



# 安装**Tie**破碎 锤**1.5** ONTAP MetroCluster

NetApp  
April 25, 2024

# 目录

安装Tieb破碎 锤1.5 .....	1
配置对ONTAP API和SSH的管理员访问权限 .....	1
安装MetroCluster Tieber1.5依赖关系 .....	1
安装或升级到Tieb破碎 机1.5 .....	21

# 安装Tieb破碎 锤1.5

## 配置对ONTAP API和SSH的管理员访问权限

您可以配置对ONTAP API和SSH的管理员访问权限。

### 步骤

1. 创建具有ONTAP API访问权限的管理员用户：`security login create -user-or-group-name mcctb -application ontapi -authentication-method password`
2. 创建具有SSH访问权限的管理员用户：`security login create -user-or-group-name mcctb -application ssh -authentication-method password`
3. 验证是否已创建新的管理员用户：`security login show`
4. 在配对集群上重复上述步骤。



"管理员身份验证和 RBAC" 已实施。

## 安装MetroCluster Tieber1.5依赖关系

根据您的主机Linux操作系统、您必须先安装MySQL或MariaDB服务器、然后再安装或升级Tieb破碎 机软件。

### 步骤

1. [安装JDK](#)
2. [安装和配置存储](#)
3. 安装 MySQL 或 MariaDB 服务器：

如果 Linux 主机为	那么 ...
Red Hat Enterprise Linux 7/CentOS 7.	<a href="#">在Red Hat Enterprise Linux 7或CentOS 7上安装MySQL Server 5.5.30或更高版本以及5.6.x版本</a>
Red Hat Enterprise Linux 8	<a href="#">在Red Hat Enterprise Linux 8上安装MariaDB服务器</a>

## 安装JDK

在安装或升级Tieb破碎 机软件之前、您必须在主机系统上安装JDK。Tieber1.5及更高版本支持OpenJDK 17、18或19。

### 步骤

1. 以"root"用户或可更改为高级权限模式的sudo用户身份登录。

```
login as: root
root@mcctb's password:
Last login: Fri Jan  8 21:33:00 2017 from host.domain.com
```

## 2. 检查可用的JDK版本:

```
yum search openjdk
```

## 3. 安装JDK 17、18或19。

以下命令将安装JDK 17:

```
yum install java-17-openjdk
```

## 4. 验证安装。

```
java -version
```

成功安装将显示以下输出:

```
openjdk version "17.0.2" 2022-01-18 LTS
OpenJDK Runtime Environment 21.9 (build 17.0.2+8-LTS)
OpenJDK 64-Bit Server VM 21.9 (build 17.0.2+8-LTS, mixed mode, sharing)
```

## 安装和配置存储

如果您没有或不想使用本地存储服务器、则必须安装存储。您可以参考此标准操作步骤 来安装存储、也可以参考Hashicorp安装说明来了解其他准则。



如果网络中有存储服务器、则可以将MetroCluster Tiebreaker主机配置为使用该存储安装。如果执行此操作、则不需要在主机上安装存储。

### 步骤

#### 1. 导航到 /bin 目录:

```
[root@mcctb] cd /bin
```

#### 2. 下载存储zip文件。

```
[root@mcctb /bin]# curl -sO
https://releases.hashicorp.com/vault/1.12.2/vault_1.12.2_linux_amd64.zip
```

#### 3. 解压缩存储文件。

```
[root@mcctb /bin]# unzip vault_1.12.2_linux_amd64.zip
Archive:  vault_1.12.2_linux_amd64.zip
  inflating: vault
```

#### 4. 验证安装。

```
[root@mcctb /bin]# vault -version
Vault v1.12.2 (415e1fe3118eebd5df6cb60d13defdc01aa17b03), built 2022-11-23T12:53:46Z
```

#### 5. 导航到 /root 目录：

```
[root@mcctb /bin] cd /root
```

#### 6. 在下创建存储配置文件 /root 目录。

在 [root@mcctb ~] 提示符下、复制并运行以下命令以创建 config.hcl 文件：

```
# cat > config.hcl << EOF
storage "file" {
  address = "127.0.0.1:8500"
  path    = "/mcctb_vdata/data"
}
listener "tcp" {
  address      = "127.0.0.1:8200"
  tls_disable = 1
}
EOF
```

#### 7. 启动存储服务器：

```
[root@mcctb ~] vault server -config config.hcl &
```

#### 8. 导出存储地址。

```
[root@mcctb ~]# export VAULT_ADDR="http://127.0.0.1:8200"
```

#### 9. 初始化存储。

```
[root@mcctb ~]# vault operator init
```

```
2022-12-15T14:57:22.113+0530 [INFO] core: security barrier not
initialized
2022-12-15T14:57:22.113+0530 [INFO] core: seal configuration missing,
not initialized
2022-12-15T14:57:22.114+0530 [INFO] core: security barrier not
initialized
2022-12-15T14:57:22.116+0530 [INFO] core: security barrier initialized:
stored=1 shares=5 threshold=3
2022-12-15T14:57:22.118+0530 [INFO] core: post-unseal setup starting
2022-12-15T14:57:22.137+0530 [INFO] core: loaded wrapping token key
2022-12-15T14:57:22.137+0530 [INFO] core: Recorded vault version: vault
version=1.12.2 upgrade time="2022-12-15 09:27:22.137200412 +0000 UTC"
build date=2022-11-23T12:53:46Z
2022-12-15T14:57:22.137+0530 [INFO] core: successfully setup plugin
catalog: plugin-directory=""
2022-12-15T14:57:22.137+0530 [INFO] core: no mounts; adding default
mount table
2022-12-15T14:57:22.143+0530 [INFO] core: successfully mounted backend:
type=cubbyhole version="" path=cubbyhole/
2022-12-15T14:57:22.144+0530 [INFO] core: successfully mounted backend:
type=system version="" path=sys/
2022-12-15T14:57:22.144+0530 [INFO] core: successfully mounted backend:
type=identity version="" path=identity/
2022-12-15T14:57:22.148+0530 [INFO] core: successfully enabled
credential backend: type=token version="" path=token/ namespace="ID:
root. Path: "
2022-12-15T14:57:22.149+0530 [INFO] rollback: starting rollback manager
2022-12-15T14:57:22.149+0530 [INFO] core: restoring leases
2022-12-15T14:57:22.150+0530 [INFO] expiration: lease restore complete
2022-12-15T14:57:22.150+0530 [INFO] identity: entities restored
2022-12-15T14:57:22.150+0530 [INFO] identity: groups restored
2022-12-15T14:57:22.151+0530 [INFO] core: usage gauge collection is
disabled
2022-12-15T14:57:23.385+0530 [INFO] core: post-unseal setup complete
2022-12-15T14:57:23.387+0530 [INFO] core: root token generated
2022-12-15T14:57:23.387+0530 [INFO] core: pre-seal teardown starting
2022-12-15T14:57:23.387+0530 [INFO] rollback: stopping rollback manager
2022-12-15T14:57:23.387+0530 [INFO] core: pre-seal teardown complete
Unseal Key 1: <unseal_key_1_id>
Unseal Key 2: <unseal_key_2_id>
Unseal Key 3: <unseal_key_3_id>
Unseal Key 4: <unseal_key_4_id>
Unseal Key 5: <unseal_key_5_id>

Initial Root Token: <initial_root_token_id>
```

Vault initialized with 5 key shares and a key threshold of 3. Please securely distribute the key shares printed above. When the Vault is re-sealed, restarted, or stopped, you must supply at least 3 of these keys to unseal it before it can start servicing requests.

Vault does not store the generated root key. Without at least 3 keys to reconstruct the root key, Vault will remain permanently sealed!

It is possible to generate new unseal keys, provided you have a quorum of existing unseal keys shares. See "vault operator rekey" for more information.



您必须将密钥ID和初始根令牌记录并存储在一个安全位置、以供日后在操作步骤中使用。

#### 10. 导出存储根令牌。

```
[root@mcctb ~]# export VAULT_TOKEN="<initial_root_token_id>"
```

#### 11. 使用创建的五個密鑰中的任意三個來打開存儲。

您必須運行 `vault operator unseal` 命令：

##### a. 使用第一個密鑰打開存儲：

```
[root@mcctb ~]# vault operator unseal
Unseal Key (will be hidden):
Key                Value
---                -
Seal Type          shamir
Initialized        true
Sealed             true
Total Shares       5
Threshold          3
Unseal Progress    1/3
Unseal Nonce       <unseal_key_1_id>
Version            1.12.2
Build Date         2022-11-23T12:53:46Z
Storage Type       file
HA Enabled         false
```

b. 使用第二个密钥打开存储:

```
[root@mcctb ~]# vault operator unseal
Unseal Key (will be hidden):
Key          Value
---          -
Seal Type    shamir
Initialized  true
Sealed       true
Total Shares 5
Threshold    3
Unseal Progress 2/3
Unseal Nonce <unseal_key_2_id>
Version      1.12.2
Build Date   2022-11-23T12:53:46Z
Storage Type file
HA Enabled   false
```

c. 使用第三个密钥打开存储:



```

[root@mcctb ~]# vault operator unseal
Unseal Key (will be hidden):
2022-12-15T15:15:00.980+0530 [INFO] core.cluster-listener.tcp:
starting listener: listener_address=127.0.0.1:8201
2022-12-15T15:15:00.980+0530 [INFO] core.cluster-listener: serving
cluster requests: cluster_listen_address=127.0.0.1:8201
2022-12-15T15:15:00.981+0530 [INFO] core: post-unseal setup starting
2022-12-15T15:15:00.981+0530 [INFO] core: loaded wrapping token key
2022-12-15T15:15:00.982+0530 [INFO] core: successfully setup plugin
catalog: plugin-directory=""
2022-12-15T15:15:00.983+0530 [INFO] core: successfully mounted
backend: type=system version="" path=sys/
2022-12-15T15:15:00.984+0530 [INFO] core: successfully mounted
backend: type=identity version="" path=identity/
2022-12-15T15:15:00.984+0530 [INFO] core: successfully mounted
backend: type=cubbyhole version="" path=cubbyhole/
2022-12-15T15:15:00.986+0530 [INFO] core: successfully enabled
credential backend: type=token version="" path=token/ namespace="ID:
root. Path: "
2022-12-15T15:15:00.986+0530 [INFO] rollback: starting rollback
manager
2022-12-15T15:15:00.987+0530 [INFO] core: restoring leases
2022-12-15T15:15:00.987+0530 [INFO] expiration: lease restore
complete
2022-12-15T15:15:00.987+0530 [INFO] identity: entities restored
2022-12-15T15:15:00.987+0530 [INFO] identity: groups restored
2022-12-15T15:15:00.988+0530 [INFO] core: usage gauge collection is
disabled
2022-12-15T15:15:00.989+0530 [INFO] core: post-unseal setup complete
2022-12-15T15:15:00.989+0530 [INFO] core: vault is unsealed
Key          Value
---          -
Seal Type    shamir
Initialized  true
Sealed       false
Total Shares 5
Threshold    3
Version      1.12.2
Build Date   2022-11-23T12:53:46Z
Storage Type file
Cluster Name vault-cluster
Cluster ID   <cluster_id>
HA Enabled   false

```

## 12. 验证存储密封状态是否为false。

```
[root@mcctb ~]# vault status
Key          Value
---          -
Seal Type    shamir
Initialized  true
Sealed       false
Total Shares 5
Threshold    3
Version      1.12.2
Build Date   2022-11-23T12:53:46Z
Storage Type file
Cluster Name vault-cluster
Cluster ID   <cluster_id>
HA Enabled   false
```

13. 将存储服务配置为在引导时启动。

- a. 运行以下命令：`cd /etc/systemd/system`

```
[root@mcctb ~]# cd /etc/systemd/system
```

- b. 在 `[root@mcctb system]` 提示符下、复制并运行以下命令以创建存储服务文件。

```
# cat > vault.service << EOF
[Unit]
Description=Vault Service
After=mariadb.service

[Service]
Type=forking
ExecStart=/usr/bin/vault server -config /root/config.hcl &
Restart=on-failure

[Install]
WantedBy=multi-user.target
EOF
```

- c. 运行以下命令：`systemctl daemon-reload`

```
[root@mcctb system]# systemctl daemon-reload
```

- d. 运行以下命令：`systemctl enable vault.service`

```
[root@mcctb system]# systemctl enable vault.service
Created symlink /etc/systemd/system/multi-
user.target.wants/vault.service → /etc/systemd/system/vault.service.
```



在安装MetroCluster Tiebreaker期间、系统会提示您使用此功能。如果要更改此方法以取消存储密封、则需要卸载并重新安装MetroCluster Tiebreaker软件。

## 在Red Hat Enterprise Linux 7或CentOS 7上安装MySQL Server 5.5.30或更高版本以及5.6.x版本

在安装或升级 Tiebreaker 软件之前，必须在主机系统上安装 MySQL Server 5.5.30 或更高版本以及 5.6.x 版本。对于Red Hat Enterprise Linux 8、[安装MariaDB服务器](#)。

### 步骤

1. 以root用户或可更改为高级权限模式的sudo用户身份登录。

```
login as: root
root@mcctb's password:
Last login: Fri Jan  8 21:33:00 2016 from host.domain.com
```

2. 将 MySQL 存储库添加到主机系统:

```
`根@mcctb ~ ]# yum localinstall https://dev.mysql.com/get/mysql57-community-release-el6-11.noarch.rpm`
```

```

Loaded plugins: product-id, refresh-packagekit, security, subscription-
manager
Setting up Local Package Process
Examining /var/tmp/yum-root-LLUw0r/mysql-community-release-el6-
5.noarch.rpm: mysql-community-release-el6-5.noarch
Marking /var/tmp/yum-root-LLUw0r/mysql-community-release-el6-
5.noarch.rpm to be installed
Resolving Dependencies
--> Running transaction check
---> Package mysql-community-release.noarch 0:el6-5 will be installed
--> Finished Dependency Resolution
Dependencies Resolved

=====
=====
Package                Arch    Version
                        Repository

Size
=====
=====
Installing:
mysql-community-release
                        noarch el6-5 /mysql-community-release-el6-
5.noarch 4.3 k
Transaction Summary
=====
=====
Install                1 Package(s)
Total size: 4.3 k
Installed size: 4.3 k
Is this ok [y/N]: y
Downloading Packages:
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : mysql-community-release-el6-5.noarch
1/1
  Verifying  : mysql-community-release-el6-5.noarch
1/1
Installed:
  mysql-community-release.noarch 0:el6-5
Complete!

```

### 3. 禁用MySQL 57存储库:

```
` 根@mcctb ~ ]# yam-config-manager -disable mysql57-community`
```

#### 4. 启用MySQL 56存储库:

```
` 根@mcctb ~ ]# yam-config-manager -enable mysql56-community`
```

#### 5. 启用存储库:

```
` 根@mcctb ~ ]# yum repolist enabled | grep "mysql.-community."
```

```
mysql-connectors-community      MySQL Connectors Community
21
mysql-tools-community           MySQL Tools Community
35
mysql56-community               MySQL 5.6 Community Server
231
```

#### 6. 安装 MySQL 社区服务器:

```
` 根@mcctb ~ ]# yum install mysql-commune-server`
```

```
Loaded plugins: product-id, refresh-packagekit, security, subscription-
manager
This system is not registered to Red Hat Subscription Management. You
can use subscription-manager
to register.
Setting up Install Process
Resolving Dependencies
--> Running transaction check
.....Output truncated.....
---> Package mysql-community-libs-compat.x86_64 0:5.6.29-2.el6 will be
obsoleting
--> Finished Dependency Resolution
Dependencies Resolved

=====
=====
Package                               Arch    Version      Repository
Size
=====
=====
Installing:
mysql-community-client                 x86_64  5.6.29-2.el6  mysql56-community
18 M
replacing mysql.x86_64 5.1.71-1.el6
mysql-community-libs                   x86_64  5.6.29-2.el6  mysql56-community
1.9 M
```

```
replacing mysql-libs.x86_64 5.1.71-1.el6
mysql-community-libs-compat x86_64 5.6.29-2.el6 mysql56-community
1.6 M
replacing mysql-libs.x86_64 5.1.71-1.el6
mysql-community-server x86_64 5.6.29-2.el6 mysql56-community
53 M
replacing mysql-server.x86_64 5.1.71-1.el6
Installing for dependencies:
mysql-community-common x86_64 5.6.29-2.el6 mysql56-community
308 k
```

#### Transaction Summary

=====

=====

Install 5 Package(s)

Total download size: 74 M

**Is this ok [y/N]: y**

Downloading Packages:

```
(1/5): mysql-community-client-5.6.29-2.el6.x86_64.rpm | 18 MB
00:28
(2/5): mysql-community-common-5.6.29-2.el6.x86_64.rpm | 308 kB
00:01
(3/5): mysql-community-libs-5.6.29-2.el6.x86_64.rpm | 1.9 MB
00:05
(4/5): mysql-community-libs-compat-5.6.29-2.el6.x86_64.rpm | 1.6 MB
00:05
(5/5): mysql-community-server-5.6.29-2.el6.x86_64.rpm | 53 MB
03:42
```

-----

-----

```
Total 289 kB/s | 74 MB
04:24
```

warning: rpmts\_HdrFromFdno: Header V3 DSA/SHA1 Signature, key ID  
<key\_id> NOKEY

Retrieving key from file:/etc/pki/rpm-gpg/RPM-GPG-KEY-mysql

Importing GPG key 0x5072E1F5:

  Userid : MySQL Release Engineering <mysql-build@oss.oracle.com>

  Package: mysql-community-release-el6-5.noarch

          (@/mysql-community-release-el6-5.noarch)

  From : file:/etc/pki/rpm-gpg/RPM-GPG-KEY-mysql

**Is this ok [y/N]: y**

Running rpm\_check\_debug

Running Transaction Test

Transaction Test Succeeded

Running Transaction

  Installing : mysql-community-common-5.6.29-2.el6.x86\_64

```
....Output truncated....
```

```
1.el6.x86_64
```

```
7/8
```

```
Verifying : mysql-5.1.71-1.el6.x86_64
```

```
8/8
```

```
Installed:
```

```
mysql-community-client.x86_64 0:5.6.29-2.el6
```

```
mysql-community-libs.x86_64 0:5.6.29-2.el6
```

```
mysql-community-libs-compat.x86_64 0:5.6.29-2.el6
```

```
mysql-community-server.x86_64 0:5.6.29-2.el6
```

```
Dependency Installed:
```

```
mysql-community-common.x86_64 0:5.6.29-2.el6
```

```
Replaced:
```

```
mysql.x86_64 0:5.1.71-1.el6 mysql-libs.x86_64 0:5.1.71-1.el6
```

```
mysql-server.x86_64 0:5.1.71-1.el6
```

```
Complete!
```

## 7. 启动 MySQL 服务器:

```
`根@mcctb ~ ]# service mysqld start`
```

```
Initializing MySQL database: 2016-04-05 19:44:38 0 [Warning] TIMESTAMP
with implicit DEFAULT value is deprecated. Please use
--explicit_defaults_for_timestamp server option (see documentation
for more details).
2016-04-05 19:44:38 0 [Note] /usr/sbin/mysqld (mysqld 5.6.29)
      starting as process 2487 ...
2016-04-05 19:44:38 2487 [Note] InnoDB: Using atomics to ref count
      buffer pool pages
2016-04-05 19:44:38 2487 [Note] InnoDB: The InnoDB memory heap is
disabled
....Output truncated....
2016-04-05 19:44:42 2509 [Note] InnoDB: Shutdown completed; log sequence
      number 1625987
```

PLEASE REMEMBER TO SET A PASSWORD FOR THE MySQL root USER!  
To do so, start the server, then issue the following commands:

```
/usr/bin/mysqladmin -u root password 'new-password'
/usr/bin/mysqladmin -u root -h mcctb password 'new-password'
```

Alternatively, you can run:

```
/usr/bin/mysql_secure_installation
```

which will also give you the option of removing the test  
databases and anonymous user created by default. This is  
strongly recommended for production servers.

.....Output truncated.....

```
WARNING: Default config file /etc/my.cnf exists on the system
This file will be read by default by the MySQL server
If you do not want to use this, either remove it, or use the
--defaults-file argument to mysqld_safe when starting the server
```

```
Starting mysqld: [ OK ]
```

## 8. 确认 MySQL 服务器正在运行:

```
`根@mcctb ~]# service mysqld status`
```

```
mysqld (pid 2739) is running...
```

## 9. 配置安全性和密码设置:

```
`根@mcctb ~]# mysql_secure_installation`
```



NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MySQL  
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MySQL to secure it, we'll need the current password for the root user. If you've just installed MySQL, and you haven't set the root password yet, the password will be blank, so you should just press enter here.

**Enter current password for root (enter for none):** <== on default  
install

hit enter here

OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MySQL root user without the proper authorization.

**Set root password? [Y/n] y**

**New password:**

**Re-enter new password:**

Password updated successfully!

Reloading privilege tables..

... Success!

By default, a MySQL installation has an anonymous user, allowing anyone to log into MySQL without having to have a user account created for them. This is intended only for testing, and to make the installation go a bit smoother. You should remove them before moving into a production environment.

**Remove anonymous users? [Y/n] y**

... Success!

Normally, root should only be allowed to connect from 'localhost'. This ensures that someone cannot guess at the root password from the network.

**Disallow root login remotely? [Y/n] y**

... Success!

By default, MySQL comes with a database named 'test' that anyone can access. This is also intended only for testing, and should be removed before moving into a production environment.

**Remove test database and access to it? [Y/n] y**

- Dropping test database...

ERROR 1008 (HY000) at line 1: Can't drop database 'test';

```
database doesn't exist
... Failed! Not critical, keep moving...
- Removing privileges on test database...
... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] y
... Success!

All done! If you've completed all of the above steps, your MySQL
installation should now be secure.

Thanks for using MySQL!

Cleaning up...
```

#### 10. 验证 MySQL 登录是否正常工作:

```
`根@mcctb ~ ]# mysql -u root -p`
```

```
Enter password: <configured_password>
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 17
Server version: 5.6.29 MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights
reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.
mysql>
```

如果 MySQL 登录正常，输出将在 `mysql>` 提示符处结束。

#### 启用MySQL自动启动设置

您应验证是否已为MySQL守护进程启用自动启动功能。如果 MetroCluster Tiebreaker 软件所在的系统重新启动，则打开 MySQL 守护进程会自动重新启动 MySQL。如果 MySQL 守护进程未运行，Tiebreaker 软件将继续运行，但无法重新启动，并且无法更改配置。

## 步骤

1. 验证是否已启用 MySQL 在启动时自动启动：

```
`根@mcctb ~ ]# systemctl list-unit-files mysqld.service`
```

```
UNIT FILE           State
-----
mysqld.service     enabled
```

如果在启动时未启用 MySQL 自动启动，请参见 MySQL 文档为您的安装启用自动启动功能。

## 在Red Hat Enterprise Linux 8上安装MariaDB服务器

在安装或升级 Tiebreaker 软件之前，必须在主机系统上安装 MariaDB 服务器。对于 Red Hat Enterprise Linux 7 或 CentOS 7、[安装MySQL Server](#)。

### 开始之前

主机系统必须运行在 Red Hat Enterprise Linux (RHEL) 8 上。

## 步骤

1. 以登录身份 root 可通过sudo进入高级权限模式的用户。

```
login as: root
root@mcctb's password:
Last login: Fri Jan  8 21:33:00 2017 from host.domain.com
```

2. 安装MariaDB服务器：

```
`根@mcctb ~ ]# yum install MariaDB-server.x86_64`
```

```
[root@mcctb ~]# yum install mariadb-server.x86_64
Loaded plugins: fastestmirror, langpacks
...
...
=====
===
Package                Arch  Version           Repository
Size
=====
===
Installing:
mariadb-server         x86_64  1:5.5.56-2.el7   base
11 M
```

```
Installing for dependencies:
```

```
Transaction Summary
```

```
=====
===
```

```
Install 1 Package (+8 Dependent packages)
Upgrade          ( 1 Dependent package)
```

```
Total download size: 22 M
```

```
Is this ok [y/d/N]: y
```

```
Downloading packages:
```

```
No Presto metadata available for base warning:
```

```
/var/cache/yum/x86_64/7/base/packages/mariadb-libs-5.5.56-2.e17.x86_64.rpm:
```

```
Header V3 RSA/SHA256 Signature,
```

```
key ID f4a80eb5: NOKEY] 1.4 MB/s | 3.3 MB 00:00:13 ETA
```

```
Public key for mariadb-libs-5.5.56-2.e17.x86_64.rpm is not installed
```

```
(1/10): mariadb-libs-5.5.56-2.e17.x86_64.rpm | 757 kB 00:00:01
```

```
..
```

```
..
```

```
(10/10): perl-Net-Daemon-0.48-5.e17.noarch.rpm | 51 kB 00:00:01
```

```
-----
-----
```

```
Installed:
```

```
  mariadb-server.x86_64 1:5.5.56-2.e17
```

```
Dependency Installed:
```

```
  mariadb.x86_64 1:5.5.56-2.e17
```

```
  perl-Compress-Raw-Bzip2.x86_64 0:2.061-3.e17
```

```
  perl-Compress-Raw-Zlib.x86_64 1:2.061-4.e17
```

```
  perl-DBD-MySQL.x86_64 0:4.023-5.e17
```

```
  perl-DBI.x86_64 0:1.627-4.e17
```

```
  perl-IO-Compress.noarch 0:2.061-2.e17
```

```
  perl-Net-Daemon.noarch 0:0.48-5.e17
```

```
  perl-PlRPC.noarch 0:0.2020-14.e17
```

```
Dependency Updated:
```

```
  mariadb-libs.x86_64 1:5.5.56-2.e17
```

```
Complete!
```

### 3. 启动 MariaDB 服务器:

```
`根@mcctb ~ ]# systemctl start MariaDB`
```

### 4. 验证MariaDB服务器是否已启动:

```
根@mcctb ~]# systemctl status MariaDB`
```

```
[root@mcctb ~]# systemctl status mariadb
mariadb.service - MariaDB database server
...
Nov 08 21:28:59 mcctb systemd[1]: Starting MariaDB database server...
...
Nov 08 21:29:01 mcctb systemd[1]: Started MariaDB database server.
```

## 5. 配置安全性和密码设置:



当系统提示您输入root密码时、请将其留空、然后按Enter继续配置安全性和密码设置。

```
根@mcctb ~]# mysql_secure_installation`
```

```
root@localhost systemd]# mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.

Set root password? [Y/n] y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!

By default, a MariaDB installation has an anonymous user, allowing
anyone
to log into MariaDB without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.
```

```
Remove anonymous users? [Y/n] y
```

```
... Success!
```

Normally, root should only be allowed to connect from 'localhost'. This ensures that someone cannot guess at the root password from the network.

```
Disallow root login remotely? [Y/n] y
```

```
... Success!
```

By default, MariaDB comes with a database named 'test' that anyone can access. This is also intended only for testing, and should be removed before moving into a production environment.

```
Remove test database and access to it? [Y/n] y
```

```
- Dropping test database...
```

```
... Success!
```

```
- Removing privileges on test database...
```

```
... Success!
```

Reloading the privilege tables will ensure that all changes made so far will take effect immediately.

```
Reload privilege tables now? [Y/n]
```

```
... Success!
```

```
Cleaning up...
```

All done! If you've completed all of the above steps, your MariaDB installation should now be secure.

```
Thanks for using MariaDB!
```

## 为MariaDB服务器启用自动启动设置

您应验证是否已为MariaDB服务器启用自动启动功能。如果不启用自动启动功能，并且 MetroCluster Tiebreaker 软件所在的系统必须重新启动，则 Tiebreaker 软件将继续运行，但无法重新启动 MariaDB 服务，也无法更改配置。

### 步骤

1. 启用自动启动服务：

```
`根@mcctb ~ ]# systemctl enable mariadb.service`
```

2. 验证启动时 MariaDB 是否已启用自动启动：

```
根@mcctb ~ ]# systemctl list-unit-files mariadb.service`
```

```
UNIT FILE           State
-----
mariadb.service    enabled
```

## 安装或升级到Tiebreak 机1.5

在主机Linux操作系统上全新安装或升级到Tiebreak 机1.5、以监控MetroCluster配置。

关于此任务

- 存储系统必须运行受支持的ONTAP版本。请参见 ["软件要求"](#) 表以了解更多详细信息。
- 您必须已使用安装OpenJDK `yum install java-x.x.x-openjdk` 命令：Tiebreak 1.5及更高版本支持OpenJDK 17、18或19。
- 您可以使用具有足够管理权限的非root用户身份安装MetroCluster Tiebreak 机、以便执行Tiebreak 机安装、创建表和用户以及设置用户密码。

步骤

1. 下载MetroCluster Tiebreak 机软件和MetroCluster\_Tiebreak 机\_RPM\_GPG密钥。



可以从NetApp 支持站点 上下载Tiebreak 机1.5软件包的同一页面下载MetroCluster\_Tiebreak 机\_RPM\_GPG密钥。

["MetroCluster Tiebreak 机\(下载\)—NetApp 支持站点"](#)

2. 以 root 用户身份登录到主机。
3. 创建非root用户和 mcctbgrp 组。
  - a. 创建非root用户并设置密码。

以下示例命令将创建一个名为的非root用户 mcctbuser1:

```
[root@mcctb ~]# useradd mcctbuser1
[root@mcctb ~]# passwd mcctbuser1
Changing password for user mcctbuser1.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

- b. 创建一个名为的组 mcctbgrp:

```
[root@mcctb ~]# groupadd mcctbgrp
```

- c. 将您创建的非root用户添加到 mcctbgrp 组。

以下命令将添加 mcctbuser1 到 mcctbgrp 组：

```
[root@mcctb ~]# usermod -a -G mcctbgrp mcctbuser1
```

#### 4. 验证RPM文件。

从包含RPM密钥的目录运行以下子步骤。

##### a. 下载并导入RPM密钥文件：

```
[root@mcctb ~]# rpm --import MetroCluster_Tiebreaker_RPM_GPG.key
```

##### b. 通过检查指纹来验证是否导入了正确的密钥。

以下示例显示了正确的密钥指纹：

```
root@mcctb:~/signing/mcctb-rpms# gpg --show-keys --with-fingerprint
MetroCluster_Tiebreaker_RPM_GPG.key
pub   rsa3072 2022-11-17 [SCEA] [expires: 2025-11-16]
       65AC 1562 E28A 1497 7BBD  7251 2855 EB02 3E77 FAE5
uid           MCCTB-RPM (mcctb RPM production signing)
<mcctb-rpm@netapp.com>
```

##### a. 验证签名： rpm --checksig NetApp-MetroCluster-Tiebreaker-Software-1.5-1.x86\_64.rpm

```
NetApp-MetroCluster-Tiebreaker-Software-1.5-1.x86_64.rpm: digests OK
```



只有在成功验证签名后、才能继续安装。

#### 5. 【安装- Tiebreaker】 安装或升级Tiebreaker软件：



只有在从Tiebreaker 1.4版升级时、才能升级到Tiebreaker 1.5版。不支持从早期版本升级到Tiebreaker 1.5。

根据您是执行新安装还是升级现有安装、选择正确的操作步骤。



## 执行新安装

- a. 检索并记录Java的绝对路径:

```
[root@mcctb ~]# readlink -f /usr/bin/java  
/usr/lib/jvm/java-19-openjdk-19.0.0.0.36-  
2.rolling.el8.x86_64/bin/java
```

- b. 运行以下命令: `rpm -ivh NetApp-MetroCluster-Tiebreaker-Software-1.5-1.x86_64.rpm`

成功安装时, 系统将显示以下输出:



在安装期间出现提示时、请提供您先前创建并分配给的非root用户 `mcctbgrp` 组。

```
Verifying...
##### [100%]
Preparing...
##### [100%]
Updating / installing...
  1:NetApp-MetroCluster-Tiebreaker-
So##### [100%]
Enter the absolute path for Java : /usr/lib/jvm/java-19-openjdk-
19.0.0.0.36-2.rolling.el8.x86_64/bin/java
Verifying if Java exists...
Found Java. Proceeding with the installation.
Enter host user account to use for the installation:
mcctbuser1
User account mcctbuser1 found. Proceeding with the installation
Enter database user name:
root
Please enter database password for root
Enter password:
Sealed          false
Do you wish to auto unseal vault(y/n)?y
Enter the key1:
Enter the key2:
Enter the key3:
Success! Uploaded policy: mcctb-policy
Error enabling approle auth: Error making API request.
URL: POST http://127.0.0.1:8200/v1/sys/auth/approle
Code: 400. Errors:
* path is already in use at approle/
Success! Enabled the kv secrets engine at: mcctb/
Success! Data written to: auth/approle/role/mcctb-app
Password updated successfully in the vault.
Synchronizing state of netapp-metrocluster-tiebreaker-
software.service with SysV service script with
/usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable netapp-
metrocluster-tiebreaker-software
Created symlink /etc/systemd/system/multi-
user.target.wants/netapp-metrocluster-tiebreaker-software.service
→ /etc/systemd/system/netapp-metrocluster-tiebreaker-
software.service.
Attempting to start NetApp MetroCluster Tiebreaker software
services
Started NetApp MetroCluster Tiebreaker software services
Successfully installed NetApp MetroCluster Tiebreaker software
version 1.5.
```

## 升级现有安装

- a. 验证是否已安装受支持的OpenJDK版本、以及是否为主机上的当前Java版本。



要升级到Tiebreaker 1.5、您必须安装OpenJDK 17、18或19版。

```
[root@mcctb ~]# readlink -f /usr/bin/java
/usr/lib/jvm/java-19-openjdk-19.0.0.0.36-
2.rolling.el8.x86_64/bin/java
```

- b. 验证存储服务是否已取消密封并正在运行: `vault status`

```
[root@mcctb ~]# vault status
Key          Value
---          -
Seal Type    shamir
Initialized   true
Sealed       false
Total Shares  5
Threshold    3
Version      1.12.2
Build Date   2022-11-23T12:53:46Z
Storage Type  file
Cluster Name  vault
Cluster ID    <cluster_id>
HA Enabled    false
```

- c. 升级Tiebreaker软件。

```
[root@mcctb ~]# rpm -Uvh NetApp-MetroCluster-Tiebreaker-Software-
1.5-1.x86_64.rpm
```

成功升级后，系统将显示以下输出：

```

Verifying...
##### [100%]
Preparing...
##### [100%]
Updating / installing...
  1:NetApp-MetroCluster-Tiebreaker-
So##### [ 50%]

Enter the absolute path for Java : /usr/lib/jvm/java-19-openjdk-
19.0.0.0.36-2.rolling.el8.x86_64/bin/java
Verifying if Java exists...
Found Java. Proceeding with the installation.
Enter host user account to use for the installation:
mcctbuser1
User account mcctbuser1 found. Proceeding with the installation
Sealed          false
Do you wish to auto unseal vault(y/n)?y
Enter the key1:
Enter the key2:
Enter the key3:
Success! Uploaded policy: mcctb-policy
Error enabling approle auth: Error making API request.
URL: POST http://127.0.0.1:8200/v1/sys/auth/approle
Code: 400. Errors:
* path is already in use at approle/
Success! Enabled the kv secrets engine at: mcctb/
Success! Data written to: auth/approle/role/mcctb-app
Enter database user name : root
Please enter database password for root
Enter password:
Password updated successfully in the database.
Password updated successfully in the vault.
Synchronizing state of netapp-metrocluster-tiebreaker-
software.service with SysV service script with
/usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable netapp-
metrocluster-tiebreaker-software
Attempting to start NetApp MetroCluster Tiebreaker software
services
Started NetApp MetroCluster Tiebreaker software services
Successfully upgraded NetApp MetroCluster Tiebreaker software to
version 1.5.
Cleaning up / removing...
  2:NetApp-MetroCluster-Tiebreaker-
So##### [100%]

```



如果输入的 MySQL root 密码不正确，Tiebreaker 软件会指示已成功安装该密码，但会显示 Access Denied 消息。要解决问题描述问题，您必须使用 `rpm -e` 命令卸载 Tiebreaker 软件，然后使用正确的 MySQL root 密码重新安装该软件。

6. 通过打开从Tiebreaker主机到每个节点管理LIF和集群管理LIF的SSH连接、检查Tiebreaker与MetroCluster软件的连接。

相关信息

["NetApp 支持"](#)

## 版权信息

版权所有 © 2024 NetApp, Inc.。保留所有权利。中国印刷。未经版权所有者事先书面许可，本档中受版权保护的任何部分不得以任何形式或通过任何手段（图片、电子或机械方式，包括影印、录音、录像或存储在电子检索系统中）进行复制。

从受版权保护的 NetApp 资料派生的软件受以下许可和免责声明的约束：

本软件由 NetApp 按“原样”提供，不含任何明示或暗示担保，包括但不限于适销性以及针对特定用途的适用性的隐含担保，特此声明不承担任何责任。在任何情况下，对于因使用本软件而以任何方式造成的任何直接性、间接性、偶然性、特殊性、惩罚性或后果性损失（包括但不限于购买替代商品或服务；使用、数据或利润方面的损失；或者业务中断），无论原因如何以及基于何种责任理论，无论出于合同、严格责任或侵权行为（包括疏忽或其他行为），NetApp 均不承担责任，即使已被告知存在上述损失的可能性。

NetApp 保留在不另行通知的情况下随时对本文档所述的任何产品进行更改的权利。除非 NetApp 以书面形式明确同意，否则 NetApp 不承担因使用本文档所述产品而产生的任何责任或义务。使用或购买本产品不表示获得 NetApp 的任何专利权、商标权或任何其他知识产权许可。

本手册中描述的产品可能受一项或多项美国专利、外国专利或正在申请的专利的保护。

有限权利说明：政府使用、复制或公开本文档受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中“技术数据权利 — 非商用”条款第 (b)(3) 条规定的限制条件的约束。

本文档中所含数据与商业产品和/或商业服务（定义见 FAR 2.101）相关，属于 NetApp, Inc. 的专有信息。根据本协议提供的所有 NetApp 技术数据和计算机软件具有商业性质，并完全由私人出资开发。美国政府对这些数据的使用权具有非排他性、全球性、受限且不可撤销的许可，该许可既不可转让，也不可再许可，但仅限在与交付数据所依据的美国政府合同有关且受合同支持的情况下使用。除本文档规定的情形外，未经 NetApp, Inc. 事先书面批准，不得使用、披露、复制、修改、操作或显示这些数据。美国政府对国防部的授权仅限于 DFARS 的第 252.227-7015(b)（2014 年 2 月）条款中明确的权利。

## 商标信息

NetApp、NetApp 标识和 <http://www.netapp.com/TM> 上所列的商标是 NetApp, Inc. 的商标。其他公司和产品名称可能是其各自所有者的商标。