

Migration Admin Guide - from Stash to Gerrit

Create repository on Stash for testing the replication



Needed only for the PoC, as in the real flow, we will already have the cloudnode repository

1. Create new project on Stash - GerritPOC (<http://stash:7990/projects/GPOC>)
2. Create a new repository – gerrit_main (http://stash:7990/projects/GPOC/repos/gerrit_main/browse)
3. Go to a local folder of cloudnode git repository on your dev machine
4. Execute git fetch - in order to make sure your local cloudnode repo is fully updated
5. Change the origin to the new gerrit_main repository

```
git remote set-url origin  
ssh://git@stash.cloud-band.com:7999/gpoc/gerrit_main.git
```

6. Create a local branch for all remote branches

```
for branch in `git branch -r | grep -v '\->' | sed "s/origin\/\///"; do git  
checkout -b $branch origin/$branch; done
```

7. Push it to the new gerrit_main

```
git push origin --all  
git push origin --tags
```

8. In Stash web UI - configure the gerrit_main repository. go to **Settings**
9. Go to **Branching model**, set the development branch to **dev**
10. Set the production branch to **master**
11. Tick the **automated merge** and **save**
12. Go to **Settings > [PERMISSIONS] Repository**
13. Add **jenkins-admins** to the Group access and set it as Admin.
14. The result is this repository on stash http://stash:7990/projects/gpoc/repos/gerrit_main/commits
15. on the local repository, revert the remote to cloudnode on stash

```
git remote set-url origin ssh://git@stash.cloud-band.com:7999/cnode/cloudnode.git
```

Copy this whole repository (cloudnode repo) AS-IS from stash into Gerrit

First I will create a project on Gerrit with same name and with Groups

1. Sign in to gerrit Web UI - <http://prod-gerrit.alucloud.local:8080/>
2. Go to Projects > Create Project
3. Give it a name (such gerrit_main)
4. Go to **People > Create New group** and create at least the following groups (name it as you wish)
 - a. gerrit_admin_admins
 - b. gerrit_admin_developers
 - c. gerrit_admin_ci

Set the following permissions

1. Go to the main page of you gerrit project - http://prod-gerrit.alucloud.local:8080/#/admin/projects/gerrit_main
2. Click on the **Edit Config** Button
3. Paste the following lines to the editor

```
[access]
[access "refs/*"]
  read = group Registered Users
  push = group gerrit_main_ci
  label-Verified = -1..+1 group gerrit_main_ci
  viewDrafts = group gerrit_main_admins
  viewDrafts = group gerrit_main_developers
  deleteDrafts = group gerrit_main_admins
  publishDrafts = group gerrit_main_admins
  label-Workflow = -1..+1 group gerrit_main_admins
  create = group gerrit_main_admins
[access "refs/drafts/*"]
  push = group gerrit_main_developers
  create = group gerrit_main_developers
[access "refs/for/*"]
  push = group gerrit_main_developers
  create = group gerrit_main_developers
[access "refs/heads/*"]
  read = group gerrit_main_ci
  push = group gerrit_main_ci
[access "refs/tags/*"]
  read = group gerrit_main_ci
  push = +force group gerrit_main_ci
  pushTag = +force group gerrit_main_ci
  pushSignedTag = group gerrit_main_ci
  create = group gerrit_main_ci
  forgeCommitter = group gerrit_main_ci
```

4. On a dev machine - clone this project locally

```
git clone ssh://dshai@prod-gerrit.alucloud.local:29418/gerrit_main gerrit_main &&
scp -p -P 29418 dshai@prod-gerrit.alucloud.local:hooks/commit-msg
gerrit_main/.git/hooks/
```

5. Now I will add a new remote to this repository (in the real flow it will be the cloudnode)

```
cd gerrit_main
git remote set-url origin
ssh://git@stash.cloud-band.com:7999/gpoc/gerrit_main.git
```

6. Change the fetch part in the **.git/config** file

```
fetch = +refs/heads/*:refs/origin/*
## THE ORIGINAL WAS: fetch = +refs/heads/*:refs/remotes/origin/*
```

7. Now pull from Stash to get the whole **gerrit_main** repository

```
git pull origin
git checkout master
```

8. Ensure you got the whole log

```
git log
```

9. Change the origin target repository back

```
git remote set-url origin  
ssh://dshai@prod-gerrit.alucloud.local:29418/gerrit_main
```

10. Push everything to the **gerrit_main** project on **gerrit**
git push origin refs/origin/*:refs/heads/*
(Don't use "git push origin --all" - as this will push branches with "origin" in the name...)
git push origin --tags
11. Copy the pre-recieve hook form stash to this repo

```
scp -r  
root@172.29.38.40:/opt/stash_data/data/repositories/115/hooks/pre-receive.d.SAMPLE  
E pre-receive.d  
scp -r  
root@172.29.38.40:/opt/stash_data/data/repositories/115/hooks/pre-receive.SAMPLE  
pre-receive
```

12. Now we would like to set the repository on stash to read-only (lock to everyone) for all users except the admin which will perform the replication

=====

The following settings and configuration is needed if we would like developers still to fetch from stash instead from gerrit. Replication from gerrit repo to stash can be done immediately or after a delay you configure (in seconds).

Replication from Gerrit to target stash repository

Now, we have an active repository on Gerrit. If we would like to have all the changes replicated to the stash server, then we will apply the following steps



Gerrit is an open source and there are many Plugins that were developed for it. One of them is the Replication Plugin which you chose to Install as part of the Gerrit Installation.



Prerequisites Apply the steps listed on [this page](#), on section **Setup local environment to work with Gerrit CLI**

1. SSH to the the Gerrit server
2. go to the installation folder of Gerrit (usually it is **/opt/gerrit2**)
3. Config the plugin by adding the following section to the **etc/replication.config** file

```
[remote "stash"]
url = ssh://jenkins@stash.cloud-band.com:7999/gpoc/${name}.git           # <- This
is the place to replicate to
push = +refs/heads/*:refs/heads/*
push = +refs/changes/*:refs/changes/*
push = +refs/tags/*:refs/tags/*
authGroup = Registered Users
threads = 1
replicationDelay = 0                                                     # <- the
delay you want to wait after a change was done on Gerrit side till it is
replicated to target
```

4. Config the ssh connection by adding the following section to the `/home/gerrit2/.ssh/config` file

```
Host stash.cloud-band.com
IdentityFile /home/gerrit2/.ssh/id_rsa
StrictHostKeyChecking no
UserKnownHostsFile /dev/null
BatchMode yes
```

5. Login with the jenkins user to stash account web UI (<http://stash:7990/account>)
6. Click on **SSH Keys**
7. Add the public key that was created at the beginning of this section (as a prerequisite)
8. In stash server, go to the repository hooks folder (the location is mentioned on http://stash:7990/projects/GPOC/repos/gerrit_main/settings)
s> Location on disk)

```
cd /opt/stash_data/data/repositories/115/hooks
```

9. To prevent the pre-recieve hook, Change the name of both pre-recive and pre-recieve.d under the target repository

```
mv pre-receive pre-receive.DEACTIVATED
mv pre-receive.d pre-receive.d.DEACTIVATED
```

10. You need to reload the plugin

```
ssh gerrit-admin gerrit plugin reload replication
```

11. Then start the replication

```
ssh gerrit-admin replication start --all
```

12. you can tail the log of the replication

```
tail -f /opt/gerrit2/logs/replication_log
```