|  |
| --- |
|  |
| GitBlit User Guide |
| Agile DevOps Lab |
| Author: Vivek Gole  Created On: 21-May-2015  Summary: This document will guide users to configure and use GitBlit as Git repository manager tool.  Source : gitblit.com |

Contents

[Introduction 3](#_Toc420075072)

[About 3](#_Toc420075073)

[Features 3](#_Toc420075074)

[Architecture 3](#_Toc420075075)

[Audience 4](#_Toc420075076)

[Prerequisites 4](#_Toc420075077)

[Installation 5](#_Toc420075078)

[Configuration 5](#_Toc420075079)

[Directory Storage 5](#_Toc420075080)

[Properties 5](#_Toc420075081)

[Create Certificate 5](#_Toc420075082)

[Configure Gitblit 7](#_Toc420075083)

[Install as Windows Service 7](#_Toc420075084)

[GitBlit Dashboard 9](#_Toc420075085)

[New Repository 9](#_Toc420075086)

[Managing Users & Teams 11](#_Toc420075087)

[Manage Profile 11](#_Toc420075088)

[Manage Users 12](#_Toc420075089)

[Manage Teams 15](#_Toc420075090)

[Manage Repositories 17](#_Toc420075091)

[Repository Dashboard 17](#_Toc420075092)

[Repository Commit History 18](#_Toc420075093)

[Repository Edit 19](#_Toc420075094)

[Appendix 21](#_Toc420075095)

[References 21](#_Toc420075096)

[Study Materials 21](#_Toc420075097)

[Downloads 21](#_Toc420075098)

# Introduction

This document explains the use and configure of GitBlit administrative tool which is used as a Git repositories manager.

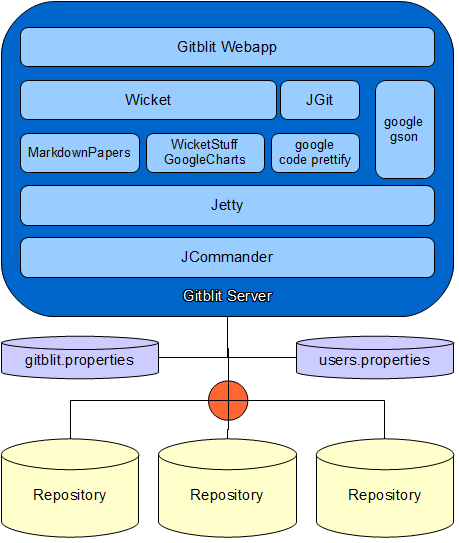
## About

Gitblit is an open-source, pure Java stack for managing, viewing, and serving [Git](http://git-scm.com/) repositories. It's designed primarily as a tool for small workgroups who want to host centralized repositories.

### Features

1. [Simple,](http://en.wikipedia.org/wiki/KISS_principle) easy to use.
2. Offer useful features for serving Git repositories.

### Architecture



## Audience

As this document only highlights the configuration and use part of GitBlit, we expect the intended audience should have the following skills:

* Familiar with Git version controlling.
* Familiar with Visual Studio 2012(+) and Team Foundation Server.

## Prerequisites

Following are the software required to configure GitBlit.

* **GitBlit**
* **Git**
* **Java 7 Runtime Environment (JRE) or a Java 7 Development Kit (JDK).**

# Installation

This document will explain how to install GitBlit as a Windows service.

Download the windows version for Gitblit from <http://www.gitblit.com/>

For more details, follow the installation steps mentioned in <http://www.gitblit.com/setup_go.html>

## Configuration

Download the latest stable GitBlit Go for windows setup which comes as a zip file. Following are the important terms and steps user needs to know while configuring GitBlit on windows.

### Directory Storage

The location where Gitblit files are extracted (this is called baseFolder): *E:\GitBlit*

### Properties

Properties/Configuration settings are defined in file *E:\GitBlit\data\gitblit.properties*

If Gitblit is installed as windows service, then the service needs to be restarted after any changes to the properties file.

Following are the properties modified with modified values.

1. server.httpPort = 88 (Make sure the port you want to use is open and listening on HTTP port)
2. server.certificateAlias = cch1wdbas2.corp.hexaware.com (Use server HostName)
3. web.siteName = cch1wdbas2.corp.hexaware.com (Use server HostName)
4. server.storePassword = gitblit

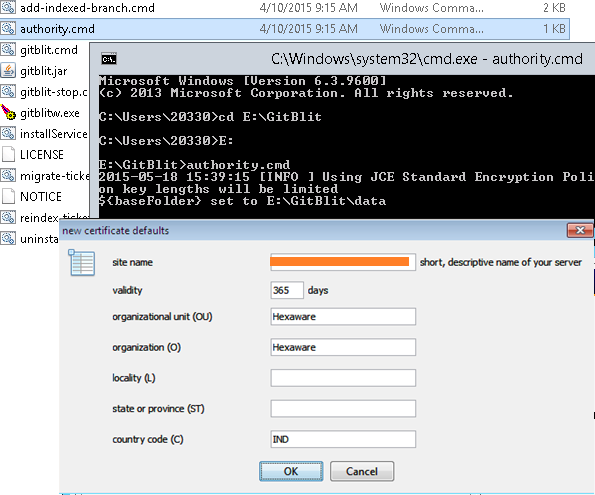
### Create Certificate

Gitblit provides a batch file inside baseFolder *authority.cmd* which can be used to create SSL certificate.

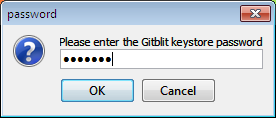
The Authority is a Swing GUI application. Use of this tool is not required as Gitblit GO will startup and create SSL certificates itself, BUT use of this tool allows you to control the identification metadata used in the generated certificates. Skipping this step will result in certificates with default metadata.

1. From baseFolder execute *authority.cmd* or *java -cp gitblit.jar com.gitblit.authority.Launcher --baseFolder data* from a command-line
2. Enter the details:

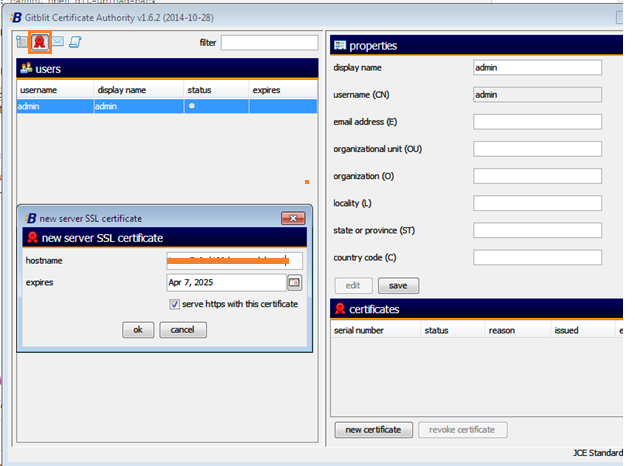
Name of website: *cch1wdbas2.corp.hexaware.com*



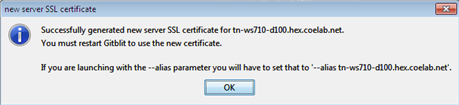
1. Enter password from gitblit.properties- property *server.storePassword = gitblit*



1. Create new server certificate- click icon in orange box.
   1. Enter host name - *cch1wdbas2.corp.hexaware.com*
   2. *Enter expiry date*
   3. *Enter other details.*



1. Save the details and this will create new certificate.



### Configure Gitblit

To install the gitblit configurations from the baseFolder execute *gitblit.cmd* or *java -jar gitblit.jar --baseFolder data* from a command-line.

### Install as Windows Service

Gitblit provides two batch files to install and un-install gitblit service from windows services.

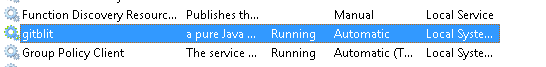
* installService.cmd – To install as windows service
* uninstallService.cmd – To uninstall from windows service

Before executing these services make sure before setting are proper as per server machine architecture.

Edit both files in notepad (notepad++), and update the ARCH property as per server architecture.

In our case it was updated as “*SET ARCH=x86*”

Check windows services and start gitblit service in automatic mode.

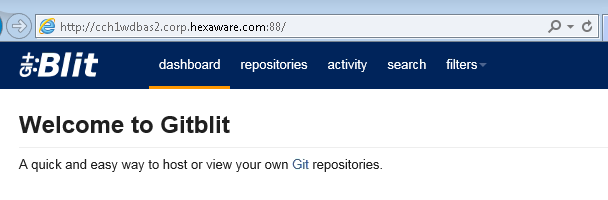


# GitBlit Dashboard

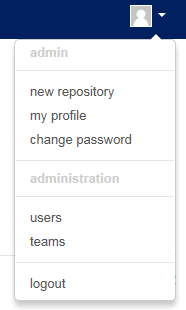
Access the gitblit from browser with provided website/hostname.

If you are accessing the site without SSL then enter the server.httpPort provided in gitblit.properties.

Default credential is admin/admin. It is recommended to change this are logging.



Right side top most show a dropdown settings link which shows options to create new repository, manage profiles, users and teams.

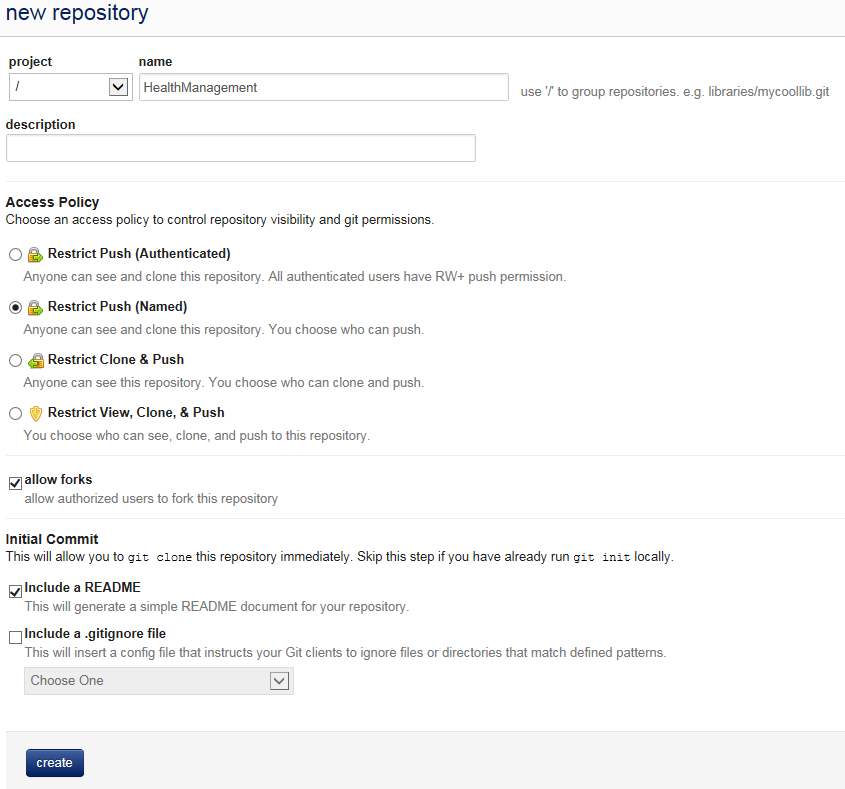


## New Repository

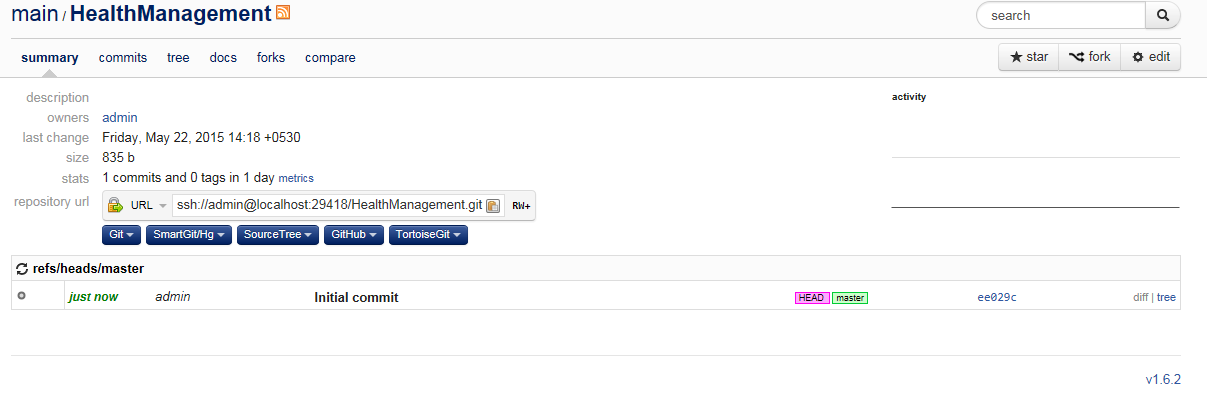
From dropdown settings click new repository to create new repository. By default the user who created the repository will be the owner of it.

It is recommended to use default options but for access policy you can restrict code/data pull/pushed to/from repository. Access policy provides various options to define the repository permissions. *Restrict View, Clone & Push* is the most secure option as it restricts view and clone for anonymous users.

By default new repositories are created as EMPTY repository, hence you will not see the expected dashboard with commit details for new repository. Hence make sure ‘Include a README’ is selected while creating the repository. With this it will check in a simple readme txt file and you can see the Dashboard will history of commits made to it.



Below is the dashboard for newly created repository.



## Managing Users & Teams

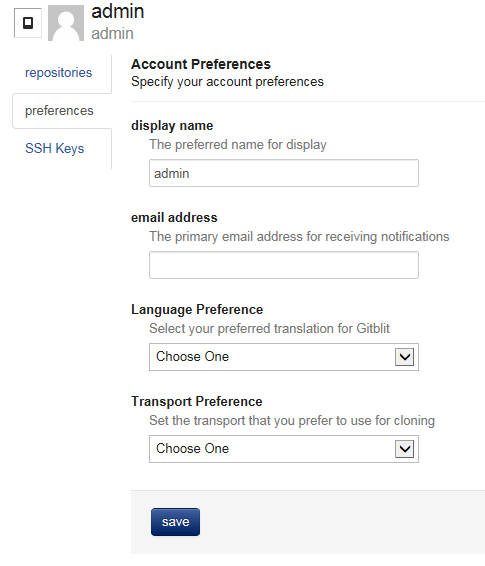
GitBlit dashboard provides UIs to manage the user profiles and teams for git repositories created by tool.

### Manage Profile

Here you can manage users profile details

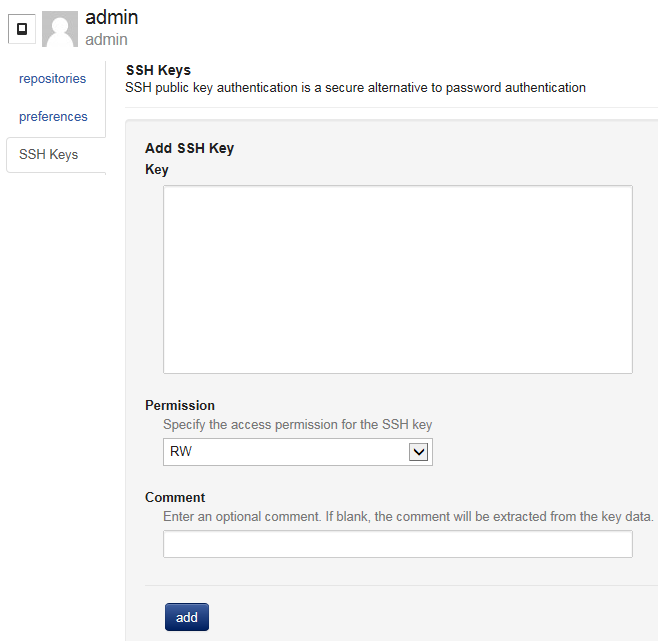
#### Preferences

In the preferences option it allows to change the display name, email address, language and Transport preferences.



#### SSH Keys

SSH Keys option, allows to added secure key for transport preference. Permissions can be of 3 types (V-View, R-Read and RW-Read Write).



### Manage Users

After selecting Users from dropdown settings link, it opens the dashboard for managing Users.

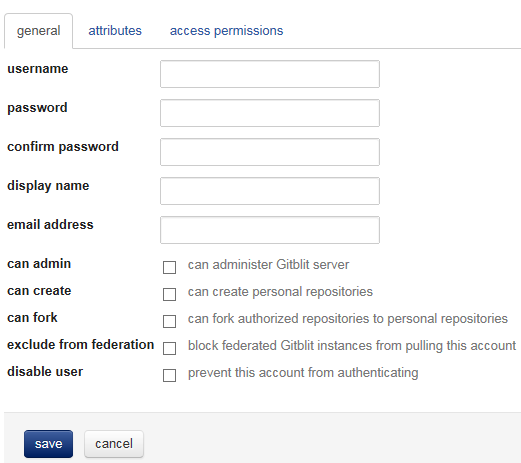
Here you can create new users, modify or remove it.



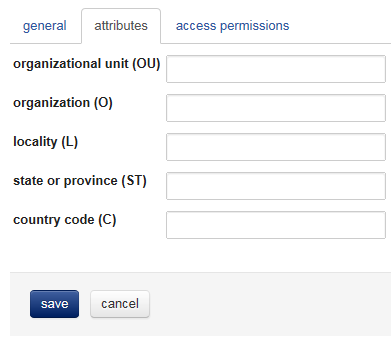
#### Create/Edit Users

To make the user as administrator, select below options. Otherwise it is recommended to use the default settings while creating users.

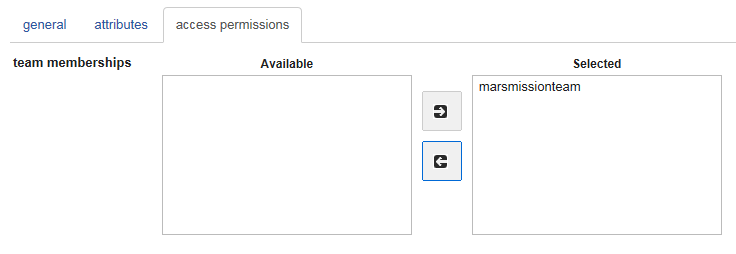
* can admin- Creates admin users for managing GitBlit server.
* can create- Allows user to create personal repositories.



Attributes are options details which add more description to user’s profile.

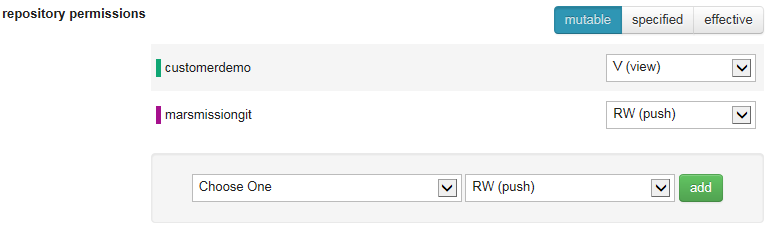


User can be added to available teams from below UI. Here user is added to existing marsmissionteam.



User permissions for repository can be changed from repository permissions option.

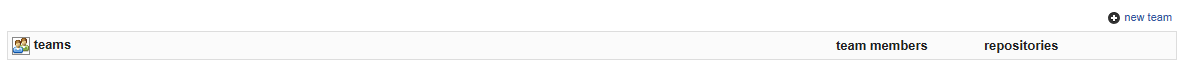
Below screen shows, user has View permissions for customerdemo repository and Read Write permissions for marsmissiongit repository.



### Manage Teams

After selecting Teams from dropdown settings link, it opens the dashboard for managing Teams.

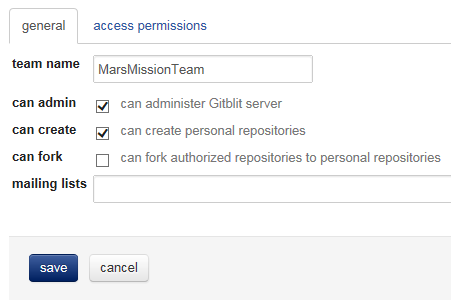
Here you can create new teams, modify or remove it.



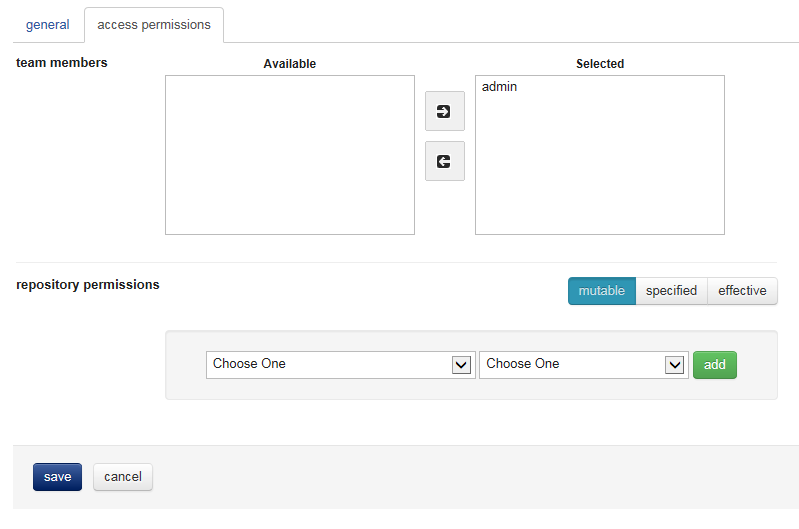
#### Create/Edit Teams

To make the all the users from team as administrator, select below options. Otherwise it is recommended to use the default settings while creating users.

* can admin- Creates admin users for managing GitBlit server.
* can create- Allows user to create personal repositories.

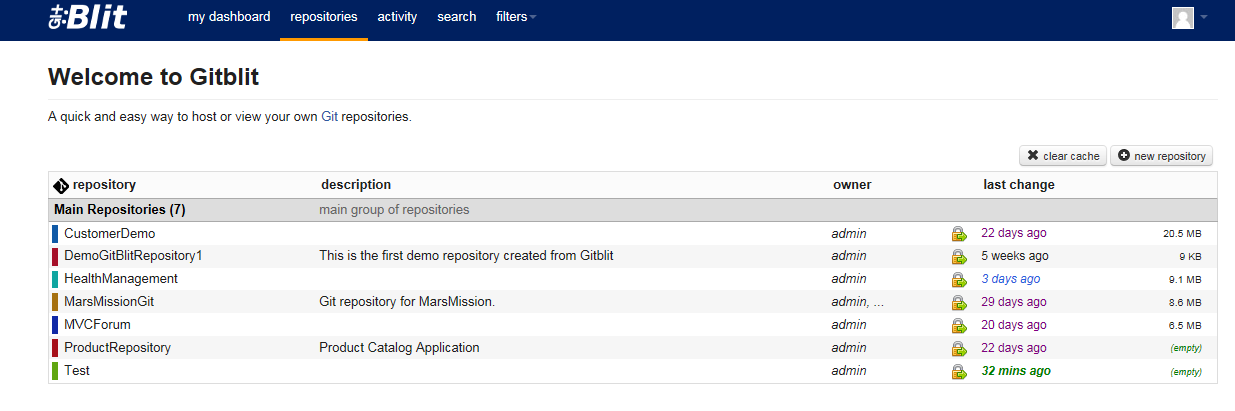


Access Permissions option allows to add/remove users from Team. It allows managing repository permissions for all teams. This option is similar to what we seen while configuring Users repository permissions.



## Manage Repositories

Repositories tab shows all the repositories created from GitBlit. On selecting any repository it will show detail summary of changes happened for selected repository.

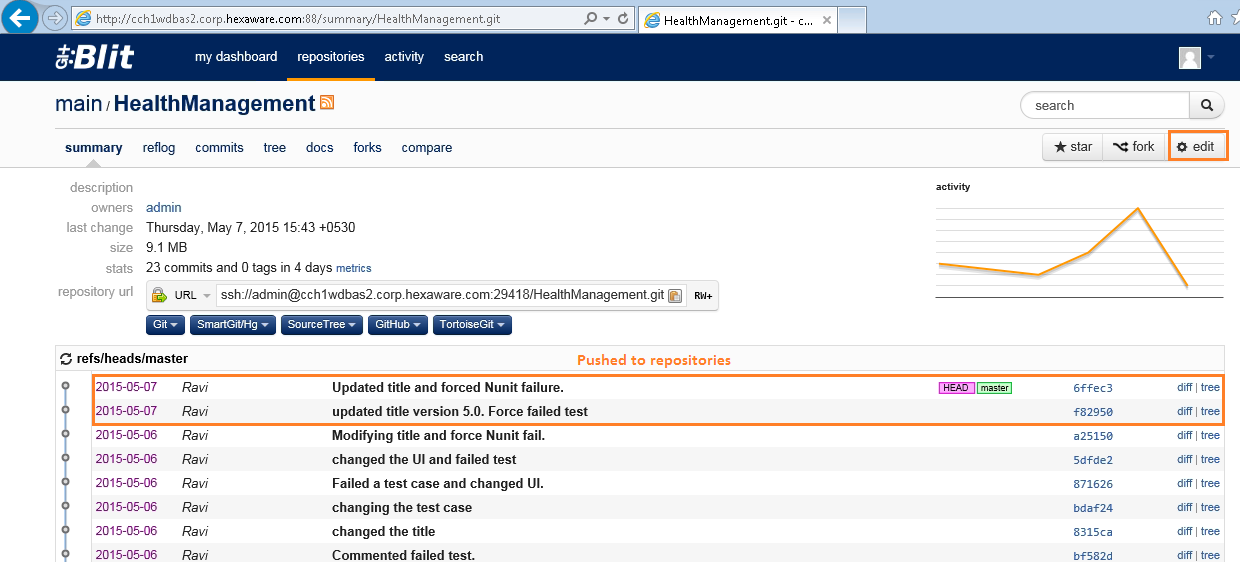


### Repository Dashboard

Click on repository from Repositories will open the selected repository dashboard.

Below image shows dashboard for HealthManagement repository. It list all the changes pushed to the repository.

Highlighted in orange box shows the changes pushed to repository. To see more details about the push changes, click on the comment.



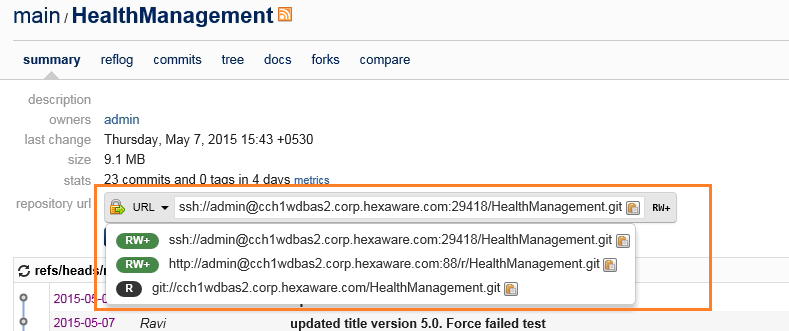
Repository URLs can be accessed with SSL or without SSL with normal http port.

Default URLs are of 2 types,

1. RW- URL with read/write permissions.
2. R- URL with only read permissions.

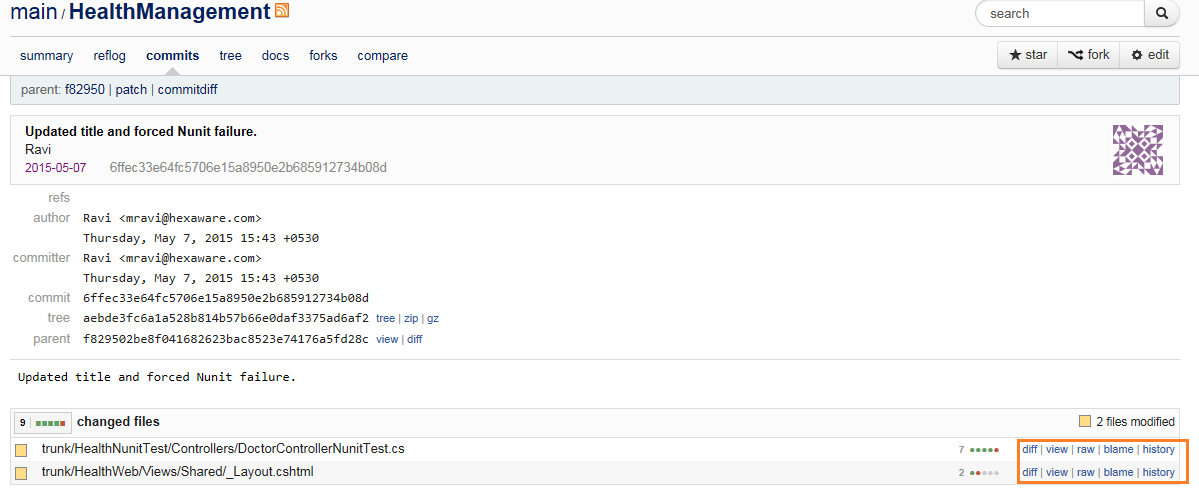
Use URL with RW if you want to push changes to repository. USE URL with only read right if when application only needs to access the repository changes.

Click on icon  to copy the URL.

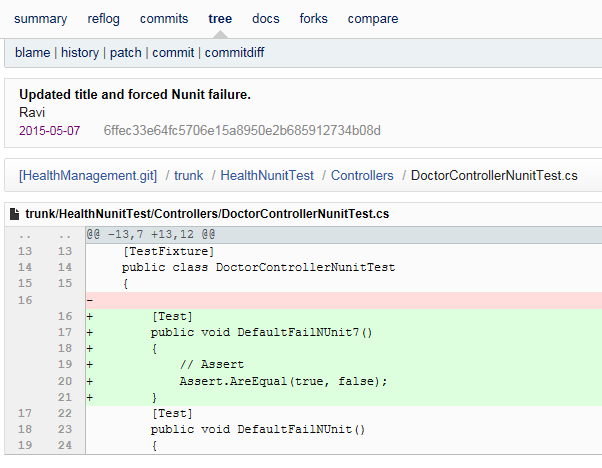


### Repository Commit History

Below image shows detailed view of changes for selected commit/push. It allows options to compare the file differences with previous commits.



After click on **diff** link (highlighted in orange above image), shows compare history of selected file.



### Repository Edit

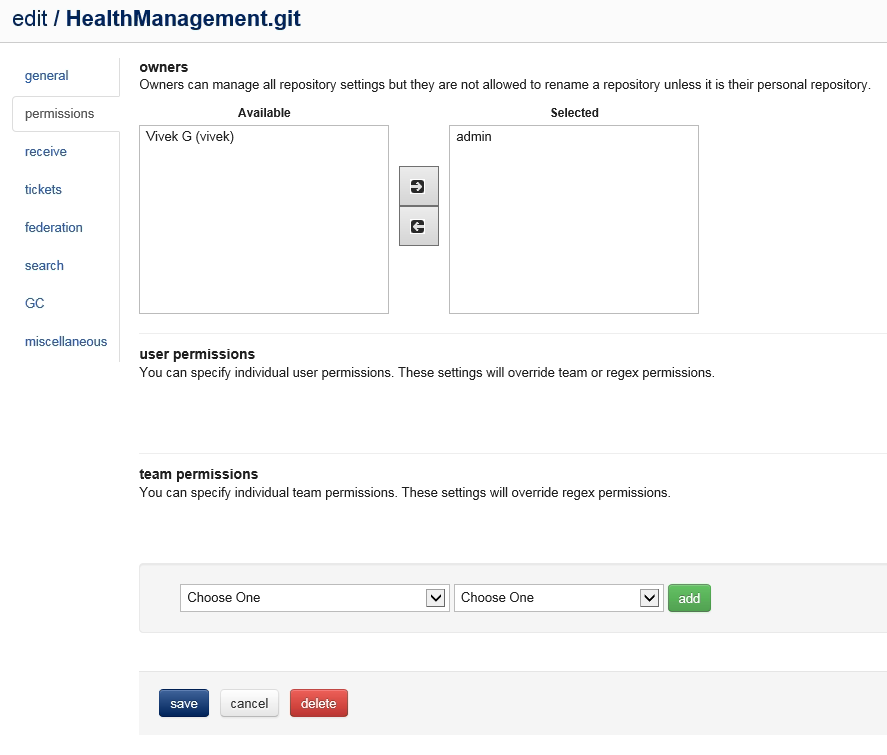
After opening repository, click Edit to modify the selected repository.

Here you can update the repository name, manage permissions for users and teams, create pre and post receives hooks.

General option will allow you to modify the repository name and access policy.

Permissions option will allow you to add users to the repository. If you have selected restricted access policies, then these selected users will have permissions to make changes to the repositories.

Permissions can also be set as per team.



# Appendix

## References

### Study Materials

Git- [http://www.git-scm.com](http://www.git-scm.com/)

GitBlit- [http://www.gitblit.com](http://www.gitblit.com/)

GitBlit Go- <http://gitblit.com/setup_go.html>

GitBlit Features - <http://gitblit.com/features.html>

### Downloads

Download latest stable version for software’s from below source.

Git: <http://www.git-scm.com>

GitBlit: [http://www.gitblit.com](http://www.gitblit.com/)