

KIP-302 - Enable Kafka clients to use all DNS resolved IP addresses

- [Status](#)
- [Motivation](#)
- [Public Interfaces](#)
 - [Client configuration](#)
- [Compatibility, Deprecation, and Migration Plan](#)
- [Rejected Alternatives](#)

Status

Current state: *"Adopted"*

Discussion thread: [here](#)

Vote thread: [here](#)

JIRA:

A screenshot of a JIRA error message. It features a yellow warning triangle icon on the left, followed by the text "Unable to render Jira issues macro, execution error." in a standard sans-serif font.

Released: 2.1.0

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

co-authored-by: [Mickael Maison](#) <mickael.maison@gmail.com>

Motivation

Currently the Kafka clients - in `org.apache.kafka.clients.NetworkClient.initiateConnect()` - resolve a symbolic hostname using :

```
new InetSocketAddress(String hostname, int port)
```

which only picks one IP address even if the DNS has multiple A records for the hostname, as it in turn calls :

```
InetAddress.getAllByName(hostname)[0]
```

For some environments, where the broker hostnames are mapped by the DNS to multiple IPs, it is desirable that clients, on failing to connect to one of the IPs, try the other ones before giving up the connection.

Our use case is for multiple load balancers fronting the Kafka cluster. The Kafka advertised listeners advertise hostnames for which the DNS server holds multiple A records, corresponding to the IPs of all LBs .

If one LB isn't available, but the client is able to use another IP for the same hostname (it connects to the 2nd LB for example) the service stays available.

Another case would be where brokers are fronted by two proxies in active/standby mode. This KIP would enable using a standby proxy for HA if connection to the active proxy fails.

Although this KIP and [KIP-235: Add DNS alias support for secured connection](#) both deal with multiple DNS records, they address separate concerns.

Public Interfaces

Client configuration

Add a new allowed value for the configuration parameter `client.dns.lookup` introduced by [KIP-235](#)

```
client.dns.lookup = "use_all_dns_ips"
```

The other values for this parameter, including the default, maintain the existing behavior of only attempting to connect to the first resolved IP, so there will be no backwards compatibility issue. Setting the parameter to "use_all_dns_ips" will have the client try to connect to all resolved IPs and will not try the canonical hostname resolution enabled by [KIP-235](#)

Proposed Changes

If the configuration parameter `client.dns.lookup` is set to "use_all_dns_ips" the network client code will use

```
InetAddress.getAllByName(hostname)
```

to obtain from the DNS server all IPs for the broker hostnames.

The NIO client will attempt a connection to one of the IPs. If the connection is refused or times out and the DNS had returned multiple IPs, then rather than failing the connection it will try to connect to one of the other IPs.

If they all fail to connect then the behavior remains like the current client when the only obtained IP had failed to connect:

- at bootstrap, move to the next hostname
- past bootstrap, retry the connection to the given node starting from the resolution of the hostname

Note that the above Java API can return a set containing both IPv4 and IPv6 addresses. The type returned first is determined by the JVM - see the Java System property "java.net.preferIPv6Addresses". This KIP proposes to only use multiple IPs of the same type (IPv4/IPv6) as the first one, to avoid any change in the network stack while trying multiple IPs.

Compatibility, Deprecation, and Migration Plan

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

By default the multiple IP resolution is disabled so there will be no impact. Even when enabled, no switch between IPv6 and IPv4 will be attempted as only one type of address will be used

Rejected Alternatives

- Making the client use all IPs by default as it may have impacted some users
- Introducing a separate configuration entry, independent of the one introduced by [KIP-235](#), so the DNS resolution enhancement suggested by this KIP is alternative and not conflicting with [KIP-235](#)