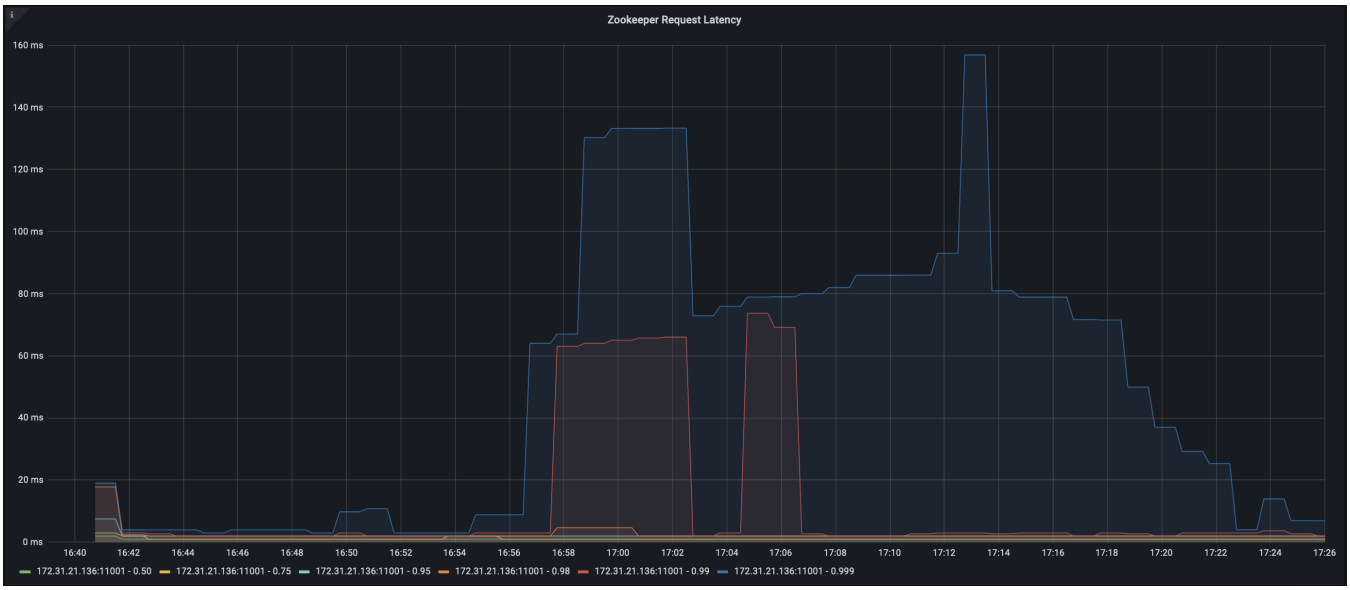
2) Start 1 Kafka broker on a different m5.4xlarge machine

3) Using 4 admin clients sequentially create up to 2000 topics with 1 partition

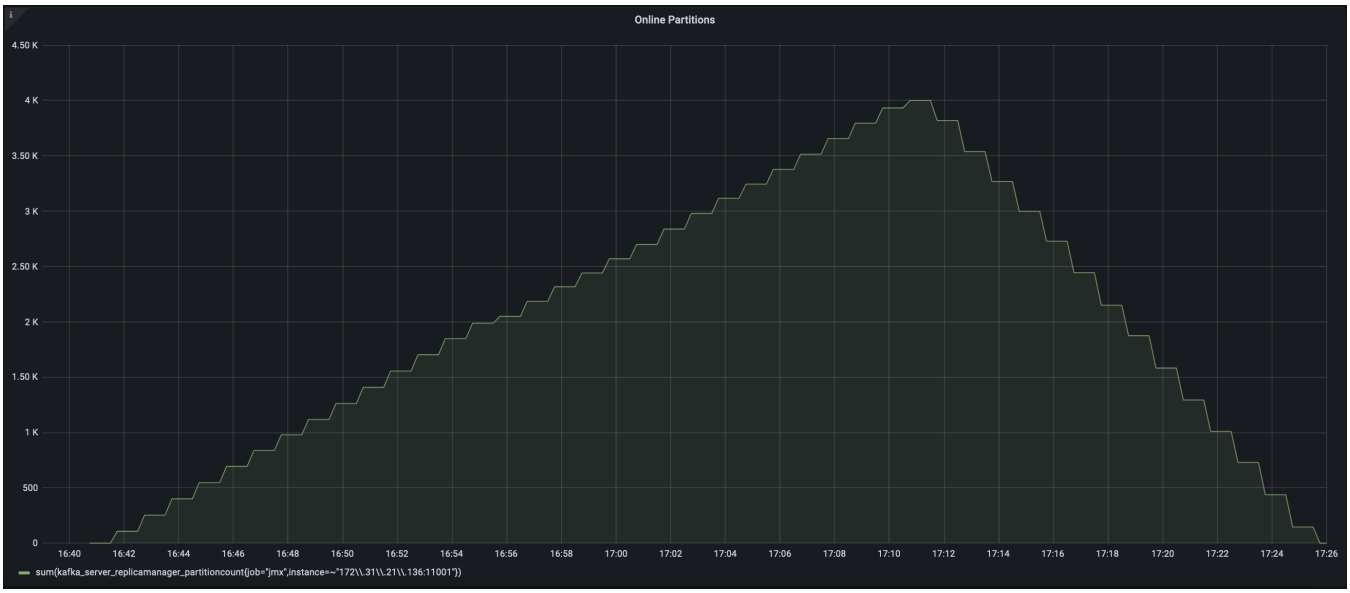
4) Using 4 admin clients sequentially change the number of partitions on all 2000 topics to 2

5) Using 4 admin clients sequentially delete all topics

Zookeeper 3.8.2 request latency (PROPOSED)

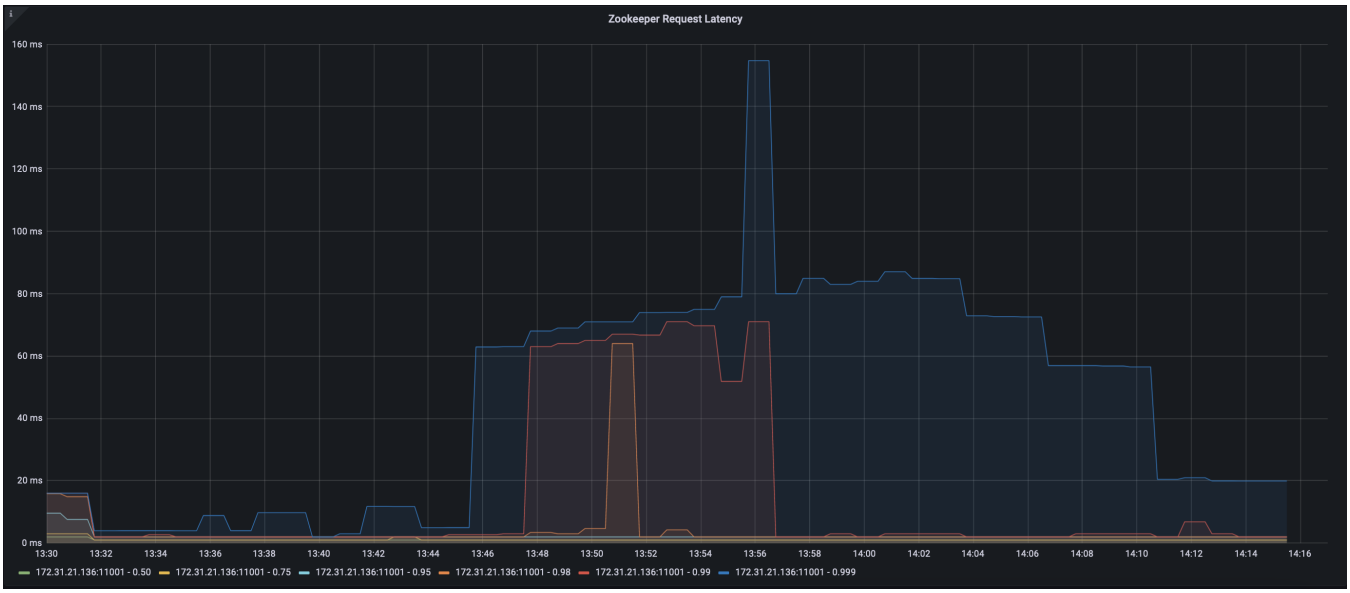


https://g-576b9cd7b5.grafana-workspace.us-east-1.amazonaws.com/dashboard/snapshot/zmXo3V1hC7MkNsGTfD2lZNe6AREk2DwZ?orgId=1

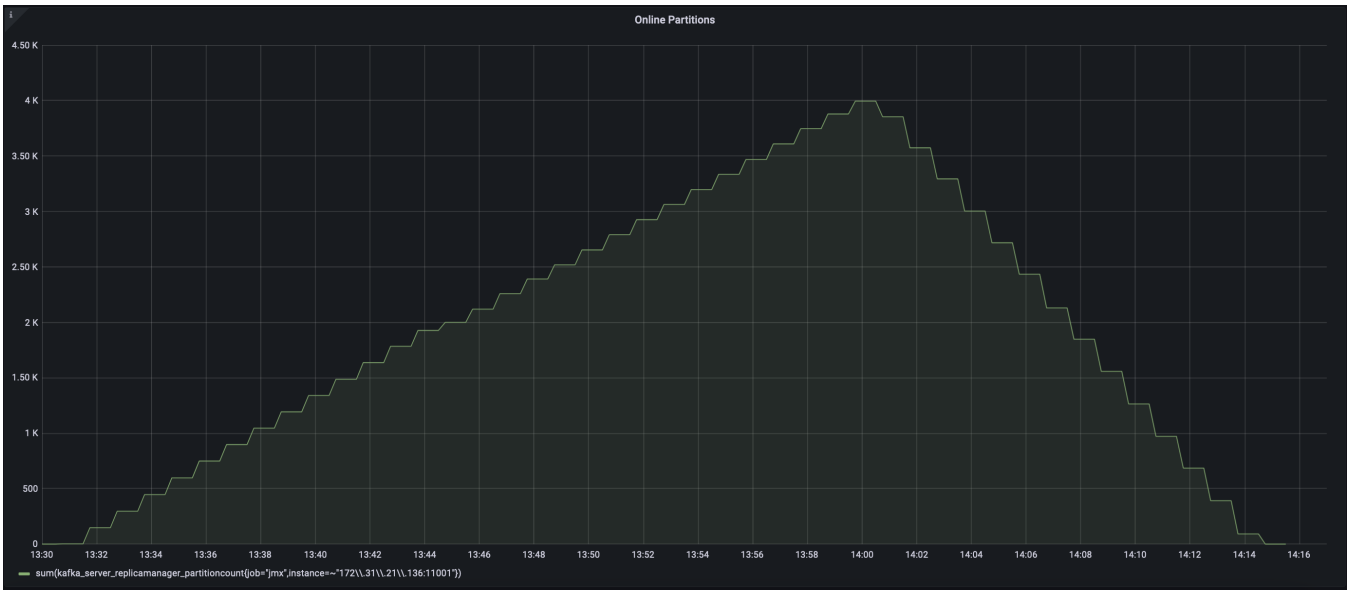


https://g-576b9cd7b5.grafana-workspace.us-east-1.amazonaws.com/dashboard/snapshot/naH2y9jFJsg9greTv72DUu22Q0WvVE2P?orgId=1

Zookeeper 3.6.3 request latency (CURRENT)



https://g-576b9cd7b5.grafana-workspace.us-east-1.amazonaws.com/dashboard/snapshot/yF2EoSNMeSK7BALUdSmqOPAv6QljK2Yn?orgId=1



https://g-576b9cd7b5.grafana-workspace.us-east-1.amazonaws.com/dashboard/snapshot/Vgqt3I8OuPm9upqNvpvMBTs5TrtFY2k5?orgId=1

Rejected Alternatives

If there are alternative ways of accomplishing the same thing, what were they? The purpose of this section is to motivate why the design is the way it is and not some other way. N/A