New Deploy Procedure by Prakash Varadarajulu on Sep 26, 2016

* Deployment of WebApp
* Deployment of Web
* Check Deploy version on servers

Deployment of WebApp

Prerequisite

Before deployment is started make sure to check if there has been changes to the database or changes to the environments in Puppet.

If there are changes to database it should be announced by the developer or we should have received an email telling us that there are changes:

Email:

----------------------------------------------------------------------------------------------

Jenkins <jenkins@resound.com>

Fri 9/9/2016 10:05 AM

To:

Endo - Operations;

Cc:

Endo - Server Development;

You forwarded this message on 9/9/2016 10:06 AM.

--preconditions onFail:MARK\_RAN onError:HALT

--precondition-sql-check expectedResult:0 SELECT count(\*) FROM INFORMATION\_SCHEMA.STATISTICS WHERE TABLE\_SCHEMA = 'inbox' AND TABLE\_NAME = 'Inbox' AND INDEX\_NAME = 'IX\_legacyId'

ALTER TABLE inbox.Inbox ADD INDEX IX\_legacyId (legacyId);

--------------------------------------------------------------------------------------------------

The puppet changes should have been done by us, but if not we can check:

“root@zabbix:/etc/puppet# git log”

If there are no changes then no need to run puppet agent on the servers.

When above is done/checked, deployment can be started.

1. Merging into release branch

* Run "PRODUCTION merge <Web/WebApp> master into release branch" job in Jenkins. After this job is finished will automatically start the “Build branch release” job
* Once build is finished run one or more of the staging jobs "Production stage ..."

2. For translations release (This step is only done on request, but should be done on a monthly basis).

* Run “TRANSLATIONS upload sources from resound-webapp” job
* Once build is finished run “TRANSLATIONS download from resound-webapp”    job
* Run "PRODUCTION merge into release branch" job
* Run “TRANSLATIONS download from resound-ng-webapp”

3. On Jenkins.

Run “Production stage <WebApp/OSGi/Web>” depending on the release. Normally we choose the release branch.

4.  From Zabbix server (zabbix.ec2.resound.com) run deploy script. Start with **api** and **rest** servers, but remember to log on to these servers and do the deployment manually. (#:/root/deploy.sh)

./root/deploy.sh -g api

./root/deploy.sh -g mobile -n 1

./root/deploy.sh -g web -n 1

./root/deploy.sh -g job

./root/deploy.sh -g rest

./root/deploy.sh -g graph-util

./root/deploy.sh -g swift

./root/deploy.sh -g inbox

For **service** log into each server and run deploy.sh from there. Wait till one is back up **and give it a few minutes to fully initialize** before deploying to next.

For **graph-legacy** log into each server and run deploy.sh from there. Wait till one is back up before deploying to next.

Run on zabbix

mco facts -F ec2\_security\_groups=graph-legacy deploy\_version

mco facts -F ec2\_security\_groups=service deploy\_version

To check that all hosts are found with latest version

* Deploy script for web & mobile adds two instances before the deployment (can be overridden with "-n 0" flag for no new instances).
* To deploy on selected server run: /root/deploy.sh on the particular server
* To update reference for new JS application files staged in S3 run: /root/deploy-assets.py (This is only done when releasing Web project).
* When deployment is done on a server group, you should be able to check that everything has been deployed as intended.

E.g.

If we deploy to the API servers we should check the access log (root@api-x:~# tail -f logs/access\_log.resound.com.2015-xx-xx.txt) and the catalina.out log (root@api-x:~# tail -f logs/catalina.out) to make sure that the servers are running and has started as intended. We should also check the “zabbix” web interface and make sure that “CloudWatch ELB Healthy Host Count” is as intended. This can be accessed by clicking on the Monitoring/Screens tabs and then choose -AWS.

5. All graph servers are being deployed manually at the moment, i.e. by running /root/deploy.sh on the individual servers

 6. I case there are problems with the servers, we might need to change the settings for AWS auto-scaling. If somethings happens during deployment and too few servers are left “alive” and the system is unresponsive. We can change the auto-scaling settings for the servers. Go-to the AWS web portal and click the EC2 link. There you click the “Auto Scaling Groups” link under the “Auto Scaling” menu point to the left of the screen.

Deployment of Web

1. Merging into release branch

* Run "PRODUCTION merge <Web> master into release branch" job in Jenkins. After this job is finished will automatically start the “Build branch release” job.
* Once build is finished run one or more of the staging jobs "Production stage ..."

1. On Jenkins

* Run “Production stage <Web>” depending on the release. Normally we choose the release branch.

1. From Zabbix server (zabbix.ec2.resound.com) run deploy script.

* ~# /root/**deploy-assets.py**

Check Deploy version on servers

 Using the mcollective agent installed on zabbix and clients on the servers, run following command from zabbix. Deploy version is build number of the release build from Jenkins.

root@zabbix:~# mco facts deploy\_version

Report for fact: deploy\_version

       Deploy-Version: 542, web                 found 1 times

       Deploy-Version: 556, graph-legacy        found 2 times

       Deploy-Version: 556, graph-util          found 2 times

       Deploy-Version: 556, mobile              found 4 times

       Deploy-Version: 557, api                 found 2 times

       Deploy-Version: 557, graph-util          found 6 times

       Deploy-Version: 557, job                 found 4 times

       Deploy-Version: 557, mobile              found 3 times

       Deploy-Version: 557, rest                found 3 times

       Deploy-Version: 557, service             found 2 times

       Deploy-Version: 557, swift               found 3 times

       Deploy-Version: 557, web                 found 2 times

       Deploy-date: 2016-01-14, osm-rest        found 1 times

Finished processing 68 / 68 hosts in 455.13 ms