CoCCA - Recommend Practices | Continuity and Infrastructure

Overview:

In order to mitigate various risks to a TLD registry, the database (postgreSQL) should be:

1. Backed up several times a day using “snapshot” backups. The CoCCA backups use the postgres pg_dump tool, pg_dump backs up the database structure *and* data in a single SQL file (these backups can also be made using the postgres command line, see example below).

   sudo -u postgres [path to pg_dump/pg_dump -b [database name] | gzip -9 > registry-$ (date +%Y-%m-%d-%H.%M.%S).sql.gz

   Example:

   sudo -u postgres /usr/lib/postgresql/14/bin/pg_dump -b registry | gzip -9 > registry-$(date +%Y-%m-%d-%H.%M.%S).sql.gz

   To re-create or restore a registry and import all data from a snapshot backup, one need simply use the following commands.

   Create empty database:

   sudo -u postgres psql postgres
   CREATE database registry;
   \q

   Create db structure and import data:

   psql -U postgres -d registry -f backup.sql

2. Configured to use postgres streaming replication to replicate all transactions in real-time to one or more servers / virtual machines. CoCCA recommends using postgres replication as it does not require users to buy any specific virtualisation tools and is vendor neutral (it will work for all users regardless of OS or virtualisation environment). Users may also supplement the postgres tools with other commercial backup solutions if they wish.

   https://www.postgresql.org/docs/current/warm-standby.html#STREAMING-REPLICATION

3. Where possible, the servers should have both public and private IP addresses, and the replication should be configured to use the private IP ranges.

4. Backup to an offline solution (external drive / USB) for disaster recovery. The CoCCA ROOT.war file that matches the current version used for production should be backup up with the offline db backups. An SQL backup made using the postgres pg_dump tool and the CoCCA application (contained in the CoCCA ROOT.war file, a java zip format) are all that is required to restore or clone the registry.

   Encrypt and backup to the cloud. There should also be a daily backup on infrastructure controlled by a trusted third party (or cloud provider) for disaster recovery. CoCCA has tools to easily encrypt and make ICANN and/or CoCCA formatted escrow deposits to third parties (including ICANN approved escrow agents).

Specific database backup advice:

Excluding escrow arrangements, where possible database should be backed up and replicated up to a minimum three instances under control of the registry operator. Two at the primary site (on different physical appliances) and one at a second site. The CoCCA DR tool will also make snapshot backups and zones at the replication sites.

CoCCA - Recommend Practices | Continuity and Infrastructure

DNS Recommendations:

The master DNS server should be a non-public “hidden master” that is used only to distribute the zone files to the public facing DNS servers. Distribution should be secured using TSIG and IP allow lists. Only trusted IP should be notified / able to pull the zones.

Multiple DNS anycast clouds run by different vendors should be used to compliment one or more unicast DNS servers under the direct control of the TLD manager. It is recommended that at least one public facing DNS server in the ROOT be under the direct control of the TLD manager.

To simplify the DNS configuration and DNSSEC signing, all registry zones that are delegated to the TLD servers in the ROOT should be compressed into the top level zone . This makes it possible to add or remove zones without co-ordination / re-configuration with DNS providers, it also means that for DNSSEC, only one zone needs to be signed.

What TTL to use ? There is some debate re this issue, CoCCA recommends following the advice in the following RIPE document.

https://labs.ripe.net/author/giovane_moura/how-to-choose-dns-ttl-values

Public facing WHOIS and RDAP services:

Where possible and practical, WHOIS and RDAP services (which require only read access to the database), should be run on a read-only replicated copy of the database. Using a read only postgres user, on a read only db, minimise the risk of unauthorised access to the registry from public facing services.

Keep it simple:

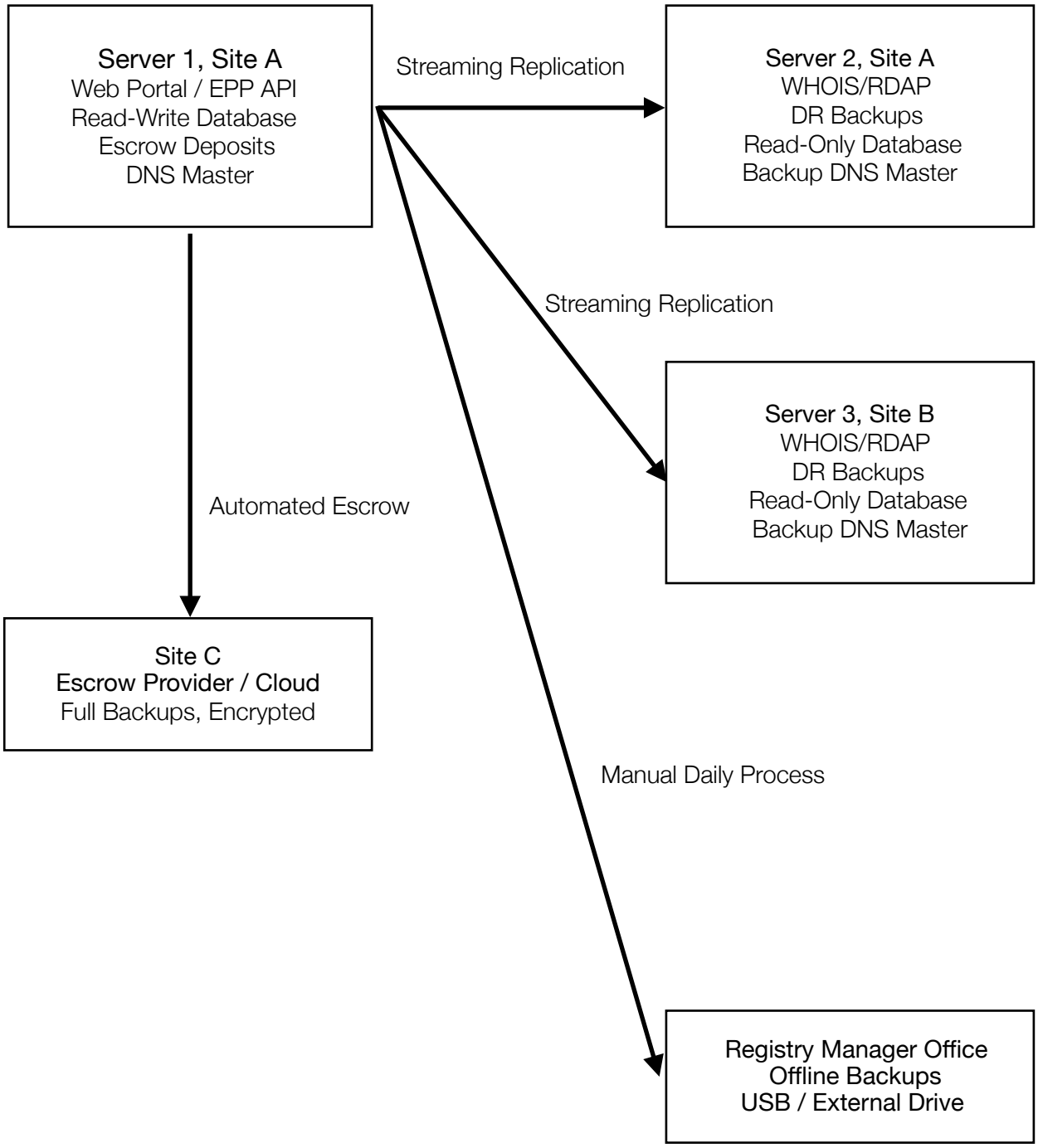
Continuous availability is critical on the public facing DNS services, this can be achieved by having multiple TLD DNS servers some operated by the TLD manager and some by third parties.

High availability or any kind of automated failover of the registry systems is not critical and likely to cause more problems than it solves. A temporary outage of the registry does not impact resolution of already registered domains, it simply prevents create and update transactions.

The decision initiate a failover — make one of the replicated registry instances the primary, will always require a review of the issue that caused the outage and manual intervention. Once a decision is taken to failover from one instance site to another, the process only takes a few mins.

The process:

- Stop Resin on the failover site.
- make a read-only postgres instance that contains a replicated copy of the database a read-write (stop postgres, start postgres)
- Change the IP address A / AAAA records of the registry and whois portal — or if the issue is appliance failure at the primary site, and you have a second appliance there with replicated data, simple more the IPs to the second appliance.
- Start Resin


Server 1, Site A
Web Portal / EPP API
Read-Write Database
Escrow Deposits
DNS Master
Streaming Replication
Server 2, Site A
WHOIS/RDAP
DR Backups
Read-Only Database
Backup DNS Master
Streaming Replication
Server 3, Site B
WHOIS/RDAP
DR Backups
Read-Only Database
Backup DNS Master
Automated Escrow
Site C
Escrow Provider / Cloud
Full Backups, Encrypted
Manual Daily Process
Registry Manager Office
Offline Backups
USB / External Drive