CodeArts Repo

FAQ

Issue	01
Date	2023-09-05

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Security Declaration

Vulnerability

Huawei's regulations on product vulnerability management are subject to "Vul. Response Process". For details about the policy, see the following website:https://www.huawei.com/en/psirt/vul-response-process For enterprise customers who need to obtain vulnerability information, visit:https://securitybulletin.huawei.com/enterprise/en/security-advisory

Contents

1 Managing Repositories

1.1 How Can I Set Multiple SSH Keys on My Computer?

Scenario

Developers generate only one public SSH key named **id_rsa** and submit it to multiple websites.

However, one public SSH key cannot be shared by two accounts in a website. In this case, you need to generate two SSH keys.

Procedure

Step 1 Generate two different SSH keys in the local Git repository.

```
ssh-keygen -t rsa -C "email"
Generating public/private rsa key pair.
Enter file in which to save the key (~/.ssh/id_rsa):< Type two file names before pressing Enter. >
Enter passphrase(empty for no passphrase):<Enter your custom password before pressing Enter.
```

NOTE

To generate two SSH keys, name two different file names **github_1** and **github_2**.

Step 2 Upload the two keys to the two accounts respectively.

Print the content of the ***.pub** file and paste it to the service website. Take note of the two usernames and their mappings to the keys.

Step 3 Edit the **~/.ssh/config** file.

```
Host dc_1
HostName *******.com
IdentityFile ~/.ssh/dc_1
PreferredAuthentications publickey
User username1
Host dc_2
HostName *******.com
IdentityFile ~/.ssh/dc_2
PreferredAuthentications publickey
User username2
```

Note that **Host** and **HostName** should be set to different values.

- **HostName**: Enter the service address.
- **Host**: Enter an alias for each key. The aliases will be used when you access the service.
- **IdentityFile**: Enter the location of each SSH key file. SSH key files can be stored in any directory you want.

Step 4 Read and write code.

In cases where only one SSH key is set, the SSH URL obtained from the service web page can be directly used to communicate with the service.

```
git@*****.com:name/repo.git
```

However, with two SSH keys on your computer, you need to replace the service address in the URL with a corresponding alias configured previously based on which account you are using.

```
git@dc_1:name/repo.git or git@dc_2:name/repo.git
```

----End

1.2 How Can I Prevent Unauthorized Code Replication?

Symptom

- Many development activities take place in the cloud.
- Cloud can be accessed anytime, anywhere.

Analysis

- The common process of using CodeArts is as follows:
  a. Create a repository in CodeArts and maintain it as the latest version.
  b. All developers involved can download the code for local development, and push the code to the server after development and testing is completed.
  c. Pushed code is merged to the main branch after passing code reviews.
- Potential risks:

A version of code is stored on the local computer of each developer involved. Code leak is possible if the code is copied to other devices or uploaded to web disks without permissions.

It is difficult to control operations on local computers.

Solution

- Block USB ports and disable Bluetooth usage.
- Install monitoring software to restrict uploads.
  - Monitor and prevent uploads of specific files, especially code files that are renamed, compressed, or mixed with other files.
  - Monitor and prevent uploads to specific websites, such as unknown network services, emails, and self-built servers.
  - Monitor and record all computer operations as evidences for lawsuits if code leaks happen.

1.3 How Do I Upgrade TLS?

Symptom

CodeArts Repo aims to provide secure, reliable, stable, and fast code hosting services, and always puts the code security first.

In 2018, CodeArts Repo announced the drop of TLSv1.0. Since January 11, 2019, TLsv1.0 is no longer supported in the following aspects:

- TLSv1: All HTTPS web, API, and Git connections to CodeArts Repo.

Analysis

- Many clients can automatically switch to use new algorithms.
- The few clients affected are legacy systems that are no longer maintained and continue to connect to Git or CodeArts Repo APIs using TLSv1.0.

Solution

- If the Git Bash installed on your computer is earlier than 2.6.0, upgrade it to the **latest version**. The latest Git supports TLSv1.2 by default.
- You can run the following command to check the TLS version of the local Git Bash:

  ```
  git config http.sslVersion
  ```

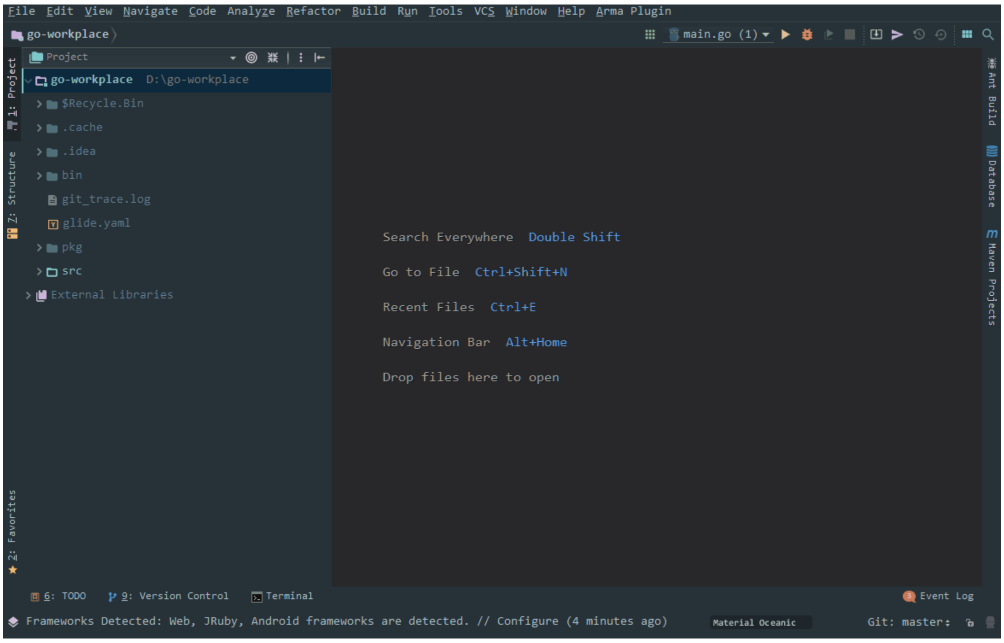
  If the version is TLSv1.0, run the following command to upgrade it to TLSv1.2:

  ```
  git config --global http.sslVersion tlsv1.2
  ```

Third-party Git GUIs

If you use a third-party Git client, such as Eclipse, JetBrains, SourceTree, VSCode, Tower, or TortoiseGit, you are advised to download the latest Git Bash and use the native Git instead of the built-in type in the tool.

- Settings in IDEA



Search Everywhere Double Shift

Go to File Ctrl+Shift+N

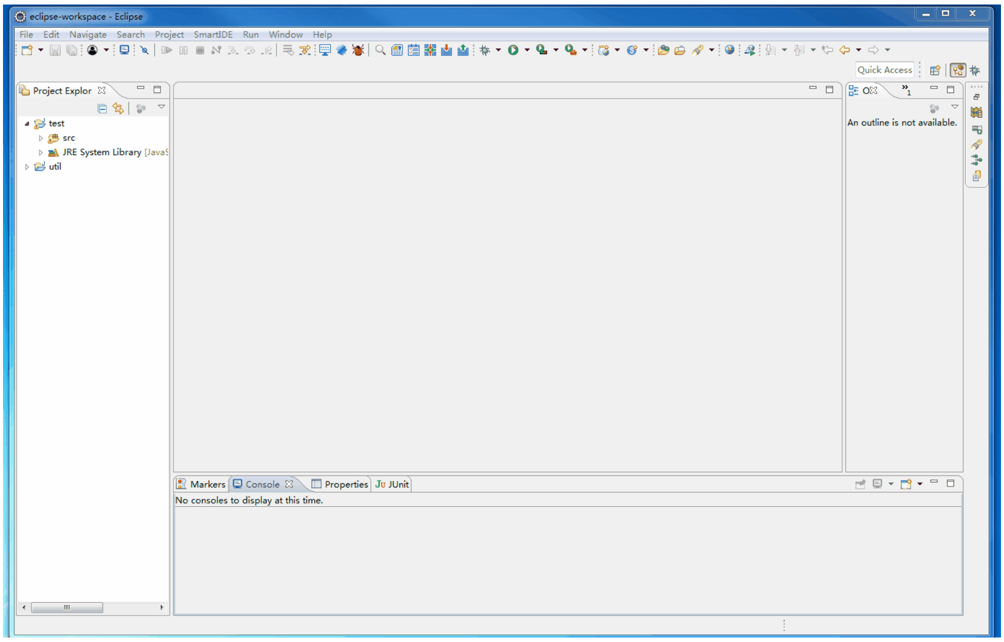
Recent Files Ctrl+E

Navigation Bar Alt+Home

Drop files here to open

Frameworks Detected: Web, JRuby, Android frameworks are detected. // Configure (4 minutes ago)

- Settings in Eclipse


eclipse-workspace - Eclipse
File Edit Navigate Search Project SmartIDE Run Window Help
Quick Access
Project Explor
test
src
JRE System Library [JavaS
util
O
Markers
Console
Properties
JUnit

An outline is not available.

No consoles to display at this time.

1.4 Should I Set a Public SSH Key for Each Repository?

No.

An SSH key is used to establish a secure connection between a computer and CodeArts Repo. After you create an SSH key pair on your computer and add the public SSH key to your account in CodeArts Repo, you can use the key to access all repositories under the account on your computer.

1.5 Can All Users Upload and Download Code with the Same SSH Key?

No.

An SSH key is used to establish a secure connection between a computer and CodeArts Repo. Different users should configure an SSH key on their own computers before connecting to CodeArts Repo via SSH.

1.6 Will the System Remove Deleted Project Members from Repositories and SSH Keys Created by Them?

No.

Deleted project members are removed from repositories, but the SSH keys created by them are not. Nonetheless, the removed members can no longer access the project and repositories.

1.7 How Can I Make a Repository Private or Public?

Yes. On the repository details page, choose **Settings** > **General Settings** > **Repository Information** and set **Visibility**.

1.8 SSH Function Upgraded

Some insecure encryption algorithms have been removed from the SSH function of CodeArts Repo.

Key Exchange (KEX) and Message Authentication Code (MAC) no longer support the following algorithms:

KEX:

- diffie-hellman-group18-sha512
- diffie-hellman-group-exchange-sha1
- diffie-hellman-group-exchange-sha256

MAC:

- hmac-md5
- hmac-md5-96
- hmac-sha1-etm@openssh.com
- hmac-sha1-96-etm@openssh.com
- hmac-md5-etm@openssh.com
- hmac-md5-96-etm@openssh.com
- umac-64@openssh.com

- umac-128@openssh.com

Upgrade your code commit tool (such as Git Bash, Eclipse, and Idea) to the latest version. By default, the new version uses algorithms that are more secure.

If you have any questions, contact technical support.

2 Using Repositories

2.1 Do I Need to Set Both an SSH Public Key and HTTPS Password Before Uploading or Downloading Code?

No.

You can use either SSH or HTTPS to communicate with CodeArts Repo for code upload or download.

- If you select SSH, **generate a key add it to the system**.
- If you select HTTPS, generate a password using CodeArts Repo. **Directly obtain the HTTPS password**.

2.2 How Can I Secure the SSH Keys?

1. You can set a passphrase for SSH keys during the key generation.
2. In CodeArts Repo, an SSH key corresponds to only one username and is visible only to that user.

2.3 Can I Set an SSH Key for Each Branch?

No. This export is not supported in CodeArts Repo.

You cannot add an SSH key for each branch in CodeArts Repo.

SSH keys are global credentials for accessing code repositories. Therefore, you cannot restrict access to a specific project or branch by setting SSH keys.

For example, if each of project members pulls a branch for development, it is impossible to set an SSH key for each branch to control the operation permissions on it.

2.4 How Do I Clear Repository Resources?

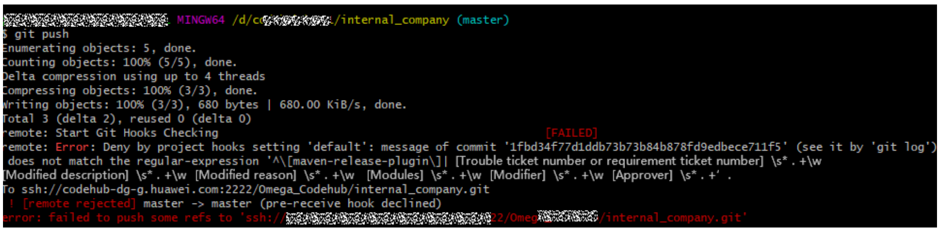
You can use the following four methods to clear repository resources.

- On the repository details page, choose **Code** > **branches**. In the branches list, select an unnecessary branch and click 🗑 to delete the branch from CodeArts Repo. Then clone or pull code from the cloud repository to delete the branch from the local host. You can also manually run the **-d** command in local repository to delete the branch.
- On the repository details page, choose **Code** > **Tags**. In the tag list, select an unnecessary tag and click 🗑 to delete the tag from CodeArts Repo. Then clone or pull code from the cloud repository to delete the tag from the local host. You can also manually run the **-d** command in local repository to delete the tag.
- On the repository details page, choose **Settings** > **Repository Management** > **Space Freeing** to run background clean-up tasks, including compressing files and removing unused objects.
- Choose **Settings** > **Repository Management** > **Submodules** in the repository details to delete unnecessary submodules.

2.5 How Do I Modify a Historical Commit Message?

Symptom

The commit message of a local commit is incorrect. The following error information is displayed:

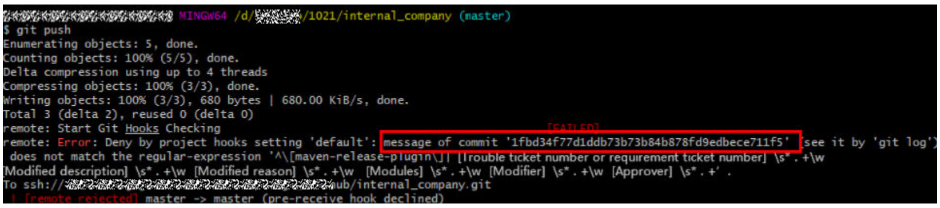


```
[redacted] MINGW64 /d/c[redacted]/internal_company (master)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 680 bytes | 680.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0)
remote: Start Git Hooks Checking                                           [FAILED]
remote: Error: Deny by project hooks setting 'default': message of commit '1fbd34f77d1ddb73b73b84b878fd9edbece711f5' (see it by 'git log')
 does not match the regular-expression '^\[maven-release-plugin\]| [Trouble ticket number or requirement ticket number] \s* . +\w
[Modified description] \s* . +\w  [Modified reason] \s* . +\w   [Modules] \s* . +\w  [Modifier] \s* . +\w  [Approver] \s* . +' .
To ssh://codehub-dg-g.huawei.com:2222/Omega_Codehub/internal_company.git
 ! [remote rejected] master -> master (pre-receive hook declined)
error: failed to push some refs to 'ssh://[redacted]22/Omeg[redacted]/internal_company.git'
```

Analysis

The error message "message of commit" is displayed, followed by a string of 40 characters consisting of digits and lowercase letters. This character string indicates that the commit message is incorrect.

The commit ID 1fbd34f77d1ddb73b73b84b878fd9edbece711f5 is incorrect. The commit message does not match the regular-expression: ^\[maven-release-plugin \]|[Trouble ticket number or requirement ticket number]\s*.+\W[Modified description]\s*.+\W[Modified reason]\s*.+\W[Modules] \s*.+\W[Modifier]\s*.+\W[Approver]\s*.+



```
[redacted] MINGW64 /d/[redacted]/1021/internal_company (master)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 680 bytes | 680.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0)
remote: Start Git Hooks Checking                                  [FAILED]
remote: Error: Deny by project hooks setting 'default': message of commit '1fbd34f77d1ddb73b73b84b878fd9edbece711f5' (see it by 'git log')
 does not match the regular-expression '^\[maven-release-plugin\]| [Trouble ticket number or requirement ticket number] \s*.+\w
[Modified description] \s*.+\w [Modified reason] \s*.+\w  [Modules] \s*.+\w [Modifier] \s*.+\w [Approver] \s*.+' .
To ssh://[redacted]ub/internal_company.git
 ! [remote rejected] master -> master (pre-receive hook declined)
```

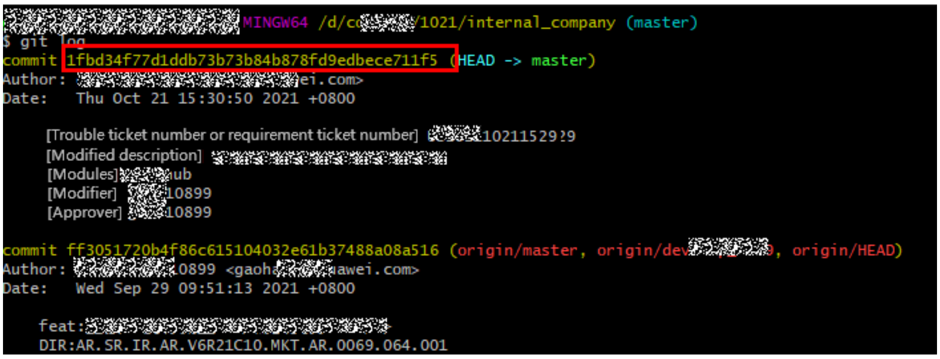
Solution

- To modify the latest incorrect commit record, perform the following operations.

  a. In the root directory of the local code workspace, that is, the level where the .git hidden folder exists, right-click and choose **git bash here** from the shortcut menu.

  b. Run the following command to query the latest commit record.

  ```
  git log
  ```

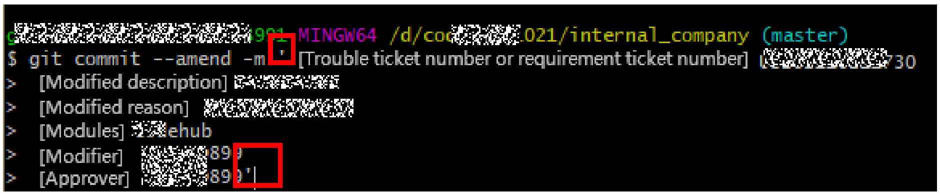
  

  ```
  [redacted] MINGW64 /d/c[redacted]/1021/internal_company (master)
  $ git log
  commit 1fbd34f77d1ddb73b73b84b878fd9edbece711f5 (HEAD -> master)
  Author: [redacted]ei.com>
  Date:   Thu Oct 21 15:30:50 2021 +0800

      [Trouble ticket number or requirement ticket number] [redacted]10211529?9
      [Modified description] [redacted]
      [Modules][redacted]ub
      [Modifier] [redacted]10899
      [Approver] [redacted]10899

  commit ff3051720b4f86c615104032e61b37488a08a516 (origin/master, origin/dev[redacted], origin/HEAD)
  Author: [redacted]0899 <gaoh[redacted]awei.com>
  Date:   Wed Sep 29 09:51:13 2021 +0800

      feat:[redacted]
      DIR:AR.SR.IR.AR.V6R21C10.MKT.AR.0069.064.001
  ```

  c. Run the following command to open the vi text editor.

  ```
  git commit --amend
  ```

  d. Enter **git commit --amend -m '**, enter the information in the first line, and press **Enter**. Enter the information in the second line, and repeat the process to the last line. Add the other single quotation mark at the end of the last line, and press **Enter**. The modification is successful.

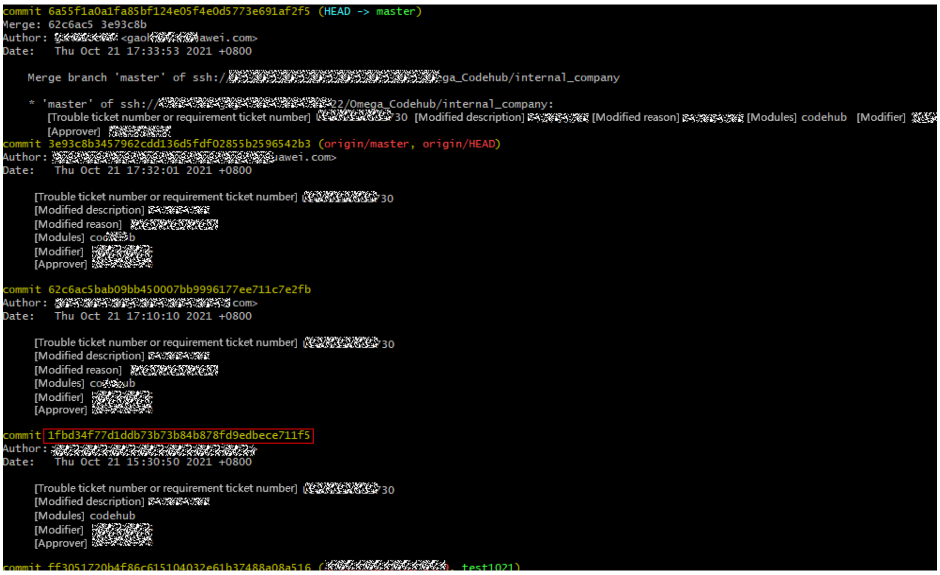
  

  ```
  [redacted]991 MINGW64 /d/coc[redacted].021/internal_company (master)
  $ git commit --amend -m '[Trouble ticket number or requirement ticket number] [redacted]30
  >   [Modified description] [redacted]
  >   [Modified reason]  [redacted]
  >   [Modules] [redacted]ehub
  >   [Modifier]  [redacted]899
  >   [Approver]  [redacted]899'
  ```

- To modify a commit record that is not the latest one, perform the following operations.

  a. Run the following command on the git bash to search for incorrect commit records.

  ```
  git log
  ```



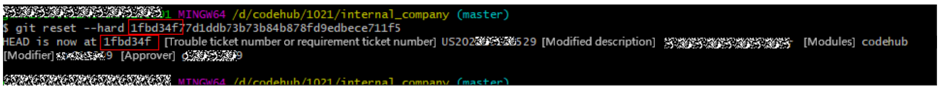
```
commit 6a55f1a0a1fa85bf124e05f4e0d5773e691af2f5 (HEAD -> master)
Merge: 62c6ac5 3e93c8b
Author: [redacted] <gaok[redacted]awei.com>
Date:   Thu Oct 21 17:33:53 2021 +0800

    Merge branch 'master' of ssh:/[redacted]ga_Codehub/internal_company

    * 'master' of ssh://[redacted]22/Omega_Codehub/internal_company:
      [Trouble ticket number or requirement ticket number] [redacted]30 [Modified description] [redacted] [Modified reason] [redacted] [Modules] codehub  [Modifier] [redacted]
      [Approver] [redacted]
commit 3e93c8b3457962cdd136d5fdf02855b2596542b3 (origin/master, origin/HEAD)
Author: [redacted]uawei.com>
Date:   Thu Oct 21 17:32:01 2021 +0800

     [Trouble ticket number or requirement ticket number] [redacted]30
     [Modified description] [redacted]
     [Modified reason] [redacted]
     [Modules] cod[redacted]b
     [Modifier] [redacted]
     [Approver]

commit 62c6ac5bab09bb450007bb9996177ee711c7e2fb
Author: [redacted]com>
Date:   Thu Oct 21 17:10:10 2021 +0800

     [Trouble ticket number or requirement ticket number] [redacted]30
     [Modified description] [redacted]
     [Modified reason] [redacted]
     [Modules] co[redacted]ub
     [Modifier] [redacted]
     [Approver]

commit 1fbd34f77d1ddb73b73b84b878fd9edbece711f5
Author: [redacted]
Date:   Thu Oct 21 15:30:50 2021 +0800

     [Trouble ticket number or requirement ticket number] [redacted]30
     [Modified description] [redacted]
     [Modules] codehub
     [Modifier] [redacted]
     [Approver]

commit ff3051720b4f86c615104032e61b37488a08a516 ([redacted], test1021)
```

b. Run the following command to roll back to the previous commit record.

```
git reset --hard commitID
```

For example, run the following command.

```
git reset --hard 1fbd34f77d1ddb73b73b84b878fd9edbece711f5
```



```
[redacted]1 MINGW64 /d/codehub/1021/internal_company (master)
$ git reset --hard 1fbd34f77d1ddb73b73b84b878fd9edbece711f5
HEAD is now at 1fbd34f  [Trouble ticket number or requirement ticket number] US202[redacted]529 [Modified description] [redacted]  [Modules] codehub
[Modifier] [redacted]9  [Approver] [redacted]9

[redacted] MINGW64 /d/codehub/1021/internal_company (master)
```

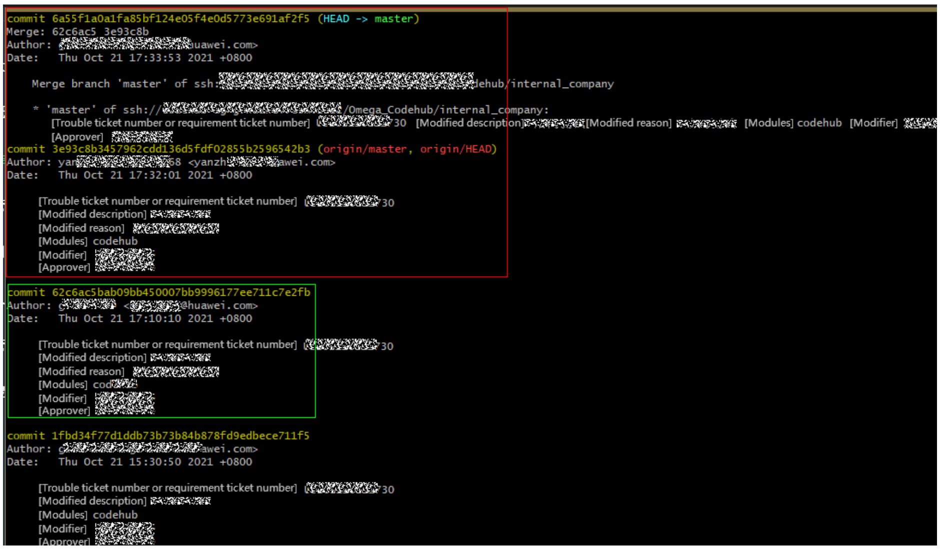
c. Modify the previous commit record by referring to 2 or 3, and then press **Enter** to save the modification.

d. Find your commit record and run cherry-pick.

NOTE

If the commit message belongs to you and you have not performed a merge operation, you can perform cherry-pick.

As shown in the following figure, the commit message in the green box can be cherry-picked, and that in the red box cannot.



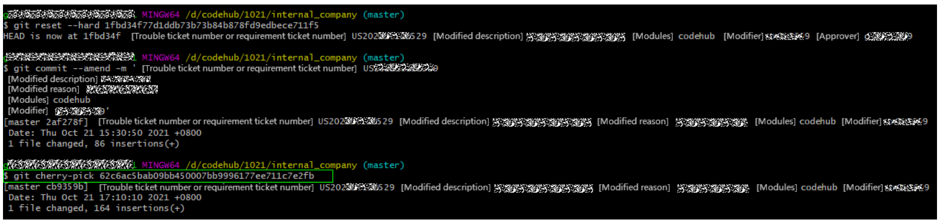
```
commit 6a55f1a0a1fa85bf124e05f4e0d5773e691af2f5 (HEAD -> master)
Merge: 62c6ac5 3e93c8b
Author: [redacted]uawei.com>
Date:   Thu Oct 21 17:33:53 2021 +0800

    Merge branch 'master' of ssh:[redacted]dehub/internal_company

    * 'master' of ssh://[redacted]/Omega_Codehub/internal_company:
      [Trouble ticket number or requirement ticket number] [redacted]30 [Modified description][redacted][Modified reason] [redacted]  [Modules] codehub  [Modifier] [redacted]
      [Approver] [redacted]
commit 3e93c8b3457962cdd136d5fdf02855b2596542b3 (origin/master, origin/HEAD)
Author: yan[redacted]68 <yanzh[redacted]awei.com>
Date:   Thu Oct 21 17:32:01 2021 +0800

     [Trouble ticket number or requirement ticket number] [redacted]30
     [Modified description] [redacted]
     [Modified reason] [redacted]
     [Modules] codehub
     [Modifier] [redacted]
     [Approver]

commit 62c6ac5bab09bb450007bb9996177ee711c7e2fb
Author: [redacted] <[redacted]@huawei.com>
Date:   Thu Oct 21 17:10:10 2021 +0800

     [Trouble ticket number or requirement ticket number] [redacted]30
     [Modified description] [redacted]
     [Modified reason] [redacted]
     [Modules] cod[redacted]
     [Modifier] [redacted]
     [Approver]

commit 1fbd34f77d1ddb73b73b84b878fd9edbece711f5
Author: [redacted]awei.com>
Date:   Thu Oct 21 15:30:50 2021 +0800

     [Trouble ticket number or requirement ticket number] [redacted]30
     [Modified description] [redacted]
     [Modules] codehub
     [Modifier] [redacted]
     [Approver]
```

e. Run the following command to select a branch.

```
git cherry-pick commitID
```

For example, the commitID is **62c6ac5bab09bb450007bb9996177ee711c7e2fb**.

```
git cherry-pick 62c6ac5bab09bb450007bb9996177ee711c7e2fb
```



```
[redacted] MINGW64 /d/codehub/1021/internal_company (master)
$ git reset --hard 1fbd34f77d1ddb73b73b84b878fd9edbece711f5
HEAD is now at 1fbd34f  [Trouble ticket number or requirement ticket number] US202[redacted]529 [Modified description] [redacted] [Modules] codehub  [Modifier] [redacted]9 [Approver] [redacted]9

[redacted] MINGW64 /d/codehub/1021/internal_company (master)
$ git commit --amend -m ' [Trouble ticket number or requirement ticket number] US[redacted]
 [Modified description] [redacted]
 [Modified reason] [redacted]
 [Modules] codehub
 [Modifier] [redacted]9'
[master 2af278f]  [Trouble ticket number or requirement ticket number] US202[redacted]529 [Modified description] [redacted] [Modified reason] [redacted] [Modules] codehub [Modifier] [redacted]9
 Date: Thu Oct 21 15:30:50 2021 +0800
 1 file changed, 86 insertions(+)

[redacted] MINGW64 /d/codehub/1021/internal_company (master)
$ git cherry-pick 62c6ac5bab09bb450007bb9996177ee711c7e2fb
[master cb9359b]  [Trouble ticket number or requirement ticket number] US202[redacted]529 [Modified description] [redacted] [Modified reason] [redacted] [Modules] codehub [Modifier] [redacted]9
 Date: Thu Oct 21 17:10:10 2021 +0800
 1 file changed, 164 insertions(+)
```

NOTE

1. If the incorrect commit message has been merged into a repository, you need to run the **git push -f** command to forcibly push the new one.
2. If the commit fails, run the **git push** command to push the new one.
3. If the error is caused by someone else and the commit has been merged into a repository, no further modification is required. You can directly delete the commit rules of the repository and push the new commit message to the repository by running the **git push** command.

3 Migrating Repositories

3.1 Can I Download Repositories in Batches in CodeArts?

No.

CodeArts Repo does not support batch download or upload of multiple code repositories. You need to perform operations on each code repository one by one. To back up local repositories, an administrator can use Shell or batch processing commands to download multiple repositories.

4 FAQs

4.1 Why Can't I View Repositories Created by Other Project Members?

To enhance code security, project members must be set as repository members before they can access to the project repositories.

To add members to a repository, access the repository details page, click the **Members** tab page and click **Add Member**.

4.2 Is There Any File Size Limit in Repositories?

Yes. Pay attention to the limits when you upload and edit code.

- When uploading code on the console, ensure that the size of a single file does not exceed 50 MB. Otherwise, the upload fails. (The client supports upload of files larger than 50 MB.)
- A maximum of 5000 lines can be edited at a time when you modify files online. Otherwise, the modification fails. (This limit does not apply to files uploaded from the client.)
- The storage space available of each repository is 2 GB.
- The maximum size of a single local file to be pushed is 200 MB. If the size exceeds 200 MB, you are advised to use the SSH protocol.

4.3 Can I Export the Statistics on Code Lines Committed by Members?

No. This export is not supported in CodeArts Repo.

However, you can check the number of committed lines on the **Repository Statistics** tab page of a repository.

4.4 How Do I Prevent Files Containing Secrets from Being Pushed to a Repository?

Background

- You can go to the repository details page, choose **Settings** > **Commit Rules**, and select **Files containing secrets cannot be committed**.
- CodeArts Repo will refuse push attempts when the name of file to be pushed matches the regular expressions listed below.

NOTICE

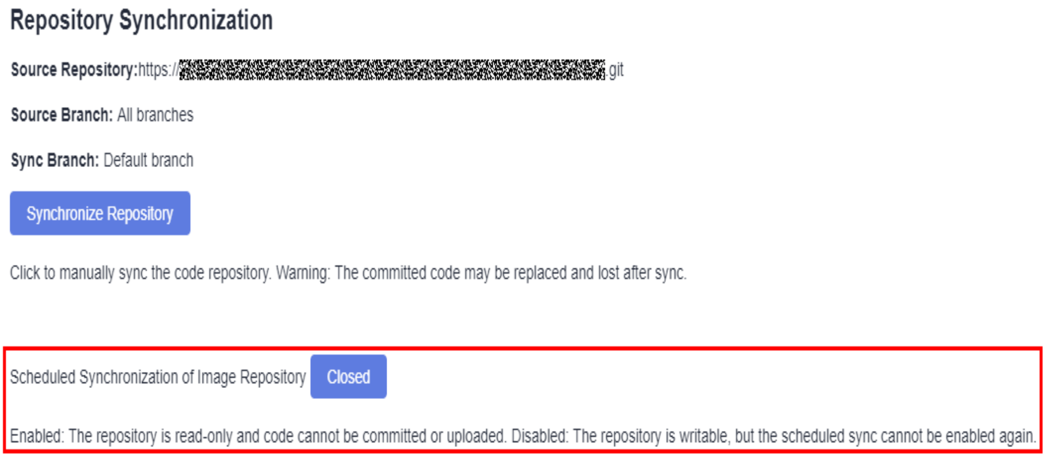
This rule does not apply to files that have been pushed.

The following table lists the regular expressions for restricting pushes of files containing secrets and provides some file name examples.

Regular Expression	File Name Example
(ssh|config)\/(personal|server)_(rsa|dsa|ed\d+|ecdsa)	ssh_server_rsa
_rsa$	id_rsa
_dsa$	id_dsa
_ed25519$	id_ed25519
_ecdsa$	id_ecdsa
\.(pem|key)$	secret.key privatekey.pem
"[._]history$"	.bash_history

4.5 Why Can't I See the Repository Synchronization Tab?

Symptom: There is no **Repository Synchronization** tab under **Repository Management** in a repository.



Repository Synchronization

Source Repository:https://[redacted]git

Source Branch: All branches

Sync Branch: Default branch

Click to manually sync the code repository. Warning: The committed code may be replaced and lost after sync.

Scheduled Synchronization of Image Repository

Enabled: The repository is read-only and code cannot be committed or uploaded. Disabled: The repository is writable, but the scheduled sync cannot be enabled again.

Analysis: This tab is displayed only in imported repositories.

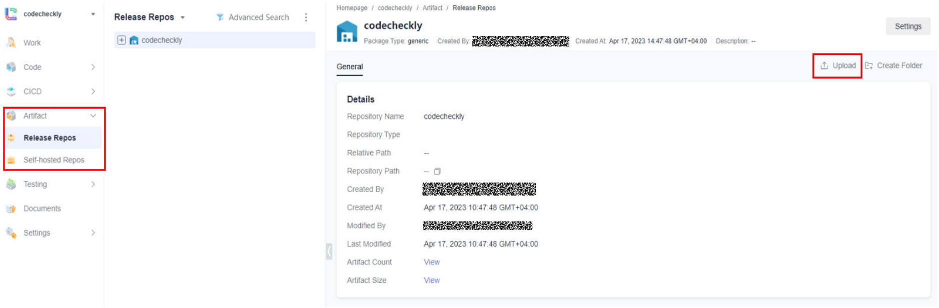
4.6 Will Related Git Code Branches Be Automatically Deleted When the Creator of a Git Branch Is Deleted?

No. After the creator of the git branch is deleted, the branch is handed over to other repository members, such as the repository manager. To view the records of deleted branches in a repository, choose **Settings** > **Security Management** > **Operation Logs**. To view history of a branch, click the **Branches** tab page, click a branch name, and click the **History** tab page.

4.7 What Do I Do If a Binary File Fails to Be Pushed to CodeArts Repo?

CodeArts Repo supports the binary file commit control function. Check whether the repository forbids binary file commit. For details about the configuration function, see **Commit Rules**.

You are not advised storing binary files in the CodeArts Repo repository. Otherwise, the performance and stability will be affected. You are advised to upload the binary file to CodeArts Artifact for storage.



codecheckly

Package Type: generic Created By: [redacted] Created At: Apr 17, 2023 14:47:48 GMT+04:00 Description: --

Details

Repository Name	codecheckly
Repository Type	
Relative Path	--
Repository Path	--
Created By	[redacted]
Created At	Apr 17, 2023 10:47:48 GMT+04:00
Modified By	[redacted]
Last Modified	Apr 17, 2023 10:47:48 GMT+04:00
Artifact Count	View
Artifact Size	View

5 Git Issues

5.1 How Does Git Identify Repository Administrators?

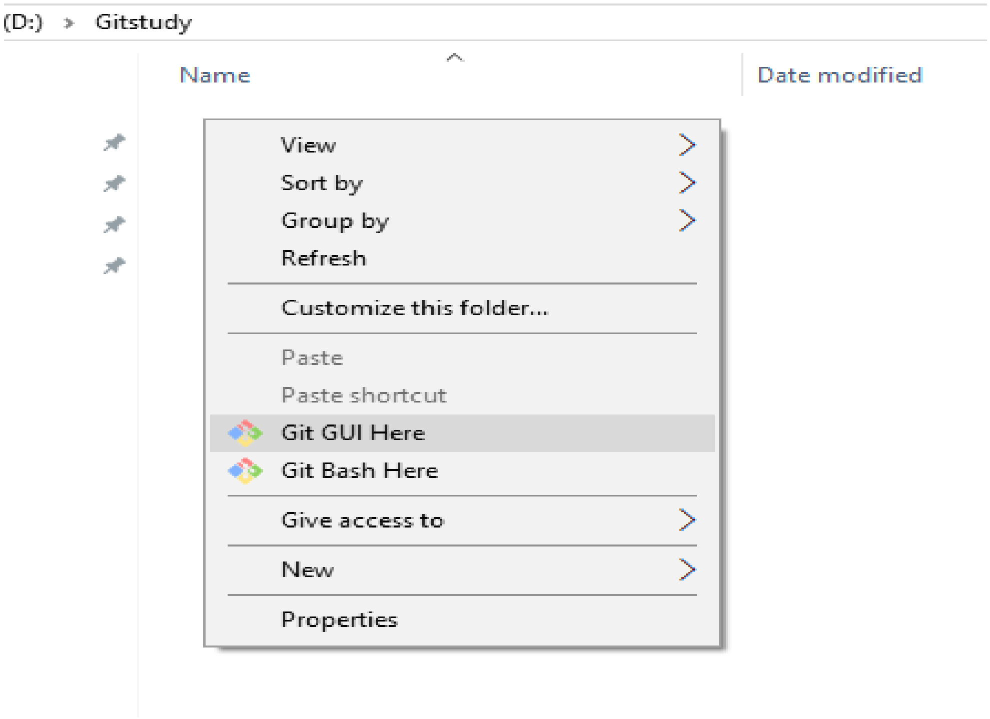
- A user needs to enter the username and password when connecting to a cloud repository via HTTPS.

  A username is in the format of account name/IAM username, or account name/account name (for Huawei Cloud). Git can judge the role of the user in a repository based on the username. For example, a username in the latter format indicates that the user is a repository administrator.
- A user needs to set an SSH key before connecting to a cloud repository via SSH.

  In CodeArts Repo, one SSH key maps to only one username. Git records the mapping between a key and username when a user accesses the cloud repository, and uses the mapping to judge the repository role of the user.

5.2 How Do I Obtain the Path Where Downloaded Code Is Stored?

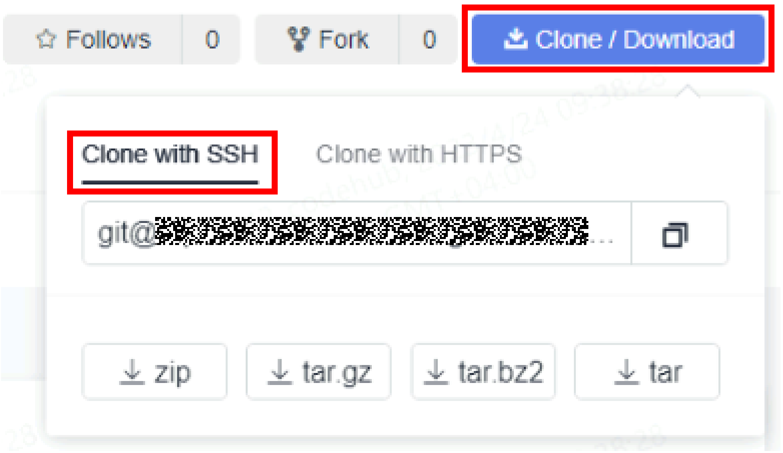
- If Git is installed in the default path and you open Git Bash from the start menu, downloaded code is stored in the default path **C:/User/*XX user***.
- If you open Git Bash by right-clicking in a directory and choosing **Git bash Here**, downloaded code is stored in this directory.



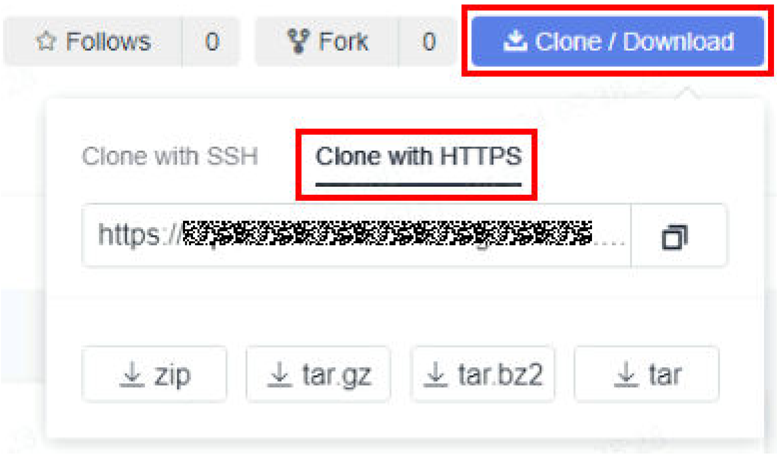
5.3 How Do I Obtain the Code Repository Address?

Step 1 On the CodeArts Repo homepage, click the name of a repository in the repository list. The repository details page is displayed.

Step 2 Click **Clone/Download** in the navigation pane. Click **Clone with SSH** to obtain the SSH protocol address.



Step 3 Click **Clone with HTTPS** to obtain the HTTPS protocol address.



https://[redacted]

----End

NOTICE

- The clone address is used only for cloning and cannot be directly accessed on the page.
- You can use the URL of the repository page to access the page.

5.4 What Are Repository URLs Used for?

Repository URLs are used for access to cloud repositories.

- You need to enter the URL of a cloud repository when running **$git clone *repository URL*** to clone the cloud repository to your local computer.
- Before you connect a local repository that is just initialized to a cloud repository, you also need to obtain the URL of the cloud repository to run **git remote add origin *repository URL***.

5.5 Does CodeArts Repo Support SVN?

No. SVN is not supported in CodeArts Repo.

CodeArts Repo is a Git-based service to enable easier remote collaboration.

However, you can import an SVN repository to CodeArts Repo. For details, see **Migrating an SVN Repository to CodeArts Repo**.

5.6 Can I Decompress a Compressed Package Uploaded to a Cloud Repository?

No.

Online decompression is not supported now. You are advised to decompress the file package locally and run the git command to upload the files.

5.7 How Do I Clear, Save, and Change the Username and Password in Git?

- To facilitate later use, you can perform the following operations to save the username and password.

  Open the Git client, configure the username and password, and run the following command to save them:

  ```
  git config --global --unset credential.helper store
  ```

- To clear the username and password, perform the following operations.

  Open the Git client and run the following command:

  ```
  git config --global --unset credential.helper
  ```

- To change the username and password, perform the following operations.

  Open the Git client, clear the previous username and password, and run the following command to reconfigure them:

  Username

  ```
  git config --global --user.name"username"
  ```

  Email

  ```
  git config --global --user.email"user@email"
  ```

  Password

  ```
  git config --global credential.helper store
  ```

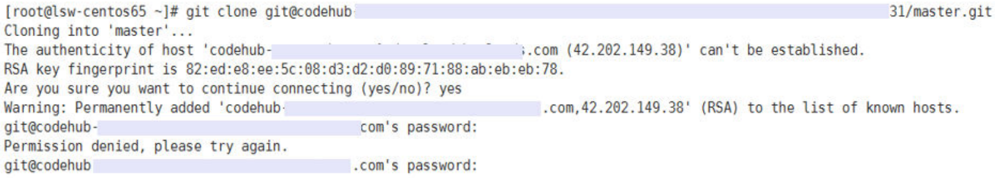
6 Troubleshooting

6.1 Private Key Is Lost

Symptom

git clone fails because the private key is incorrect or is lost. When the key authentication fails, the system switches to password-based authentication. Since no password is configured previously, the password authentication fails, either, as shown in **Figure 6-1**.

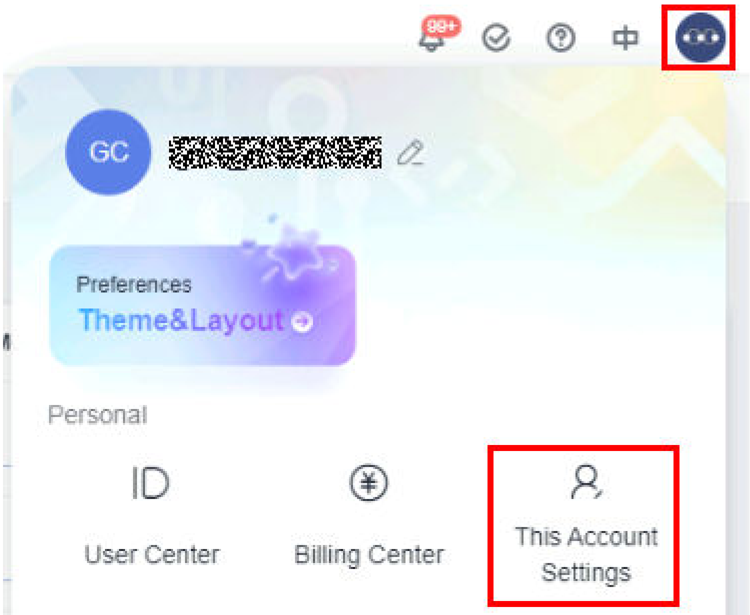
Figure 6-1 Authentication failure due to private key loss



```
[root@lsw-centos65 ~]# git clone git@codehub[redacted]31/master.git
Cloning into 'master'...
The authenticity of host 'codehub-[redacted].com (42.202.149.38)' can't be established.
RSA key fingerprint is 82:ed:e8:ee:5c:08:d3:d2:d0:89:71:88:ab:eb:eb:78.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'codehub[redacted].com,42.202.149.38' (RSA) to the list of known hosts.
git@codehub-[redacted]com's password:
Permission denied, please try again.
git@codehub[redacted].com's password:
```

Solution

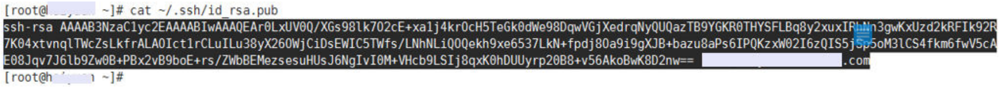
1. Log in to the CodeArts Repo service repository list page, click the alias in the upper right corner, and choose **This Account Settings** > **SSH Keys**. The page is displayed.



2. On the **SSH Keys** page displayed, click 🗑 to delete the existing key.
3. Generate a new key pair.

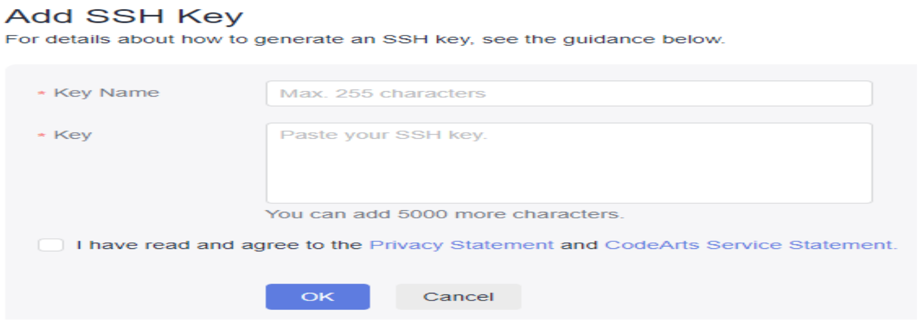
   Run **ssh-keygen -t rsa -C "abc123@***.com"** to generate a key pair.
4. Copy the generated public key and run **cat ~/.ssh/id_rsa.pub**.

Figure 6-2 Copying a public key



```
[root@[redacted] ~]# cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAr0LxUV0Q/XGs98lk702cE+xa1j4kr0cH5TeGk0dWe98DqwVGjXedrqNyQUQazTB9YGKR0THYSFLBq8y2xuxIRhNn3gwKxUzd2kRFIk92R
7K04xtvnqlTWcZsLkfrALA0Ict1rCLuILu38yX260WjCiDsEWIC5TWfs/LNhNLiQ0Qekh9xe6537LkN+fpdj80a9i9gXJB+bazu8aPs6IPQKzxW02I6zQIS5jSp5oM3lCS4fkm6fwV5cA
E08Jqv7J6lb9Zw0B+PBx2vB9boE+rs/ZWbBEMezsesuHUsJ6NgIvI0M+VHcb9LSIj8qxK0hDUUyrp20B8+v56AkoBwK8D2nw== [redacted].com
[root@h[redacted] ~]#
```

5. On the **SSH Keys** page, click **Add SSH Key**. As shown below.



Add SSH Key

For details about how to generate an SSH key, see the guidance below.

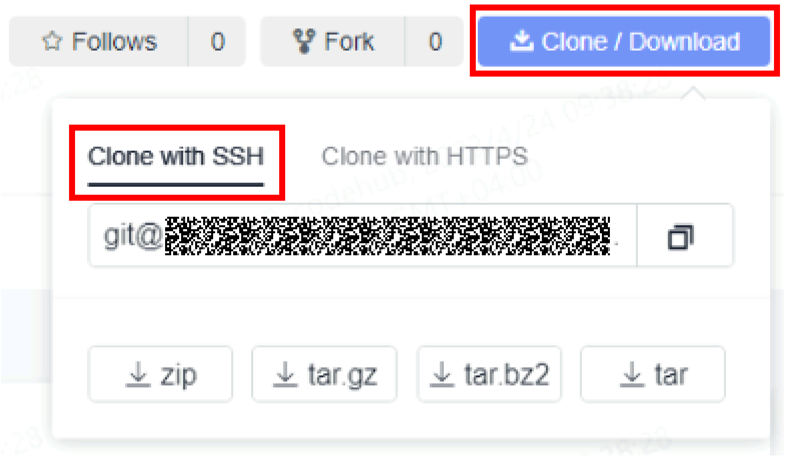
* Key Name: Max. 255 characters

* Key: Paste your SSH key.

You can add 5000 more characters.

□ I have read and agree to the Privacy Statement and CodeArts Service Statement.

6. On the CodeArts Repo homepage, click **SSH** in the **Repository URL** column of the repository to be cloned.

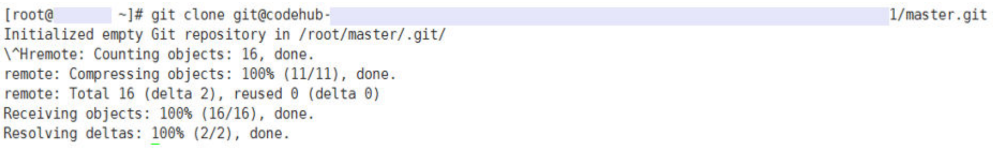


git@[redacted]

7. Clone the code.

   Run **git clone *SSH URL***.

Figure 6-3 Cloning code



```
[root@[redacted] ~]# git clone git@codehub-[redacted]1/master.git
Initialized empty Git repository in /root/master/.git/
\^Hremote: Counting objects: 16, done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 16 (delta 2), reused 0 (delta 0)
Receiving objects: 100% (16/16), done.
Resolving deltas: 100% (2/2), done.
```

6.2 SSH Key Already Exits

Symptom

When a user attempts to add an SSH key to CodeArts, a message is displayed, indicating that the key already exists and asking the user to generate a new key.

Analysis

One key maps to only one username. The error occurs because the SSH key has been added by another user.

Solution

- Ask the user who has added the key to delete the key from CodeArts.
- Alternately, generate an SSH key pair and add the new public key to CodeArts.

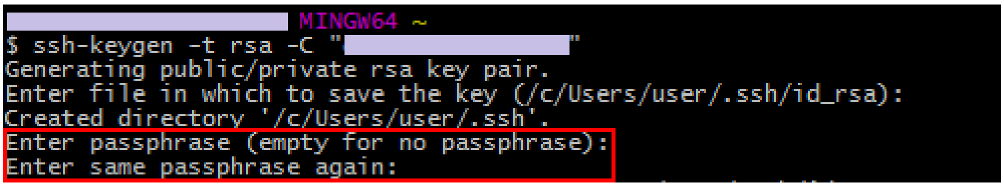
6.3 Git Keeps Asking for Passphrase During Code Clone via SSH

Symptom

When a user attempts to clone code using SSH, Git keeps prompting for the passphrase.

Analysis

A passphrase is set during the generation of the SSH key. Therefore, the passphrase is mandatory during code clone.



```
[redacted] MINGW64 ~
$ ssh-keygen -t rsa -C "[redacted]"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/user/.ssh/id_rsa):
Created directory '/c/Users/user/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
```

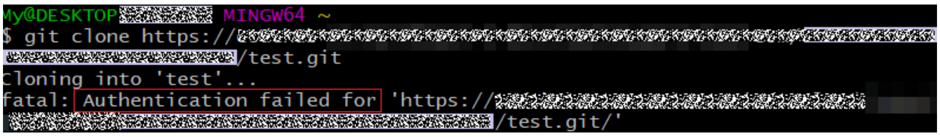
Solution

- Enter the passphrase set during the generation of the SSH key.
- To avoid entering a passphrase in each clone, delete the original SSH key from CodeArts Repo and generate a new one. Press **Enter** at the position marked in the preceding figure instead of setting a passphrase. Then, add the new key to CodeArts Repo.

6.4 Password Incorrect During Code Clone via HTTPS

Symptom

When a user attempts to clone code using HTTPS, Git prompts incorrect password after the user enters a password.



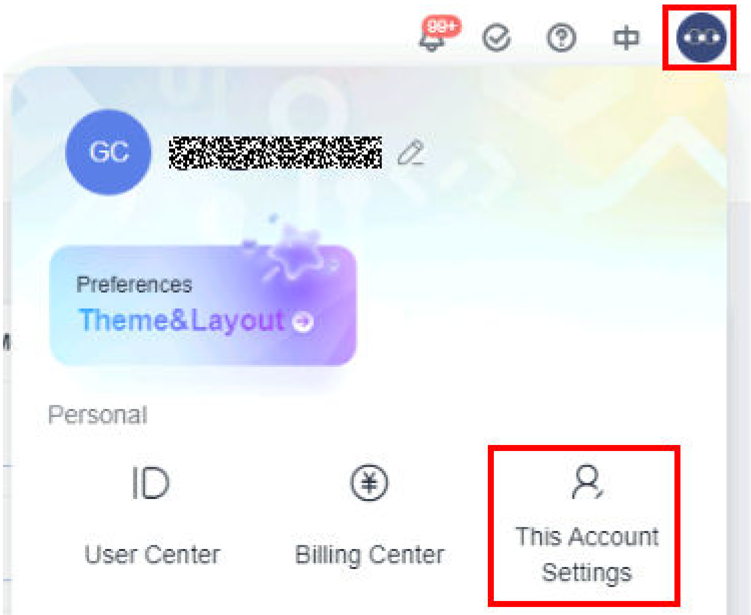
```
My@DESKTOP[redacted] MINGW64 ~
$ git clone https://[redacted]
[redacted]/test.git
Cloning into 'test'...
fatal: Authentication failed for 'https://[redacted]
[redacted]/test.git/'
```

Analysis

The password is incorrect.

Solution

Log in to the CodeArts Repo repository list page, click the alias in the upper right corner, and choose **This Account Settings > HTTPS Password**. On the page that is displayed, reset the HTTPS password.



NOTE

1. Enter a complete user name in the format of **xxx/xxx** during code clone via HTTPS.
2. To obtain the email verification code, click **Send Email**.
3. If you forget the password, set a new HTTPS password.

6.5 Committed Files Cannot Be Found in the Cloud Repository

Symptom

After a user runs **git commit** to commit a modified file, the file cannot be found in the cloud repository.

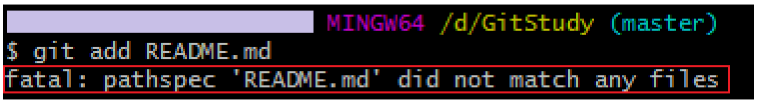
Solution

The **git commit** command is used to commit the locally modified file to the local repository. To push the file to the cloud repository, run the **git push** command.

6.6 Message "pathspec *XXX* did not match any files" Is Displayed

Symptom

When a user runs **git add**, the message **pathspec *XXX* did not match any files** is displayed.



```
[redacted] MINGW64 /d/GitStudy (master)
$ git add README.md
fatal: pathspec 'README.md' did not match any files
```

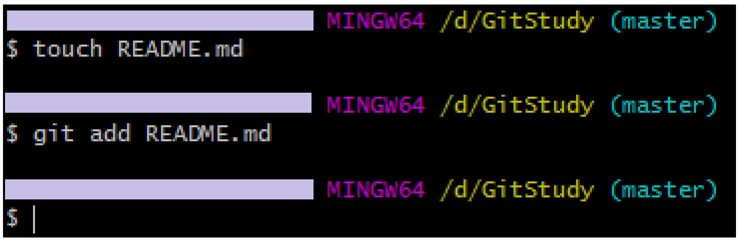
Analysis

There is no **README.md** file on the local computer. The **git add** command is used to add an existing file to the stage. It cannot create a file.

Solution

Method 1: Manually create the **README.md** file in the local repository and run **git add**.

Method 2: Run **touch README.md** to create the **README.md** file, and run **git add**.



```
[redacted] MINGW64 /d/GitStudy (master)
$ touch README.md

[redacted] MINGW64 /d/GitStudy (master)
$ git add README.md

[redacted] MINGW64 /d/GitStudy (master)
$ |
```

6.7 Message "Transport Error: cannot get remote repository refs. XXX.git: cannot open git-upload-pack" Is Displayed

Symptom

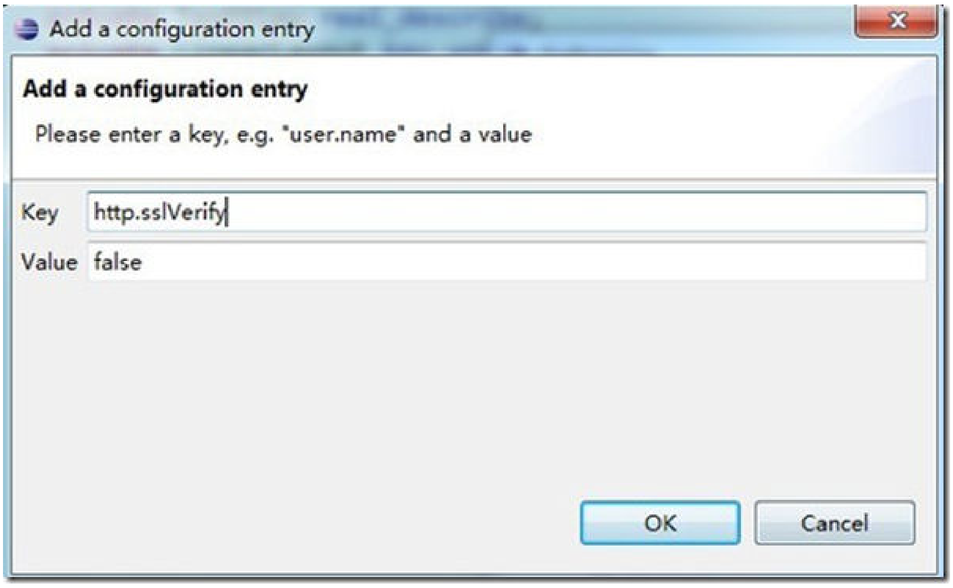
When a user attempts to access a cloud repository via HTTPS on Eclipse, the message **Transport Error: cannot get remote repository refs. *XXX*.git: cannot open git-upload-pack** is displayed.

Analysis

The configuration of the EGit on Eclipse is incorrect.

Solution

1. In the local development tool Eclipse, choose **Windows** > **Preferences** > **Team** > **Git** > **Configuration** > **User Settings**.
2. Click **Add Entry** to add a key-value pair **http.sslVerify=false**.



Add a configuration entry

Add a configuration entry

Please enter a key, e.g. "user.name" and a value

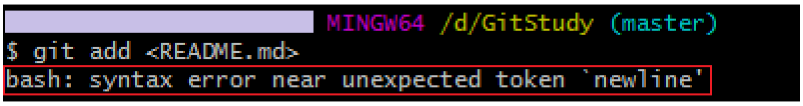
Key: http.sslVerify

Value: false

6.8 Message "syntax error near unexpected token `newline'" Is Displayed

Symptom

The message **syntax error near unexpected token `newline'** is displayed after the **git add** command is executed.



```
[redacted] MINGW64 /d/GitStudy (master)
$ git add <README.md>
bash: syntax error near unexpected token `newline'
```

Analysis

The command contains a diamond operator (<>).

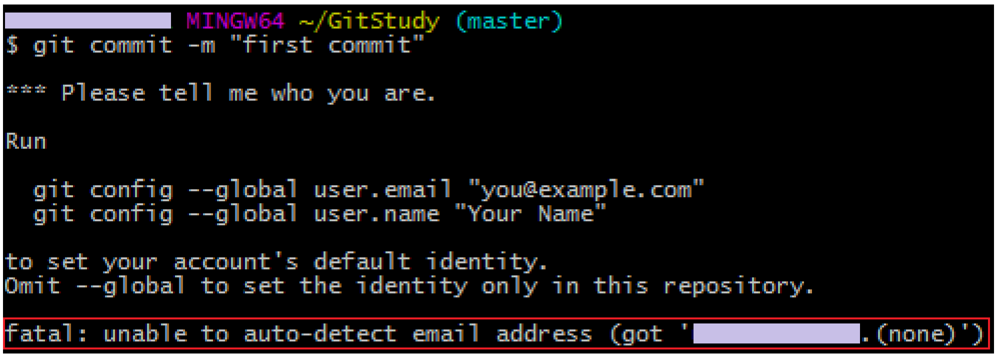
Solution

Delete the diamond operator (<>) in the command, for example, **git add README.md**.

6.9 Message "unable to auto-detect email address" Is Displayed

Symptom

The message **unable to auto-detect email address** is displayed after the **git commit** command is executed.



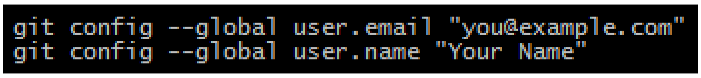
```
[redacted] MINGW64 ~/GitStudy (master)
$ git commit -m "first commit"

*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got '[redacted].(none)')
```

Analysis

The user name and email address have not set.

Solution

1. Run the following commands to complete the configurations:



```
git config --global user.email "you@example.com"
git config --global user.name "Your Name"
```

2. Run **git commit** again.

6.10 Message "fatal:Authentication failed" Is Displayed

Symptom

When a user run **git clone** to clone code using HTTPS, the message **fatal:Authentication failed** is displayed.

Analysis

The user has entered an incorrect HTTPS username or password.

Solution

Check whether the entered username and password are correct.

6.11 Message "'origin' does not appear to be a git repository" Is Displayed

Symptom

When a user attempts to push code to a cloud repository, the message **origin' does not appear to be a git repository** is displayed.

Analysis

The cloud repository has failed to be initialized.

Solution

Run **git push -u origin master** to forcibly associate with the cloud repository and push the code.

6.12 Message "You are not allowed to push code to protected branches on this project" is Displayed

Symptom

When you push code to a repository, the message **You are not allowed to push code to protected branches on this project** is displayed.

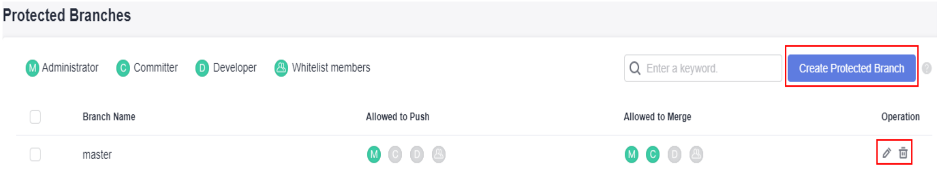
Analysis

The branch to be pushed is a protected, and the user does not have the permissions to push code to the branch.

Solution 1: Modify the Protected Branch Settings

On the repository details page, choose **Settings** > **Policy Settings** > **Protected Branches** to unprotect the branch. For details, see **Protected Branches** in the *User Guide*.

Figure 6-4 Editing Protected Branches

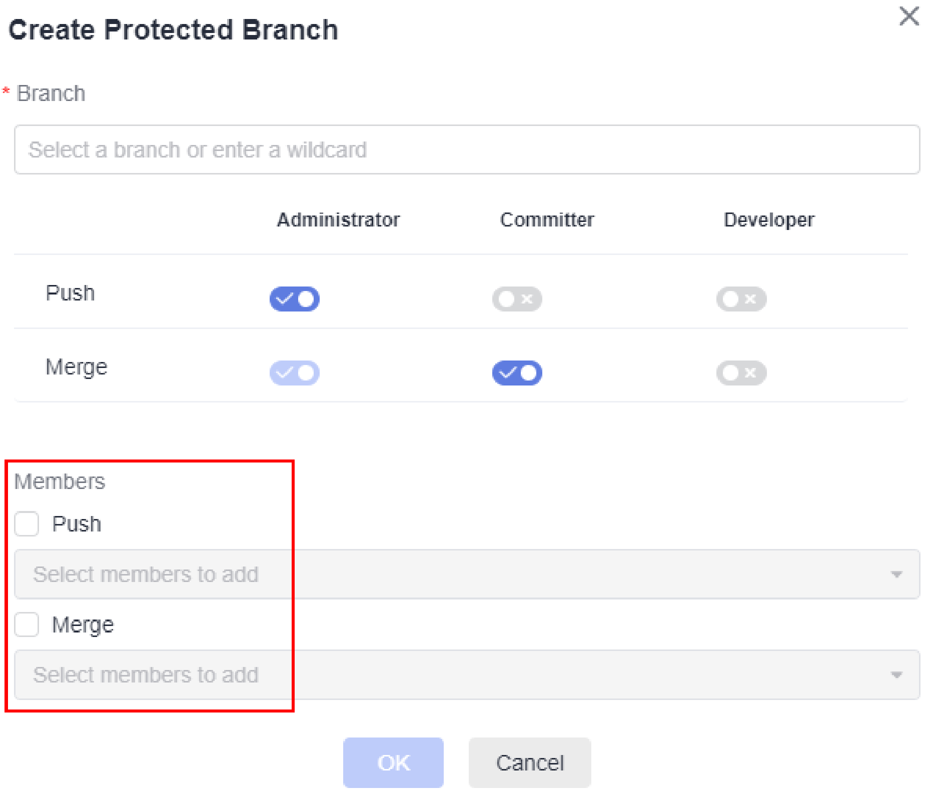


Protected Branches

M Administrator C Committer D Developer Whitelist members

□	Branch Name	Allowed to Push	Allowed to Merge	Operation
□	master	M C D	M C D	✎ 🗑

Figure 6-5 Setting a whitelist



Create Protected Branch

* Branch

Select a branch or enter a wildcard

	Administrator	Committer	Developer
Push	✓	✗	✗
Merge	✓	✓	✗

Members

□ Push

Select members to add

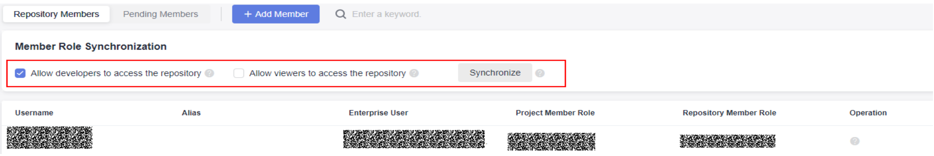
□ Merge

Select members to add

Solution 2: Change the Role of the User in the Repository

Go to the CodeArts Repo details page, click the **Members** tab page, search for the target user, click the administrator setting icon 👤, and modify the repository member role to set the member as the repository administrator. For details, see **Configuring Member Management** in the *User Guide*.

Figure 6-6 Member list



Member Role Synchronization

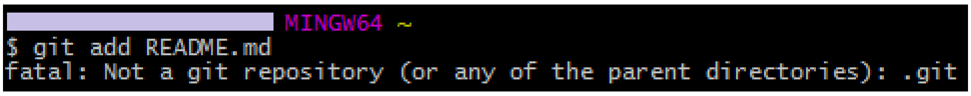
☑ Allow developers to access the repository □ Allow viewers to access the repository

Username	Alias	Enterprise User	Project Member Role	Repository Member Role	Operation
[redacted]		[redacted]	[redacted]	[redacted]	

6.13 Message "Not a git repository" Is Displayed

Symptom

The message **Not a git repository** is displayed after the **git add** command is executed.



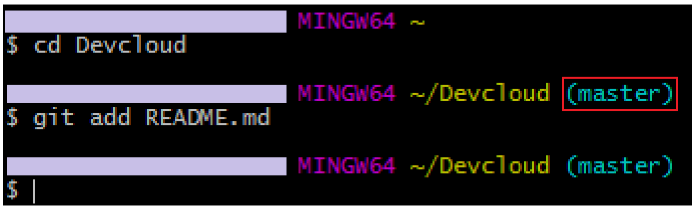
```
[redacted] MINGW64 ~
$ git add README.md
fatal: Not a git repository (or any of the parent directories): .git
```

Analysis

The current directory is not the repository directory.

Solution 1

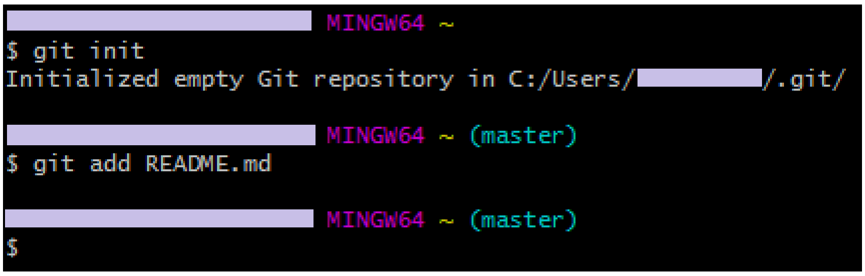
Obtain the path of the repository directory. Run **cd *repository name*** to check out the repository and run **add** again.



```
[redacted] MINGW64 ~
$ cd Devcloud

[redacted] MINGW64 ~/Devcloud (master)
$ git add README.md

[redacted] MINGW64 ~/Devcloud (master)
$ |
```

Solution 2

Run **git init** to initialize the current directory as a repository, and then run **git add** again.



```
[redacted] MINGW64 ~
$ git init
Initialized empty Git repository in C:/Users/[redacted]/.git/

[redacted] MINGW64 ~ (master)
$ git add README.md

[redacted] MINGW64 ~ (master)
$
```

6.14 Message "src refspec master does not match any" Is Displayed

Symptom

The message **src refspec master does not match any** is displayed during code push.

Analysis

No file can be uploaded from the local repository.

Solution

Run **git add** to add modified files to the stage, and **git commit** to commit the files to the local repository. Push the file to the cloud repository by running **git push**.

6.15 Message "destination path '*XXX*' already exists and is not an empty directory" Is Displayed

Symptom

The message **destination path'*XXX*' already exists and is not empty directory** is displayed during code clone from a cloud repository.

Analysis

The repository directory already exists and is not empty.

Solution

Clone the project to a new directory.

1. Enter the directory:
   ```
   cd local repository directory
   ```
2. Clone the cloud repository to the **tmp** directory.
   ```
   git clone --no-checkout repository URL tmp
   ```
3. Move the **.git** directory from the **tmp** directory to the current directory.
   ```
   mv tmp/.git name of the local repository directory
   ```
4. Delete the **tmp** directory.
   ```
   rmdir tmp
   ```
5. Roll back to the earlier version.
   ```
   git reset --hard HEAD
   ```

6.16 Message "The requested URL returned error: 401" Is Displayed

Symptom

When a user attempts to clone code using HTTPS in CentOS, the message **The requested URL returned error: 401** is displayed.

Analysis

An old Git version is installed in CentOS.

Solution

Run **git --version** to check the Git version. **CentOS 6.5** usually has the Git 1.7.1 in-built.

1. Uninstall the in-built Git 1.7.1.
   ```
   # yum remove git
   ```
2. Download the latest version from the **Git website** and add Git to the environment variables.

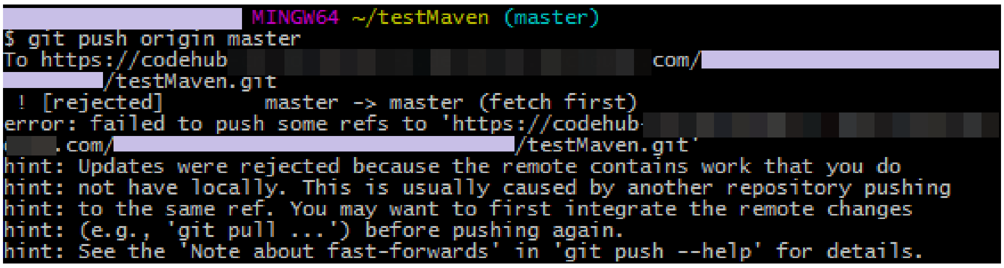
```
# wget https://github.com/git/git/archive/version.tar.gz
# tar zxvf version.tar.gz
# cd git-version
# make configure
# ./configure --prefix=/usr/local/git --with-iconv=/usr/local/libiconv
# make all doc
# make install install-doc install-html
# echo "export PATH=$PATH:/usr/local/git/bin" >> /etc/bashrc
# source /etc/bashrc
```

6.17 Code Push Fails

Symptom

The code push fails and the following error message is displayed, as shown in the following figure.

Figure 6-7 Error message



```
[redacted] MINGW64 ~/testMaven (master)
$ git push origin master
To https://codehub[redacted]com/[redacted]
[redacted]/testMaven.git
 ! [rejected]        master -> master (fetch first)
error: failed to push some refs to 'https://codehub[redacted]
[redacted].com/[redacted]/testMaven.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

Analysis

Changes have been pushed to the cloud repository from other developers. Therefore, the code in the cloud repository is different from that in the local repository.

Solution

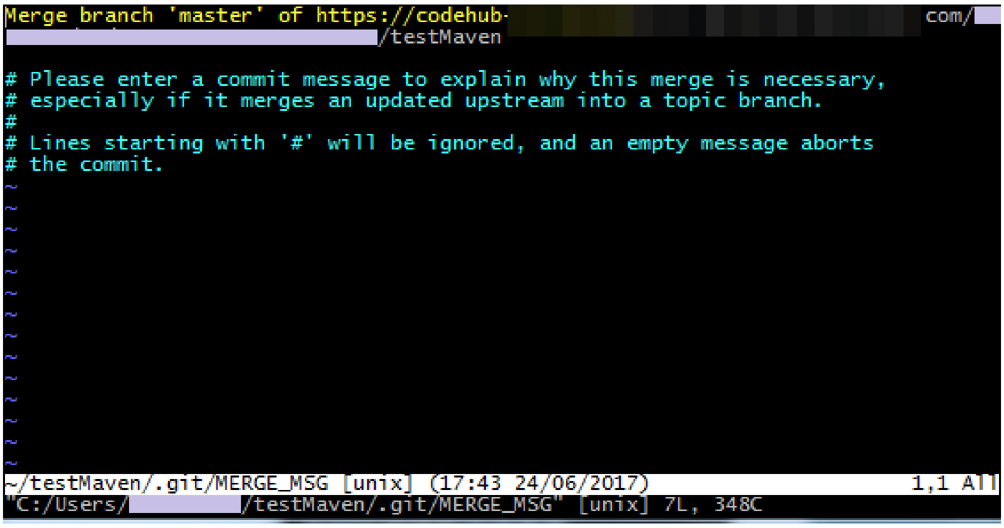
Run **git pull** to pull code from the cloud repository, merge the code, and then run **push** again.

6.18 Code Pull Fails

Symptom

git pull fails and the following message is displayed, as shown in the following figure.

Figure 6-8 Error message



```
Merge branch 'master' of https://codehub-[redacted]com/[redacted]
[redacted]/testMaven

# Please enter a commit message to explain why this merge is necessary,
# especially if it merges an updated upstream into a topic branch.
#
# Lines starting with '#' will be ignored, and an empty message aborts
# the commit.
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~/testMaven/.git/MERGE_MSG [unix] (17:43 24/06/2017)                1,1 All
"C:/Users/[redacted]/testMaven/.git/MERGE_MSG" [unix] 7L, 348C
```

Analysis

The code differs between the cloud and local repositories. Therefore, when **git pull** is executed, the remote code will be merged to the local code. The dialog box displayed asks if you confirm the merge, and prompts you to enter a commit message.

Solution

Perform the following operations on the Git client:

1. Press **i** to enter the **insert** mode.
2. Enter a commit message.
3. Enter **:wq** and press **Enter**.

6.19 Message "fatal: refusing to merge unrelated histories" Is Displayed

Symptom

The message **refusing to merge unrelated histories** is displayed after **git pull** is executed.

Analysis

The objects of the merge attempt are two unrelated repositories or branches.

Solution

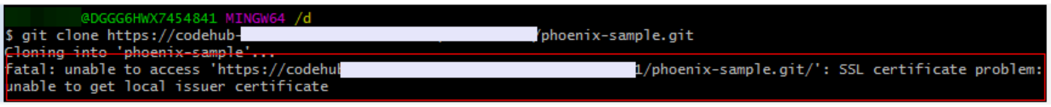
Run **git pull origin master --allow-unrelated-histories** for force merge.

6.20 Message "SSL certificate problem" Is Displayed

Symptom

During code clone, the message **SSL certificate problem: Invalid certificate chain** or **SSL certificate problem: unable to get local issuer certificate** is displayed.

Figure 6-9 Error message



```
[redacted]@DGGG6HWX7454841 MINGW64 /d
$ git clone https://codehub-[redacted]/phoenix-sample.git
Cloning into 'phoenix-sample'...
fatal: unable to access 'https://codehub[redacted]1/phoenix-sample.git/': SSL certificate problem:
unable to get local issuer certificate
```

Analysis

The local issuer certificate fails to be obtained, probably because the computer connects to an intranet using a proxy.

Solution

Run the following command:

```
git config --global http.sslVerify false
```