SUSE® Manager High-Availability HOW-TO

TUT 7951

Jeff Price
Principal Architect - SUSE Consulting
jprice@suse.com

Agenda

- Overview
- Objectives - Goals for HA enabled infrastructure
- HA Designs - Active-Passive, Active-Active
- Database Concerns - HA for DB
- Software Components – SLES®, SLESHA, SUSE® Manager
- Environment Setup - Storage, OS, Network, SUMa, HAE…
- Results - Tests and Evaluation, Deployment
- Links - For more Information
- Q & A

Logistics…

- Cell Phones - silent
- Laptops for notes
- Bathrooms
- General Information

Overview

Question : Why are you here…what is this all about?

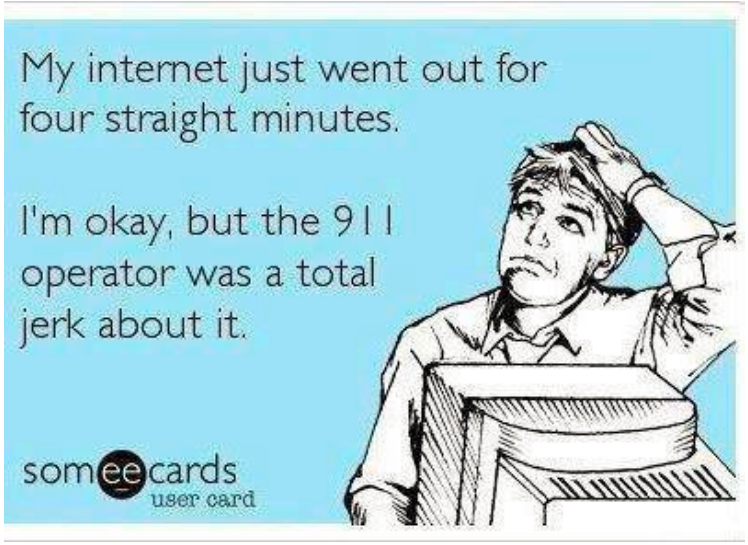
- How important is your key infrastructure?
- LOB owners demand high-availability for their applications, what about the tools that manage it?
- Do critical patches or updates require a critical uptime level for your delivery infrastructure?
- How good are YOUR sneakers?

Sneaker Net?

Infrastructure - can’t live without it.

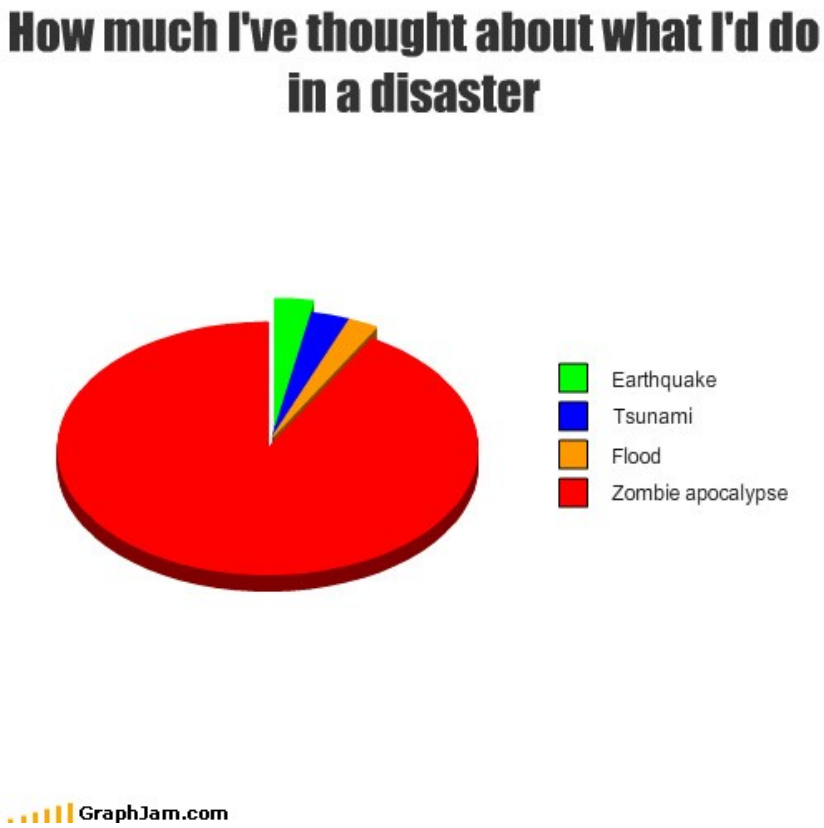
- SUSE® Manager is “infrastructure”.

- Although there are times when critical infrastructure becomes “unavailable”, we take considerable steps to avoid those “times”.
- Power, Water, Heat…
- …INTERNET?!?!



IRL - Be Realistic...

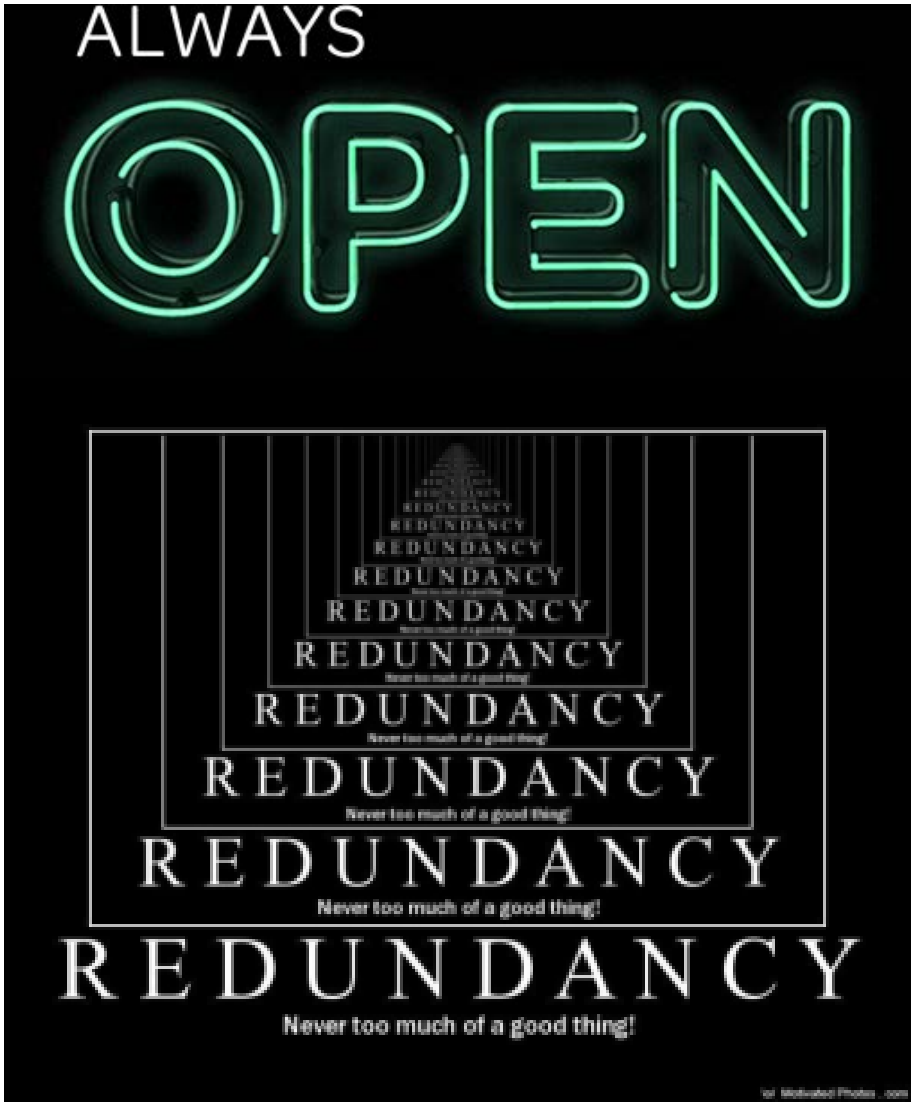
- SUSE® Manager can be made resilient without going to extremes
- Backup your DB and package-store (/var/spacewalk)
- A good plan should including testing…
- …Testing is always a good part of a plan



Always Open!

- A perfect combination :
  - SUSE® Manager (1.7- 2.1)
  - SUSE® Linux Enterprise Server 11 (sp2-sp3)
  - SUSE® Linux Enterprise Server 11 - High Availability Extension (sp2-sp3)

  Future:
  - SUSE® Cloud/Storage ??



Objectives

Fatherly Wisdom…

- Dad always said, “Plan your work, THEN work your plan”.
- Not all HA designs are the same - nor are the requirements.
- What are your uptime goals and BUDGET?
- Uptime has a “cost”.
- Cost, Quality and Speed … pick 2.

HA Terms:

- Active / Passive
- Active / Active
- Scale UP
- Scale OUT
- Cluster
- Load Balancer
- Geo / Site / Metro

Part Smart, Part Lucky

"…building highly available sites is part architecture and part luck. You have to design for failures and hope that when failures occur they are the types of failures you anticipated and built for."

Nobody can ever provide 100% uptime and nobody should ever expect it either.

"***Everything can and will fail. You have to design for failure but every failure is different. Sometimes you have an answer for the failure(s) that occur and sometimes you may not***."

http://www.kavistechnology.com/blog/availability-is-all-about-trade-offs-a-story-about-uptime/

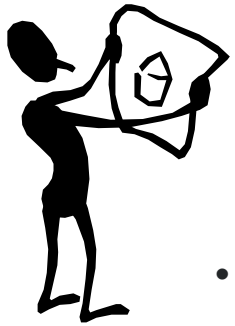
Don’t forget to bring a FULL bottle of PATIENCE!!

- Populating your channels with packages can take a ***very long time*** depending on the number of channels and the number of packages IN them
- The RES 6 (expanded support channel) has about **20,000 PACKAGES!!!**
- **“ARE WE THERE YET?!?!”**

HA Design Patterns

“Plan Your Work” - Decisions, Decisions…

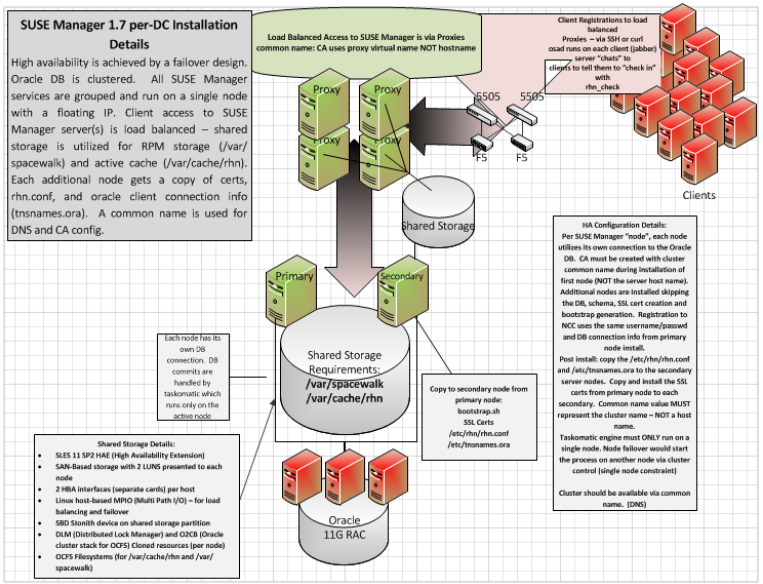
- Consider the number of hosts and their importance…
  - General Servers
  - Line-of-Business
  - SAP
  - Critical Systems
- What is the uptime target/SLA?
- Where is the weak link in the infrastructure chain?

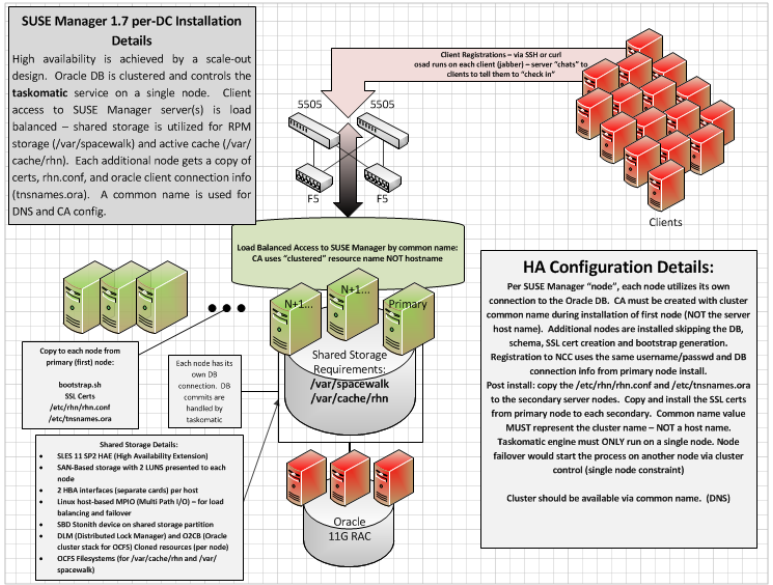


- What is SUSE® Manager responsible for?
  - Patch Management
  - Configuration Management
  - Provisioning
  - Monitoring
- Do you have SUSE Manager Proxies?
- Do all dependent components have the same SLA?
  - Set expectations appropriately

HA Examples - Part of the “plan”…

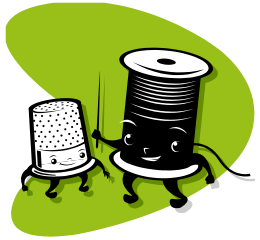
- Active / Passive - “Failover / Redundant”
- Active / Active - Scale Out / N+1





Common Threads… (Storage & Network)

- Each design has “shared storage” requirements for several of the key components:
  - Database
  - Packages (/var/spacewalk and /var/cache/rhn)
  - Squid cache (SUSE® Manager Proxies)



- Every design has network resiliency as part of the “plan”
  - Load balancers
  - Split-Layer2 (pathing/redundancy)
  - Bonding
  - SSL Certs that leverage “subject alternative names” - aliases

Common Threads… (Database & Cluster)

- Database HA is also a concern and potentially a cost
  - Oracle RAC presents some challenges - both from a cost perspective and configuration/setup
  - Consider PostgreSQL as a viable alternative - it reduces cost but not necessarily the complexity
  - Shared storage - sufficient for active/passive
- Not all applications are “cluster-ready”
  - SUSE Manager is no exception - careful configuration is required to ensure a cluster can effectively control the services that make up the SUSE® Manager application and also the network services.
  - Cluster configuration will include constraints/policies
    - Location and Single Service Instance

Database Concerns

PostgreSQL or Oracle???

Oracle

- Supported database for SUSE® Manager (for now)
- Clustering for Failover or Load-Balancing
- $$$ for the DB
- $$$ for cluster enablement (RAC / Grid)
- Become good friends with an Oracle DBA
- UTF8 is still required and is NOT the same thing as AL32UTF8
- Setup SUSE Manager (phase 2) using a single node/vip in /etc/tnsnames.ora (SID = vs. SERVICE_NAME =)
- Script the installation (see appendix)

PostgreSQL

- Default database in SUSE® Manager Appliance
- Low-cost option
- On-par performance with Oracle
- Variety of Clustering Designs
  - Failover
    - Warm Standby (easiest) - consider as an option for continuous backup too
    - DRBD
  - Load Balancing
    - Hot Standby
    - Synchronous (pgpool-I or pgpool-II)
    - or Asynchronous (slony-I, Bucardo or Londiste)
    - Memcached for enhancing performance

Software Components

Putting it together…

- Finalized Design
  - Physical or Virtual
  - Network Caveats
  - Storage layout
    - Shared
    - Parallel
    - Mirrored
  - Database HA setup
  - Software Components
    - SLES®
    - SLES HAE
    - SUSE® Manager
- Installation Details
  - SUSE Manager Activation Code
  - Mirror Credentials
  - Database Details
    - DNS, IP (vip), id/password, port
  - HTTP Proxy Creds
  - NTP Server
  - DNS Info
  - IP Addresses (nodes, vip)
  - Inbound/Outbound Firewall info for network team

Active/Passive

Remote DB Example:

- Setting up remote postgresql is pretty straightforward:
  - Install SLES® 11 SP3 with postgresql91 (not 8)
  - Register and Update the server (post install)
  - Add SUSE® Manager repos (manually OR additional registration command)
  - suse_register -a email=<your email> -a regcode-sms=<reg key>
  - Install postgresql91-pltcl from SUSE Manager Pool repo
  - https://nu.novell.com/repo/$RCE/SUSE-Manager-Server-2.1-Pool/sle-11-x86_64/rpm/x86_64/postgresql91-pltcl-9.1.12-0.3.1.x86_64.rpm
  - Install pgtune from SUSE Manager Pool repo
  - https://nu.novell.com/repo/$RCE/SUSE-Manager-Server-2.1-Pool/sle-11-x86_64/rpm/noarch/pgtune-0.9.3-0.10.2.noarch.rpm

Remote DB Example:

- Start and stop the PostgreSQL server to initialize the data directory contents:
  - rcpostgresql start
  - rcpostgresql stop
- Modify the **/var/lib/pgsql/data/postgresql.conf** file to allow remote connections (owned by postgres:postgres (600)):
  - Modify **listen_addresses = 'localhost'** to allow connections from other hosts

  The simplest (but least secure) approach is to make it **listen_addresses = '0.0.0.0'**
- Modify the **/var/lib/pgsql/data/pg_hba.conf** file to allow remote connections (owned by postgres:postgres (600)):
  - Make it look something like this (keeping in mind that what is **easy** is not necessarily **secure**) :

Remote DB Example:

- Initialize a susemanager DB on the remote Postgres server. This will be populated when we start “phase 2” setup of the SUSE® Manager first node:

```
su - postgres -c 'PGPASSWORD=susemanager;
createdb -E UTF8 susemanager ; createlang
plpgsql susemanager ; createlang pltclu
susemanager ; yes $PGPASSWORD | createuser -P
-sDR susemanager’
```

You may get an error regarding a character set already existing, but this can safely be ignored.

The database should now be ready for the SUSE Manager phase 2 setup (yast susemanager_setup)

Active Passive - General Steps…

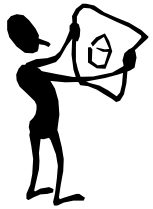
- Install SUMA on Node1
  - Phase 1 Installs on both nodes (before running yast susemanager_setup)
  - Patch (zypper -n up -l)
  - Setup storage
    - LVM lvs for database, packages and apache
  - Phase 2 on primary node (yast susemanager_setup)
  - Finalize DNS
  - Turn off SUSE® Manager services (spacewalk-service disable)
- Repeat for Node 2
  - Phase 1 Installs on both nodes (before running yast susemanager_setup)
  - Patch (zypper -n up -l)
  - Phase 2 on secondary node (yast susemanager_setup)
  - Stop SUSE Manager (spacewalk-service stop and rcpostgresql stop)
  - Clean up install - removing/recreating directory content for packages, database and apache

Active Passive - General Steps…(cont.)

- Disable SUSE® Manager services on node 2
  - Spacewalk-service disable
  - Finalize DNS
- Pre-Cluster preparation :
  - SSL preparation, Apache and jabber modifications
  - Copy node 1 data to alternate node (2)
- Cluster Configuration
  - Install SLEHA on each node
- Add SLEHA repos and patch again
- Setup stonith and corosync configurations
- Configure LVM for cluster volumes
- Start openais
- Configure cib with primitives, groups and constraints
- See appendix

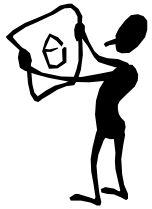
Active/Active

Active/Active - Plan Ahead…



- Scale out clusters need special consideration and tuning.
  - PostgreSQL connections : depending on the client contact method (SSH, OSAD) doing a large patch operation can overload the number of available DB connections. Using pgtune can help
  - Taskomatic is the job scheduler in SUSE® Manager (think cron). It is java-based and consumes resources as such. JVM tuning might be needed and can impact available resources (see above). This service should only be run on 1 node as the job queue is stored within the DB.
  - The SUSE Manager web application leverages Tomcat. As such you might need to tune yet another JVM and again consider the available resources (see above and above)…

Active/Active - Plan Ahead… (continued)



- SSL over load balanced networks can be problematic - carefully review load balance designs and SSL stickness capabilities - and set expectations ahead of time
- SSL configuration for clustering can be painful
- “Officially” - SUSE® supports a self-signed server infrastucture and using external certs and even setting up for clustering can be tricky - host renames are also not supported
- Use mgr-ssl-tool to create CA and server certs and add node names and cluster name as “subject alt-names”
- Copy certs, keys and pem into place (see appendix)
- Cert Locations:
  - /etc/ssl/servercerts/
  - /etc/apache2/ssl.crt/
  - /etc/ssl/private/
  - /etc/apache2/ssl.key/
  - /etc/apache2/ssl.csr/
  - /etc/pki/spacewalk/jabberd/

Setting it up… DB First

- Define nodes and get your database HA setup first
  - Often takes more time to get the DB cluster setup than the application cluster
  - Become good friends with your DBA (and their boss!) ;-)
  - For Oracle - provide a base installation script and stress the importance of UTF8 (not AL32UTF8!)
  - For Oracle RAC - ask your DBA for a single node to do the first SUSE Manager node installation - tnsnames.ora contains ‘SID = …’ not ‘SERVICE_NAME = …’ and will be modified later
  - Validate installation with DBA - sometimes the DB ‘grants’/’revokes’ need to be checked - this can keep stored procedures from compiling correctly - DBAs can help (see above statement about ‘friends’) ;-)

Setting it up … SUSE® Manager - first node …

- Collect your install information:
  - SLES®, SUMa and SLEHA activation codes
  - IP addresses, http-proxy creds (if needed)
  - SCC Mirroring Credentials
  - DB credentials / ports

- Install the first node using either the SUSE Manager appliance install or a SLES 11 SP3 with HAE add-on.
- Configure alternative storage locations - /var/spacewalk, /var/cache/rhn on shared storage with OCFS2
- Phase 2 installation populates DB

Setting it up… SUSE® Manager - first node (continued) …

- SSL rework begins:
  - Collect all cluster node names/IPs
  - Generate new CA
  - Generate new host certs
  - Deploy new certs as mentioned previously
- Cluster controls
  - OCFS mounts
  - SBD - disk-based stonith
  - DLM - distributed lock manager
  - O2cb - cluster stack for OCFS2 filesystem

OCFS2 Setup

- Make sure all mount points are setup and working prior to adding them to cluster controlled resources.
- mkfs -t ocfs2 -N 2 -L cache /dev/mapper/smrepo_part2
- mkfs -t ocfs2 -N 2 -L spacewalk dev/mapper/smrepo_part3
- DLM and O2CB will run on each node

Taskomatic

- Since the cluster will control the SUSE® Manager services - you want to chkconfig them ‘off’
  - You can do this using ‘spacewalk-service disable’ or chkconfig, but make sure Taskomatic doesn’t get re-enabled… ONLY the cluster should start/stop it
- HA resource for Taskomatic will have constraints to limit it to a single node (it can move, but should only run on a single node) on active/active cluster
- If you ever need to do channel adds or channel maintenance from the command line - find out where taskomatic is running and do it there.

Taskomatic (cont.)

- Customize spacewalk-service script to remove Taskomatic as a grouped service - /usr/sbin/spacewalk-service

BEFORE:

```
SERVICES=“jabberd $SUSE_AUDITLOG $DB_SERVICE
$TOMCAT $HTTPD osa-dispatcher Monitoring
MonitoringScout rhn-search cobblerd
taskomatic”
```

AFTER:

```
SERVICES=“jabberd $SUSE_AUDITLOG $DB_SERVICE
$TOMCAT $HTTPD osa-dispatcher Monitoring
MonitoringScout rhn-search cobblerd”
```

Setup

Phase 2 Setup - Primary Node

- Yast2 susemanager_setup will ask questions to create the CA - make sure you use the cluster name - you might find it easier to go back and recreate it later using the spacewalk tools…
- After the DB is populated, and phase 2 installation completes… shutdown the spacewalk services which have started automatically “spacewalk-service stop”
- This won’t shutdown the DB - leave it running…
- For Oracle - copy the /etc/tnsnames.ora to the other node

Secondary Node

- On the secondary nodes, use the spacewalk-setup utility to get them into the mix:
  - `“spacewalk-setup --skip-db-population --ncc”`
- Make sure you have entitlements available. In some cases, support has seen validation issues if the database has more entitlements than what is purchased.
- “spacewalk-service disable” on secondary nodes - cluster will control them - also modify the spacewalk-service script to remove taskomatic as on the primary node for safety
- Replace the certs with the cluster ready ones…

Creating Cluster-ready Certs

Cluster Certs

- Initial build asks to setup a CA … creates /root/ssl_build directory
- Each node will have its own self-signed certificate - you can use mgr-ssl-tool to adapt the server certificates to include alternative cnames. This is important for connection failovers/reconnects
- The osa-dispatcher service is very particular to the server name - not alt-cname aware. So you need to create certs with the node names and alias names for the cluster name and alternate nodes.
- mgr-ssl-tool can be used to recreate the ca and/or the certs

Cert and CA Generation

- Phase 2 setup (yast susemanager_setup) will create a CA and create the /root/.ssl_build/ directory. If you forget to reference the cluster name in the CA - you can recreate it and the certs and redeploy them:
- All keys/certs are built from the /root/.ssl_build/ directory
- Use the mgr-ssl-tool to generate these
- Use the rhn-deploy-ca-cert.pl, rhn-install-ssl-cert.pl and rhn-generate-pem.pl tools

Examples Follow…

Cert and CA Examples

- Generate a new CA:
  - *mgr-ssl-tool --gen-ca --set-org=<client site>*
  - *mgr-ssl-tool --gen-ca --set-country='US' --set-state='MN' --set-city='Minneapolis' --set-org='SUSE Consulting Services' --set-org-unit='MN SM'*
- Generate server certs:
  - *mgr-ssl-tool --gen-server --set-hostname=<> --set-orgname=<> --set-org-unit=<> --set-city=<> --set-state+<> --set-country=US --set-cname=<node1> --set-cname=<node2> --set-cname=<node3>*

SEE Following PAGE

Server Cert Generation

- Server Certificate generation with ‘subject alternative names’ - include cluster name (fqdn) and each node as a cname… this buffers against connection pathing issues.
- Validate these through the browser…

- ```
  rhn-ssl-tool --gen-server --set-country='US' --set-state=‘MN' --set-city=‘Minneapolis' --set-hostname=‘suma21cl.sm.susetest.com' --set-org=‘SUSE Consulting NA' --set-org-unit=‘MSP SM' --set-email=‘jprice@suse.com' --set-cname=‘suma21cl.sm.susetest.com' --set-cname=' suma21cl-node1.sm.susetest.com ' --set-cname=' suma21cl-node2.sm.susetest.com ' --set-cname=‘suma21cl-node3.sm.susetest.com' --set-cname=' sumaproxy1.sm.susetest.com' --set-cname=' sumaproxy2.sm.susetest.com' --set-cname=' sumaproxy3.sm.susetest.com' --set-cname=' sumaproxy4.sm.susetest.com'
  ```
  ```

Propagate the Certs

- Once the certs are generated - you can copy the /root/.ssl_build/ directory over to the other nodes and deploy (copy) the certs into place
- Since the certs have each of the cluster node names /info in it, any connection (or reconnection) will be validated via SSL CA and Cert

Certs continued… Active/Passive

- The jabber service also has some sensitivity to host names and certificates. The client-to-server config (c2s.xml) and the session manager (sm.xml) both need to reference the cluster name (fqdn) - floater.
- The jabber server referenced in /etc/rhn/rhn.conf needs to reference the local server name and the osa-dispatcher entry uses the cluster name.
- Certs need to be put into place:
  - cp server.crt /etc/ssl/servercerts/spacewalk.crt
  - cp server.key /etc/ssl/private/spacewalk.key
  - cp server.crt /etc/apache2/ssl.crt/spacewalk.crt
  - cp server.key /etc/apache2/ssl.key/spacewalk.key

Certs - Active/Active

- For active/active there is no aliasing, so everything should have an fqdn in it, including /etc/jabber/xml files and /etc/rhn/rhn.conf
- Outside of letting HAE control the SUSE® Manager services - testing will tell you if there are problems
- Watch for errors like :

```
– SUSE Manager 4861 2013/06/17 16:16:10 -05:00: ('Not able to reconnect',)
  SUSE Manager 4861 2013/06/17 16:16:10 -05:00: ('Traceback (most recent call
  last):\n  File "/usr/share/rhn/osad/jabber_lib.py", line 252, in
  setup_connection\n
   c = self._get_jabber_client(js)\n  File
  "/usr/share/rhn/osad/jabber_lib.py", line 309, in _get_jabber_client\n
  c.connect()\n  File "/usr/share/rhn/osad/jabber_lib.py",
   line 589, in connect\n    raise SSLDisabledError\nSSLDisabledError\n',)
```

Certs

- SSL connect errors:
  - Check to see if you are using the short names (floater DNS names - cluster name) in jabber config files s2s.xml and sm.xml - in /etc/jabberd/
  - Check /etc/rhn/rhn.conf
    - There is an entry for the jabber server and it should use the FQDN
    - There is an entry for the OSA dispatcher and it should use the short name (cluster name)
    - Proxy servers should point to the cluster name for the parent server vs. the individual node names.

Demonstration

VM demonstrations…

Active / Passive

- Shared storage for
  - Remote DB
  - Package Repo
  - Web
- Failover / Resume

Active / Active

- Shared storage for
  - Remote DB
  - Package Repo
- Scale Out / SSL Reconnects
- Load Balance
- Resource HA

Appendix

HA Configuration

SLEHA Init and Join work wonders - leverage them for quick work of getting your nodes working cib 'Primitives' will include various SUSE® Manager services like :

- Monitoring
- MonitoringScout
- Apache
- Cobbler
- Postgresql
- /var/spacewalk mount
- /srv/www/htdocs/pub mount
- Virtual IP
- Jabber
- LVM Filesystem groups
- OSA Dispatcher
- RHN Search
- Taskomatic
- Tomcat

Basically every service you find in /sbin/spacewalk-service script - PLUS IP addresses and filesystems

HA Configuration

- If you recall, we basically disable the spacewalk-service script from being able to control/start/stop things - which adds some complexity when it comes to patching/updating SUSE® Manager itself:
  - Use Hawk or CLI commands to stop services in the cluster when updating SUSE Manager
  - Most times the DB stays up and other services come down - get patched - then you update the DB schema (if needed)
  - Finally, bring the SUSE Manager services back up.
  - Remember - with an external DB - you only need to upgrade the schema once, but you might need to patch/update the SUSE Manager services on more than 1 node

HA Configuration

- Example CIBs, Installation Procedures, etc:
  - Michael Brookhuis from SUSE® Consulting in Germany put together a wiki entry on SUSE Manager HAE clustering and you can find it here:
  - http://wiki.novell.com/index.php/SUSE_Manager/HAE

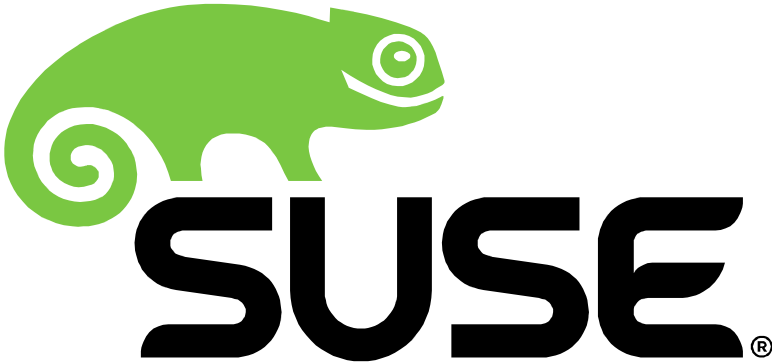
(This is for an active/passive (failover) cluster.)

Active/Active differences leverage HAE “constraints” to ensure Taskomatic runs as a single instance (on any 1 node).

Q & A

Questions?

Thank you.


SUSE®
We adapt. You succeed.

Corporate Headquarters
Maxfeldstrasse 5
90409 Nuremberg
Germany

+49 911 740 53 0 (Worldwide)
www.suse.com

Join us on:
www.opensuse.org