



Innovative replication management in FreeIPA

Ludwig Krispenz <lkrispen@redhat.com>

Petr Vobornik <pvoborni@redhat.com>

Edinburgh, Nov, 12th, 2015

Replication configuration

Goals:

- Identical data on any server participating in replication
- Concurrent updates and any server

Method in 389-ds:

- Replay modifications between servers (defined by Replication Agreements)
- Perform update resolution to resolve conflicts

Replication Agreements:

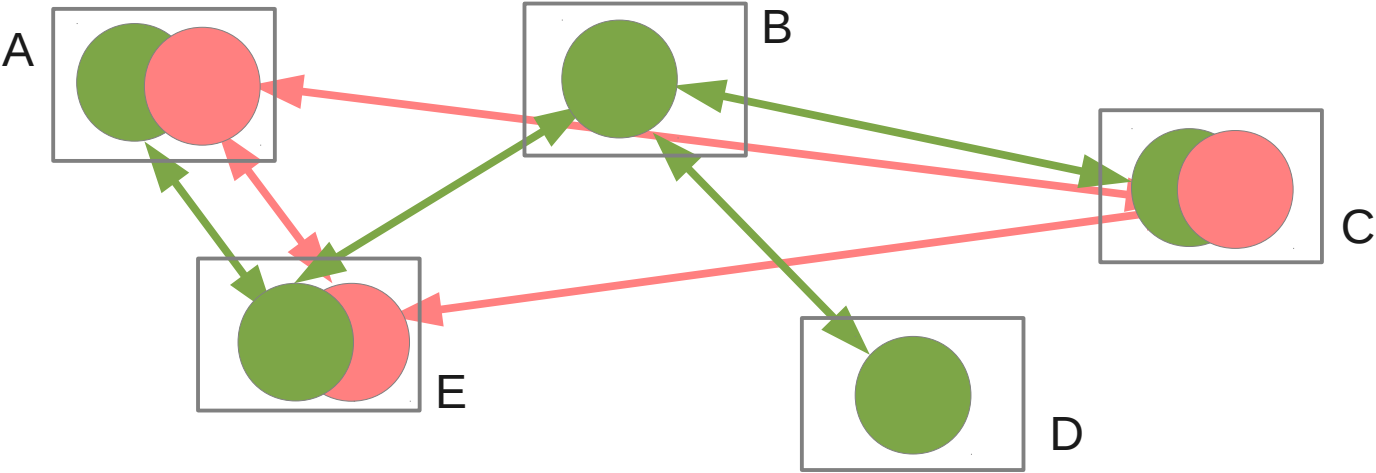
- Target server: replication endpoint for push replication
- Transport: secure, startTLS, unsecure,...
- Authorization: bind dn, principal,....
- Content: fractional replication




Replication agreements are similar on all servers, but

- Number of agreements can differ
- Content definition can differ, eg full replication in one subset, fractional to the rest
- Transport can differ: different security levels between different servers

So, replicating cn=config does not help.

Replication topology



-  A Managed server
-  Replicated suffix
-  Replication connection

Server B is single point of failure in the green topology

Challenges in IPA replication management

Situation:

- two suffixes (or more)
 - one for user, hosts, services, ...
 - one for certificates for certificate servername
 - additional user defined
- one suffix available on all servers, the other on a subset only
- deployments with high number of replicas (20-30 quite common)
- highly dynamic, frequent addition and removal of replicas

Questions:

- are all servers connected ?
- is there a single point of failure ?
- can a server safely be removed without disconnecting the topology ?

Problem: Required information is distributed across the topology

Requirements for replication management

- allow management of the replication topology on a single server
- verify the degree of connectivity
- initiate online initialization from one remote server to another
- add and delete connections between any servers
- check if removal of a segment would disconnect the topology or if a segment to be added already exists.
- simplify the topology management via command line interface and Web UI.
- allow replication monitoring by querying a single server

New replication management

Represent replication configuration in objects in a replicated database

- managed servers, managed suffixes, replication connections
- either in dedicated suffix or suffix available on all servers

==> complete information available on any server

Deploy 389-ds plugin to manage local replication configuration based on data in shared tree.

- config in shared tree is authoritative, overrides local conf
- enables sanity checks for any attempted configuration change

==> topology config is consistent and valid across all servers

Changes to entries in cn=config

Add topology plugin:

```
dn: cn=IPA Toplogy Configuration,cn=plugins,cn=config
objectClass: top
objectClass: nsSlapdPlugin
....
nsslapd-pluginEnabled: on
nsslapd-topo-plugin-shared-config-base: cn=topology,cn=etc,dc=example,dc=com
```

The only required configuration setting is the base entry for topology information

Replication agreements are marked, managed by the topology plugin:

```
dn: cn=meTovm-072.idm.lab,cn=replica,cn=dc\3Dexample\2Cdc\3Dcom,cn=mapping tree,cn=config
objectClass: nsds5replicationagreement
objectClass: top
objectClass: ipaReplTopoManagedAgreement
.....
ipaReplTopoManagedAgreementState: managed agreement - controlled by topology plugin
```

Topology management objects: Managed servers

For each server in the topology an entry defines which suffixes should be managed by the topology plugin.

[Unmanaged suffixes may exist, if instances are used to contain backends without relation to IPA, but they could use the topology management as well.]

```
dn: cn=vm-072.idm.lab,cn=masters,cn=ipa,cn=etc,dc=example,dc=com
objectClass: top
objectClass: nsContainer
objectClass: ipaReplTopoManagedServer
objectClass: ipaConfigObject
cn: vm-072.abgc.idm.lab.eng.brq.redhat.com
ipaReplTopoManagedSuffix: dc=example,dc=com
paReplTopoManagedSuffix: o=ipaca
```


Topology management objects: Suffixes

cn=topology, <topology management based dn> is a container for all managed suffixes

A managed suffix is a container for all replication connections

A managed suffix defines

- Default replication connection settings, eg
 - use GSSAPI
 - strip specific attributes from replication
 - ...
- Connectivity requirements
 - How many independent paths between servers must exist
- Monitoring control
 - Continuous monitoring
 - Ad hoc monitoring requests, which will then be propagated across the topology

```
dn: cn=realm,cn=topology,cn=ipa,cn=etc,dc=abc,dc=example,dc=com
objectClass: top
objectClass: iparepltopoconf
ipaReplTopoConfRoot: dc=example,dc=com
nsDS5ReplicatedAttributeList: xxxxxxxx
```

Topology management objects: Segments

A segment is the representation of one or two replication agreements between two servers

It is defined by:

- Endpoints: leftnode, rightnode (fqdn of two managed servers)
- Connectivity: left-right, right-left, both

In addition, properties of the represented agreements can be set

- Transport
- Fractional attributes
- Init: adding a “refresh” attribute will trigger online initialization, will be reset automatically

Segments are also used in monitoring, as segments can track the status of corresponding agreements

Topology management objects: Example

```
dn: cn=topology,dc=example,dc=com
  └─ objectclass: nsContainer
├─ cn=replica example,cn=topology,dc=example,dc=com
  └─ objectclass: ipaReplTopoConf
     └─ ipaReplTopoConfRoot: dc=example,dc=com
├─ cn=111-to-102,cn=replica example,cn=topology,dc=example,dc=com
  └─ objectclass: ipaReplTopoSegment
     └─ ipaReplTopoSegmentDirection: both
        └─ ipaReplTopoSegmentLeftNode: vm-111.idm.lab
           └─ ipaReplTopoSegmentRightNode: vm-102.idm.lab
              └─ nsds5ReplicaEnabled;right: on
                 └─ nsds5ReplicaEnabled;left: off
                    └─ nsDS5ReplicatedAttributeList: (objectclass=*) $ EXCLUDE krbblastfailedauth krbloginfailedcount
                       └─ nsds5ReplicaStripAttrs;left: modifiersName modifyTimestamp
├─ cn=111-to-191,cn=replica example,cn=topology,dc=example,dc=com
  └─ objectclass: ipaReplTopoSegment
     └─ ipaReplTopoSegmentDirection: left-to-right
        └─ ipaReplTopoSegmentLeftNode: vm-111.idm.lab
           └─ ipaReplTopoSegmentRightNode: vm-191.idm.lab
              └─ ....
├─ cn=replica ipaca,cn=topology,dc=example,dc=com
  └─ objectclass: ipaReplTopoConf
     └─ ipaReplTopoConfRoot: o=ipaca
├─ cn=111-to-191,cn=replica example,cn=topology,dc=example,dc=com
  └─ ipaReplTopoSegmentDirection: both
     └─ ipaReplTopoSegmentLeftNode: vm-111.idm.lab
        └─ ipaReplTopoSegmentRightNode: vm-191.idm.lab
```

Topology plugin operation: startup and preop

Startup

- Build internal graph of replication topology based on data in shared tree
- Compare topology information with local config
 - shared tree is authoritative, remove local agreements not covered by topology
 - In upgrade scenarios or when servers become managed, local config is auto transformed into topology data

Preop

- Reject direct modification of local agreements in cn=config
- Reject adding duplicate connections
- Reject removal of segments which would result in disconnected topology

Topology plugin operation: postop

Successful modification of topology objects triggers update of local configuration

- New segment ==> new agreement(s)
- If segment is one directional and segment for opposite direction exists ==> merge
- If suffix becomes managed and local agreements exist ==> autogenerate segments
- If managed server is removed ==> remove all segments connecting this server ==> remove corresponding agreements

If monitoring request for suffix is received ==> check status of local agreements and update segments

Topology plugin use: monitoring

Replicated topology configuration can be used for replication monitoring.

Scenario for checking: replication status in the complete topology:

- Add monitoring request attribute in managed suffix entry on server X
- This is replicated to all servers
- Each server checks the status of the local agreements and updates the corresponding segments
- Status update will be replicated to all servers, including X
- X can compare received update status with topology graph and determine connection failures

Topology plugin use: graphical user interface

As all topology information is available on all servers and change in topology can be triggered on any server this simplifies the topology management from the command line and from a web interface.

The topology graph can be easily visualized by contacting only one server, update can directly be applied.

< demo >