



EPAM Cloud Orchestrator

Gerrit Service Activation and Jenkins Integration

User Guide

December 2016

Version 1.1

Legal Notice: This document is property of EPAM and may not be disclosed, distributed or reproduced without the prior written permission of EPAM®.

Contents

Contents	2
1 Introduction	3
2 Prerequisites	3
3 Services Activation and Integration	4
3.1 Jenkins Activation	4
3.2 Gerrit Activation	4
3.3 Adding Gerrit SSH Key to Git Repository	6
3.4 Repository Cloning	7
3.5 Verification.....	9
Table of Figures.....	11
Version History.....	12

1 INTRODUCTION

This how-to document is intended to help you activate and integrate Gerrit and Jenkins services. Please note that the procedure described below is done on a Linux workstation, so some commands may not be applicable for Windows users. You should have Maestro CLI installed to use the procedure described in this guide.

2 PREREQUISITES

Before you activate Gerrit as a service, make sure that the following pre-requisites are met:

- There is an empty Git repository which will be served by Gerrit.
- You have an SSH key created in the same project and region where Gerrit service will be run. If no key is available, use the **or2-create-keypair** command to obtain an SSH key.



The `-p` (`--project`) and `-r` (`--region`) parameters are project-specific, please use the parameters of your project.

Create an empty Git repository for Gerrit to replicate its projects.

The screenshot shows the GitLab interface for a project named 'EPMC-CLO / test-gerrit'. The main content area displays the message "The repository for this project is empty" and provides "Command line instructions" for setting up Git globally and creating a new repository. The instructions are as follows:

```

git config --global user.name "User User"
git config --global user.email "user_user@epam.com"

Create a new repository

git clone git@git.epam.com:epmc-clo/test-gerrit.git
cd test-gerrit
touch README.md
git add README.md
  
```

Figure 1 - Repository creation

Create an SSH keypair which would be used by the administrator user for interaction with Gerrit's 29418 port via SSH:

```
or2-create-keypair -p EPMC-CLO -r EPAM-BY1 -k gerrit_test
```

```
Response:
=====
name          | owner          | project |
=====
gerrit_test   | User User     | EPMC-CLO |
=====
Generated key is saved to /home/mprod/maestro-cli/out/EPMC-CLO/EPAM-BY1/gerrit_test.pem
```

Figure 2 - SSH key generation

Please note where the private key is saved, for example:

```
Generated key is saved to /home/mprod/maestro-cli/out/EPMC-CLO/EPAM-BY1/gerrit_test.pem
```

3 SERVICES ACTIVATION AND INTEGRATION

3.1 JENKINS ACTIVATION

Activate the Jenkins service using the following command:

```
or2-manage-service -p EPMC-CLO -r epam-by1 -a -s jenkins -h medium
```

Notice, that the **-h** (--shape) parameter is applicable for all services.

```
Response: Service activated
Response:
=====
stackName      | stackId        | status  |
=====
EPMC-CLOJenkins | EPMC-CLOJenkins-ddb0df71 | PENDING_TO_CREATE |
=====
```

Figure 3 - Jenkins activation

3.2 GERRIT ACTIVATION

Activate the Gerrit service using the following command:

```
or2-manage-service -p EPMC-CLO -r epam-by1 -a -s gerrit -h large -k gerrit_test
```

Please, pay attention to the **-h** (desired instance shape, MEDIUM by default) and **-k** (created SSH keypair name) parameters.

In addition, you will be asked to choose the Git repository to configure replication for it. Please, copy and paste the SSH address from your project homepage as shown on picture below:

EPAM Cloud Orchestrator – Gerrit as a Service

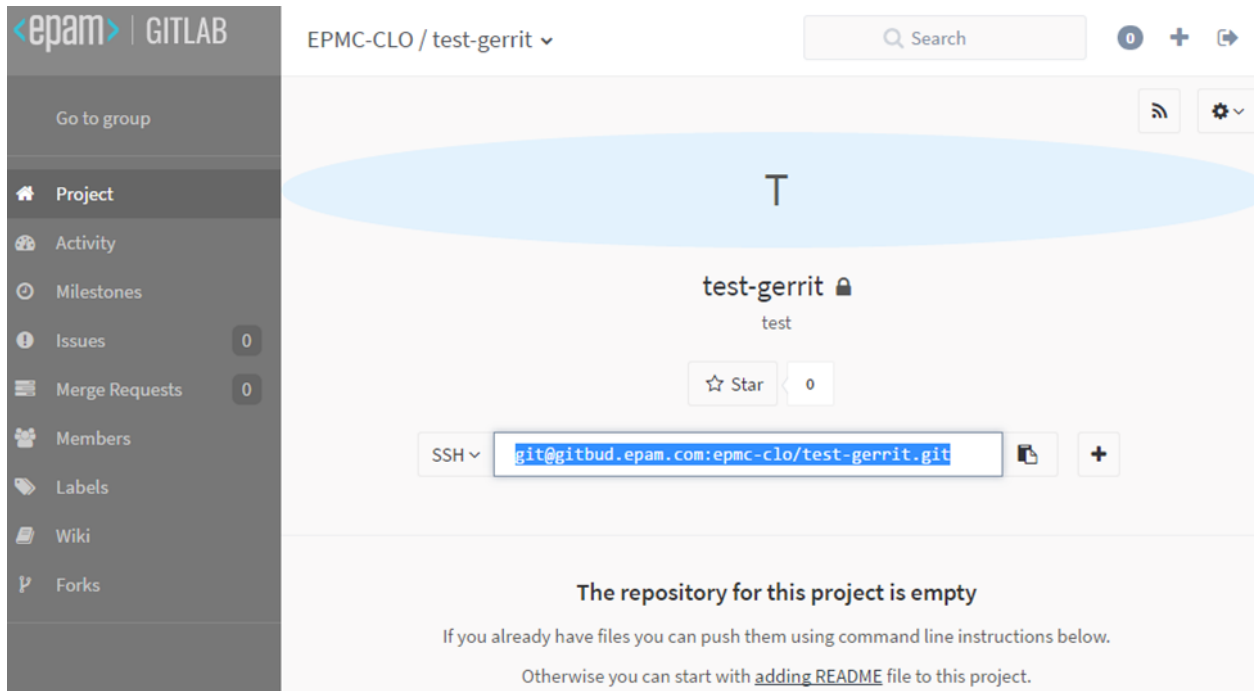


Figure 4 - Repository address

The system will respond with the service activation details:

```
or2-manage-service -p epmc-clo -r epam-by1 -a -s gerrit -h large -k gerrit_test
Git repository to be served by Gerrit. Example: git@git.epam.com:SOME-PRJ/repo.git: git@git.epam.com:epmc-clo/test-gerrit.git
Response: Service activated
Response:

=====
stackName      | stackId          | status          |
=====
EPMC-CLOGerrit | EPMC-CLOGerrit-48461347 | PENDING_TO_CREATE |
=====
```

Please note, that the user activating the Gerrit service becomes the Gerrit administrator.

You can check the service status using the following command:

```
or2-describe-services -p EPMC-CLO -r epam-by1
```

```
Response:
=====
serviceId      | serviceName | instanceSearchId | availability | stackId      | ip |
=====
service-c746f60b | gerrit      | i-2e1525e9       | unavailable | EPMC-CLOGerrit-48461347 |   |
service-24b0f214 | jenkins    | i-ebb9b81a       | available  | EPMC-CLOJenkins-ddb0df71 | 10.***** |
=====

=====
| dns                | keyName      | webUiUrl          | user | password |
=====
|                    | gerrit_test |                   |      |          |
=====
| ECS *****.epam.com |             | http://ecs *****.epam.com:8080/ | admin | JenkinsPassword123 |
=====
```

Figure 5 - Service data (shown in two lines for better visibility)

Wait until both services are available:

```
=====
serviceId      | serviceName | instanceSearchId | availability | stackId      |
=====
service-c746f60b | gerrit      | i-2e1525e9       | available  | EPMC-CLOGerrit-48461347 |
service-24b0f214 | jenkins    | i-ebb9b81a       | available  | EPMC-CLOJenkins-ddb0df71 |
=====
```

Figure 6 - Gerrit and Jenkins available

3.3 ADDING GERRIT SSH KEY TO GIT REPOSITORY

Add Gerrit public SSH key to the Git repository.

After the Gerrit service becomes available, you will get an email containing an SSH key:

Project: EPMC-CLO
Region: EPAM-BY1

EPMC-CLO. Maestro Stack EPMC-CLOGerrit-48461347 Executed

Maestro Stack Info	
Stack ID:	EPMC-CLOGerrit-48461347
Stack Name:	EPMC-CLOGerrit
Template Name:	gerrit.json
Stack Status:	CREATE_COMPLETE
Owner:	User User

Maestro Stack Outputs	
Public SSH key:	ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDlxjzfcLSLJXS3f0U5G4nT7wBAInbN3H8cdJHtYKbQcHqsrPUv++Ffqtpe+u0LF+EZYAve0XUoe1lv+WYwu0Qb2WDukZObyigUkLAc4JaCW5N gerrit@ECSC0010482B

Figure 7 - Email containing SSH key

Copy this key and add it to any account that has sufficient permissions to replicate in the Git repository.

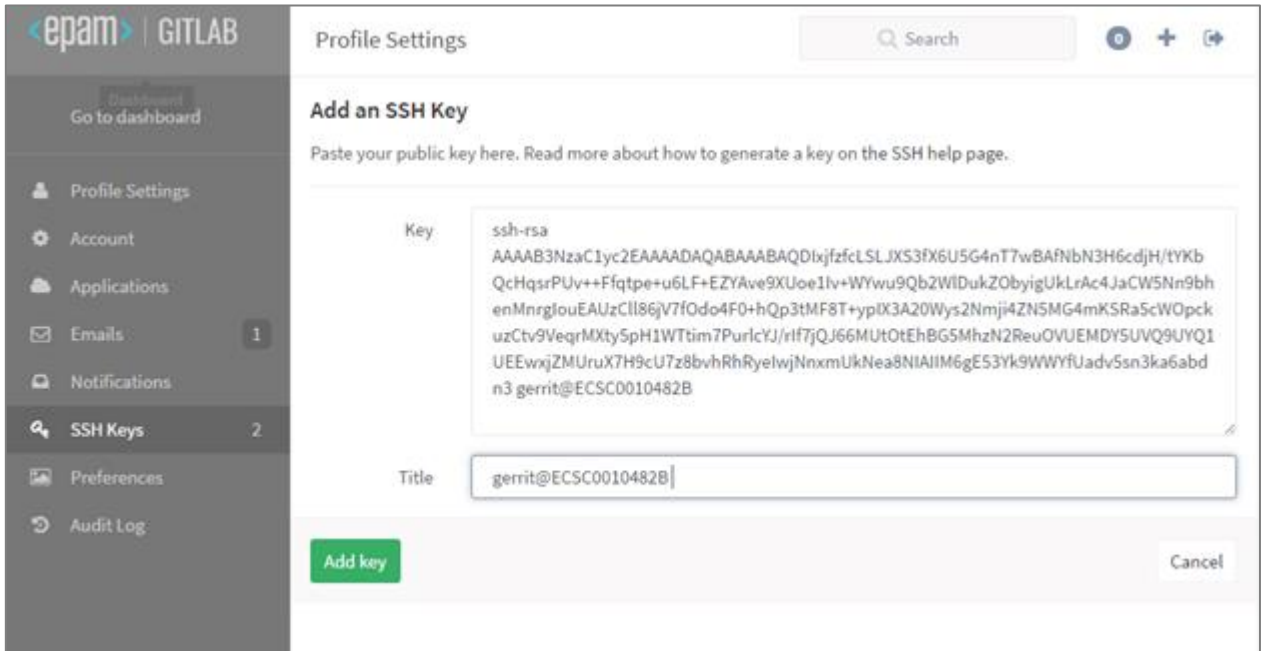


Figure 8 - Adding SSH key

Now you can manage Gerrit via ssh:

```
ssh -i /home/mprod/maestro-cli/out/EPMC-CLO/EPAM-BY1/gerrit_test.pem -p 29418 firstname_lastname@ECSC*****.epam.com gerrit
```

3.4 REPOSITORY CLONING

Clone the repository and push first change for review:

Access the Gerrit UI via the link in the 'webUrl' column of the service description (response to **or2dser** command, please see p. 5 above) and find the link to clone the project with commit-hook. Use your PMC credentials (firstname_lastname, without @epam.com) in lowercase. Please note that your login must be up to 20 characters. If your firstname_lastname is longer than 20 characters, truncate your last name. Password is your Active Directory password:

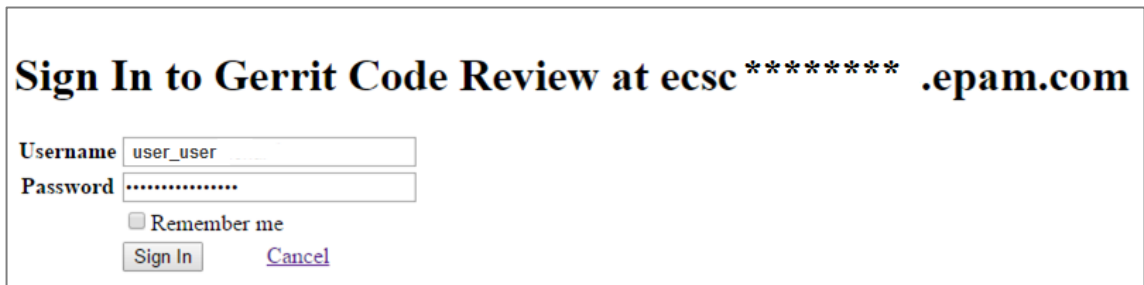


Figure 9 - Logging in to Gerrit

Copy the link to clone your repository (Projects->List->your-project), choose 'Clone with commit-msg hook' and 'ssh':

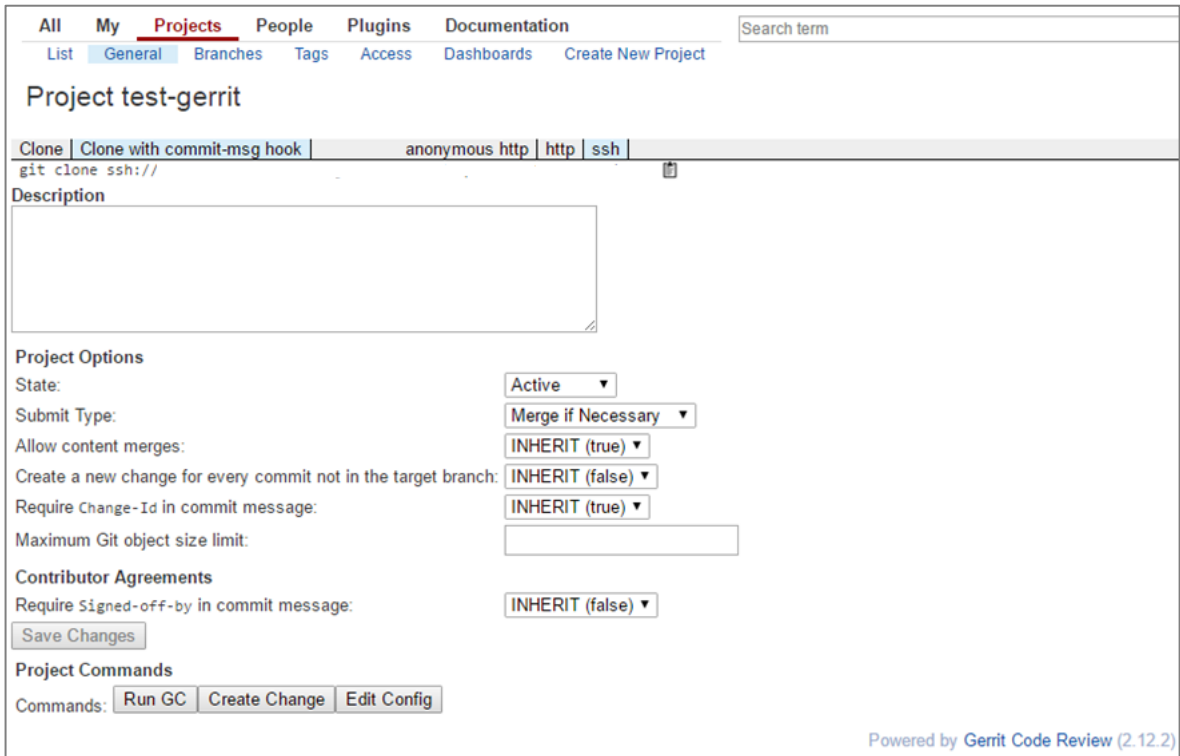


Figure 10 - Repository cloning

```
ssh-agent bash -c 'ssh-add /home/mprod/maestro-cli/out/EPMC-CLO/EPAM-BY1/gerrit_test.pem; git clone
ssh://user_user@ECSC*****.epam.com:29418/test-gerrit && scp -p -P
29418 user_user@ECSC*****.epam.com:hooks/commit-msg test-
gerrit/.git/hooks/'
```

```
Identity added: /home/mprod/maestro-cli/out/EPMC-CLO/EPAM-BY1/gerrit_test.pem (/home/mprod/maestro-cli/out/EPMC-CLO/EPAM-BY1/gerrit_test.pem)
Cloning into 'test-gerrit'...
The authenticity of host '[ecsc *****.epam.com]:29418 ([10.*****]:29418)' can't be established.
RSA key fingerprint is 17:4f:a1:59:4f:0f:5d:57:80:8a:26:c4:b2:21:8f:19.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[ecsc *****.epam.com]:29418' (RSA) to the list of known hosts.
Warning: the RSA host key for '[ecsc *****.epam.com]:29418' differs from the key for the IP address '[10.*****]:29418'
Offending key for IP in /home/sefs/.ssh/known_hosts:871
Are you sure you want to continue connecting (yes/no)? yes
Checking connectivity... done.
warning: remote HEAD refers to nonexistent ref, unable to checkout.

Warning: the RSA host key for '[ecsc *****.epam.com]:29418' differs from the key for the IP address '[10.*****]:29418'
Offending key for IP in /home/sefs/.ssh/known_hosts:871
Matching host key in /home/sefs/.ssh/known_hosts:1032
Are you sure you want to continue connecting (yes/no)? yes
commit-msg
```

Figure 11 – Repository cloning

In this example, the 'ssh-agent' command is used to substitute the private SSH key generated according to the instructions in p.2 above.

Let's push something to check the Gerrit performance:

```
$ cd test-gerrit/
$ date > current.date
$ git add current.date
$ git commit -m "Current date"
```


EPAM Cloud Orchestrator – Gerrit as a Service

```
$ ssh-agent bash -c 'ssh-add /home/mprod/maestro-cli/out/EPMC-CLO/EPAM-BY1/gerrit_test.pem; git push origin HEAD:refs/for/master'
```



The commands above are for illustration purposes only. Please do not copy them to use with your repositories.

3.5 VERIFICATION

Now you can verify that integration is working properly:

The screenshot shows the Jenkins web interface for a project named 'example_gerrit_job'. The interface includes a sidebar with navigation options such as 'Back to Dashboard', 'Status', 'Changes', 'Workspace', 'Build Now', 'Delete Project', and 'Configure'. The main content area displays the project name, a 'Workspace' folder icon, 'Recent Changes', and a 'Permalinks' section with links for 'Last build (#1)', 'Last stable build (#1)', 'Last successful build (#1)', and 'Last completed build (#1)'. A 'Build History' table shows a single build from May 23, 2016, at 3:50 PM. There are also 'RSS for all' and 'RSS for failures' links at the bottom of the build history section.

Figure 12 - Jenkins job

The screenshot shows the Gerrit web interface for a code review. The main area displays the change details, including the author 'User_User', the commit message 'current.date', and the change ID. The right sidebar shows the review status, including 'Code-Review+2', 'Reviewed by jenkins', and 'Verified +1 jenkins'. The bottom section shows a table of files with columns for 'File Path', 'Comments', and 'Size'.

File Path	Comments	Size
Commit Message		
current.date	+1, -1	+1, -1

Figure 13 - Gerrit

Now you can merge this change to master and check replication:

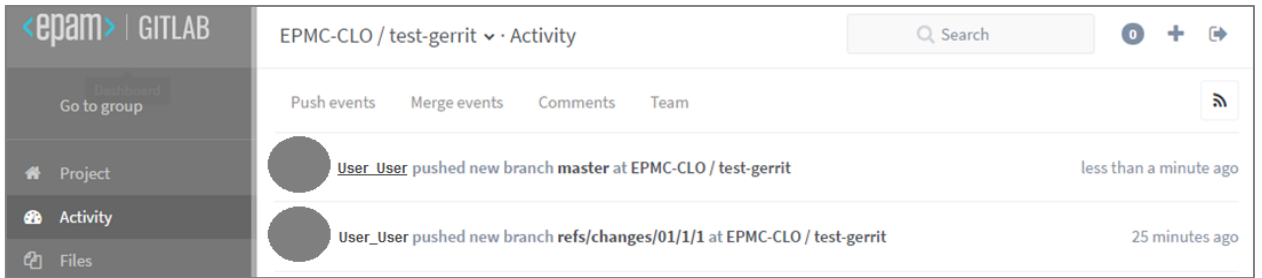


Figure 14 - Replication

If you want to grant administrator privileges to another user (or add them to another group), this user should login to Gerrit to register their account in database.

TABLE OF FIGURES

Figure 1 - Repository creation	3
Figure 2 - SSH key generation	4
Figure 3 - Jenkins activation.....	4
Figure 4 - Repository address	5
Figure 5 - Service data (shown in two lines for better visibility).....	6
Figure 6 - Gerrit and Jenkins available	6
Figure 7 - Email containing SSH key	6
Figure 8 - Adding SSH key	7
Figure 9 - Logging in to Gerrit.....	7
Figure 10 - Repository cloning.....	8
Figure 11 – Repository cloning.....	8
Figure 12 - Jenkins job.....	9
Figure 13 - Gerrit.....	9
Figure 14 - Replication	10

VERSION HISTORY

Version	Date	Summary
1.1	December 16, 2016	Classification changed from Confidential to Public, approved by Dzmitry Pliushch
1.0	May 23, 2016	First published