

Lesson 8 练习: TiDB Cluster 部署

Lesson 8 练习：概述

概述

在本课练习中，您将会用到 TiUP 工具来部署 TiDB 数据库集群，并且连接到 TiDB 数据库执行 SQL 语句，之后您会练习到关闭和启动 TiDB 数据库集群。

实验环境要求

1. 注意，您的实验环境（包括 IP，端口号，用户名，密码和目录）可能和手册不同。
2. 可以连接到实验用虚拟机。

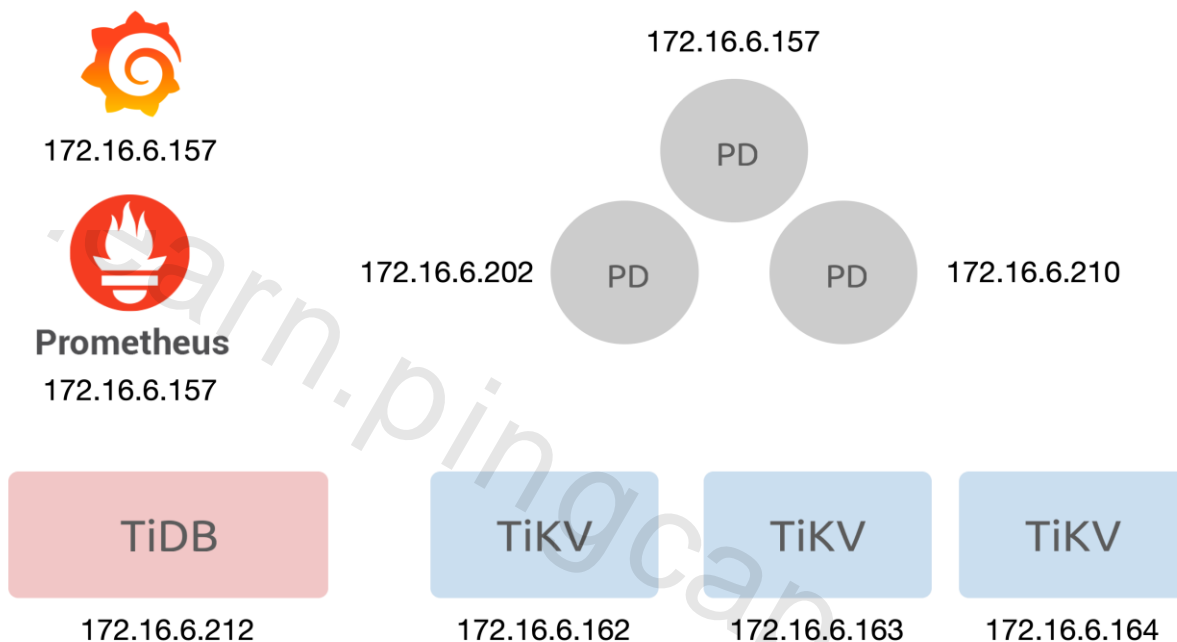
learn.pingcap.cc

Lesson 8 练习-1: 使用 TiUP 来部署 TiDB

概述

在这个练习中, 您将会用到 **TiUP** 工具来部署 **TiDB** 数据库集群, 并且连接到 **TiDB** 执行 **SQL** 语句。

拓扑结构介绍



Lesson 8 练习-1: 使用 TiUP 来部署 TiDB

任务

1. 下载安装 TiUP 工具:

```
[root@centos76_vm ~]# curl --proto 'https' --tlsv1.2 -sSf
https://tiup-mirrors.pingcap.com/install.sh | sh
```

2. 重新声明全局环境变量:

```
[root@centos76_vm ~]# source /root/.bash_profile
```

3. 安装 TiUP cluster 组件:

```
[root@centos76_vm ~]# tiup cluster

Starting component `cluster`: /root/.tiup/components/cluster/v1.4.3/tiup-cluster
Deploy a TiDB cluster for production

Usage:
tiup cluster [command]

Available Commands:
check    Perform preflight checks for the cluster.
deploy   Deploy a cluster for production
start    Start a TiDB cluster
... (省略中间内容)
--wait-timeout uint Timeout in seconds to wait for an operation to complete, ignored for operations that don't fit. (default 120)
-y, --yes           Skip all confirmations and assumes 'yes'

Use "tiup cluster help [command]" for more information about a command
```

4. 更新 TiUP cluster 组件至最新版本:

```
[root@centos76_vm ~]# tiup update --self && tiup update cluster

download https://tiup-mirrors.pingcap.com/tiup-v1.4.3-linux-amd64.tar.gz 1023.02 KiB / 6.73 MiB 14.84% 1.22 Mdownload
https://tiup-mirrors.pingcap.com/tiup-v1.4.3-linux-amd64.tar.gz 1.32 MiB / 6.73 MiB 19.59% 1.22 MiB download
https://tiup-mirrors.pingcap.com/tiup-v1.4.3-linux-amd64.tar.gz 1.65 MiB / 6.73 MiB 24.47% 1.22 MiB download
https://tiup-mirrors.pingcap.com/tiup-v1.4.3-linux-amd64.tar.gz 1.93 MiB / 6.73 MiB 28.64% 1.24 MiB download
https://tiup-mirrors.pingcap.com/tiup-v1.4.3-linux-amd64.tar.gz 2.14 MiB / 6.73 MiB 31.77% 1.24 MiB download
... (省略中间内容)
https://tiup-mirrors.pingcap.com/tiup-v1.4.3-linux-amd64.tar.gz 4.81 MiB / 6.73 MiB 71.45% 1.25 MiB download
https://tiup-mirrors.pingcap.com/tiup-v1.4.3-linux-amd64.tar.gz 5.12 MiB / 6.73 MiB 76.09% 1.27 MiB download
https://tiup-mirrors.pingcap.com/tiup-v1.4.3-linux-amd64.tar.gz 5.39 MiB / 6.73 MiB 80.04% 1.27 MiB download
https://tiup-mirrors.pingcap.com/tiup-v1.4.3-linux-amd64.tar.gz 5.70 MiB / 6.73 MiB 84.68% 1.27 MiB download
https://tiup-mirrors.pingcap.com/tiup-v1.4.3-linux-amd64.tar.gz 5.89 MiB / 6.73 MiB 87.46% 1.27 MiB download
https://tiup-mirrors.pingcap.com/tiup-v1.4.3-linux-amd64.tar.gz 6.12 MiB / 6.73 MiB 90.94% 1.27 MiB download
https://tiup-mirrors.pingcap.com/tiup-v1.4.3-linux-amd64.tar.gz 6.45 MiB / 6.73 MiB 95.82% 1.27 MiB download
https://tiup-mirrors.pingcap.com/tiup-v1.4.3-linux-amd64.tar.gz 6.73 MiB / 6.73 MiB 100.00% 1.41 MiB p/s
Updated successfully!
component cluster version v1.4.3 is already installed
Updated successfully!
```

Lesson 8 练习-1: 使用 TiUP 来部署 TiDB

5. 验证当前 TiUP cluster 版本信息。查看 TiUP cluster 组件版本:

```
[root@centos76_vm ~]# tiup --binary cluster  
  
/root/.tiup/components/cluster/v1.4.3/tiup-cluster
```

6. 根据不同的集群拓扑, 编辑 TiUP 所需的集群初始化配置文件:

```
[root@centos76_vm ~]# tiup cluster template > topology.yaml  
  
Starting component `cluster`: /root/.tiup/components/cluster/v1.4.3/tiup-cluster template
```

7. 显示拓扑文件并编辑拓扑文件:

```
[root@centos76_vm ~]# ls  
  
anaconda-ks.cfg topology.yaml
```

接下来, 我们打开拓扑文件并编辑, 如下:

```
[root@centos76_vm ~]# vi topology.yaml  
  
## Global variables are applied to all deployments and used as the default value of  
## the deployments if a specific deployment value is missing.  
global:  
  ## The user who runs the tidb cluster.  
  
... (省略中间内容)  
  
  ## Alertmanager log file storage directory.  
  # log_dir: "/tidb-deploy/alertmanager-9093/log"
```

修改内容如下:

(1) 根据实验环境设置 PD 节点 IP 地址, 如下:

```
pd_servers:  
  ## The ip address of the PD Server.  
  - host: 172.16.6.202  
  ...  
  - host: 172.16.6.157  
  ...  
  - host: 172.16.6.210
```

(2) 根据实验环境设置 TiKV 节点 IP 地址, 如下:

```
tikv_servers:  
  ## The ip address of the TiKV Server.  
  - host: 172.16.6.162  
  ...
```

Lesson 8 练习-1: 使用 TiUP 来部署 TiDB

- host: 172.16.6.163
 - ...
 - host: 172.16.6.164
- (3) 根据实验环境设置 TiDB 节点 IP 地址, 如下:
- ```
tidb_servers:
The ip address of the TiDB Server.
- host: 172.16.6.212
```
- (4) 在此练习中关闭 TiFlash 节点, 如下:
- ```
tiflash_servers:
# # The ip address of the TiFlash Server.
# - host: 10.0.1.20
```
- (5) 根据实验环境设置 monitoring 节点 IP 地址, 如下:
- ```
monitoring_servers:
The ip address of the Monitoring Server.
- host: 172.16.6.157
```
- (5) 根据实验环境设置 Grafana 节点 IP 地址, 如下:
- ```
grafana_servers:
# # The ip address of the Grafana Server.
- host: 172.16.6.157
```
- (5) 根据实验环境设置 alertmanager 节点 IP 地址, 如下:
- ```
Server configs are used to specify the configuration of Alertmanager Servers.
alertmanager_servers:
The ip address of the Alertmanager Server.
- host: 172.16.6.157
```

## Lesson 8 练习-1: 使用 TiUP 来部署 TiDB

### 8. 检查和自动修复集群存在的潜在风险:

```
[root@centos76_vm ~]# tiup cluster check ./topology.yaml --apply --user root -p

Starting component `cluster`: /root/.tiup/components/cluster/v1.4.3/tiup-cluster check ./topology.yaml --apply --user root -p
Input SSH password:
+ Download necessary tools
 - Downloading check tools for linux/amd64 ... Done
+ Collect basic system information
 - Getting system info of 172.16.6.202:22 ... :: CopyComponent: component=insight, version=, remote=172.16.6.202:/tmp/tiup
os=lin...
+ Collect basic system information
 - Getting system info of 172.16.6.202:22 ... :: CopyComponent: component=insight, version=, remote=172.16.6.202:/tmp/tiup
os=lin...
 - Getting system info of 172.16.6.157:22 ... :: Shell: host=172.16.6.157, sudo=false, command='/tmp/tiup/bin/insight`
+ Collect basic system information
+ Collect basic system information
+ Collect basic system information
 - Getting system info of 172.16.6.202:22 ... Done
 - Getting system info of 172.16.6.157:22 ... Done
 - Getting system info of 172.16.6.210:22 ... Done
... (省略中间内容)
+ Try to apply changes to fix failed checks
 - Applying changes on 172.16.6.202 ... Done
 - Applying changes on 172.16.6.157 ... Done
 - Applying changes on 172.16.6.210 ... Done
 - Applying changes on 172.16.6.162 ... Done
 - Applying changes on 172.16.6.163 ... Done
 - Applying changes on 172.16.6.164 ... Done
 - Applying changes on 172.16.6.212 ... Done
```

## Lesson 8 练习-1: 使用 TiUP 来部署 TiDB

### 9. 部署 TiDB 集群:

```
[root@centos76_vm ~]# tiup cluster deploy tidb-test v5.0.0 ./topology.yaml --user root -p
```

Starting component `cluster`: /root/.tiup/components/cluster/v1.4.3/tiup-cluster deploy tidb-test v5.0.0 ./topology.yaml --user root -p

Please confirm your topology:

Cluster type: tidb

Cluster name: tidb-test

---

Cluster version: v5.0.0

| Role         | Host         | Ports       | OS/Arch      | Directories                                                 |
|--------------|--------------|-------------|--------------|-------------------------------------------------------------|
| pd           | 172.16.6.202 | 2379/2380   | linux/x86_64 | /tidb-deploy/pd-2379,/tidb-data/pd-2379                     |
| pd           | 172.16.6.157 | 2379/2380   | linux/x86_64 | /tidb-deploy/pd-2379,/tidb-data/pd-2379                     |
| pd           | 172.16.6.210 | 2379/2380   | linux/x86_64 | /tidb-deploy/pd-2379,/tidb-data/pd-2379                     |
| tikv         | 172.16.6.162 | 20160/20180 | linux/x86_64 | /tidb-deploy/tikv-20160,/tidb-data/tikv-20160               |
| tikv         | 172.16.6.163 | 20160/20180 | linux/x86_64 | /tidb-deploy/tikv-20160,/tidb-data/tikv-20160               |
| tikv         | 172.16.6.164 | 20160/20180 | linux/x86_64 | /tidb-deploy/tikv-20160,/tidb-data/tikv-20160               |
| tidb         | 172.16.6.212 | 4000/10080  | linux/x86_64 | /tidb-deploy/tidb-4000                                      |
| prometheus   | 172.16.6.157 | 9090        | linux/x86_64 | /tidb-deploy/prometheus-9090,/tidb-data/prometheus-9090     |
| grafana      | 172.16.6.157 | 3000        | linux/x86_64 | /tidb-deploy/grafana-3000                                   |
| alertmanager | 172.16.6.157 | 9093/9094   | linux/x86_64 | /tidb-deploy/alertmanager-9093,/tidb-data/alertmanager-9093 |

Attention:

1. If the topology is not what you expected, check your yaml file.
2. Please confirm there is no port/directory conflicts in same host.

Do you want to continue? [y/N]: (default=N) y

Input SSH password:

+ Generate SSH keys ... Done

+ Download TiDB components

... (省略中间内容)

Enabling component node\_exporter

Enabling component blackbox\_exporter

Cluster `tidb-test` deployed successfully, you can start it with command: `tiup cluster start tidb-test`

### 10. 查看 TiUP 管理的集群情况:

```
[root@centos76_vm ~]# tiup cluster list
```

Starting component `cluster`: /root/.tiup/components/cluster/v1.4.3/tiup-cluster list

| Name      | User | Version | Path                                           | PrivateKey                                                |
|-----------|------|---------|------------------------------------------------|-----------------------------------------------------------|
| tidb-test | tidb | v5.0.0  | /root/.tiup/storage/cluster/clusters/tidb-test | /root/.tiup/storage/cluster/clusters/tidb-test/ssh/id_rsa |



## Lesson 8 练习-1: 使用 TiUP 来部署 TiDB

### 11. 检查 tidb-test 集群情况:

```
[root@centos76_vm ~]# tiup cluster display tidb-test

Starting component `cluster`: /root/.tiup/components/cluster/v1.4.3/tiup-cluster display tidb-test
Cluster type: tidb
Cluster name: tidb-test
Cluster version: v5.0.0
SSH type: builtin
ID Role Host Ports OS/Arch Status Data Dir Deploy Dir

172.16.6.157:9093 alertmanager 172.16.6.157 9093/9094 linux/x86_64 inactive /tidb-data/alertmanager-9093 /tidb-deploy/alertmanager-9093
172.16.6.157:3000 grafana 172.16.6.157 3000 linux/x86_64 inactive - /tidb-deploy/grafana-3000
172.16.6.157:2379 pd 172.16.6.157 2379/2380 linux/x86_64 Down /tidb-data/pd-2379 /tidb-deploy/pd-2379
172.16.6.202:2379 pd 172.16.6.202 2379/2380 linux/x86_64 Down /tidb-data/pd-2379 /tidb-deploy/pd-2379
172.16.6.210:2379 pd 172.16.6.210 2379/2380 linux/x86_64 Down /tidb-data/pd-2379 /tidb-deploy/pd-2379
172.16.6.157:9090 prometheus 172.16.6.157 9090 linux/x86_64 inactive /tidb-data/prometheus-9090 /tidb-deploy/prometheus-9090
172.16.6.212:4000 tidb 172.16.6.212 4000/10080 linux/x86_64 Down - /tidb-deploy/tidb-4000
172.16.6.162:20160 tikv 172.16.6.162 20160/20180 linux/x86_64 N/A /tidb-data/tikv-20160 /tidb-deploy/tikv-20160
172.16.6.163:20160 tikv 172.16.6.163 20160/20180 linux/x86_64 N/A /tidb-data/tikv-20160 /tidb-deploy/tikv-20160
172.16.6.164:20160 tikv 172.16.6.164 20160/20180 linux/x86_64 N/A /tidb-data/tikv-20160 /tidb-deploy/tikv-20160
Total nodes: 10
```

### 12. 启动集群:

```
[root@centos76_vm ~]# tiup cluster start tidb-test

Starting component `cluster`: /root/.tiup/components/cluster/v1.4.3/tiup-cluster start tidb-test
Starting cluster tidb-test...
+ [Serial] - SSHKeySet: privateKey=/root/.tiup/storage/cluster/clusters/tidb-test/ssh/id_rsa,
publicKey=/root/.tiup/storage/cluster/clusters/tidb-test/ssh/id_rsa.pub
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.157
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.164
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.212
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.157
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.157
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.163
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.210
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.157
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.162
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.202
+ [Serial] - StartCluster
Starting component pd
 Starting instance pd 172.16.6.202:2379
 Starting instance pd 172.16.6.157:2379
 Starting instance pd 172.16.6.210:2379
... (省略中间内容)
+ [Serial] - UpdateTopology: cluster=tidb-test
Started cluster `tidb-test` successfully
```

## Lesson 8 练习-1: 使用 TiUP 来部署 TiDB

---

### 13. 连接到 TiDB: (请根据自己实验环境输入密码)

```
[root@centos76_vm ~]# mysql -h 172.16.6.212 -P 4000 -uroot
```

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 5

Server version: 5.7.25-TiDB-v5.0.0 TiDB Server (Apache License 2.0) Community Edition, MySQL 5.7 compatible

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

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> show databases;
```

| Database           |
|--------------------|
| INFORMATION_SCHEMA |
| METRICS_SCHEMA     |
| PERFORMANCE_SCHEMA |
| mysql              |
| test               |

5 rows in set (0.00 sec)

## Lesson 8 练习-2: 关闭和启动 TiDB 集群

### 概述

您会练习到关闭和启动 TiDB 数据库集群。

### 任务

#### 1. 关闭当前 TiDB 数据库集群，如下：

```
[root@centos76_vm ~]# tiup cluster stop tidb-test
Starting component `cluster`: /root/.tiup/components/cluster/v1.4.4/tiup-cluster stop tidb-test
+ [Serial] - SSHKeySet: privateKey=/root/.tiup/storage/cluster/clusters/tidb-test/ssh/id_rsa,
publicKey=/root/.tiup/storage/cluster/clusters/tidb-test/ssh/id_rsa.pub
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.202
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.163
... (省略中间内容)
Stopping component blackbox_exporter
Stopping component node_exporter
Stopping component blackbox_exporter
Stopped cluster `tidb-test` successfully
```

#### 2. 查看当前 TiDB 数据库集群状态，如下：

```
[root@centos76_vm ~]# tiup cluster display tidb-test
Starting component `cluster`: /root/.tiup/components/cluster/v1.4.4/tiup-cluster display tidb-test
Cluster type: tidb
Cluster name: tidb-test
Cluster version: v5.0.0
SSH type: builtin
```

| ID                 | Role         | Host         | Ports       | OS/Arch      | Status   | Data Dir                     | Deploy Dir                     |
|--------------------|--------------|--------------|-------------|--------------|----------|------------------------------|--------------------------------|
| 172.16.6.157:9093  | alertmanager | 172.16.6.157 | 9093/9094   | linux/x86_64 | inactive | /tidb-data/alertmanager-9093 | /tidb-deploy/alertmanager-9093 |
| 172.16.6.157:3000  | grafana      | 172.16.6.157 | 3000        | linux/x86_64 | inactive | -                            | /tidb-deploy/grafana-3000      |
| 172.16.6.157:2379  | pd           | 172.16.6.157 | 2379/2380   | linux/x86_64 | Down     | /tidb-data/pd-2379           | /tidb-deploy/pd-2379           |
| 172.16.6.202:2379  | pd           | 172.16.6.202 | 2379/2380   | linux/x86_64 | Down     | /tidb-data/pd-2379           | /tidb-deploy/pd-2379           |
| 172.16.6.210:2379  | pd           | 172.16.6.210 | 2379/2380   | linux/x86_64 | Down     | /tidb-data/pd-2379           | /tidb-deploy/pd-2379           |
| 172.16.6.157:9090  | prometheus   | 172.16.6.157 | 9090        | linux/x86_64 | inactive | /tidb-data/prometheus-9090   | /tidb-deploy/prometheus-9090   |
| 172.16.6.212:4000  | tidb         | 172.16.6.212 | 4000/10080  | linux/x86_64 | Down     | -                            | /tidb-deploy/tidb-4000         |
| 172.16.6.162:20160 | tikv         | 172.16.6.162 | 20160/20180 | linux/x86_64 | N/A      | /tidb-data/tikv-20160        | /tidb-deploy/tikv-20160        |
| 172.16.6.163:20160 | tikv         | 172.16.6.163 | 20160/20180 | linux/x86_64 | N/A      | /tidb-data/tikv-20160        | /tidb-deploy/tikv-20160        |
| 172.16.6.164:20160 | tikv         | 172.16.6.164 | 20160/20180 | linux/x86_64 | N/A      | /tidb-data/tikv-20160        | /tidb-deploy/tikv-20160        |

Total nodes: 10

#### 3. 启动当前 TiDB 数据库集群，如下：

```
[root@centos76_vm ~]# tiup cluster start tidb-test
Starting component `cluster`: /root/.tiup/components/cluster/v1.4.4/tiup-cluster start tidb-test
Starting cluster tidb-test...
... (省略中间内容)
+ [Serial] - UpdateTopology: cluster=tidb-test
Started cluster `tidb-test` successfully
```

## Lesson 8 练习-2: 关闭和启动 TiDB 集群

### 4. 查看当前 TiDB 数据库集群状态，如下：

```
[root@centos76_vm ~]# tiup cluster display tidb-test
Starting component `cluster`: /root/.tiup/components/cluster/v1.4.4/tiup-cluster display tidb-test
Cluster type: tidb
Cluster name: tidb-test
Cluster version: v5.0.0
SSH type: builtin
Dashboard URL: http://172.16.6.157:2379/dashboard
ID Role Host Ports OS/Arch Status Data Dir Deploy Dir
--
172.16.6.157:9093 alertmanager 172.16.6.157 9093/9094 linux/x86_64 Up /tidb-data/alertmanager-9093 /tidb-deploy/alertmanager-9093
172.16.6.157:3000 grafana 172.16.6.157 3000 linux/x86_64 Up - /tidb-deploy/grafana-3000
172.16.6.157:2379 pd 172.16.6.157 2379/2380 linux/x86_64 Up|UI /tidb-data/pd-2379 /tidb-deploy/pd-2379
172.16.6.202:2379 pd 172.16.6.202 2379/2380 linux/x86_64 Up|L /tidb-data/pd-2379 /tidb-deploy/pd-2379
172.16.6.210:2379 pd 172.16.6.210 2379/2380 linux/x86_64 Up /tidb-data/pd-2379 /tidb-deploy/pd-2379
172.16.6.157:9090 prometheus 172.16.6.157 9090 linux/x86_64 Up /tidb-data/prometheus-9090 /tidb-deploy/prometheus-9090
172.16.6.212:4000 tidb 172.16.6.212 4000/10080 linux/x86_64 Up - /tidb-deploy/tidb-4000
172.16.6.162:20160 tikv 172.16.6.162 20160/20180 linux/x86_64 Up /tidb-data/tikv-20160 /tidb-deploy/tikv-20160
172.16.6.163:20160 tikv 172.16.6.163 20160/20180 linux/x86_64 Up /tidb-data/tikv-20160 /tidb-deploy/tikv-20160
172.16.6.164:20160 tikv 172.16.6.164 20160/20180 linux/x86_64 Up /tidb-data/tikv-20160 /tidb-deploy/tikv-20160
Total nodes: 10
```

### 5. 退出所有窗口。

## Lesson 9 练习: TiDB 的连接管理

## Lesson 9 练习：概述

---

### 概述

在本课练习中，您将尝试使用 MySQL 的各种客户端连接之前部署好的 TiDB 数据库，并执行 SQL 语句，最后监控 MySQL 数据库的连接状态。

### 实验环境要求

1. 注意，您的实验环境（包括 IP，端口号，用户名，密码和目录）可能和手册不同。
2. 可以连接到实验用虚拟机。
3. TiDB 数据库之前部署完毕。

learn.pingcap.cc

## Lesson 9 练习-1: 使用命令行工具连接 TiDB 数据库

---

### 概述

在这个练习中，您将会使用 MySQL 客户端连接 TiDB 数据库

### 任务

1. 下载安装 MySQL 客户端，请参考 MySQL 官方网站：<https://dev.mysql.com>。
2. 使用 MySQL 客户端连接 TiDB 数据库：（请根据自己实验环境输入密码）

```
[root@centos76_vm ~]# mysql -h172.16.6.212 -P 4000 -uroot -p
```

```
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 19
Server version: 5.7.25-TiDB-v5.0.0 TiDB Server (Apache License 2.0) Community Edition, MySQL 5.7 compatible

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

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
```

3. 查看数据库版本：

```
mysql>select tidb_version()\G

***** 1. row *****
tidb_version(): Release Version: v5.0.0
Edition: Community
Git Commit Hash: bdac0885cd11bdf571aad9353bfc24e13554b91c
Git Branch: heads/refs/tags/v5.0.0
UTC Build Time: 2021-04-06 16:36:29
GoVersion: go1.13
Race Enabled: false
TiKV Min Version: v3.0.0-60965b006877ca7234adaced7890d7b029ed1306
Check Table Before Drop: false
1 row in set (0.00 sec)
```

## Lesson 9 练习-1: 使用 命令行工具连接 TiDB 数据库

### 4. 查看默认数据库:

```
mysql>show databases;
```

| Database           |
|--------------------|
| INFORMATION_SCHEMA |
| METRICS_SCHEMA     |
| PERFORMANCE_SCHEMA |
| mysql              |
| test               |

5 rows in set (0.00 sec)

### 5. 创建数据库 pingcap:

```
mysql>create database tidb;
```

Query OK, 0 rows affected (0.11 sec)

### 查看新创建数据库:

```
mysql>show databases;
```

| Database           |
|--------------------|
| INFORMATION_SCHEMA |
| METRICS_SCHEMA     |
| PERFORMANCE_SCHEMA |
| mysql              |
| test               |
| tidb               |

6 rows in set (0.00 sec)

### 6. 进入 tidb 数据库:

```
mysql>use tidb
```

Database changed

### 7. 执行建表语句, 创建表 tab\_tidb:

```
mysql>CREATE TABLE `tab_tidb` (
 `id` int(11) NOT NULL AUTO_INCREMENT,
 `name` varchar(20) NOT NULL DEFAULT '',
 `age` int(11) NOT NULL DEFAULT 0,
 `version` varchar(20) NOT NULL DEFAULT '',
 PRIMARY KEY (`id`),
 KEY `idx_age` (`age`));
```

Query OK, 0 rows affected (0.10 sec)



## Lesson 9 练习-1: 使用 命令行工具连接 TiDB 数据库

---

### 8. 向表中插入数据:

```
mysql>insert into `tab_tidb` values (1,'TiDB',5,'TiDB-v5.0.0');
```

Query OK, 1 row affected (0.01 sec)

### 9. 查看数据:

```
mysql>select * from tab_tidb;
```

| id | name | age | version     |
|----|------|-----|-------------|
| 1  | TiDB | 5   | TiDB-v5.0.0 |

1 row in set (0.00 sec)

### 10. 退出客户端:

```
mysql>Exit
```

Bye

## Lesson 9 练习-2: 使用 GUI 工具连接 TiDB 数据库

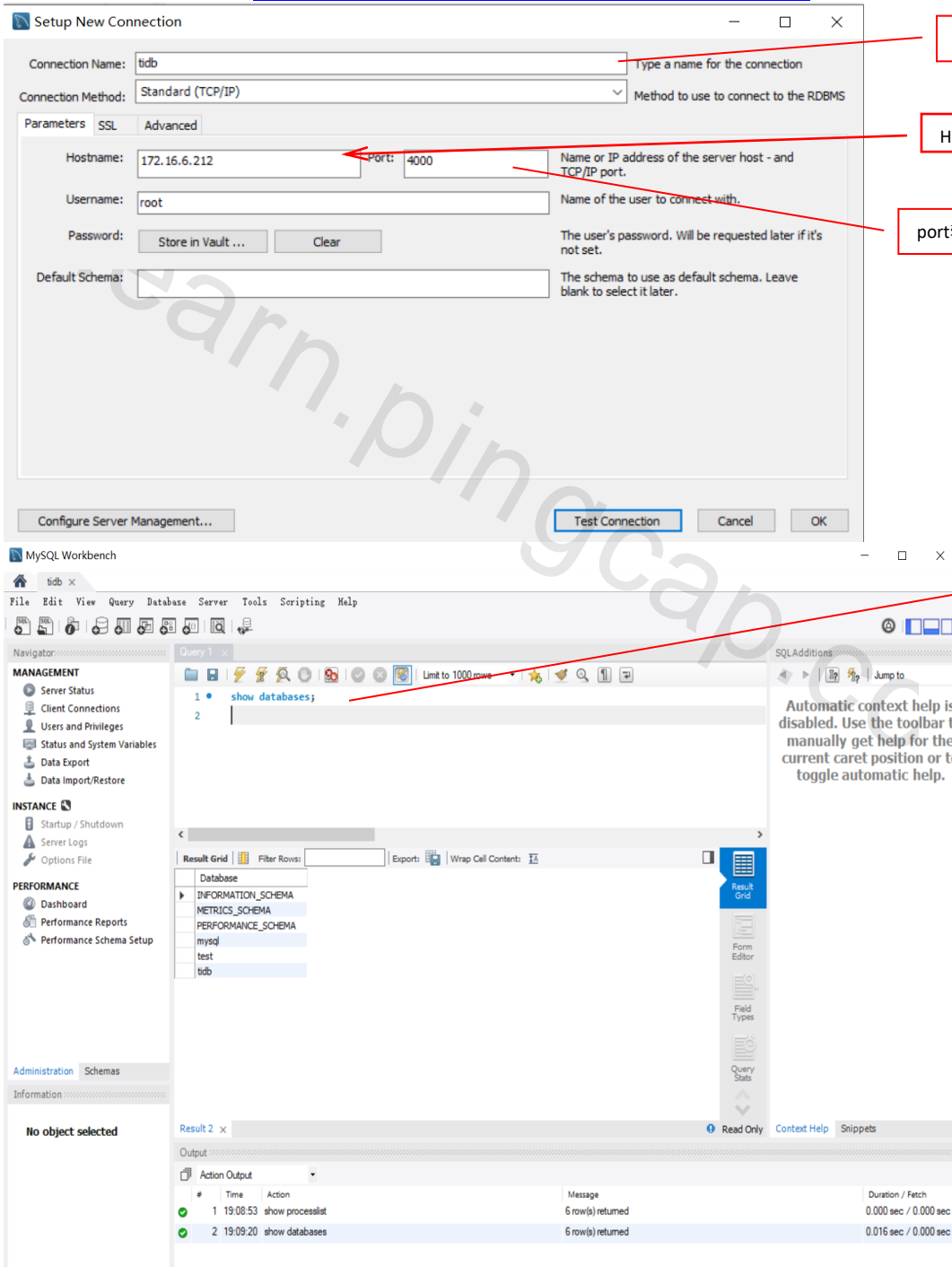
### 概述

在这个练习中，您将会使用 **MySQL Workbench** 客户端连接 **TiDB** 数据库。

### 任务

#### 1. 使用MySQL连接到TiDB:

下载mysql (参考 <https://dev.mysql.com/downloads/workbench/> )



The image shows two screenshots of MySQL Workbench. The top screenshot is the 'Setup New Connection' dialog box. The bottom screenshot is the main MySQL Workbench interface.

**Setup New Connection Dialog:**

- Connection Name: tidb (labeled: Connection Name 输入tidb)
- Connection Method: Standard (TCP/IP)
- Parameters tab selected.
- Hostname: 172.16.6.212 (labeled: Hostname输入172.16.6.212)
- Port: 4000 (labeled: port输入4000)
- Username: root
- Password: (empty, with 'Store in Vault...' and 'Clear' buttons)
- Default Schema: (empty)
- Buttons: Configure Server Management..., Test Connection, Cancel, OK

**MySQL Workbench Main Interface:**

- Navigator pane on the left shows the 'tidb' connection selected.
- Query Editor shows the query: `show databases;`
- Result Grid shows the output of the query.
- Output pane at the bottom shows the execution log.
- Annotations: A red arrow points to the query editor with the text '进入MySQL后输入show databases查看数据库'.

| # | Time     | Action           | Message           | Duration / Fetch      |
|---|----------|------------------|-------------------|-----------------------|
| 1 | 19:08:53 | show processlist | 6 row(s) returned | 0.000 sec / 0.000 sec |
| 2 | 19:09:20 | show databases   | 6 row(s) returned | 0.016 sec / 0.000 sec |

## Lesson 9 练习-3:监控 TiDB 数据库的连接

### 概述

在这个练习中，您将会使用 **MySQL** 客户端连接 **TiDB** 数据库，并且监控 **TiDB** 数据库的已有连接状态。

### 任务

#### 1. 查询当前数据库的连接状态:

```
mysql>show processlist;
```

| Id | User | Host               | db   | Command | Time | State      | Info             |
|----|------|--------------------|------|---------|------|------------|------------------|
| 29 | root | 172.16.6.212:46316 | NULL | Query   | 0    | autocommit | show processlist |

1 row in set (0.00 sec)

（对于结果的解释如下：

**Id:** 连接的 ID，每个连接不同。

**User:** 连接的用户名。

**Host :** 连接的客户端主机名。

**db :** 连接用户所在的数据库，**NULL** 代表没有在任何 **database** 中。

**Command:** 连接用户当前的命令动作，**Query** 代表正在执行，**Sleep** 代表没有任何操作。

**Time:** 连接时长，单位为秒。

**State:** 连接的提交状态。

**Info:** 命令信息，一般为正在执行的命令，**NULL** 代表没有执行任何命令。）

## Lesson 10 练习: TiDB 的配置

## Lesson 10 练习：概述

---

### 概述

在本课练习中，您将会修改 TiDB 数据库的系统参数（不同作用域）和集群配置。

### 实验环境要求

1. 注意，您的实验环境（包括 IP，端口号，用户名，密码和目录）可能和手册不同。
2. 可以连接到实验用虚拟机。
3. TiDB 数据库之前部署完毕。

learn.pingcap.cc

## Lesson 10 练习-1: 在不同作用域修改 TiDB 数据库的系统参数

### 概述

在这个练习中，您将在不同作用域下对于 TiDB 数据库的系统参数进行修改。

### 任务

6. 首先连接到 TiDB ,用户为 root: 进入 test 数据库: (请根据自己实验环境输入密码)

```
[root@centos76_vm ~]# mysql -h172.16.6.212 -P 4000 -uroot -p
```

Enter password:

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 29

Server version: 5.7.25-TiDB-v5.0.0 TiDB Server (Apache License 2.0) Community Edition, MySQL 5.7 compatible

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

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.

进入 test 数据库:

```
mysql> use test;
```

Reading table information for completion of table and column names

You can turn off this feature to get a quicker startup with -A

Database changed

2. 创建测试表 t1 (t1 表只有一列 a, 并且是自增主键列):

```
mysql> CREATE TABLE t1 (a int not null primary key auto_increment);
```

Query OK, 0 rows affected (0.52 sec)

3. 查看系统参数 auto\_increment\_increment (auto\_increment\_increment 默认值为 1.):

```
mysql> show session variables like 'auto_increment_increment';
```

| Variable_name            | Value |
|--------------------------|-------|
| auto_increment_increment | 1     |

1 row in set (0.56 sec)

## Lesson 10 练习-1: 在不同作用域修改 TiDB 数据库的系统参数

### 4. 插入 2 行测试数据（增量变为 1）：

```
mysql> insert into t1 values ();
```

Query OK, 1 row affected (0.02 sec)

```
mysql> insert into t1 values ();
```

Query OK, 1 row affected (0.00 sec)

```
mysql> select * from t1;
```

|   |
|---|
| a |
| 1 |
| 2 |

2 rows in set (0.01 sec)

### 5. 在会话级别修改参数auto\_increment\_increment为10:

```
mysql> set auto_increment_increment = 10;
```

Query OK, 0 rows affected (0.00 sec)

### 6. 继续插入 2 行测试数据(增量变为10):

```
mysql> insert into t1 values ();
```

Query OK, 1 row affected (0.00 sec)

```
mysql> insert into t1 values ();
```

Query OK, 1 row affected (0.01 sec)

```
mysql> select * from t1;
```

|    |
|----|
| a  |
| 1  |
| 2  |
| 11 |
| 21 |

4 rows in set (0.01 sec)

## Lesson 10 练习-1: 在不同作用域修改 TiDB 数据库的系统参数

7. 另外启动一个终端窗口 (称为终端2), 连接 TiDB数据库: (请根据自己实验环境输入密码)

```
[root@centos76_vm ~]# mysql -h 172.16.6.212 -P4000 -uroot -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 55
Server version: 5.7.25-TiDB-v5.0.0 TiDB Server (Apache License 2.0) Community Edition, MySQL 5.7 compatible

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

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.
```

进入test库:

```
mysql> use test;

Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
```

8. 在终端2查看参数auto\_increment\_increment (默认值依然为1):

```
mysql> show variables like 'auto_increment_increment';

+-----+-----+
| Variable_name | Value |
+-----+-----+
| auto_increment_increment | 1 |
+-----+-----+
1 row in set (0.59 sec)
```



## Lesson 10 练习-1: 在不同作用域修改 TiDB 数据库的系统参数

### 9. 在终端2插入 2 行测试数据(增量为1):

```
mysql> insert into t1 values ();
Query OK, 1 row affected (0.01 sec)

mysql> insert into t1 values ();
Query OK, 1 row affected (0.01 sec)

mysql> select * from t1;
+----+
| a |
+----+
| 1 |
| 2 |
| 11 |
| 21 |
| 22 |
| 23 |
+----+
6 rows in set (0.00 sec)
```

### 10. 将第一个会话退出:

```
mysql> exit

Bye
```

### 11. 再次连接会话, 并进入 test 数据库: (请根据自己实验环境输入密码)

```
[root@centos76_vm ~]# mysql -h172.16.6.212 -P 4000 -uroot -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 59
Server version: 5.7.25-TiDB-v5.0.0 TiDB Server (Apache License 2.0) Community Edition, MySQL 5.7 compatible

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

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> use test

Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
```

## Lesson 10 练习-1: 在不同作用域修改 TiDB 数据库的系统参数

12. 查询参数 `auto_increment_increment` (默认值变为 1):

```
mysql> show session variables like 'auto_increment_increment';
```

| Variable_name            | Value |
|--------------------------|-------|
| auto_increment_increment | 1     |

```
1 row in set (0.61 sec)
```

13. 继续插入 2 行测试数据 (增量为 1):

```
mysql> insert into t1 values ();
```

Query OK, 1 row affected (0.00 sec)

```
mysql> insert into t1 values ();
```

Query OK, 1 row affected (0.00 sec)

```
mysql> select * from t1;
```

|    |
|----|
| a  |
| 1  |
| 2  |
| 11 |
| 21 |
| 22 |
| 23 |
| 24 |
| 25 |

8 rows in set (0.01 sec)

(通过刚才的实验，我们发现在 `SESSION` 级别修改参数，只会影响到当前会话，对于其他会话或者本会话退出后，对于参数的修改就无效了。下面让我们看一看 `GLOBAL` 级别的参数，`GLOBAL` 级别的参数，对于当前会话无效，但是对于新的会话是起作用的。)

14. 在 `GLOBAL` 范围进行修改:

```
mysql> set global auto_increment_increment=10;
```

Query OK, 0 rows affected (0.01 sec)

## Lesson 10 练习-1: 在不同作用域修改 TiDB 数据库的系统参数

15. 在当前会话查询会话级别，查询参数 `auto_increment_increment`:

```
mysql> show session variables like 'auto_increment_increment';
```

| Variable_name            | Value |
|--------------------------|-------|
| auto_increment_increment | 1     |

```
1 row in set (0.00 sec)
```

(发现并没有改变。得出结论，参数 `auto_increment_increment` 在 GLOBAL 范围进行修改，并不会影响当前会话。)

16. 启动一个终端窗口(称为终端3)，连接 TiDB数据库，并进入test库：  
(请根据自己实验环境输入密码)

```
[root@centos76_vm ~]# mysql -h 172.16.6.212 -P4000 -uroot -p
```

Enter password:

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 107

Server version: 5.7.25-TiDB-v5.0.0 TiDB Server (Apache License 2.0) Community Edition, MySQL 5.7 compatible

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

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> use test;
```

Reading table information for completion of table and column names

You can turn off this feature to get a quicker startup with -A

Database changed

17. 在终端3查看参数 `auto_increment_increment` (默认值变为 10):

```
mysql> show variables like 'auto_increment_increment';
```

| Variable_name            | Value |
|--------------------------|-------|
| auto_increment_increment | 10    |

```
1 row in set (0.62 sec)
```

## Lesson 10 练习-1: 在不同作用域修改 TiDB 数据库的系统参数

18. 继续插入 2 行测试数据(增量为10):

```
mysql> insert into t1 values ();
Query OK, 1 row affected (0.01 sec)

mysql> insert into t1 values ();
Query OK, 1 row affected (0.01 sec)

mysql> select * from t1;
+----+
| a |
+----+
| 1 |
| 2 |
| 11 |
| 21 |
| 22 |
| 23 |
| 24 |
| 25 |
| 31 |
| 41 |
+----+
10 rows in set (0.00 sec)
```

(得出结论, 参数`auto_increment_increment`在 GLOBAL 范围进行修改, 会影响新连接会话。)

19. 回到终端1, 退出会话再次登录(发现变量`auto_increment_increment`为10.):  
(请根据自己实验环境输入密码)

```
mysql> exit
Bye

[root@centos76_vm ~]# mysql -h172.16.6.212 -P 4000 -uroot -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 109
Server version: 5.7.25-TiDB-v5.0.0 TiDB Server (Apache License 2.0) Community Edition, MySQL 5.7 compatible

Copyright (c) 2000, 2021, Oracle and/or its affiliates.
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> show variables like 'auto_increment_increment';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| auto_increment_increment | 10 |
+-----+-----+
1 row in set (0.64 sec)
```

## Lesson 10 练习-1: 在不同作用域修改 TiDB 数据库的系统参数

20. 重启 tidb 数据库，验证GLOBAL范围的参数修改是否会被持久化  
首先关闭 tidb 数据库：

```
[root@centos76_vm ~]# tiup cluster stop tidb-test
Starting component `cluster`: /root/.tiup/components/cluster/v1.4.3/tiup-cluster stop tidb-test
+ [Serial] - SSHKeySet: privateKey=/root/.tiup/storage/cluster/clusters/tidb-test/ssh/id_rsa,
publicKey=/root/.tiup/storage/cluster/clusters/tidb-test/ssh/id_rsa.pub
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.163
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.202
... (省略中间内容)
Stopped cluster `tidb-test` successfully
```

启动 tidb 数据库：

```
[root@centos76_vm ~]# tiup cluster start tidb-test
Starting component `cluster`: /root/.tiup/components/cluster/v1.4.3/tiup-cluster start tidb-test
Starting cluster tidb-test...
+ [Serial] - SSHKeySet: privateKey=/root/.tiup/storage/cluster/clusters/tidb-test/ssh/id_rsa,
publicKey=/root/.tiup/storage/cluster/clusters/tidb-test/ssh/id_rsa.pub
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.157
... (省略中间内容)
Started cluster `tidb-test` successfully
```

21. 登录数据库后，发现参数被持久化，依然为 10：（请根据自己实验环境输入密码）

```
[root@centos76_vm ~]# mysql -h172.16.6.212 -P 4000 -uroot -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 5
Server version: 5.7.25-TiDB-v5.0.0 TiDB Server (Apache License 2.0) Community Edition, MySQL 5.7 compatible

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

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> show variables like 'auto_increment_increment';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| auto_increment_increment | 10 |
+-----+-----+
1 row in set (0.59 sec)
```

## Lesson 10 练习-2: 修改集群配置

### 概述

在这个实验中，我们会使用 `tiup config` 和 `tiup reload` 修改所有 TiKV 节点的配置，我们修改的配置参数名为 `log-level`，其默认值为 `info`，我们将其修改为 `warning`。

### 任务

1. 进入 TiKV-Server，打开配置文件，一般位置为 `/tidb-deploy/tikv-20160/conf`（注意：tikv-20160 可能不同），打开文件 `tikv.toml`：

```
[root@centos76_vm ~]# vi tikv.toml
```

其内容如下（可能不同），发现并没有 `log-level` 参数。

```
WARNING: This file is auto-generated. Do not edit! All your modification will be overwritten!
You can use 'tiup cluster edit-config' and 'tiup cluster reload' to update the configuration
All configuration items you want to change can be added to:
server_configs:
tikv:
aa.b1.c3: value
aa.b2.c4: value
```

2. 进入到安装 `tiup` 的中控机或者节点，执行配置文件编辑命令：

```
[root@centos76_vm ~]# tiup cluster edit-config tidb-test
```

```
global:
 user: tidb
 ssh_port: 22
 ssh_type: builtin
...（省略中间内容）
arch: amd64
os: linux
```

3. 输入 `i`，进入编辑模式：

```
server_configs:
 tidb: {}
 tikv:
 log-level: warning
```

4. 输入 `ESC` 键，输入：`qw`，出现提示如下，输入 `y` 继续：

```
:wq
Please check change highlight above, do you want to apply the change? [y/N]:(default=N) y
Applying changes...
Applied successfully, please use `tiup cluster reload tidb-test [-N <nodes>] [-R <roles>]` to reload config.
```

## Lesson 10 练习-2: 修改集群配置

### 5. 使用 `tiup cluster reload` 命令来载入修改的参数 (这一步会重启所有 TiKV 节点):

```
[root@centos76_vm ~]# tiup cluster edit-config tidb-test
Starting component `cluster`: /root/.tiup/components/cluster/v1.4.3/tiup-cluster reload tidb-test -R tikv
+ [Serial] - SSHKeySet: privateKey=/root/.tiup/storage/cluster/clusters/tidb-test/ssh/id_rsa,
publicKey=/root/.tiup/storage/cluster/clusters/tidb-test/ssh/id_rsa.pub
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.157
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.162
... (省略中间内容)
Still waiting for 2 store leaders to transfer...
 Restarting instance 172.16.6.164
 Restart 172.16.6.164 success
Reloaded cluster `tidb-test` successfully
```

### 6. 再次进入TiKV-Server, 打开配置文件, 一般位置为 `/tidb-deploy/tikv-20160/conf` (注意: `tikv-20160` 可能不同), 打开文件 `tikv.toml`, 发现 `log-level = "warning"` 配置已经被持久化到配置文件中:

```
[root@centos76_vm ~]# vi tikv.toml

WARNING: This file is auto-generated. Do not edit! All your modification will be overwritten!
You can use 'tiup cluster edit-config' and 'tiup cluster reload' to update the configuration
All configuration items you want to change can be added to:
server_configs:
tikv:
aa.b1.c3: value
aa.b2.c4: value
log-level = "warning"
```

## Lesson 11 练习: 用户管理与安全



## Lesson 11 练习：概述

---

### 概述

在本课练习中，你将会创建 TiDB 数据库的用户和角色，并对用户和角色进行管理。之后您还会管理用户和角色的权限，理解用户与角色的关系。

### 实验环境要求

1. 注意，您的实验环境（包括 IP，端口号，用户名，密码和目录）可能和手册不同。
2. 可以连接到实验用虚拟机。
3. TiDB 数据库之前部署完毕。

learn.pingcap.cc

## Lesson 11 练习-1: 创建用户和角色

---

### 概述

在这个练习中，您将会创建用户并为用户设置密码，之后创建 2 个角色，最后将用户和角色删除。

### 任务

#### 7. 启动一个 **mysql** 客户端会话：（请根据自己实验环境输入密码）

```
[root@centos76_vm ~]# mysql -h172.16.6.212 -P4000 -uroot -p
```

Enter password:

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 15

Server version: 5.7.25-TiDB-v5.0.0 TiDB Server (Apache License 2.0) Community Edition, MySQL 5.7 compatible

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

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.

#### 2. 创建用户名为 'jack'@'172.16.6.212'：

```
mysql> create user 'jack'@'172.16.6.212' identified by 'pingcap';
```

Query OK, 0 rows affected (0.07 sec)

#### 3. 创建 2 个新的角色，名为 **r\_manager** 和 **r\_staff**：

```
mysql> create role r_manager, r_staff;
```

Query OK, 0 rows affected (0.03 sec)

## Lesson 11 练习-1: 创建用户和角色

4. 查询 `mysql.user` 表的 `user`, `host` 和 `authentication_string` 列，确认新的用户和角色已经被创建了：

```
mysql> select user, host, authentication_string from mysql.user\G;
***** 1. row *****
 user: root
 host: %
authentication_string:
***** 2. row *****
 user: r_mgr
 host: %
authentication_string:
***** 3. row *****
 user: r_emp
 host: %
authentication_string:
***** 4. row *****
 user: jack
 host: 172.16.6.212
authentication_string: *926E4B88EB93FD344DF0870EE025D6EB153C02DE
***** 5. row *****
 user: r_manager
 host: %
authentication_string:
***** 6. row *****
 user: r_staff
 host: %
authentication_string:
6 rows in set (0.00 sec)

ERROR:
No query specified
```

(这里我们可以注意到：

- (1) 用户和角色都被存储在 `mysql.user` 表中。
- (2) 角色是没有密码的。)

## Lesson 11 练习-1: 创建用户和角色

5. 在 mysql.user 表中查询角色 r\_staff 的详细信息:

```
mysql> select * from mysql.user where user='r_staff'\G;
***** 1. row *****
 Host: %
 User: r_staff
authentication_string:
 Select_priv: N
 Insert_priv: N
 Update_priv: N
 Delete_priv: N
 Create_priv: N
 Drop_priv: N
... (省略中间内容)
 Repl_client_priv: N
1 row in set (0.00 sec)
ERROR:
No query specified
```

(我们会发现角色的特点:

- (1) 角色是被锁定的 (Account\_locked: Y)
- (2) 角色没有密码 (authentication\_string 为空)

## Lesson 11 练习-1: 创建用户和角色

---

6. 修改 'jack'@'172.16.6.212' 的密码为 tidb, 并退出:

```
mysql> alter user 'jack'@'172.16.6.212' identified by 'tidb';

Query OK, 0 rows affected (0.03 sec)

mysql> exit

Bye
```

7. 使用用户 'jack'@'172.16.6.212' 的新密码重新连接, 验证密码是否修改成功:  
(请根据自己实验环境输入密码)

```
[root@centos76_vm ~]# mysql -h172.16.6.212 -P 4000 -ujack -ptidb

mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 19
Server version: 5.7.25-TiDB-v5.0.0 TiDB Server (Apache License 2.0) Community Edition, MySQL 5.7 compatible

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

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.
```

8. 退出当前会话:

```
mysql> exit

Bye
```

## Lesson 11 练习-1: 创建用户和角色

---

9. 以 root 用户登录数据库：（请根据自己实验环境输入密码）

```
[root@centos76_vm ~]# mysql -h172.16.6.212 -P 4000 -uroot -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 21
Server version: 5.7.25-TiDB-v5.0.0 TiDB Server (Apache License 2.0) Community Edition, MySQL 5.7 compatible

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

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.
```

10. 删除角色 r\_staff, 并且通过mysql.user表进行确认:

```
mysql> drop role r_staff;
Query OK, 0 rows affected (0.04 sec)

mysql> select user, host from mysql.user where user='r_staff';
Empty set (0.00 sec)
```

11. 删除角色 r\_manager, 并且通过 mysql.user 表进行确认:

```
mysql> drop role r_manager;
Query OK, 0 rows affected (0.04 sec)

mysql> select user, host from mysql.user where user='r_manager';
Empty set (0.01 sec)
```

12. 删除用户 'jack'@'172.16.6.212', 并且通过 mysql.user 表进行确认:

```
mysql> drop user 'jack'@'172.16.6.212';
Query OK, 0 rows affected (0.04 sec)

mysql> select user, host from mysql.user where user='jack';
Empty set (0.00 sec)
```

## Lesson 11 练习-2: 权限管理

---

### 概述

在这个练习中，您将会管理用户的权限并且创建角色，将角色赋予用户。

### 任务

1. 以 root 用户登录，创建 test 库下测试表 emp:

```
mysql> create table emp(id int, name varchar(20));
Query OK, 0 rows affected (0.53 sec)

mysql> insert into emp values(1,'tom');
Query OK, 1 row affected (0.02 sec)

mysql> insert into emp values(2,'jack');
Query OK, 1 row affected (0.00 sec)
```

2. 创建新用户 'jack'@'172.16.6.212'，密码为 pingcap:

```
mysql> create user 'jack'@'172.16.6.212' identified by 'pingcap';
Query OK, 0 rows affected (0.02 sec)
```

3. 创建新的角色 r\_mgr 和 r\_emp:

```
mysql> create role r_mgr, r_emp;
Query OK, 0 rows affected (0.03 sec)
```

4. 将对 test 库下的表 emp 的读权限赋权给角色 r\_emp:

```
mysql> grant select on test.emp to r_emp;
Query OK, 0 rows affected (0.03 sec)
```

5. 将对 test 库下的所有表的 insert, update 和 delete 权限赋权给角色 r\_mgr:

```
mysql> grant insert, update, delete on test.* to r_mgr;
Query OK, 0 rows affected (0.03 sec)
```

6. 将角色 r\_emp 赋予角色 r\_mgr 和用户 'jack'@'172.16.6.212' :

```
mysql> grant r_emp to r_mgr, 'jack'@'172.16.6.212';
Query OK, 0 rows affected (0.04 sec)
```

## Lesson 11 练习-2: 权限管理

---

7. 以 root 用户登录, 创建 test 库下测试表 dept:

```
mysql> create table dept(id int, dname varchar(20));
Query OK, 0 rows affected (1.03 sec)

mysql> insert into dept values(1, 'dev');
Query OK, 1 row affected (0.02 sec)

mysql> insert into dept values(2, 'sales');
Query OK, 1 row affected (0.00 sec)
```

8. 将对 test 库下的 dept 表的 select 权限赋予用户 'jack'@'172.16.6.212':

```
mysql> grant select on test.dept to 'jack'@'172.16.6.212';
Query OK, 0 rows affected (0.04 sec)
```

9. 开启新的会话, 以用户 'jack'@'172.16.6.212' 连接数据库:  
(请根据自己实验环境输入密码)

```
[root@centos76_vm ~]# mysql -h172.16.6.212 -P4000 -ujack -ppingcap
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 29
Server version: 5.7.25-TiDB-v5.0.0 TiDB Server (Apache License 2.0) Community Edition, MySQL 5.7 compatible

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

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.
```



## Lesson 11 练习-2: 权限管理

10. 切换当前数据库到 test，之后查询表 emp 的第一行和 表 dept 的第一行：

```
mysql> use test;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from emp limit 1;
ERROR 1142 (42000): SELECT command denied to user 'jack'@'172.16.6.212' for table 'emp'
mysql> select * from dept limit 1;
+----+-----+
| id | dname |
+----+-----+
| 1 | dev |
+----+-----+
1 row in set (0.00 sec)
```

(结论是：

- (1) 用户 'jack'@'172.16.6.212' 无法查询表 emp，因为权限是通过角色 r\_emp 赋予的，但是这个角色并没有在会话中开启。
- (2) 用户 'jack'@'172.16.6.212' 可以查询表 dept，因为权限是直接赋予用户的。)

11. 检查用户 'jack'@'172.16.6.212' 所在会话的权限：

```
mysql> select current_role();
+-----+
| current_role() |
+-----+
| |
+-----+
1 row in set (0.00 sec)

mysql> show grants;
+-----+
| Grants for User |
+-----+
| GRANT USAGE ON *.* TO 'jack'@'172.16.6.212' |
| GRANT Select ON test.dept TO 'jack'@'172.16.6.212' |
| GRANT 'r_emp'@'%' TO 'jack'@'172.16.6.212' |
+-----+
3 rows in set (0.00 sec)
```

(我们发现：

- (1) 会话没有开启任何角色。
- (2) 用户虽然被赋予了角色 r\_emp，但是这个角色并没有开启。)

## Lesson 11 练习-2: 权限管理

12. 我们接下来为用户 'jack'@'172.16.6.212' 开启角色 r\_emp, 命令如下:

```
mysql> set role all;
Query OK, 0 rows affected (0.00 sec)

mysql> select current_role();
+-----+
| current_role() |
+-----+
| 'r_emp'@'%' |
+-----+
1 row in set (0.01 sec)

mysql> show grants;
+-----+
| Grants for User |
+-----+
| GRANT USAGE ON *.* TO 'jack'@'172.16.6.212' |
| GRANT Select ON test.emp TO 'jack'@'172.16.6.212' |
| GRANT Select ON test.dept TO 'jack'@'172.16.6.212' |
| GRANT 'r_emp'@'%' TO 'jack'@'172.16.6.212' |
+-----+
4 rows in set (0.01 sec)
```

(我们看到, 当开启角色后, show grants的结果显示了用户 'jack'@'172.16.6.212' 所有的权限, 包括通过角色赋予的权限。)

13. 接下来继续查询 test 库下面的表 emp 的第一行:

```
mysql> select * from emp limit 1;
+----+-----+
| id | name |
+----+-----+
| 1 | tom |
+----+-----+
```

(我们发现用户 'jack'@'172.16.6.212' 通过角色 r\_emp 赋予了查询 test 库下面的表 emp 的第一行的权限。)

## Lesson 11 练习-2: 权限管理

---

14. 尝试删除 test 库下的表 emp 的第一行:

```
mysql> delete from emp where id=1;
```

```
ERROR 8121 (HY000): privilege check fail
```

(发现用户 'jack'@'172.16.6.212' 并没有删除权限。)

15. 最后, 退出终端:

```
mysql> exit
```

```
Bye
```

16. 关闭所有窗口, 实验结束。

learn.pingcap.cc

## Lesson 12 练习: TiDB 文件与 日志管理

## Lesson 12 练习：概述

---

### 概述

在本课练习中，您将熟悉 TiDB 数据库集群中各个节点的数据文件、日志文件和配置文件位置。

### 实验环境要求

1. 注意，您的实验环境（包括 IP，端口号，用户名，密码和目录）可能和手册不同。
2. 可以连接到实验用虚拟机。
3. TiDB 数据库之前部署完毕。

learn.pingcap.cc

## Lesson 12 练习-1: 查看 TiDB 各个节点的数据文件、日志文件和配置文件

### 概述

在这个练习中，您将熟悉 TiDB 数据库集群中各个节点的数据文件、日志文件和配置文件位置。

### 任务

8. 登录到部署了 TiUP 的节点，执行 `tiup cluster edit-config tidb-test`，查看集群配置文件汇总：

```
[root@centos76_vm ~]# tiup cluster edit-config tidb-test
global:
 user: tidb
 ssh_port: 22
 ssh_type: builtin
 deploy_dir: /tidb-deploy
 data_dir: /tidb-data
 os: linux
 arch: amd64
 ... (省略中间内容)
alertmanager_servers:
 - host: 172.16.6.157
 ssh_port: 22
 web_port: 9093
 cluster_port: 9094
 deploy_dir: /tidb-deploy/alertmanager-9093
 data_dir: /tidb-data/alertmanager-9093
 arch: amd64
 os: linux
```

(在结果中找到：`tidb_servers`，`tikv_servers` 和 `pd_servers` 三个选项，我们可以看到每个选项下面有文件夹 `deploy_dir` 和 `data_dir`。其中文件夹 `deploy_dir` 表示节点的软件目录，`data_dir` 表示节点的数据目录。)

9. 参考 `tidb_servers` 的目录，连接入其中一个 `tidb` 节点，查看里面的软件目录和数据目录：查看 `tidb` 节点的软件目录：

```
[root@centos76_vm ~]# cd /tidb-deploy/
[root@centos76_vm tidb-deploy]# ls
monitor-9100 tidb-4000
[root@centos76_vm tidb-deploy]# cd tidb-4000/
[root@centos76_vm tidb-4000]# ls
bin conf log scripts
```

(其中, /tidb-deploy/tidb-4000/conf 目录下的文件 tidb.toml 为配置文件;  
/tidb-deploy/tidb-4000/log 目录下的文件为日志文件。)

## Lesson 12 练习-1: 查看 TiDB 各个节点的数据文件、日志文件和配置文件

查看 tidb 节点的数据目录:

```
[root@centos76_vm ~]# cd /tidb-data/
[root@centos76_vm tidb-data]# ls
monitor-9100
```

(发现只有监控数据, 数据库数据不存储在 tidb 节点上。)

3. 参考 tikv\_servers 的目录, 连接入其中一个 tikv 节点, 查看里面的软件目录和数据目录:

```
[root@copy-of-vm-ee-centos76-v1~]# cd /tidb-deploy/
[root@copy-of-vm-ee-centos76-v1 tidb-deploy]# ls
monitor-9100 tikv-20160
[root@copy-of-vm-ee-centos76-v1 tidb-deploy]# cd tikv-20160/
[root@copy-of-vm-ee-centos76-v1 tikv-20160]# ls
bin conf log scripts
```

(其中, /tidb-deploy/tikv-20160/conf 目录下的文件 tikv.toml 为配置文件;  
/tidb-deploy/tikv-20160/log 目录下的文件为日志文件。)

查看 tikv 节点的数据目录:

```
[root@copy-of-vm-ee-centos76-v1 ~]# cd /tidb-data/
[root@copy-of-vm-ee-centos76-v1 tidb-data]# ls
monitor-9100 tikv-20160
[root@copy-of-vm-ee-centos76-v1 tidb-data]# cd tikv-20160/
[root@copy-of-vm-ee-centos76-v1 tikv-20160]# ls
db raftdb.info.2021-05-27-12:43:11.620945690
import rocksdb.info
last_tikv.toml rocksdb.info.2021-05-27-12:12:29.269924605
LOCK snap
raft space_placeholder_file
raftdb.info
```

## Lesson 12 练习-1: 查看 TiDB 各个节点的数据文件、日志文件和配置文件

其中 db 目录下为数据文件:

```
[root@copy-of-vm-ee-centos76-v1]# cd db/
[root@copy-of-vm-ee-centos76-v1 db]# ls
000019.sst 000058.sst 000545.sst 000552.sst 000559.sst OPTIONS-000572
000021.sst 000059.sst 000546.sst 000553.sst 000560.log OPTIONS-000574
000024.sst 000071.sst 000547.sst 000554.sst 000562.sst
000054.sst 000118.sst 000548.sst 000555.sst CURRENT
000055.sst 000123.log 000549.sst 000556.sst IDENTITY
000056.sst 000126.sst 000550.sst 000557.sst LOCK
000057.sst 000127.sst 000551.sst 000558.sst MANIFEST-000119
```

4. 参考 `pd_servers` 的目录，连接入其中一个 `pd` 节点，查看里面的软件目录和数据目录：  
查看 `pd` 节点的软件目录：

```
[root@centos76_vm ~]# cd /tidb-deploy/
[root@centos76_vm tidb-deploy]# ls
monitor-9100 pd-2379
[root@centos76_vm tidb-deploy]# cd pd-2379/
[root@centos76_vm pd-2379]# ls
bin conf log scripts
```

(其中，`/tidb-deploy/pd-2379/conf` 目录下的文件 `pd.toml` 为配置文件；

`/tidb-deploy/pd-2379/log` 目录下的文件为日志文件。)

查看 `pd` 节点的数据目录：

```
[root@centos76_vm ~]# cd /tidb-data/
[root@centos76_vm tidb-data]# ls
monitor-9100 pd-2379
[root@centos76_vm tidb-data]# cd pd-2379/
[root@centos76_vm pd-2379]# ls
member region-meta
```



## Lesson 13 练习: TiDB 的监控

© 北京平凯星辰科技发展有限公司，2021。北京平凯星辰科技发展有限公司及其关联公司（“PingCAP 集团”）保留所有权利。未经 PingCAP 集团事先书面许可，严禁复制、改编、汇编、传播或翻译本文全部或部分内容。

## Lesson 13 练习：概述

---

### 概述

在本课练习中，您将使用 TiDB 数据库提供的 Prometheus + Grafana 监控和 PD 节点自带的 TiDB 的 Dashboard 监控，并观察常见的监控指标

### 实验环境要求

1. 注意，您的实验环境（包括 IP，端口号，用户名，密码和目录）可能和手册不同。
2. 可以连接到实验用虚拟机。
3. TiDB 数据库之前部署完毕。

## Lesson 13 练习-1: TiDB 的监控

### 概述

在本课练习中，您将使用 TiDB 数据库提供的 Prometheus + Grafana 监控和 PD 节点自带的 TiDB 的 Dashboard 监控，并观察常见的监控指标

### 任务

10. 确认 TiDB 集群状态，如下：

```
[root@centos76_vm ~]# tiup cluster display tidb-test
Starting component `cluster`: /root/.tiup/components/cluster/v1.4.4/tiup-cluster display tidb-test
Cluster type: tidb
Cluster name: tidb-test
Cluster version: v5.0.0
SSH type: builtin
Dashboard URL: http://172.16.6.157:2379/dashboard
ID Role Host Ports OS/Arch Status Data Dir Deploy Dir
-- -
172.16.6.157:9093 alertmanager 172.16.6.157 9093/9094 linux/x86_64 Up /tidb-data/alertmanager-9093 /tidb-deploy/alertmanager-9093
172.16.6.157:3000 grafana 172.16.6.157 3000 linux/x86_64 Up - /tidb-deploy/grafana-3000
172.16.6.157:2379 pd 172.16.6.157 2379/2380 linux/x86_64 Up|L|UI /tidb-data/pd-2379 /tidb-deploy/pd-2379
172.16.6.202:2379 pd 172.16.6.202 2379/2380 linux/x86_64 Up /tidb-data/pd-2379 /tidb-deploy/pd-2379
172.16.6.210:2379 pd 172.16.6.210 2379/2380 linux/x86_64 Up /tidb-data/pd-2379 /tidb-deploy/pd-2379
172.16.6.157:9090 prometheus 172.16.6.157 9090 linux/x86_64 Up /tidb-data/prometheus-9090 /tidb-deploy/prometheus-9090
172.16.6.212:4000 tidb 172.16.6.212 4000/10080 linux/x86_64 Up - /tidb-deploy/tidb-4000
172.16.6.162:20160 tikv 172.16.6.162 20160/20180 linux/x86_64 Up /tidb-data/tikv-20160 /tidb-deploy/tikv-20160
172.16.6.163:20160 tikv 172.16.6.163 20160/20180 linux/x86_64 Up /tidb-data/tikv-20160 /tidb-deploy/tikv-20160
172.16.6.164:20160 tikv 172.16.6.164 20160/20180 linux/x86_64 Up /tidb-data/tikv-20160 /tidb-deploy/tikv-20160
Total nodes: 10
```

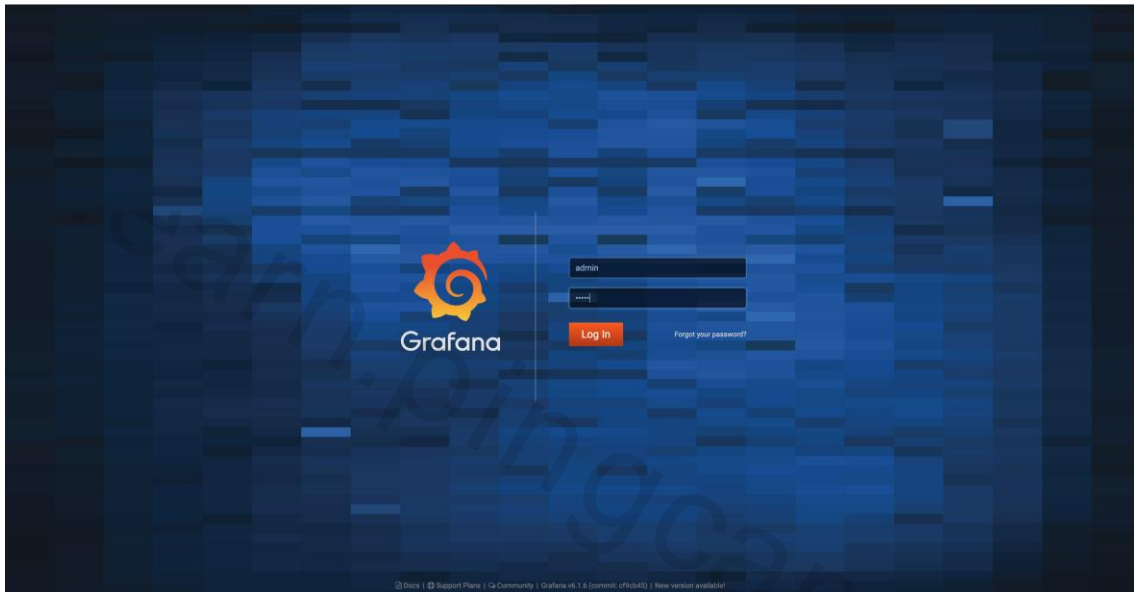
在信息中，可以发现：

Dashboard URL: <http://172.16.6.210:2379/dashboard>  
grafana 172.16.6.157 3000 (URL: <http://172.16.6.157:3000>)

## Lesson 13 练习-1: TiDB 的监控

---

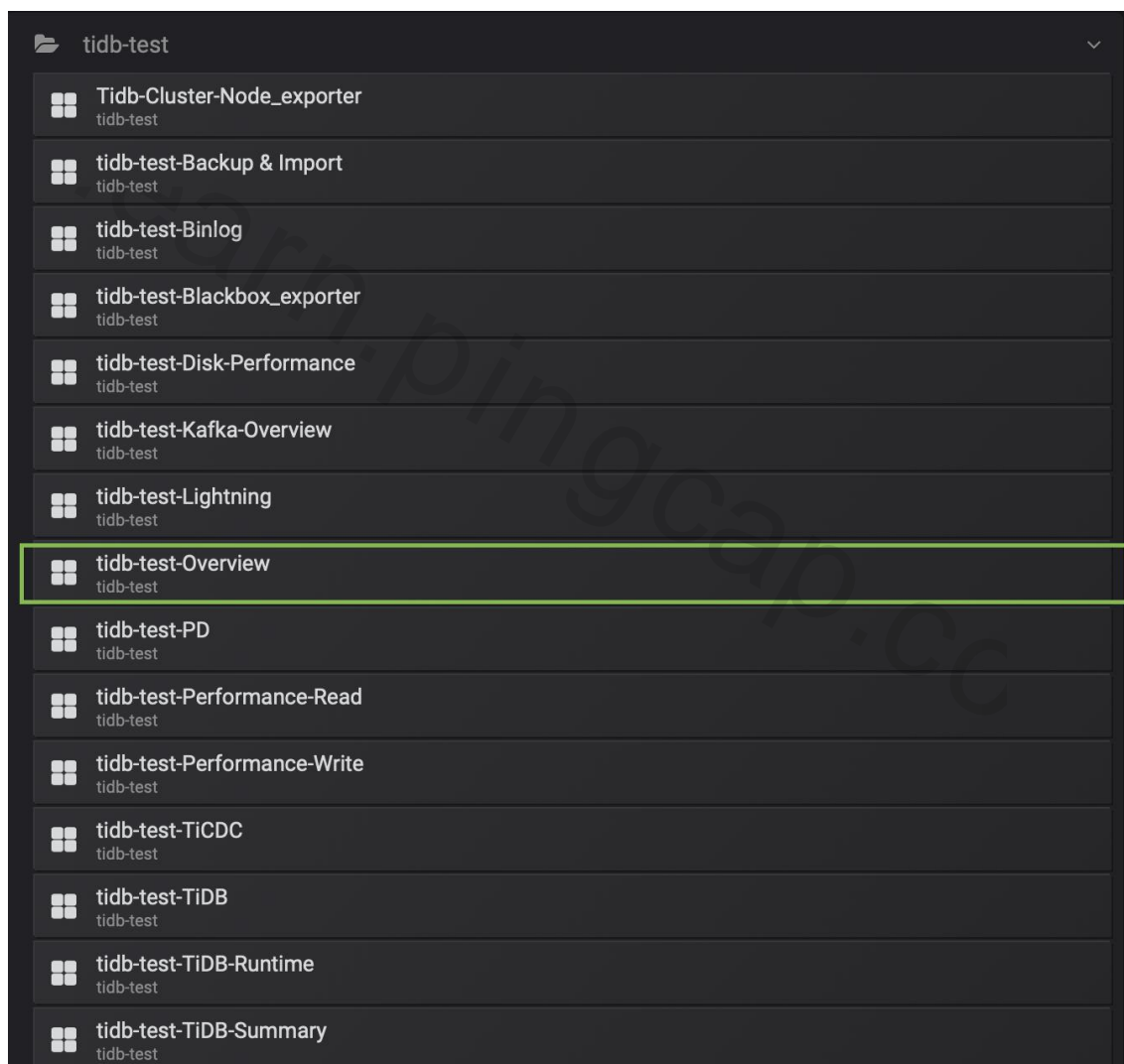
2. 访问 Prometheus + Grafana 监控，用户名/密码默认为 admin/admin，如下图：



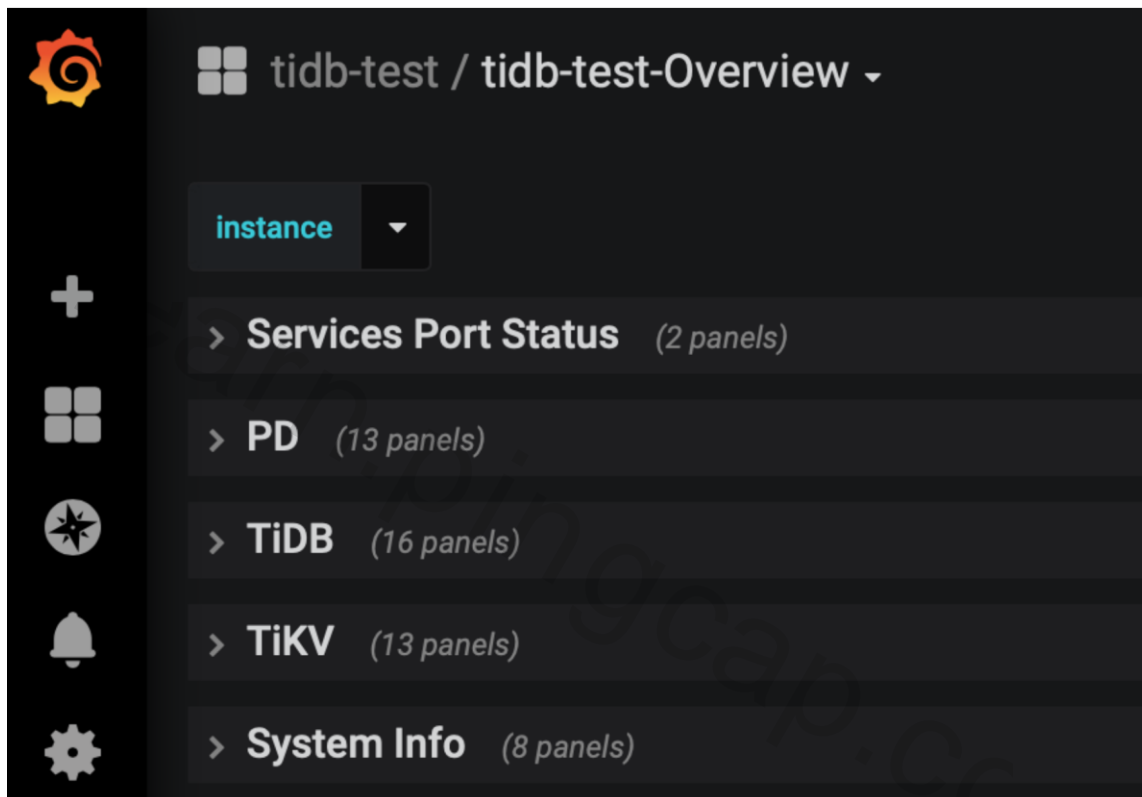
## Lesson 13 练习-1: TiDB 的监控

---

3. 选择需要监控的数据库 `tidb-test`, 展开后 找到 `tidb-test-Overview`, 查看相应内容, 如下图:



## Lesson 13 练习-1: TiDB 的监控



4. 展开 tidb-test-Overview 的 System-Info, 会看到整个 TiDB 数据库所有的服务器监控指标, 如下图:



在图中，我们可以监控到 CPU 配置，内存配置，CPU 使用率，网络状态和内存使用率等指标。

## Lesson 13 练习-1: TiDB 的监控

5. 收起 System-Info，展开 tidb-test-Overview 的 Service Port Status，会看到各个节点的在线状况，如下图：



在图中，我们可以监控到各个节点的在线状况。

6. 收起 Service Port Status，展开 tidb-test-Overview 的 PD 监控页，会看到各个 PD 节点的监控指标，如下图：



在图中，我们可以监控到整个 TiDB 存储的总大小，存储使用大小，Region 数量以及 Region 监控信息等。

## Lesson 13 练习-1: TiDB 的监控

7. 收起 PD 监控页，展开 tidb-test-Overview 的 TiDB 监控页，会看到各个 TiDB-Server 节点的监控指标，如下图：



在图中，我们可以监控到整个 TiDB 执行 SQL 语句的情况等，比如连接数量，每秒执行 SQL 的数量，SQL 的平均处理时间等。

8. 收起 TiDB 监控页，展开 tidb-test-Overview 的 TiKV 监控页，会看到各个 TiKV-Server 节点的监控指标，如下图：

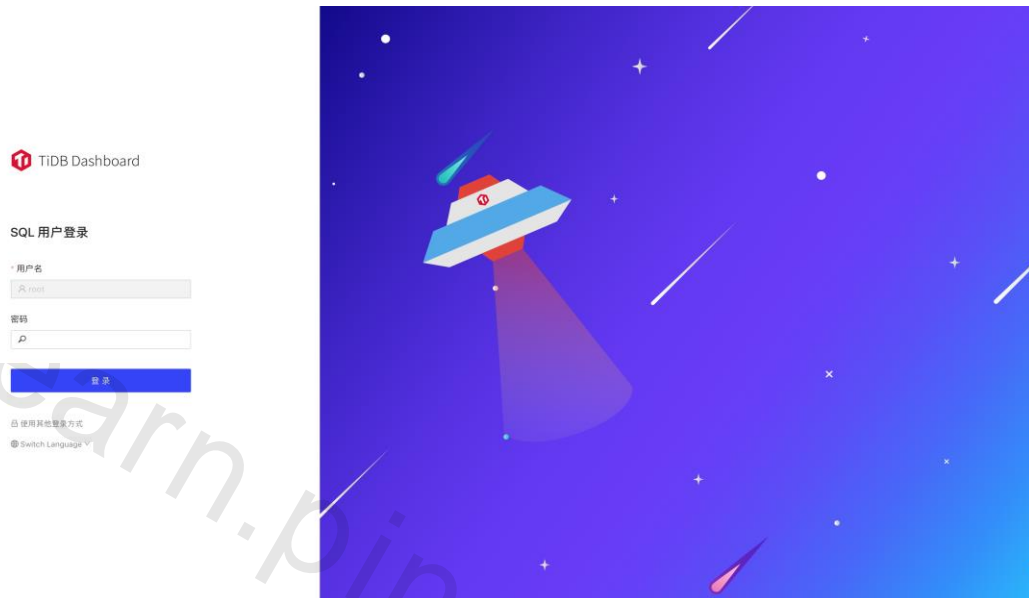


在图中，我们可以监控到所有 TiKV 节点的情况，比如 leader 的数量，region 的数量，TiKV 节点的 CPU 负载和内存使用量等。

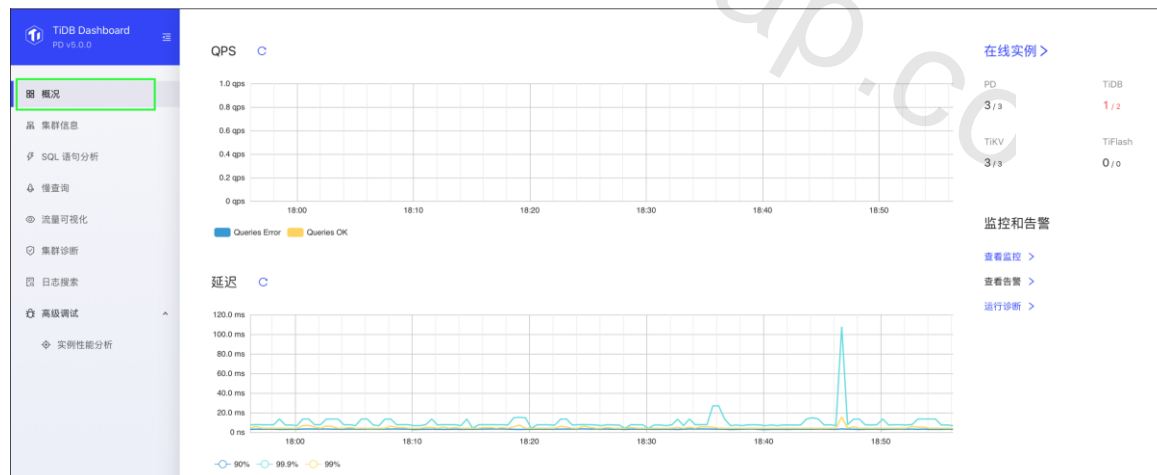


## Lesson 13 练习-1: TiDB 的监控

9. 另外开启一个浏览器窗口，输入地址：`http://172.16.6.210:2379/dashboard`，会打开 TiDB 数据库 PD 节点的 Dashboard 监控，第一次登录的用户名为 root，默认没有密码，如下图：



10. 进入 TiDB Dashboard，我们可以查看常用监控指标，比如通过左边的“概况”，可以查看 TiDB 的整体 QPS 和 SQL 延迟，如下图：



## Lesson 13 练习-1: TiDB 的监控

11. 我们通过左边的“集群信息”，可以查看当前数据库集群各个节点的监控状态，如下图：

TiDB Dashboard  
PD v5.0.0

三

概況

集群信息

SQL 语句分析

慢查询

流量可视化

集群诊断

日志搜索

高级调试

实例性能分析

实例

主机

磁盘

存储拓扑

统计

| 地址                 | 状态     | 启动时间             | 版本      | Git 哈希值                   | 部署路径                            |
|--------------------|--------|------------------|---------|---------------------------|---------------------------------|
| PD (3)             |        |                  |         |                           |                                 |
| 172.16.6.157:2379  | ● 在线   | 上星期一 21:36       | v5.0.0  | 687acfc8db8378e01e47ab... | /tidb-deploy/pd-2379/bin        |
| 172.16.6.202:2379  | ● 在线   | 上星期一 21:36       | v5.0.0  | 687acfc8db8378e01e47ab... | /tidb-deploy/pd-2379/bin        |
| 172.16.6.210:2379  | ● 在线   | 上星期一 21:36       | v5.0.0  | 687acfc8db8378e01e47ab... | /tidb-deploy/pd-2379/bin        |
| TiDB (2)           |        |                  |         |                           |                                 |
| 172.16.6.208:4000  | ❏ 无法访问 | 2021年5月26日 14:34 | v4.0.11 | 058e52ad7c1b477147dcb...  | /home/tidb-deploy/tidb-4000/bin |
| 172.16.6.212:4000  | ● 在线   | 上星期一 21:36       | v5.0.0  | bdac0885cd11bdf571aad9... | /tidb-deploy/tidb-4000/bin      |
| TiKV (3)           |        |                  |         |                           |                                 |
| 172.16.6.162:20160 | ● 在线   | 上星期一 21:36       | v5.0.0  | 7706b9634bd901c9fe8dbe... | /tidb-deploy/tikv-20160/bin     |
| 172.16.6.163:20160 | ● 在线   | 上星期一 21:36       | v5.0.0  | 7706b9634bd901c9fe8dbe... | /tidb-deploy/tikv-20160/bin     |
| 172.16.6.164:20160 | ● 在线   | 上星期一 21:36       | v5.0.0  | 7706b9634bd901c9fe8dbe... | /tidb-deploy/tikv-20160/bin     |

## Lesson 14 练习: TiDB 集群管理

## Lesson 14 练习：概述

---

### 概述

在本课练习中，您将进行整个 TiDB 数据库的扩容和缩容，之后进行集群名的修改。

### 实验环境要求

1. 注意，您的实验环境（包括 IP，端口号，用户名，密码和目录）可能和手册不同。
2. 可以连接到实验用虚拟机。
3. 部署好 1 个 TiBD 数据库集群，集群名为 tidb-test 。
4. 一个节点，可以作为 tikv 加入到现有集群中。

learn.pingcap.cc

## Lesson 14 练习-1: 对 TiDB 集群进行扩容和缩容

### 概述

在这个练习中，我们对现有的 TiDB 数据库集群进行扩容操作（加入一个 TiKV 节点），之后进行缩容操作（减少一个 TiKV 节点）。

### 任务

11. 查询现有的数据库集群，发现有 3 个 TiKV 节点，如下：

```
[root@centos76_vm ~]# tiup cluster display tidb-test
```

Found cluster newer version:

The latest version: v1.4.4  
Local installed version: v1.4.3  
Update current component: tiup update cluster  
Update all components: tiup update --all

Starting component `cluster`: /root/.tiup/components/cluster/v1.4.3/tiup-cluster display tidb-test

Cluster type: tidb  
Cluster name: tidb-test  
Cluster version: v5.0.0  
SSH type: builtin  
Dashboard URL: http://172.16.6.210:2379/dashboard

| ID                 | Role         | Host         | Ports       | OS/Arch      | Status | Data Dir                     | Deploy Dir                     |
|--------------------|--------------|--------------|-------------|--------------|--------|------------------------------|--------------------------------|
| 172.16.6.157:9093  | alertmanager | 172.16.6.157 | 9093/9094   | linux/x86_64 | Up     | /tidb-data/alertmanager-9093 | /tidb-deploy/alertmanager-9093 |
| 172.16.6.157:3000  | grafana      | 172.16.6.157 | 3000        | linux/x86_64 | Up     | -                            | /tidb-deploy/grafana-3000      |
| 172.16.6.157:2379  | pd           | 172.16.6.157 | 2379/2380   | linux/x86_64 | Up     | /tidb-data/pd-2379           | /tidb-deploy/pd-2379           |
| 172.16.6.202:2379  | pd           | 172.16.6.202 | 2379/2380   | linux/x86_64 | Up L   | /tidb-data/pd-2379           | /tidb-deploy/pd-2379           |
| 172.16.6.210:2379  | pd           | 172.16.6.210 | 2379/2380   | linux/x86_64 | Up UI  | /tidb-data/pd-2379           | /tidb-deploy/pd-2379           |
| 172.16.6.157:9090  | prometheus   | 172.16.6.157 | 9090        | linux/x86_64 | Up     | /tidb-data/prometheus-9090   | /tidb-deploy/prometheus-9090   |
| 172.16.6.212:4000  | tidb         | 172.16.6.212 | 4000/10080  | linux/x86_64 | Up     | -                            | /tidb-deploy/tidb-4000         |
| 172.16.6.162:20160 | tikv         | 172.16.6.162 | 20160/20180 | linux/x86_64 | Up     | /tidb-data/tikv-20160        | /tidb-deploy/tikv-20160        |
| 172.16.6.163:20160 | tikv         | 172.16.6.163 | 20160/20180 | linux/x86_64 | Up     | /tidb-data/tikv-20160        | /tidb-deploy/tikv-20160        |
| 172.16.6.164:20160 | tikv         | 172.16.6.164 | 20160/20180 | linux/x86_64 | Up     | /tidb-data/tikv-20160        | /tidb-deploy/tikv-20160        |

Total nodes: 10

（注意：您的环境可能和实验手册不同。）

## Lesson 14 练习-1: 对 TiDB 集群进行扩容和缩容

### 2. 编辑扩容文件，内容如下：

```
[root@centos76_vm ~]# vi scale-out-tikv.yaml
tikv_servers:
- host: 172.16.6.157
 ssh_port: 22
 port: 20160
 status_port: 20180
 deploy_dir: /tidb-deploy/tikv-20160
 data_dir: /tidb-data/tikv-20160
 log_dir: /tidb-deploy/tikv-20160/log
```

(其中，加入 TiKV 节点 IP 地址为：172.16.6.157，端口号：20160  
注意：您的环境可能和实验手册不同。)

### 3. 执行如下命令，进行扩容操作：

```
[root@centos76_vm ~]# tiup cluster scale-out tidb-test scale-out-tikv.yaml -uroot -p
Found cluster newer version:

The latest version: v1.4.4
Local installed version: v1.4.3
Update current component: tiup update cluster
Update all components: tiup update --all

Starting component `cluster`: /root/.tiup/components/cluster/v1.4.3/tiup-cluster scale-out tidb-test scale-out-tikv.yaml -uroot -p
Please confirm your topology:
Cluster type: tidb
Cluster name: tidb-test
Cluster version: v5.0.0
Role Host Ports OS/Arch Directories

tikv 172.16.6.157 20160/20180 linux/x86_64 /tidb-deploy/tikv-20160,/tidb-data/tikv-20160
Attention:
 1. If the topology is not what you expected, check your yaml file.
 2. Please confirm there is no port/directory conflicts in same host.
Do you want to continue? [y/N]: (default=N) y
Input SSH password:
... (省略中间内容)
+ [Serial] - UpdateTopology: cluster=tidb-test
Scaled cluster `tidb-test` out successfully
```

## Lesson 14 练习-1: 对 TiDB 集群进行扩容和缩容

### 4. 查看集群状态，发现新的 TiKV 节点已经加入，如下：

```
[root@centos76_vm ~]# tiup cluster display tidb-test
```

Found cluster newer version:

The latest version: v1.4.4  
 Local installed version: v1.4.3  
 Update current component: tiup update cluster  
 Update all components: tiup update --all

Starting component `cluster`: /root/.tiup/components/cluster/v1.4.3/tiup-cluster display tidb-test

Cluster type: tidb  
 Cluster name: tidb-test  
 Cluster version: v5.0.0  
 SSH type: builtin  
 Dashboard URL: http://172.16.6.210:2379/dashboard

| ID                 | Role         | Host         | Ports       | OS/Arch      | Status | Data Dir                     | Deploy Dir                     |
|--------------------|--------------|--------------|-------------|--------------|--------|------------------------------|--------------------------------|
| 172.16.6.157:9093  | alertmanager | 172.16.6.157 | 9093/9094   | linux/x86_64 | Up     | /tidb-data/alertmanager-9093 | /tidb-deploy/alertmanager-9093 |
| 172.16.6.157:3000  | grafana      | 172.16.6.157 | 3000        | linux/x86_64 | Up     | -                            | /tidb-deploy/grafana-3000      |
| 172.16.6.157:2379  | pd           | 172.16.6.157 | 2379/2380   | linux/x86_64 | Up     | /tidb-data/pd-2379           | /tidb-deploy/pd-2379           |
| 172.16.6.202:2379  | pd           | 172.16.6.202 | 2379/2380   | linux/x86_64 | Up L   | /tidb-data/pd-2379           | /tidb-deploy/pd-2379           |
| 172.16.6.210:2379  | pd           | 172.16.6.210 | 2379/2380   | linux/x86_64 | Up UI  | /tidb-data/pd-2379           | /tidb-deploy/pd-2379           |
| 172.16.6.157:9090  | prometheus   | 172.16.6.157 | 9090        | linux/x86_64 | Up     | /tidb-data/prometheus-9090   | /tidb-deploy/prometheus-9090   |
| 172.16.6.212:4000  | tidb         | 172.16.6.212 | 4000/10080  | linux/x86_64 | Up     | -                            | /tidb-deploy/tidb-4000         |
| 172.16.6.157:20160 | tikv         | 172.16.6.157 | 20160/20180 | linux/x86_64 | Up     | /tidb-data/tikv-20160        | /tidb-deploy/tikv-20160        |
| 172.16.6.162:20160 | tikv         | 172.16.6.162 | 20160/20180 | linux/x86_64 | Up     | /tidb-data/tikv-20160        | /tidb-deploy/tikv-20160        |
| 172.16.6.163:20160 | tikv         | 172.16.6.163 | 20160/20180 | linux/x86_64 | Up     | /tidb-data/tikv-20160        | /tidb-deploy/tikv-20160        |
| 172.16.6.164:20160 | tikv         | 172.16.6.164 | 20160/20180 | linux/x86_64 | Up     | /tidb-data/tikv-20160        | /tidb-deploy/tikv-20160        |

Total nodes: 11

## Lesson 14 练习-1: 对 TiDB 集群进行扩容和缩容

5. 扩容成功后，我们进行缩容操作，如下所示：

```
[root@centos76_vm ~]# tiup cluster scale-in tidb-test --node 172.16.6.157:20160
Found cluster newer version:

The latest version: v1.4.4
Local installed version: v1.4.3
Update current component: tiup update cluster
Update all components: tiup update --all

Starting component `cluster`: /root/.tiup/components/cluster/v1.4.3/tiup-cluster scale-in tidb-test --node 172.16.6.157:20160
This operation will delete the 172.16.6.157:20160 nodes in `tidb-test` and all their data.
Do you want to continue? [y/N]:(default=N) y
Scale-in nodes...
+ [Serial] - SSHKeySet: privateKey=/root/.tiup/storage/cluster/clusters/tidb-test/ssh/id_rsa,
publicKey=/root/.tiup/storage/cluster/clusters/tidb-test/ssh/id_rsa.pub
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.163
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.202
... (省略中间内容)
- Regenerate config grafana -> 172.16.6.157:3000 ... Done
- Regenerate config alertmanager -> 172.16.6.157:9093 ... Done
+ [Serial] - SystemCtl: host=172.16.6.157 action=reload prometheus-9090.service
Scaled cluster `tidb-test` in successfully
```



## Lesson 14 练习-1: 对 TiDB 集群进行扩容和缩容

6. 查询所容后的集群状态，发现 172.16.6.157 节点的状态已经变为 Tombstone，代表节点已经下线，如下所示：

```
[root@centos76_vm ~]# tiup cluster display tidb-test
Found cluster newer version:

The latest version: v1.4.4
Local installed version: v1.4.3
Update current component: tiup update cluster
Update all components: tiup update --all

Starting component `cluster`: /root/.tiup/components/cluster/v1.4.3/tiup-cluster display tidb-test
Cluster type: tidb
Cluster name: tidb-test
Cluster version: v5.0.0
SSH type: builtin
Dashboard URL: http://172.16.6.210:2379/dashboard
```

| ID                 | Role         | Host         | Ports       | OS/Arch      | Status    | Data Dir                     | Deploy Dir                     |
|--------------------|--------------|--------------|-------------|--------------|-----------|------------------------------|--------------------------------|
| 172.16.6.157:9093  | alertmanager | 172.16.6.157 | 9093/9094   | linux/x86_64 | Up        | /tidb-data/alertmanager-9093 | /tidb-deploy/alertmanager-9093 |
| 172.16.6.157:3000  | grafana      | 172.16.6.157 | 3000        | linux/x86_64 | Up        | -                            | /tidb-deploy/grafana-3000      |
| 172.16.6.157:2379  | pd           | 172.16.6.157 | 2379/2380   | linux/x86_64 | Up        | /tidb-data/pd-2379           | /tidb-deploy/pd-2379           |
| 172.16.6.202:2379  | pd           | 172.16.6.202 | 2379/2380   | linux/x86_64 | Up L      | /tidb-data/pd-2379           | /tidb-deploy/pd-2379           |
| 172.16.6.210:2379  | pd           | 172.16.6.210 | 2379/2380   | linux/x86_64 | Up UI     | /tidb-data/pd-2379           | /tidb-deploy/pd-2379           |
| 172.16.6.157:9090  | prometheus   | 172.16.6.157 | 9090        | linux/x86_64 | Up        | /tidb-data/prometheus-9090   | /tidb-deploy/prometheus-9090   |
| 172.16.6.212:4000  | tidb         | 172.16.6.212 | 4000/10080  | linux/x86_64 | Up        | -                            | /tidb-deploy/tidb-4000         |
| 172.16.6.157:20160 | tikv         | 172.16.6.157 | 20160/20180 | linux/x86_64 | Tombstone | /tidb-data/tikv-20160        | /tidb-deploy/tikv-20160        |
| 172.16.6.162:20160 | tikv         | 172.16.6.162 | 20160/20180 | linux/x86_64 | Up        | /tidb-data/tikv-20160        | /tidb-deploy/tikv-20160        |
| 172.16.6.163:20160 | tikv         | 172.16.6.163 | 20160/20180 | linux/x86_64 | Up        | /tidb-data/tikv-20160        | /tidb-deploy/tikv-20160        |
| 172.16.6.164:20160 | tikv         | 172.16.6.164 | 20160/20180 | linux/x86_64 | Up        | /tidb-data/tikv-20160        | /tidb-deploy/tikv-20160        |

```
Total nodes: 11
There are some nodes can be pruned:
Nodes: [172.16.6.157:20160]
You can destroy them with the command: `tiup cluster prune tidb-test`
```

## Lesson 14 练习-1: 对 TiDB 集群进行扩容和缩容

7. 执行 `tiup cluster prune tidb-test` , 清理节点信息, 如下:

```
[root@centos76_vm ~]# tiup cluster prune tidb-test
```

```
Found cluster newer version:
```

```
The latest version: v1.4.4
```

```
Local installed version: v1.4.3
```

```
Update current component: tiup update cluster
```

```
Update all components: tiup update --all
```

```
Starting component `cluster`: /root/.tiup/components/cluster/v1.4.3/tiup-cluster prune tidb-test
```

```
+ [Serial] - SSHKeySet: privateKey=/root/.tiup/storage/cluster/clusters/tidb-test/ssh/id_rsa,
publicKey=/root/.tiup/storage/cluster/clusters/tidb-test/ssh/id_rsa.pub
```

```
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.157
```

```
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.202
```

```
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.157
```

```
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.210
```

```
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.162
```

```
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.163
```

```
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.164
```

```
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.157
```

```
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.212
```

```
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.157
```

```
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.157
```

```
+ [Serial] - FindTombstoneNodes
```

```
Will destroy these nodes: [172.16.6.157:20160]
```

```
Do you confirm this action? [y/N]:(default=N) y
```

```
Start destroy Tombstone nodes: [172.16.6.157:20160] ...
```

```
... (省略中间内容)
```

```
- Regenerate config prometheus -> 172.16.6.157:9090 ... Done
```

```
- Regenerate config grafana -> 172.16.6.157:3000 ... Done
```

```
- Regenerate config alertmanager -> 172.16.6.157:9093 ... Done
```

```
+ [Serial] - SystemCtl: host=172.16.6.157 action=reload prometheus-9090.service
```

```
Destroy success
```

## Lesson 14 练习-1: 对 TiDB 集群进行扩容和缩容

8. 查询集群状态，发现之前扩容的节点已经消失了，如下图所示：

```
[root@centos76_vm ~]# tiup cluster display tidb-test
```

Found cluster newer version:

The latest version: v1.4.4  
Local installed version: v1.4.3  
Update current component: tiup update cluster  
Update all components: tiup update --all

Starting component `cluster`: /root/.tiup/components/cluster/v1.4.3/tiup-cluster prune tidb-test

```
[root@centos76_vm ~]# tiup cluster display tidb-test
```

Found cluster newer version:

The latest version: v1.4.4  
Local installed version: v1.4.3  
Update current component: tiup update cluster  
Update all components: tiup update --all

Starting component `cluster`: /root/.tiup/components/cluster/v1.4.3/tiup-cluster display tidb-test

Cluster type: tidb  
Cluster name: tidb-test  
Cluster version: v5.0.0  
SSH type: builtin  
Dashboard URL: http://172.16.6.210:2379/dashboard

| ID                 | Role         | Host         | Ports       | OS/Arch      | Status | Data Dir                     | Deploy Dir                     |
|--------------------|--------------|--------------|-------------|--------------|--------|------------------------------|--------------------------------|
| 172.16.6.157:9093  | alertmanager | 172.16.6.157 | 9093/9094   | linux/x86_64 | Up     | /tidb-data/alertmanager-9093 | /tidb-deploy/alertmanager-9093 |
| 172.16.6.157:3000  | grafana      | 172.16.6.157 | 3000        | linux/x86_64 | Up     | -                            | /tidb-deploy/grafana-3000      |
| 172.16.6.157:2379  | pd           | 172.16.6.157 | 2379/2380   | linux/x86_64 | Up     | /tidb-data/pd-2379           | /tidb-deploy/pd-2379           |
| 172.16.6.202:2379  | pd           | 172.16.6.202 | 2379/2380   | linux/x86_64 | Up L   | /tidb-data/pd-2379           | /tidb-deploy/pd-2379           |
| 172.16.6.210:2379  | pd           | 172.16.6.210 | 2379/2380   | linux/x86_64 | Up UI  | /tidb-data/pd-2379           | /tidb-deploy/pd-2379           |
| 172.16.6.157:9090  | prometheus   | 172.16.6.157 | 9090        | linux/x86_64 | Up     | /tidb-data/prometheus-9090   | /tidb-deploy/prometheus-9090   |
| 172.16.6.212:4000  | tidb         | 172.16.6.212 | 4000/10080  | linux/x86_64 | Up     | -                            | /tidb-deploy/tidb-4000         |
| 172.16.6.162:20160 | tikv         | 172.16.6.162 | 20160/20180 | linux/x86_64 | Up     | /tidb-data/tikv-20160        | /tidb-deploy/tikv-20160        |
| 172.16.6.163:20160 | tikv         | 172.16.6.163 | 20160/20180 | linux/x86_64 | Up     | /tidb-data/tikv-20160        | /tidb-deploy/tikv-20160        |
| 172.16.6.164:20160 | tikv         | 172.16.6.164 | 20160/20180 | linux/x86_64 | Up     | /tidb-data/tikv-20160        | /tidb-deploy/tikv-20160        |

Total nodes: 10

## Lesson 14 练习-2: 对集群进行改名

### 概述

在这个练习里，我们将集群名由 `tidb-test` 改为 `tidb-prod`，之后改回 `tidb-test`。

### 任务

1. 使用 `tiup` 进行集群名修改，如下所示：

```
[root@centos76_vm ~]# tiup cluster rename tidb-test tidb-prod
Found cluster newer version:

The latest version: v1.4.4
Local installed version: v1.4.3
Update current component: tiup update cluster
Update all components: tiup update --all

Starting component `cluster`: /root/.tiup/components/cluster/v1.4.3/tiup-cluster rename tidb-test tidb-prod
Rename cluster `tidb-test` -> `tidb-prod` successfully
+ [Serial] - SSHKeySet: privateKey=/root/.tiup/storage/cluster/clusters/tidb-prod/ssh/id_rsa,
publicKey=/root/.tiup/storage/cluster/clusters/tidb-prod/ssh/id_rsa.pub
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.157
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.157
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.163
... (省略中间过程)
- Refresh config blackbox_exporter -> 172.16.6.163 ... Done
- Refresh config blackbox_exporter -> 172.16.6.164 ... Done
- Refresh config blackbox_exporter -> 172.16.6.212 ... Done
+ [Serial] - UpgradeCluster
Upgrading component prometheus
 Restarting instance 172.16.6.157
 Restart 172.16.6.157 success
Upgrading component grafana
 Restarting instance 172.16.6.157
 Restart 172.16.6.157 success
Reloaded cluster `tidb-prod` successfully
```

## Lesson 14 练习-2: 对集群进行改名

2. 查看集群状态，发现集群名已经改为 tidb-prod，如下：

```
[root@centos76_vm ~]# tiup cluster display tidb-prod
```

Found cluster newer version:

The latest version: v1.4.4  
 Local installed version: v1.4.3  
 Update current component: tiup update cluster  
 Update all components: tiup update --all

Starting component `cluster`: /root/.tiup/components/cluster/v1.4.3/tiup-cluster display tidb-prod

Cluster type: tidb  
 Cluster name: tidb-prod  
 Cluster version: v5.0.0  
 SSH type: builtin  
 Dashboard URL: http://172.16.6.210:2379/dashboard

| ID                 | Role         | Host         | Ports       | OS/Arch      | Status | Data Dir                     | Deploy Dir                     |
|--------------------|--------------|--------------|-------------|--------------|--------|------------------------------|--------------------------------|
| 172.16.6.157:9093  | alertmanager | 172.16.6.157 | 9093/9094   | linux/x86_64 | Up     | /tidb-data/alertmanager-9093 | /tidb-deploy/alertmanager-9093 |
| 172.16.6.157:3000  | grafana      | 172.16.6.157 | 3000        | linux/x86_64 | Up     | -                            | /tidb-deploy/grafana-3000      |
| 172.16.6.157:2379  | pd           | 172.16.6.157 | 2379/2380   | linux/x86_64 | Up     | /tidb-data/pd-2379           | /tidb-deploy/pd-2379           |
| 172.16.6.202:2379  | pd           | 172.16.6.202 | 2379/2380   | linux/x86_64 | Up L   | /tidb-data/pd-2379           | /tidb-deploy/pd-2379           |
| 172.16.6.210:2379  | pd           | 172.16.6.210 | 2379/2380   | linux/x86_64 | Up UI  | /tidb-data/pd-2379           | /tidb-deploy/pd-2379           |
| 172.16.6.157:9090  | prometheus   | 172.16.6.157 | 9090        | linux/x86_64 | Up     | /tidb-data/prometheus-9090   | /tidb-deploy/prometheus-9090   |
| 172.16.6.212:4000  | tidb         | 172.16.6.212 | 4000/10080  | linux/x86_64 | Up     | -                            | /tidb-deploy/tidb-4000         |
| 172.16.6.162:20160 | tikv         | 172.16.6.162 | 20160/20180 | linux/x86_64 | Up     | /tidb-data/tikv-20160        | /tidb-deploy/tikv-20160        |
| 172.16.6.163:20160 | tikv         | 172.16.6.163 | 20160/20180 | linux/x86_64 | Up     | /tidb-data/tikv-20160        | /tidb-deploy/tikv-20160        |
| 172.16.6.164:20160 | tikv         | 172.16.6.164 | 20160/20180 | linux/x86_64 | Up     | /tidb-data/tikv-20160        | /tidb-deploy/tikv-20160        |

Total nodes: 10

## Lesson 14 练习-2: 对集群进行改名

3. 使用 tiup 进行集群名修改, 改为 tidb-test, 如下所示:

```
[root@centos76_vm ~]# tiup cluster rename tidb-prod tidb-test
```

Found cluster newer version:

The latest version: v1.4.4

Local installed version: v1.4.3

Update current component: tiup update cluster

Update all components: tiup update --all

Starting component `cluster`: /root/.tiup/components/cluster/v1.4.3/tiup-cluster rename tidb-prod tidb-test

Rename cluster `tidb-prod` -> `tidb-test` successfully

+ [ Serial ] - SSHKeySet: privateKey=/root/.tiup/storage/cluster/clusters/tidb-test/ssh/id\_rsa,  
publicKey=/root/.tiup/storage/cluster/clusters/tidb-test/ssh/id\_rsa.pub

+ [ Parallel ] - UserSSH: user=tidb, host=172.16.6.202

+ [ Parallel ] - UserSSH: user=tidb, host=172.16.6.212

+ [ Parallel ] - UserSSH: user=tidb, host=172.16.6.157

... (省略中间内容)

- Refresh config blackbox\_exporter -> 172.16.6.162 ... Done

- Refresh config blackbox\_exporter -> 172.16.6.163 ... Done

- Refresh config blackbox\_exporter -> 172.16.6.164 ... Done

+ [ Serial ] - UpgradeCluster

Upgrading component prometheus

Restarting instance 172.16.6.157

Restart 172.16.6.157 success

Upgrading component grafana

Restarting instance 172.16.6.157

Restart 172.16.6.157 success

Reloaded cluster `tidb-test` successfully

## Lesson 14 练习-2: 对集群进行改名

### 4. 查看集群状态，发现集群名已经改为 tidb-test，如下所示：

```
[root@centos76_vm ~]# tiup cluster display tidb-test
Found cluster newer version:

The latest version: v1.4.4
Local installed version: v1.4.3
Update current component: tiup update cluster
Update all components: tiup update --all

Starting component `cluster`: /root/.tiup/components/cluster/v1.4.3/tiup-cluster display tidb-test
Cluster type: tidb
Cluster name: tidb-test
Cluster version: v5.0.0
SSH type: builtin
Dashboard URL: http://172.16.6.210:2379/dashboard

ID Role Host Ports OS/Arch Status Data Dir Deploy Dir
-- -
172.16.6.157:9093 alertmanager 172.16.6.157 9093/9094 linux/x86_64 Up /tidb-data/alertmanager-9093 /tidb-deploy/alertmanager-9093
172.16.6.157:3000 grafana 172.16.6.157 3000 linux/x86_64 Up - /tidb-deploy/grafana-3000
172.16.6.157:2379 pd 172.16.6.157 2379/2380 linux/x86_64 Up /tidb-data/pd-2379 /tidb-deploy/pd-2379
172.16.6.202:2379 pd 172.16.6.202 2379/2380 linux/x86_64 Up|L /tidb-data/pd-2379 /tidb-deploy/pd-2379
172.16.6.210:2379 pd 172.16.6.210 2379/2380 linux/x86_64 Up|UI /tidb-data/pd-2379 /tidb-deploy/pd-2379
172.16.6.157:9090 prometheus 172.16.6.157 9090 linux/x86_64 Up /tidb-data/prometheus-9090 /tidb-deploy/prometheus-9090
172.16.6.212:4000 tidb 172.16.6.212 4000/10080 linux/x86_64 Up - /tidb-deploy/tidb-4000
172.16.6.162:20160 tikv 172.16.6.162 20160/20180 linux/x86_64 Up /tidb-data/tikv-20160 /tidb-deploy/tikv-20160
172.16.6.163:20160 tikv 172.16.6.163 20160/20180 linux/x86_64 Up /tidb-data/tikv-20160 /tidb-deploy/tikv-20160
172.16.6.164:20160 tikv 172.16.6.164 20160/20180 linux/x86_64 Up /tidb-data/tikv-20160 /tidb-deploy/tikv-20160
Total nodes: 10
```

### 4. 退出窗口，实验结束。

## Lesson 15 练习: TiDB Cluster 的升级



## Lesson 15 练习：概述

---

### 概述

在本练习中,我们将已经部署好的 TiDB 数据库 v4.0 版本集群升级到 v5.0 版本,并且采用不停库的升级方法。

### 实验环境要求

1. 已经部署好的 TiDB 数据库 v4.0 集群。
2. 中控机安装部署好 TiUP 组件。
3. 可以连接到外网。

learn.pingcap.cc

## Lesson 15 练习-1: 将 TiDB 集群从 v4.0 升级到 v5.0

### 概述

在本练习中, 我们将已经部署好的 TiDB 数据库 v4.0 版本集群升级到 v5.0 版本, 并且采用不停库的升级方法。

### 任务

12. 升级前, 请您检查现有的 TiDB 数据库集群, 版本为 v4.0:

```
[root@centos76_vm ~]# tiup cluster display tidb-test
Starting component `cluster`: /root/.tiup/components/cluster/v1.4.4/tiup-cluster display tidb-test
Cluster type: tidb
Cluster name: tidb-test
Cluster version: v4.0.0
SSH type: builtin
Dashboard URL: http://172.16.6.157:2379/dashboard
```

| ID                 | Role         | Host         | Ports       | OS/Arch      | Status | Data Dir                     | Deploy Dir                     |
|--------------------|--------------|--------------|-------------|--------------|--------|------------------------------|--------------------------------|
| 172.16.6.157:9093  | alertmanager | 172.16.6.157 | 9093/9094   | linux/x86_64 | Up     | /tidb-data/alertmanager-9093 | /tidb-deploy/alertmanager-9093 |
| 172.16.6.157:3000  | grafana      | 172.16.6.157 | 3000        | linux/x86_64 | Up     | -                            | /tidb-deploy/grafana-3000      |
| 172.16.6.157:2379  | pd           | 172.16.6.157 | 2379/2380   | linux/x86_64 | Up UI  | /tidb-data/pd-2379           | /tidb-deploy/pd-2379           |
| 172.16.6.202:2379  | pd           | 172.16.6.202 | 2379/2380   | linux/x86_64 | Up L   | /tidb-data/pd-2379           | /tidb-deploy/pd-2379           |
| 172.16.6.210:2379  | pd           | 172.16.6.210 | 2379/2380   | linux/x86_64 | Up     | /tidb-data/pd-2379           | /tidb-deploy/pd-2379           |
| 172.16.6.157:9090  | prometheus   | 172.16.6.157 | 9090        | linux/x86_64 | Up     | /tidb-data/prometheus-9090   | /tidb-deploy/prometheus-9090   |
| 172.16.6.212:4000  | tidb         | 172.16.6.212 | 4000/10080  | linux/x86_64 | Up     | -                            | /tidb-deploy/tidb-4000         |
| 172.16.6.162:20160 | tikv         | 172.16.6.162 | 20160/20180 | linux/x86_64 | Up     | /tidb-data/tikv-20160        | /tidb-deploy/tikv-20160        |
| 172.16.6.163:20160 | tikv         | 172.16.6.163 | 20160/20180 | linux/x86_64 | Up     | /tidb-data/tikv-20160        | /tidb-deploy/tikv-20160        |
| 172.16.6.164:20160 | tikv         | 172.16.6.164 | 20160/20180 | linux/x86_64 | Up     | /tidb-data/tikv-20160        | /tidb-deploy/tikv-20160        |

```
Total nodes: 10
```

## Lesson 15 练习-1: 将 TiDB 集群从 v4.0 升级到 v5.0

2. 使用 TiUP 组件升级 TiDB 集群到 v5.0, 此过程不停数据库, 如下

```
[root@centos76_vm ~]# tiup cluster upgrade tidb-test v5.0.0
Starting component `cluster`: /root/.tiup/components/cluster/v1.4.4/tiup-cluster upgrade tidb-test v5.0.0
This operation will upgrade tidb v4.0.0 cluster tidb-test to v5.0.0.
Do you want to continue? [y/N]:(default=N) y
Upgrading cluster...
+ [Serial] - SSHKeySet: privateKey=/root/.tiup/storage/cluster/clusters/tidb-test/ssh/id_rsa,
publicKey=/root/.tiup/storage/cluster/clusters/tidb-test/ssh/id_rsa.pub
... (省略中间内容)
Upgraded cluster `tidb-test` successfully
```

3. 升级后, 请您检查现有的 TiDB 数据库集群, 版本为 v5.0:

```
[root@centos76_vm ~]# tiup cluster display tidb-test
Starting component `cluster`: /root/.tiup/components/cluster/v1.4.4/tiup-cluster display tidb-test
Cluster type: tidb
Cluster name: tidb-test
Cluster version: v5.0.0
SSH type: builtin
Dashboard URL: http://172.16.6.157:2379/dashboard
ID Role Host Ports OS/Arch Status Data Dir Deploy Dir
--
172.16.6.157:9093 alertmanager 172.16.6.157 9093/9094 linux/x86_64 Up /tidb-data/alertmanager-9093 /tidb-deploy/alertmanager-9093
172.16.6.157:3000 grafana 172.16.6.157 3000 linux/x86_64 Up - /tidb-deploy/grafana-3000
172.16.6.157:2379 pd 172.16.6.157 2379/2380 linux/x86_64 Up|UI /tidb-data/pd-2379 /tidb-deploy/pd-2379
172.16.6.202:2379 pd 172.16.6.202 2379/2380 linux/x86_64 Up|L /tidb-data/pd-2379 /tidb-deploy/pd-2379
172.16.6.210:2379 pd 172.16.6.210 2379/2380 linux/x86_64 Up /tidb-data/pd-2379 /tidb-deploy/pd-2379
172.16.6.157:9090 prometheus 172.16.6.157 9090 linux/x86_64 Up /tidb-data/prometheus-9090 /tidb-deploy/prometheus-9090
172.16.6.212:4000 tidb 172.16.6.212 4000/10080 linux/x86_64 Up - /tidb-deploy/tidb-4000
172.16.6.162:20160 tikv 172.16.6.162 20160/20180 linux/x86_64 Up /tidb-data/tikv-20160 /tidb-deploy/tikv-20160
172.16.6.163:20160 tikv 172.16.6.163 20160/20180 linux/x86_64 Up /tidb-data/tikv-20160 /tidb-deploy/tikv-20160
172.16.6.164:20160 tikv 172.16.6.164 20160/20180 linux/x86_64 Up /tidb-data/tikv-20160 /tidb-deploy/tikv-20160
Total nodes: 10
```

4. 升级完毕, 退出所有窗口。

## Lesson 17 练习：使用备份恢复 工具 BR 进行备份恢复

## Lesson 17 练习：概述

---

### 概述：

本课练习中，我们将部署备份恢复工具 BR，之后进行 TiDB 数据库的全库备份，单库备份恢复和单表备份恢复。

### 实验环境要求：

1. 注意，您的实验环境（包括 IP，端口号，用户名，密码和目录）可能和手册不同。
2. 可以连接到实验用虚拟机。
3. TiDB 数据库之前部署完毕。
4. 清理掉之前的实验数据。

learn.pingcap.cc

## Lesson 17 练习-1: 部署备份恢复工具 BR

### 概述

在这个练习中,我们将部署备份恢复工具 BR,以为后面的 2 个练习使用。根据 BR 的最佳实践,我们最好将 BR 工具部署到 PD 节点上,这里我们选择 1 个 PD 节点进行 BR 的部署。

### 任务

13. 备份恢复工具 BR 集成在 tidb-toolkit 中,先进行下载:

```
[root@centos76_vm ~]# wget https://download.pingcap.org/tidb-toolkit-v5.0.1-linux-amd64.tar.gz
--2021-05-28 14:31:08-- https://download.pingcap.org/tidb-toolkit-v5.0.1-linux-amd64.tar.gz
Resolving download.pingcap.org (download.pingcap.org)... 124.239.158.230, 124.239.158.231, 124.239.158.232, ...
Connecting to download.pingcap.org (download.pingcap.org)|124.239.158.230|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 220298684 (210M) [application/x-compressed]
Saving to: 'tidb-toolkit-v5.0.1-linux-amd64.tar.gz.5'

100%[=====>] 220,298,684 15.8MB/s in 14s

2021-05-28 14:31:22 (14.7 MB/s) - 'tidb-toolkit-v5.0.1-linux-amd64.tar.gz.5' saved [220298684/220298684]
```

2. 解压缩下载软件包 tidb-toolkit-v5.0.1-linux-amd64.tar.gz:

```
[root@centos76_vm ~]# tar xvf tidb-toolkit-v5.0.1-linux-amd64.tar.gz
tidb-toolkit-v5.0.1-linux-amd64/
tidb-toolkit-v5.0.1-linux-amd64/bin/
tidb-toolkit-v5.0.1-linux-amd64/bin/tikv-importer
tidb-toolkit-v5.0.1-linux-amd64/bin/dumpling
tidb-toolkit-v5.0.1-linux-amd64/bin/br
tidb-toolkit-v5.0.1-linux-amd64/bin/tidb-lightning
tidb-toolkit-v5.0.1-linux-amd64/bin/mydumper
tidb-toolkit-v5.0.1-linux-amd64/bin/tidb-lightning-ctl
tidb-toolkit-v5.0.1-linux-amd64/bin/pd-tso-bench
tidb-toolkit-v5.0.1-linux-amd64/bin/sync_diff_inspector
```

3. 进入目录, 确认安装完毕。也可以将目录加入环境变量, 方便后续执行:

```
[root@centos76_vm ~]# cd tidb-toolkit-v5.0.1-linux-amd64/bin/
[root@centos76_vm bin]# ls
br nohup.out tidb-lightning tidb-lightning.toml
dumpling pd-tso-bench tidb-lightning-ctl tikv-importer
mydumper sync_diff_inspector tidb-lightning.log
```

## Lesson 17 练习-2: 使用 BR 进行全备份

### 概述

在本练习中，我们将使用备份恢复工具 BR 进行全库备份。

### 任务

1. 在所有的 TiKV 节点创建文件夹 `/tmp/backup`，用来存储本节点的备份文件(SST 文件) 并将文件夹的权限设置为可以读写。  
首先登录到 TiKV 节点，之后执行：

```
[root@copy-of-vm-ee-centos76-v1 ~]# mkdir /tmp/backup
[root@copy-of-vm-ee-centos76-v1 ~]# chmod 777 /tmp/backup
```

(循环在每一个 TiKV 节点进行操作。)

14. 连接到其中一个 PD 节点，进入到 `tidb-toolkit-v5.0.1-linux-amd64/bin/` 目录，开始进行数据库全备份：

```
[root@centos76_vm bin]# ./br backup full
--pd "172.16.6.202:2379" --storage "local:///tmp/backup"
--ratelimit 120 --log-file backupfull.log
Detail BR log in backupfull.log
Full backup <-----> 100.00%
Checksum <-----> 100.00%
[2021/05/28 14:51:45.439 +08:00] [INFO] [collector.go:62] ["Full backup success summary"] [total-ranges=26] [ranges-succeed=26]
[ranges-failed=0] [backup-checksum=1.63704543s] [backup-fast-checksum=3.739175ms] [backup-total-regions=31] [Size=82188247]
[BackupTS=425245984089440258] [total-take=12.678373151s] [data-size=428.5MB] [average-speed=39.04MB/s] [total-kv=7881232]
```

(对于参数解释如下：

- pd "172.16.6.202:2379" :连接 TiDB 数据库的 PD 节点,最好在 PD 节点上执行,即连接本节点。
- storage "local:///tmp/backup" :备份文件存储在 TiKV 节点上的位置。
- ratelimit 120 :对于备份所用存储带宽限速,以免影响线上业务。
- log-file backupfull.log : 备份日志文件。)

## Lesson 17 练习-2: 使用 BR 进行全备份

---

3. 备份结束后,可以到各个 TiKV 节点检查备份文件:

```
[root@copy-of-vm-ee-centos76-v1]# cd /tmp/backup/
[root@copy-of-vm-ee-centos76-v1 backup]# ls
5_4013_78_17850c0a0f924f1990ae86a7d32da70c17cb2d87c8178dfa50c304a8dc9aff9a_1622100885221_write.sst
5_4013_78_2507d4fa541d4d0e2c4f116adf2221c43b0481468e13e96e2816f0505e7fd6e1_1622100885695_write.sst
5_4013_78_2bbcd667caad45265d1c33dd3f497ba21482fb0c8f16b7324696478751f5eed3_1622100885205_write.sst
... (省略中间内容)
5_5301_340_6186f6c12aa634800c5e9e9a25da7fdf3416eb5a2f149b230279ac423029a287_1622184699319_write.sst
5_5329_340_24640d875d278feaf0907734395a8dd7c4d71a0705ddf87a497448c9ef046e7_1622184696097_write.sst
5_5333_340_2f4efc450a70a61427332192e0bc6e47309bd0f2d644be59b5f1bc6d6ba892c6_1622184697693_write.sst
```

4. 同时,在执行备份的节点上,也会自动创建文件夹 /tmp/backup 用来存储元数据和锁信息:

```
[root@centos76_vm bin]# cd /tmp/backup/
[root@centos76_vm backup]# ls
backup.lock backupmeta
```



## Lesson 17 练习-3: 使用 BR 进行单库备份与恢复

### 概述

在本练习中，您将使用备份恢复工具 BR 进行单库employees 的备份和恢复。

### 任务

1. 在所有的 TiKV 节点创建文件夹 /tmp/employeesbk, 用来存储本节点的备份文件 (SST 文件), 并将文件夹的权限设置为可以读写。

首先登录到 TiKV 节点, 之后执行:

```
[root@copy-of-vm-ee-centos76-v1 ~]# mkdir /tmp/employeesbk
[root@copy-of-vm-ee-centos76-v1 ~]# chmod 777 /tmp/employeesbk
```

2. 连接到其中一个 PD 节点, 进入到 tidb-toolkit-v5.0.1-linux-amd64/bin/ 目录, 开始进行单库 employees 的备份:

```
[root@centos76_vm ~]# cd tidb-toolkit-v5.0.1-linux-amd64/bin/
[root@centos76_vm bin]# ./br backup db --pd "172.16.6.202:2379"
--db employees --storage "local:///tmp/employeesbk" --ratelimit
120 --log-file backupdb.log

Detail BR log in backupdb.log
Database backup <-----> 100.00%
Checksum <-----> 100.00%
[2021/05/27 18:56:11.949 +08:00] [INFO] [collector.go:62] ["Database backup success summary"] [total-ranges=14] [ranges-succeed=14] [ranges-failed=0] [backup-checksum=1.16283122s] [backup-fast-checksum=2.269924ms] [backup-total-regions=22] [BackupTS=425227180395462663] [Size=81998918] [total-take=9.595068759s] [total-kv=7869642] [data-size=427.9MB] [average-speed=51.01MB/s]
```

(对于参数解释如下:

--pd "172.16.6.202:2379" :连接 TiDB 数据库的 PD 节点,最好在 PD 节点上执行,即连接本节点。

--storage "local:///tmp/employeesbk" :备份文件存储在 TiKV 节点上的位置。

--db employees : 备份 employees 库下面所有的表。

--ratelimit 120 :对于备份所用存储带宽限速,以免影响线上业务。

--log-file backupdb.log : 备份日志文件。)

## Lesson 17 练习-3: 使用 BR 进行单库备份与恢复

3. 接下来, 登录到 `tidb` 上, 将刚刚备份的 `employees` 数据库删除, 如下:  
首先查看 `employees` 数据库: (请根据自己实验环境输入密码)

```
[root@centos76_vm ~]# mysql -h172.16.6.212 -P4000 -uroot -p
```

Enter password:

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 139

Server version: 5.7.25-TiDB-v5.0.0 TiDB Server (Apache License 2.0) Community Edition, MySQL 5.7 compatible

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

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> show databases;
```

| Database           |
|--------------------|
| INFORMATION_SCHEMA |
| METRICS_SCHEMA     |
| PERFORMANCE_SCHEMA |
| employees          |
| mysql              |
| test               |
| tidb               |
| world              |

8 rows in set (0.00 sec)

删除数据库 `employees`:

```
mysql> drop database employees;
```

Query OK, 0 rows affected (0.52 sec)

## Lesson 17 练习-3: 使用 BR 进行单库备份与恢复

验证数据库 employees 已经被删除:

```
mysql> show databases;
```

| Database           |
|--------------------|
| INFORMATION_SCHEMA |
| METRICS_SCHEMA     |
| PERFORMANCE_SCHEMA |
| mysql              |
| test               |
| tidb               |
| world              |

7 rows in set (0.01 sec)

3. 接下来准备进行恢复, 首先, 将所有 TiKV 目录 `/tmp/employees` 下面的备份文件相拷贝到其他节点, 保证在所有的 TiKV 节点上都有所有的备份文件。如下:

第一步: 登录到 TiKV 节点, 进入目录 `cd /tmp/employeesbk/`:

```
[root@copy-of-vm-ee-centos76-v1 ~]# cd /tmp/employeesbk/
```

第二步: 将备份文件拷贝到其他节点:

```
[root@copy-of-vm-ee-centos76-v1 employeesbk]# scp *
172.16.6.164:/tmp/employeesbk
1_5021_266_44dff3146a23233c6aac8442b4b1700a7 100% 1849 2.0MB/s 00:00
1_5021_266_46c92973dd98c6b2b3fcb1a026633931a7 100% 1773 2.7MB/s 00:00
1_5021_266_ff8db3d34bd055d01c43a187846f5cf469 100% 1818 2.4MB/s 00:00
..... (省略中间过程)
5_5045_274_e1c74313c0a5c523a91f9ab5145e8bb324 100% 12MB 108.8MB/s 00:00
5_5053_274_96ce66db8d907f185b5446590881c822df 100% 11MB 112.4MB/s 00:00
5_5061_278_581ca997d499845ea7c1ccd902aca85400 100% 6876KB 101.3MB/s 00:00
5_5069_278_5f37fc845fa254efcc676081d3ecb2ca68 100% 6145KB 97.2MB/s 00:00
```

第三步: 循环操作, 更换不同节点, 使所有节点都有所有的备份文件。

## Lesson 17 练习-3: 使用 BR 进行单库备份与恢复

### 4. 开始进行恢复:

```
[root@centos76_vm ~]# cd tidb-toolkit-v5.0.1-linux-amd64/bin/
[root@centos76_vm bin]# ./br restore db --pd "172.16.6.202:2379" --db
"employees" --storage "local:///tmp/employeesbk" --log-file
restoredb.log
Detail BR log in restoredb.log
Database restore <-----> 100.00%
[2021/05/28 16:53:15.550 +08:00] [INFO] [collector.go:62] ["Database restore success summary"] [total-ranges=24] [ranges-
succeed=24] [ranges-failed=0] [split-region=317.636984ms] [restore-checksum=5.257332678s] [restore-ranges=18] [Size=81998918]
[total-take=14.878057135s] [total-kv=7869642] [data-size=427.9MB] [average-speed=27.36MB/s]
```

(对于参数解释如下:

--pd "172.16.6.202:2379" :连接 TiDB 数据库的 PD 节点,最好在 PD 节点上执行,即连接本  
节点。--storage "local:///tmp/employeesbk" :备份文件存储在 TiKV 节点上的位置。  
--log-file restoredb.log : 备份日志文件。)

### 5. 最后,验证 employees 库是否已经导入到 TiDB 数据库中:

```
mysql> show databases;
+-----+
| Database |
+-----+
| INFORMATION_SCHEMA |
| METRICS_SCHEMA |
| PERFORMANCE_SCHEMA |
| employees |
| mysql |
| test |
| tidb |
| world |
+-----+
8 rows in set (0.00 sec)

mysql> use employees;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
Database changed

mysql> show tables;
+-----+
| Tables_in_employees |
+-----+
| departments |
| dept_emp |
| dept_manager |
| employees |
| salaries |
| titles |
+-----+
6 rows in set (0.00 sec)

mysql> select count(*) from salaries;
+-----+
| count(*) |
+-----+
| 2844047 |
+-----+
1 row in set (0.92 sec)
```

## Lesson 17 练习-4: 使用 BR 进行单表备份与恢复

### 概述

在本练习中，您将使用备份恢复工具 BR 进行单表 salaries 的备份和恢复。

### 任务

1. 在所有的 TiKV 节点创建文件夹 /tmp/salariestab, 用来存储本节点的备份文件 (SST 文件), 并将文件夹的权限设置为可以读写。

首先登录到 TiKV 节点, 之后执行:

```
[root@copy-of-vm-ee-centos76-v1 ~]# mkdir /tmp/salariestab
[root@copy-of-vm-ee-centos76-v1 ~]# chmod 777 /tmp/salariestab
```

循环在每一个 TiKV 节点进行操作。

2. 连接到其中一个 PD 节点, 进入到 tidb-toolkit-v5.0.1-linux-amd64/bin/ 目录, 开始进行库 employees 下的表 salaries 的备份:

```
[root@centos76_vm ~]# cd tidb-toolkit-v5.0.1-linux-amd64/bin/
[root@centos76_vm bin]# ./br backup table --pd "172.16.6.202:2379"
--db employees --table salaries --storage "local:///tmp/salariestab"
--ratelimit 120 --log-file backuptable.log
Detail BR log in backuptable.log
Table backup <-----> 100.00%
Checksum <-----> 100.00%
[2021/05/27 20:21:09.918 +08:00] [INFO] [collector.go:62] ["Table backup success summary"] [total-ranges=2] [ranges-succeed=2]
[ranges-failed=0] [backup-checksum=1.135022468s] [backup-fast-checksum=388.953µs] [backup-total-regions=6]
[BackupTS=425228517880365058] [Size=58301077] [total-take=5.584693037s] [total-kv=5688094] [data-size=301MB] [average-
speed=69.73MB/s]
```

(对于参数解释如下:

--pd "172.16.6.202:2379" :连接 TiDB 数据库的 PD 节点,最好在 PD 节点上执行,即连接本节点。

--storage "local:///tmp/salariestab" :备份文件存储在 TiKV 节点上的位置。

--db employees --table salaries : 备份 employees 库下面的 salaries 表。

--ratelimit 120 :对于备份所用存储带宽限速,以免影响线上业务。

--log-file backupdb.log : 备份日志文件。)

## Lesson 17 练习-4: 使用 BR 进行单表备份与恢复

---

3. 登录到 tidb 上, 将刚刚备份的 employees 库下的表 salaries 删除。

首先查看 employees 库中的 salaries 表:

```
mysql> use employees
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_employees |
+-----+
| departments |
| dept_emp |
| dept_manager |
| employees |
| salaries |
| titles |
+-----+
6 rows in set (0.00 sec)

mysql> select count(*) from salaries;
+-----+
| count(*) |
+-----+
| 2844047 |
+-----+
1 row in set (0.01 sec)
```

## Lesson 17 练习-4: 使用 BR 进行单表备份与恢复

删除库 employees 下的表 salaries:

```
mysql> drop table salaries;
```

Query OK, 0 rows affected (1.01 sec)

验证库 employees 的表 salaries 已经被删除:

```
mysql> show tables;
```

| Tables_in_employees |
|---------------------|
| departments         |
| dept_emp            |
| dept_manager        |
| employees           |
| titles              |

5 rows in set (0.00 sec)

4. 接下来准备进行恢复, 首先, 将所有 TiKV 目录/tmp/salariestab 下面的备份文件相互拷贝到其他节点, 保证在所有的 TiKV 节点上都有所有的备份文件。如下:

第一步: 登录到 TiKV 节点, 进入目录 `cd /tmp/salariestab/`:

```
[root@copy-of-vm-ee-centos76-v1 ~] # cd /tmp/salariestab/
```

第二步: 将备份文件拷贝到其他节点:

```
[root@copy-of-vm-ee-centos76-v1 salariestab] # scp * 172.16.6.164:/tmp/salariestab
```

root@172.16.6.202's password:

1\_5197\_312\_cfa07537e6eadac915e62eb7b53c3f63fa35a60cf0c9 100% 6877KB 98.7MB/s 00:00

1\_5201\_312\_2f0fa07c6c6922fe65208e19beb1ee7d3a35ad2766e0 100% 6884KB 98.0MB/s 00:00

1\_5217\_312\_5e1307340bc1acb4be2f37904e0ab6ec89f4e9cc26d0 100% 13MB 107.9MB/s 00:00

4\_5221\_314\_fd1bf85f44979d625d13f47b31e229e374f997380f21 100% 11MB 102.0MB/s 00:00

5\_5205\_314\_9d85779a541b708d89983e650ee1f3119a008abd12d2 100% 6145KB 108.7MB/s 00:00

5\_5213\_312\_3c940f2a680dd03d645cfaf6622eb4e15d5d22525e26 100% 12MB 112.4MB/s 00:00

第三步: 循环操作, 连接其他节点, 使所有节点都有所有的备份文件。

## Lesson 17 练习-4: 使用 BR 进行单表备份与恢复

### 5. 开始进行恢复:

```
[root@copy-of-vm-ee-centos76-v1 salariestab]# ./br restore table --pd
"172.16.6.202:2379" --db "employees" --table "salaries"
--storage "local:///tmp/salariestab" --log-file restoreable.log
Detail BR log in restoreable.log
Table restore <-----> 100.00%
[2021/05/27 21:00:42.284 +08:00] [INFO] [collector.go:62] ["Table restore success summary"] [total-ranges=7] [ranges-succeed=7]
[ranges-failed=0] [split-region=145.519509ms] [restore-checksum=1.30030903s] [restore-ranges=6] [Size=58301077] [total-
take=10.828147528s] [total-kv=5688094] [data-size=301MB] [average-speed=35.57MB/s]
```

(对于参数解释如下:

- pd "172.16.6.202:2379" :连接 TiDB 数据库的 PD 节点,最好在 PD 节点上执行,即连接本节点。
- storage "local:///tmp/salariestab" :备份文件存储在 TiKV 节点上的位置。
- db employees --table salaries : 恢复 employees 库下面的 salaries 表。
- log-file backupdb.log : 备份日志文件。)

### 6. 最后,验证 employees 库是否已经导入到 TiDB 数据库中:

```
mysql> use employees;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_employees |
+-----+
| departments |
| dept_emp |
| dept_manager |
| employees |
| salaries |
| titles |
+-----+
6 rows in set (0.00 sec)

mysql> select count(*) from salaries;
+-----+
| count(*) |
+-----+
| 2844047 |
+-----+
1 row in set (0.58 sec)
```

### 7. 实验完毕, 关闭所有窗口。



## Lesson 18 练习: 数据导出工具 Dumpling

## Lesson 18 练习：概述

---

### 概述

在本课练习中，我们将部署数据导出工具 Duplicity，之后使用 Duplicity 从 TiDB 数据库和 MySQL 数据库中导出数据。

### 实验环境要求

1. 注意，您的实验环境（包括 IP，端口号，用户名，密码和目录）可能和手册不同。
2. 可以连接到实验用虚拟机。
3. TiDB 数据库之前部署完毕。
4. 清理掉之前的实验数据。

## Lesson 18 练习-1: 部署 Dumpling 数据导出工具

### 概述

在这个练习中, 我们将部署数据导出工具 Dumpling , 以为后面的 2 个练习使用。

### 任务

#### 15. Dumpling 工具集成在 tidb-toolkit 中, 先进行下载:

```
[root@centos76_vm ~]# wget https://download.pingcap.org/tidb-toolkit-v5.0.1-linux-amd64.tar.gz
--2021-05-27 16:10:13-- https://download.pingcap.org/tidb-toolkit-v5.0.1-linux-amd64.tar.gz
--2021-05-27 11:30:19-- Resolving download.pingcap.org (download.pingcap.org)... 42.81.118.41
Connecting to download.pingcap.org (download.pingcap.org)|42.81.118.41|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 220298684 (210M) [application/x-compressed]
Saving to: 'tidb-toolkit-v5.0.1-linux-amd64.tar.gz.2'

100%[=====>] 220,298,684 48.0MB/s in 4.3s

2021-05-27 16:10:18 (48.3 MB/s) - 'tidb-toolkit-v5.0.1-linux-amd64.tar.gz.2' saved [220298684/220298684]
```

#### 2. 解压缩下载软件包 tidb-toolkit-v5.0.1-linux-amd64.tar.gz:

```
[root@centos76_vm ~]# tar xvf tidb-toolkit-v5.0.1-linux-amd64.tar.gz

tidb-toolkit-v5.0.1-linux-amd64/
tidb-toolkit-v5.0.1-linux-amd64/bin/
tidb-toolkit-v5.0.1-linux-amd64/bin/tikv-importer
tidb-toolkit-v5.0.1-linux-amd64/bin/dumpling
tidb-toolkit-v5.0.1-linux-amd64/bin/br
tidb-toolkit-v5.0.1-linux-amd64/bin/tidb-lightning
tidb-toolkit-v5.0.1-linux-amd64/bin/mydumper
tidb-toolkit-v5.0.1-linux-amd64/bin/tidb-lightning-ctl
tidb-toolkit-v5.0.1-linux-amd64/bin/pd-tso-bench
tidb-toolkit-v5.0.1-linux-amd64/bin/sync_diff_inspector
```

#### 3. 进入目录, 确认安装完毕。也可以将目录加入环境变量, 方便后续执行:

```
[root@centos76_vm bin]# ls
br dumpling mydumper pd-tso-bench sync_diff_inspector tidb-lightning tidb-lightning-ctl tikv-importer
```

## Lesson 18 练习-2: 使用 Duplicating 从 TiDB 数据库中导出数据

### 概述

在这个练习中，我们将使用 Duplicating 工具导出 TiDB 数据库中的单张表和单个库。

### 任务

1. 连接到 TiDB 数据库, 里面有实验数据库 world, 在 world 库中有 city, country 和 countrylanguage 三张表:

```
mysql> show databases;
```

| Database           |
|--------------------|
| INFORMATION_SCHEMA |
| METRICS_SCHEMA     |
| PERFORMANCE_SCHEMA |
| mysql              |
| test               |
| tidb               |
| world              |

7 rows in set (0.00 sec)

```
mysql> use world;
```

Reading table information for completion of table and column names

You can turn off this feature to get a quicker startup with -A

Database changed

```
mysql> show tables;
```

| Tables_in_world |
|-----------------|
| city            |
| country         |
| countrylanguage |

3 rows in set (0.00 sec)

## Lesson 18 练习-2: 使用 Dumpling 从 TiDB 数据库中导出数据

### 2. 从 TiDB 集群中导出数据库 world 中的表 city:

```
[root@centos76_vm ~] # ./dumpling -uroot -ptidb -P4000
-h 172.16.6.212 --filetype sql -t 8 -o /tmp/city -r 200000
-F 256MiB -T world.city
Release version: v5.0.1
Git commit hash: 4cb115746bb658b6d1a12c0e49932bfd3a08afac
Git branch: heads/refs/tags/v5.0.1
Build timestamp: 2021-04-23 06:01:59Z
Go version: go version go1.13 linux/amd64

[2021/05/27 16:44:47.995 +08:00] [INFO] [versions.go:55] ["Welcome to dumpling"] ["Release Version"=v5.0.1] ["Git Commit Hash"=4cb115746bb658b6d1a12c0e49932bfd3a08afac] ["Git Branch"=heads/refs/tags/v5.0.1] ["Build timestamp"=2021-04-23 06:01:59"] ["Go Version"=go version go1.13 linux/amd64"]
... (省略中间内容)
[2021/05/27 16:44:48.471 +08:00] [INFO] [main.go:81] ["dump data successfully, dumpling will exit now"]
```

各个参数的解释如下:

- uroot -P4000 -h 172.16.6.212 :用户名为 root, 端口号为4000, 主机 IP 为 172.16.6.212;
- filetype sql :导出文件类型为 SQL 文件。
- t 8 :采用 8 线程同时导出。
- o /tmp/city :导出文件保存在 /tmp/city 中。
- r 200000 :每个导出文件最大容纳 200000 行数据。
- F 256MiB :每个导出文件最大 256 MiB。

## Lesson 18 练习-2: 使用 Duplicating 从 TiDB 数据库中导出数据

---

### 3. 进入导出目录, 查看导出文件:

```
[root@centos76_vm ~] # cd /tmp/city/
[root@centos76_vm city] # ls
metadata world.city.0000000010000.sql world.city-schema.sql world-schema-create.sql
```

### 4. world-schema-create.sql 为创建数据库脚本:

```
[root@centos76_vm city]# vi world-schema-create.sql
/*!40101 SET NAMES binary*/;
CREATE DATABASE `world` /*!40100 DEFAULT CHARACTER SET utf8mb4 */;
```

### 5. world.city-schema.sql 为建表脚本:

```
[root@centos76_vm city]# vi world.city-schema.sql
/*!40101 SET NAMES binary*/;
CREATE TABLE `city` (
 `ID` int(11) NOT NULL AUTO_INCREMENT,
 `Name` char(35) NOT NULL DEFAULT "",
 `CountryCode` char(3) NOT NULL DEFAULT "",
 `District` char(20) NOT NULL DEFAULT "",
 `Population` int(11) NOT NULL DEFAULT '0',
 PRIMARY KEY (`ID`) /*!T![clustered_index] CLUSTERED */,
 KEY `CountryCode` (`CountryCode`),
 CONSTRAINT `city_ibfk_1` FOREIGN KEY (`CountryCode`) REFERENCES `country` (`Code`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_bin AUTO_INCREMENT=4081;
```

### 6. world.city.0000000010000.sql 为数据插入脚本:

```
[root@centos76_vm city]# vi world.city.0000000010000.sql
/*!40101 SET NAMES binary*/;
INSERT INTO `city` VALUES
(1,'Kabul','AFG','Kabol',1780000),
..... (省略中间内容)
"world.city.0000000010000.sql" 4081L, 181247C
```

## Lesson 18 练习-2: 使用 Dumpling 从 TiDB 数据库中导出数据

---

### 7. metadata 为导出时的时间戳信息:

```
[root@centos76_vm city]# vi metadata
```

```
Started dump at: 2021-05-27 16:44:48
```

```
SHOW MASTER STATUS:
```

```
Log: tidb-binlog
```

```
Pos: 425225116181856262
```

```
GTID:
```

```
Finished dump at: 2021-05-27 16:44:48
```

### 8. 接下来, 从 TiDB 集群中导出数据库 world:

```
[root@centos76_vm ~] # ./dumpling -uroot -ptidb -P4000
-h 172.16.6.212 --filetype sql -t 8 -o /tmp/world -r 200000
-F 256MiB -B world
```

```
Release version: v5.0.1
```

```
Git commit hash: 4cb115746bb658b6d1a12c0e49932bfd3a08afac
```

```
Git branch: heads/refs/tags/v5.0.1
```

```
Build timestamp: 2021-04-23 06:01:59Z
```

```
Go version: go version go1.13 linux/amd64
```

```
[2021/05/27 17:28:54.688 +08:00] [INFO] [versions.go:55] ["Welcome to dumpling"] ["Release Version"=v5.0.1] ["Git Commit Hash"=4cb115746bb658b6d1a12c0e49932bfd3a08afac] ["Git Branch"=heads/refs/tags/v5.0.1] ["Build timestamp"=2021-04-23 06:01:59"] ["Go Version"=go version go1.13 linux/amd64"]
```

```
[2021/05/27 17:28:54.695 +08:00] [INFO] [config.go:599] ["detect server type"] [type=TiDB]
```

```
... (省略中间内容)
```

```
[2021/05/27 17:28:55.420 +08:00] [INFO] [collector.go:212] ["backup Success summary: total backup ranges: 10, total success: 10, total failed: 0, total take(backup time): 449.988329ms, total take(real time): 450.020557ms, total size(Byte): 245169, avg speed(Byte/s): 544834.13, total rows: 5302"]
```

```
[2021/05/27 17:28:55.421 +08:00] [INFO] [main.go:81] ["dump data successfully, dumpling will exit now"]
```

(参数说明请参考第 2 步, 其中 -B world 表示导出 world 数据库。)

## Lesson 18 练习-2: 使用 Dumpling 从 TiDB 数据库中导出数据

---

9. 查看导出目录 /tmp/world, 看到里面是各个表的 SQL 格式的导出文件:

```
[root@centos76_vm ~] # cd /tmp/world /
[root@centos76_vm world] # ls
metadata world.countrylanguage.0000000000000000.sql
world.city.0000000000000000.sql world.countrylanguage-schema.sql
world.city-schema.sql world.country-schema.sql
world.country.0000000000000000.sql world-schema-create.sql
```

learn.pingcap.cc



## Lesson 18 练习-3: 使用 Dumping 从 MySQL 数据库中导出数据

---

### 概述

在这个练习中，我们将使用 Dumping 工具导出 MySQL 数据库中的单张表和单个库。

### 任务

1. 进入 3306 端口处的 MySQL 实例, 其中有 employees 数据库:  
(请根据自己实验环境输入密码)

```
[root@centos76_vm ~] # mysql -uroot -pmysql -h127.0.0.1 -P3306
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 38
Server version: 5.7.34 MySQL Community Server (GPL)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

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> show databases;
+-----+
| Database |
+-----+
| information_schema |
| employees |
| mysql |
| performance_schema |
| sakila |
| sys |
| world |
+-----+
7 rows in set (0.00 sec)
```

## Lesson 18 练习-3: 使用 Duplicating 从 MySQL 数据库中导出数据

```
mysql> use employees;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_employees |
+-----+
| current_dept_emp |
| departments |
| dept_emp |
| dept_emp_latest_date |
| dept_manager |
| employees |
| salaries |
| titles |
+-----+
8 rows in set (0.00 sec)
```

2. 进入bin文件内使用 Duplicating 工具将 3306 端口的 MySQL 数据库的 employees 库导出:

```
[root@centos76_vm ~] # cd tidb-toolkit-v5.0.1-linux-amd64/bin/
[root@centos76_vm bin]# ./duplicating -uroot -P3306 -h '127.0.0.1'
-p'Pingcap!@3' --filetype sql -t 8 -o /tmp/employees -r 200000
-F 256MiB -B employees

Release version: v5.0.1
Git commit hash: 4cb115746bb658b6d1a12c0e49932bfd3a08afac
Git branch: heads/refs/tags/v5.0.1
Build timestamp: 2021-04-23 06:01:59Z
Go version: go version go1.13 linux/amd64

[2021/05/28 10:10:28.252 +08:00] [INFO] [versions.go:55] ["Welcome to duplicating"] ["Release Version"=v5.0.1] ["Git Commit Hash"=4cb115746bb658b6d1a12c0e49932bfd3a08afac] ["Git Branch"=heads/refs/tags/v5.0.1] ["Build timestamp"=2021-04-23 06:01:59"] ["Go Version"=go version go1.13 linux/amd64"]
... (省略中间内容)
backup ranges: 27, total success: 27, total failed: 0, total take(backup time): 3.954136822s, total take(real time): 3.954200379s, total size(MB): 164.31, avg speed(MB/s): 41.55, total rows: 3919015"]
[2021/05/28 10:10:32.240 +08:00] [INFO] [main.go:81] ["dump data successfully, duplicating will exit now"]
```

## Lesson 18 练习-3: 使用 Dumping 从 MySQL 数据库中导出数据

---

### 3. 查看导入文件:

```
[root@centos76_vm ~] # cd /tmp/employees/
[root@centos76_vm employees]# ls
employees.departments.00000000000000.sql employees.salaries.0000000060000.sql
employees.departments-schema.sql employees.salaries.0000000070000.sql
employees.dept_emp.00000000000000.sql employees.salaries.0000000080000.sql
employees.dept_emp-schema.sql employees.salaries.0000000110000.sql
employees.dept_manager.00000000000000.sql employees.salaries.0000000120000.sql
employees.dept_manager-schema.sql employees.salaries.0000000130000.sql
employees.employees.00000000000000.sql employees.salaries-schema.sql
employees.employees-schema.sql employees-schema-create.sql
employees.salaries.00000000000000.sql employees.titles.00000000000000.sql
employees.salaries.0000000010000.sql employees.titles.0000000010000.sql
employees.salaries.0000000020000.sql employees.titles-schema.sql
employees.salaries.0000000050000.sql metadata
```

## Lesson 19 练习: 数据导入工具 TiDB Lightning

## Lesson 19 练习：概述

---

### 概述

本课练习中，您将使用 TiDB Lightning 来向 TiDB 数据库中导入数据。

### 实验环境要求

1. 注意，您的实验环境（包括 IP，端口号，用户名，密码和目录）可能和手册不同。
2. 可以连接到实验用虚拟机。
3. TiDB 数据库之前部署完毕。
4. 清理掉之前的实验数据。

learn.pingcap.cc

## Lesson 19 练习-1: 部署 TiDB Lightning 工具

---

### 概述

TiDB Lightning 的部署方式和 Duplicating 的部署方式类似, 如果学习过 Duplicating 部署方式则请忽略此步骤。

### 任务

16. TiDB Lightning 工具集成在 tidb-toolkit 中, 先进行下载:

```
[root@centos76_vm ~]# wget https://download.pingcap.org/tidb-toolkit-v5.0.1-linux-amd64.tar.gz
--2021-05-27 17:39:46-- https://download.pingcap.org/tidb-toolkit-v5.0.1-linux-amd64.tar.gz
Resolving download.pingcap.org (download.pingcap.org)... 42.81.118.41
Connecting to download.pingcap.org (download.pingcap.org)|42.81.118.41|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 220298684 (210M) [application/x-compressed]
Saving to: 'tidb-toolkit-v5.0.1-linux-amd64.tar.gz.4'

100%[=====>] 220,298,684 31.6MB/s in 5.1s

2021-05-27 17:39:51 (41.2 MB/s) - 'tidb-toolkit-v5.0.1-linux-amd64.tar.gz.4' saved [220298684/220298684]
```

2. 解压缩下载软件包 tidb-toolkit-v5.0.1-linux-amd64.tar.gz:

```
[root@centos76_vm ~]# tar xvf tidb-toolkit-v5.0.1-linux-amd64.tar.gz
tidb-toolkit-v5.0.1-linux-amd64/
tidb-toolkit-v5.0.1-linux-amd64/bin/
tidb-toolkit-v5.0.1-linux-amd64/bin/tikv-importer
tidb-toolkit-v5.0.1-linux-amd64/bin/dumpling
tidb-toolkit-v5.0.1-linux-amd64/bin/br
tidb-toolkit-v5.0.1-linux-amd64/bin/tidb-lightning
tidb-toolkit-v5.0.1-linux-amd64/bin/mydumper
tidb-toolkit-v5.0.1-linux-amd64/bin/tidb-lightning-ctl
tidb-toolkit-v5.0.1-linux-amd64/bin/pd-tso-bench
tidb-toolkit-v5.0.1-linux-amd64/bin/sync_diff_inspector
```

## Lesson 19 练习-1: 部署 TiDB Lightning 工具

---

3. 进入目录, 确认安装完毕。也可以将目录加入环境变量, 方便后续执行:

```
[root@centos76_vm ~]# cd tidb-toolkit-v5.0.1-linux-amd64/bin/
[root@centos76_vm bin]# ls
br nohup.out tidb-lightning tidb-lightning.toml
dumping pd-tso-bench tidb-lightning-ctl tikv-importer
mydumper sync_diff_inspector tidb-lightning.log
```

learn.pingcap.cc

## Lesson 19 练习-2: 使用 TiDB Lightning 工具将数据导入到 TiDB 数据库

---

### 概述

在数据导出工具 Duplicating 的练习三中, 我们使用 Duplicating 工具将 MySQL 数据库的 employees 数据导出在实验环境的 /tmp/employees 目录中, 这里我们将使用 TiDB Lightning 工具将其导入到 TiDB 数据库中。

注意: 如果 /tmp/employees 目录中没有数据, 请参考数据导出工具 Duplicating 的练习三中的步骤进行数据导出。

### 任务

#### 1. 检查待导入数据:

```
[root@centos76_vm ~]# cd /tmp/employees/
[root@centos76_vm employees]# ls
employees.departments.0000000000000000.sql employees.salaries.00000000600000.sql
employees.departments-schema.sql employees.salaries.00000000700000.sql
employees.dept_emp.0000000000000000.sql employees.salaries.00000000800000.sql
employees.dept_emp-schema.sql employees.salaries.0000000011000000.sql
employees.dept_manager.0000000000000000.sql employees.salaries.0000000012000000.sql
employees.dept_manager-schema.sql employees.salaries.0000000013000000.sql
employees.employees.0000000000000000.sql employees.salaries-schema.sql
employees.employees-schema.sql employees-schema-create.sql
employees.salaries.0000000000000000.sql employees.titles.0000000000000000.sql
employees.salaries.0000000010000000.sql employees.titles.0000000010000000.sql
employees.salaries.0000000020000000.sql employees.titles-schema.sql
employees.salaries.0000000050000000.sql metadata
```



## Lesson 19 练习-2: 使用 TiDB Lightning 工具将数据导入到 TiDB 数据库

### 2. 编辑 TiDB Lightning 工具的配置文件, 如下所示:

```
[root@centos76_vm ~]# cd tidb-toolkit-v5.0.1-linux-amd64/bin/
[root@centos76_vm bin]# vi tidb-lightning.toml
```

内容为:

```
[lightning]
日志
level = "info"
file = "tidb-lightning.log"
[tikv-importer]
选择使用的 local 后端
backend = "local"
设置排序的键值对的临时存放地址, 目标路径需要是一个空目录
sorted-kv-dir = "/tmp"

[mydumper]
源数据目录。
data-source-dir = "/tmp/employees/"

[tidb]
目标集群的信息
host = "172.16.6.212"
port = 4000
user = "root"
表架构信息在从 TiDB 的“状态端口”获取。status-port = 10080
集群 pd 的地址
pd-addr = "172.16.6.202:2379"
```

注意如下:

- (1) backend = "local":表示直接导入到 TiKV-Server 中。
- (2) pd-addr = "172.16.6.202:2379":选择任意一个 PD 节点的 IP 和输入端口号

## Lesson 19 练习-2: 使用 TiDB Lightning 工具将数据导入到 TiDB 数据库

### 3. 连接到TiDB, 检查导入数据前的 TiDB 数据库: (请根据自己实验环境输入密码)

```
[root@centos76_vm ~]# mysql -h172.16.6.212 -P4000 -uroot -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 115
Server version: 5.7.25-TiDB-v5.0.0 TiDB Server (Apache License 2.0) Community Edition, MySQL 5.7 compatible

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

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 > show databases;
+-----+
| Database |
+-----+
| INFORMATION_SCHEMA |
| METRICS_SCHEMA |
| PERFORMANCE_SCHEMA |
| mysql |
| test |
| tidb |
| world |
+-----+
7 rows in set (0.00 sec)
```

### 4. 开始使用 TiDB Lightning 工具进行数据导入:

```
[root@centos76_vm bin]# ! /bin/bash
[root@centos76_vm bin]# nohup ./tidb-lightning -config
tidb-lightning.toml > nohup.out &
[1] 31276
[root@centos76_vm bin]# nohup: ignoring input and redirecting stderr to stdout
（这里请注意：此时屏幕光标会悬停在最后一行，我们可以点击回车键“Enter”，来使
nohup 的 tidb-lightning 在后台工作。）

[1]+ Exit 1 nohup ./tidb-lightning -config tidb-lightning.toml > nohup.out
```

## Lesson 19 练习-2: 使用 TiDB Lightning 工具将数据导入到 TiDB 数据库

5. 进入到 TiDB 数据库, 发现 employees 库被导入, 其中有 6 张数据表:

```
[root@centos76_vm]# mysql -h172.16.6.212 -P4000 -uroot -p
```

Enter password:

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 129

Server version: 5.7.25-TiDB-v5.0.0 TiDB Server (Apache License 2.0) Community Edition, MySQL 5.7 compatible

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

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 > show databases;
```

| Database           |
|--------------------|
| INFORMATION_SCHEMA |
| METRICS_SCHEMA     |
| PERFORMANCE_SCHEMA |
| employees          |
| mysql              |
| test               |
| tidb               |
| world              |

8 rows in set (0.00 sec)

```
mysql > use employees;
```

Reading table information for completion of table and column names

You can turn off this feature to get a quicker startup with -A

Database changed

## Lesson 19 练习-2: 使用 TiDB Lightning 工具将数据导入到 TiDB 数据库

```
mysql > show tables;
+-----+
| Tables_in_employees |
+-----+
| departments |
| dept_emp |
| dept_manager |
| employees |
| salaries |
| titles |
+-----+
6 rows in set (0.00 sec)

mysql > select count(*) from salaries;
+-----+
| count(*) |
+-----+
| 2844047 |
+-----+
1 row in set (0.50 sec)
```

6. 可以从 nohup.out 中监控 tidb-lightning 工具是否退出:

```
[root@centos76_vm bin]# vi nohup.out
```

内容如下:

Verbose debug logs will be written to tidb-lightning.log

tidb lightning exit

7. 可以从 tidb-lightning 工具的详细日志中查看 tidb-lightning 的详细内容:

```
[root@centos76_vm bin]# vi tidb-lightning.log
```

内容如下:

2021/05/27 12:39:37.989 +08:00 [WARN] [config.go:483] ["currently only per-task configuration can be applied, global configuration changes can only be made on startup"] ["global config changes"="[lightning.level,lightning.file]"]

[2021/05/27 12:39:37.990 +08:00] [INFO] [info.go:40] ["Welcome to TiDB-Lightning"] [release-version=v5.0.1] [git-

..... (省略中间过程)

[2021/05/27 12:40:31.828 +08:00] [INFO] [restore.go:342] ["the whole procedure completed"] [takeTime=53.818883735s] []

[2021/05/27 12:40:31.829 +08:00] [INFO] [main.go:94] ["tidb lightning exit"]

8. 退出所有窗口。

## Lesson 20 练习: TiDB Data Migration (DM)

## Lesson 20 练习：概述

---

### 概述

在本课练习中, 您将为已有的 TiDB 数据库着手部署一个 Data Migration(DM) 集群, 之后将已有的 2 个 MySQL 数据库实例中的数据表同步到 TiDB 数据库中。同时您还会练习 Data Migration(DM) 集群的扩容和缩容等维护工作。

### 实验环境要求

1. 部署好 1 个 TiDB 数据库集群, 集群名为 `tidb-test`, 并清理掉之前的实验数据。
2. 部署好 2 个 MySQL 数据库实例, 端口号为 3306 和 3307 , 并清理掉之前的实验数据。
3. tiup 组件部署完毕。
4. 可以连接到互联网。

## Lesson 20 练习-1: Data Migration (DM) 的部署

---

### 概述

在练习一中, 您将通过 TiUP 工具来为已有的 TiDB 数据库着手部署一个 Data Migration (DM) 集群

### 任务

17. 通过 tiup 安装 dm 组件, 如下:

```
[root@centos76_vm ~]# tiup install dm
download https://tiup-mirrors.pingcap.com/dm-v1.4.4-linux-amd64.tar.gz 7.75 MiB / 7.75 MiB 100.00% 4.86 MiB p/s
```

2. 通过 tiup 更新 dm 组件到最新版本, 如下:

```
[root@centos76_vm ~]# tiup update --self && tiup update dm
download https://tiup-mirrors.pingcap.com/tiup-v1.4.4-linux-amd64.tar.gz 6.73 MiB / 6.73 MiB 100.00% 4.42 MiB p/s
Updated successfully!
component dm version v1.4.4 is already installed
Updated successfully!
```

3. 生成一个初始化配置文件, 并准备编辑, 如下:

```
[root@centos76_vm ~]# tiup dm template > topology.yaml
Starting component `dm`: /root/.tiup/components/dm/v1.4.4/tiup-dm template
```

## Lesson 20 练习-1: Data Migration (DM) 的部署

4. 编辑 topology.yaml 文件, 加入 master\_servers, worker\_servers, monitoring\_servers, grafana\_servers 和 alertmanager\_servers , 注意, 您的 IP 地址可能和文档不同, 如下:

```
[root@centos76_vm ~]# vi topology.yaml
```

内容为:

```
The topology template is used deploy a minimal DM cluster, which suitable
for scenarios with only three machinescontains. The minimal cluster contains
- 3 master nodes
- 3 worker nodes
You can change the hosts according your environment

global:
 user: "tidb"
 ssh_port: 22
 deploy_dir: "/home/tidb/dm/deploy"
 data_dir: "/home/tidb/dm/data"
 # arch: "amd64"

master_servers:
 - host: 172.16.6.157
 - host: 172.16.6.202
 - host: 172.16.6.210

worker_servers:
 - host: 172.16.6.157
 - host: 172.16.6.210

monitoring_servers:
 - host: 172.16.6.157

grafana_servers:
 - host: 172.16.6.157

alertmanager_servers:
 - host: 172.16.6.157
```



## Lesson 20 练习-1: Data Migration (DM) 的部署

---

5. 查看当前可用的 Data Migration (DM) 最新版本, 或者其他可用版本, 如下:

```
[root@centos76_vm]# tiup list dm-master
```

Available versions for dm-master:

| Version                            | Installed | Release                   | Platforms               |
|------------------------------------|-----------|---------------------------|-------------------------|
| nightly -> v5.0.0-nightly-20210531 |           | 2021-05-31T21:55:15+08:00 | linux/amd64,linux/arm64 |
| v2.0.0-rc                          |           | 2020-08-21T17:49:08+08:00 | linux/amd64,linux/arm64 |
| v2.0.0-rc.2                        |           | 2020-09-01T20:51:29+08:00 | linux/amd64,linux/arm64 |
| v2.0.0                             |           | 2020-10-30T16:10:58+08:00 | linux/amd64,linux/arm64 |
| v2.0.1                             |           | 2020-12-25T13:22:29+08:00 | linux/amd64,linux/arm64 |
| v2.0.3                             |           | 2021-05-11T22:14:31+08:00 | linux/amd64,linux/arm64 |
| v5.0.0-nightly-20210531            |           | 2021-05-31T21:55:15+08:00 | linux/amd64,linux/arm64 |

## Lesson 20 练习-1: Data Migration (DM) 的部署

6. 部署 Data Migration (DM) 集群, 集群名称为 集群 dm-test, 如下所示:

```
[root@centos76_vm ~]# tiup dm deploy dm-test
v5.0.0-nightly-20210531 ./topology.yaml --user root -p
Starting component `dm`: /root/.tiup/components/dm/v1.4.4/tiup-dm deploy dm-test v5.0.0-nightly-20210531 ./topology.yaml --
user root -p
Please confirm your topology:
Cluster type: dm
Cluster name: dm-test
Cluster version: v5.0.0-nightly-20210531
Role Host Ports OS/Arch Directories

dm-master 172.16.6.157 8261/8291 linux/x86_64 /home/tidb/dm/deploy/dm-master-8261,/home/tidb/dm/data/dm-master-8261
dm-master 172.16.6.202 8261/8291 linux/x86_64 /home/tidb/dm/deploy/dm-master-8261,/home/tidb/dm/data/dm-master-8261
dm-master 172.16.6.210 8261/8291 linux/x86_64 /home/tidb/dm/deploy/dm-master-8261,/home/tidb/dm/data/dm-master-8261
dm-worker 172.16.6.157 8262 linux/x86_64 /home/tidb/dm/deploy/dm-worker-8262,/home/tidb/dm/data/dm-worker-8262
dm-worker 172.16.6.210 8262 linux/x86_64 /home/tidb/dm/deploy/dm-worker-8262,/home/tidb/dm/data/dm-worker-8262
prometheus 172.16.6.157 9090 linux/x86_64 /home/tidb/dm/deploy/prometheus-9090,/home/tidb/dm/data/prometheus-9090
grafana 172.16.6.157 3000 linux/x86_64 /home/tidb/dm/deploy/grafana-3000
alertmanager 172.16.6.157 9093/9094 linux/x86_64 /home/tidb/dm/deploy/alertmanager-9093,/home/tidb/dm/data/alertmanager-9093
Attention:
 1. If the topology is not what you expected, check your yaml file.
 2. Please confirm there is no port/directory conflicts in same host.
Do you want to continue? [y/N]: (default=N) y
Input SSH password:
+ Generate SSH keys ... Done
+ Download TiDB components
 - Download dm-master:v5.0.0-nightly-20210531 (linux/amd64) ... Done
...(省略中间内容)
Enabling component alertmanager
 Enabling instance alertmanager 172.16.6.157:9093
 Enable alertmanager 172.16.6.157:9093 success
Cluster `dm-test` deployed successfully, you can start it with command: `tiup dm start dm-test`
```

## Lesson 20 练习-1: Data Migration (DM) 的部署

7. 查看 TiUP 管理的 DM 集群情况, 如下:

```
[root@centos76_vm ~]# tiup dm list
Starting component `dm`: /root/.tiup/components/dm/v1.4.4/tiup-dm list
Name User Version Path PrivateKey

dm-test tidb v5.0.0-nightly-20210531 /root/.tiup/storage/dm/clusters/dm-test /root/.tiup/storage/dm/clusters/dm-test/ssh/id_rsa
```

8. 检查部署的集群 dm-test 的状态, 如下:

```
[root@centos76_vm ~]# tiup dm display dm-test
Starting component `dm`: /root/.tiup/components/dm/v1.4.4/tiup-dm list
Name User Version Path PrivateKey

dm-test tidb v5.0.0-nightly-20210531 /root/.tiup/storage/dm/clusters/dm-test /root/.tiup/storage/dm/clusters/dm-test/ssh/id_rsa
[root@centos76_vm ~]# tiup dm display dm-test
Starting component `dm`: /root/.tiup/components/dm/v1.4.4/tiup-dm display dm-test
Cluster type: dm
Cluster name: dm-test
Cluster version: v5.0.0-nightly-20210531
SSH type: builtin
ID Role Host Ports OS/Arch Status Data Dir Deploy Dir
--
172.16.6.157:9093 alertmanager 172.16.6.157 9093/9094 linux/x86_64 Up /home/tidb/dm/data/alertmanager-9093
/home/tidb/dm/deploy/alertmanager-9093
172.16.6.157:8261 dm-master 172.16.6.157 8261/8291 linux/x86_64 Healthy /home/tidb/dm/data/dm-master-8261
/home/tidb/dm/deploy/dm-master-8261
172.16.6.202:8261 dm-master 172.16.6.202 8261/8291 linux/x86_64 Healthy /home/tidb/dm/data/dm-master-8261
/home/tidb/dm/deploy/dm-master-8261
172.16.6.210:8261 dm-master 172.16.6.210 8261/8291 linux/x86_64 Healthy|L /home/tidb/dm/data/dm-master-8261
/home/tidb/dm/deploy/dm-master-8261
172.16.6.157:8262 dm-worker 172.16.6.157 8262 linux/x86_64 Bound /home/tidb/dm/data/dm-worker-8262
/home/tidb/dm/deploy/dm-worker-8262
172.16.6.210:8262 dm-worker 172.16.6.210 8262 linux/x86_64 Bound /home/tidb/dm/data/dm-worker-8262
/home/tidb/dm/deploy/dm-worker-8262
172.16.6.157:3000 grafana 172.16.6.157 3000 linux/x86_64 Up - /home/tidb/dm/deploy/grafana-3000
172.16.6.157:9090 prometheus 172.16.6.157 9090 linux/x86_64 Up /home/tidb/dm/data/prometheus-9090
/home/tidb/dm/deploy/prometheus-9090
Total nodes: 8
```

预期输出包括 dm-test 集群中实例 ID、角色、主机、监听端口和状态(由于还未启动, 所以状态为 Down/inactive)、目录信息。

## Lesson 20 练习-1: Data Migration (DM) 的部署

### 9. 启动集群 dm-test, 如下:

```
[root@centos76_vm ~]# tiup dm start dm-test
```

```
Starting component `dm`: /root/.tiup/components/dm/v1.4.4/tiup-dm start dm-test
```

```
Starting cluster dm-test...
```

```
+ [Serial] - SSHKeySet: privateKey=/root/.tiup/storage/dm/clusters/dm-test/ssh/id_rsa,
publicKey=/root/.tiup/storage/dm/clusters/dm-test/ssh/id_rsa.pub
```

```
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.157
```

```
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.202
```

```
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.157
```

```
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.157
```

```
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.210
```

```
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.157
```

```
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.210
```

```
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.157
```

```
+ [Serial] - StartCluster
```

```
Starting component dm-master
```

```
Starting instance dm-master 172.16.6.210:8261
```

```
Starting instance dm-master 172.16.6.157:8261
```

```
Starting instance dm-master 172.16.6.202:8261
```

```
Start dm-master 172.16.6.202:8261 success
```

```
Start dm-master 172.16.6.210:8261 success
```

```
Start dm-master 172.16.6.157:8261 success
```

```
Starting component dm-worker
```

```
Starting instance dm-worker 172.16.6.210:8262
```

```
Starting instance dm-worker 172.16.6.157:8262
```

```
Start dm-worker 172.16.6.210:8262 success
```

```
Start dm-worker 172.16.6.157:8262 success
```

```
Starting component prometheus
```

```
Starting instance prometheus 172.16.6.157:9090
```

```
Start prometheus 172.16.6.157:9090 success
```

```
Starting component grafana
```

```
Starting instance grafana 172.16.6.157:3000
```

```
Start grafana 172.16.6.157:3000 success
```

```
Starting component alertmanager
```

```
Starting instance alertmanager 172.16.6.157:9093
```

```
Start alertmanager 172.16.6.157:9093 success
```

```
Started cluster `dm-test` successfully
```

## Lesson 20 练习-1: Data Migration (DM) 的部署

10. 检查部署的集群 `dm-test` 的状态, 如下:

```
[root@centos76_vm ~]# tiup dm display dm-test
```

Starting component `dm`: /root/.tiup/components/dm/v1.4.4/tiup-dm display dm-test

Cluster type: dm

Cluster name: dm-test

Cluster version: v5.0.0-nightly-20210531

SSH type: builtin

| ID                | Role         | Host         | Ports     | OS/Arch      | Status  | Data Dir                             | Deploy Dir                             |
|-------------------|--------------|--------------|-----------|--------------|---------|--------------------------------------|----------------------------------------|
| 172.16.6.157:9093 | alertmanager | 172.16.6.157 | 9093/9094 | linux/x86_64 | Up      | /home/tidb/dm/data/alertmanager-9093 | /home/tidb/dm/deploy/alertmanager-9093 |
| 172.16.6.157:8261 | dm-master    | 172.16.6.157 | 8261/8291 | linux/x86_64 | Healthy | /home/tidb/dm/data/dm-master-8261    | /home/tidb/dm/deploy/dm-master-8261    |
| 172.16.6.202:8261 | dm-master    | 172.16.6.202 | 8261/8291 | linux/x86_64 | Healthy | /home/tidb/dm/data/dm-master-8261    | /home/tidb/dm/deploy/dm-master-8261    |
| 172.16.6.210:8261 | dm-master    | 172.16.6.210 | 8261/8291 | linux/x86_64 | Healthy | /home/tidb/dm/data/dm-master-8261    | /home/tidb/dm/deploy/dm-master-8261    |
| 172.16.6.157:8262 | dm-worker    | 172.16.6.157 | 8262      | linux/x86_64 | Bound   | /home/tidb/dm/data/dm-worker-8262    | /home/tidb/dm/deploy/dm-worker-8262    |
| 172.16.6.210:8262 | dm-worker    | 172.16.6.210 | 8262      | linux/x86_64 | Bound   | /home/tidb/dm/data/dm-worker-8262    | /home/tidb/dm/deploy/dm-worker-8262    |
| 172.16.6.157:3000 | grafana      | 172.16.6.157 | 3000      | linux/x86_64 | Up      | -                                    | /home/tidb/dm/deploy/grafana-3000      |
| 172.16.6.157:9090 | prometheus   | 172.16.6.157 | 9090      | linux/x86_64 | Up      | /home/tidb/dm/data/prometheus-9090   | /home/tidb/dm/deploy/prometheus-9090   |

Total nodes: 8

## Lesson 20 练习-1: Data Migration (DM) 的部署

---

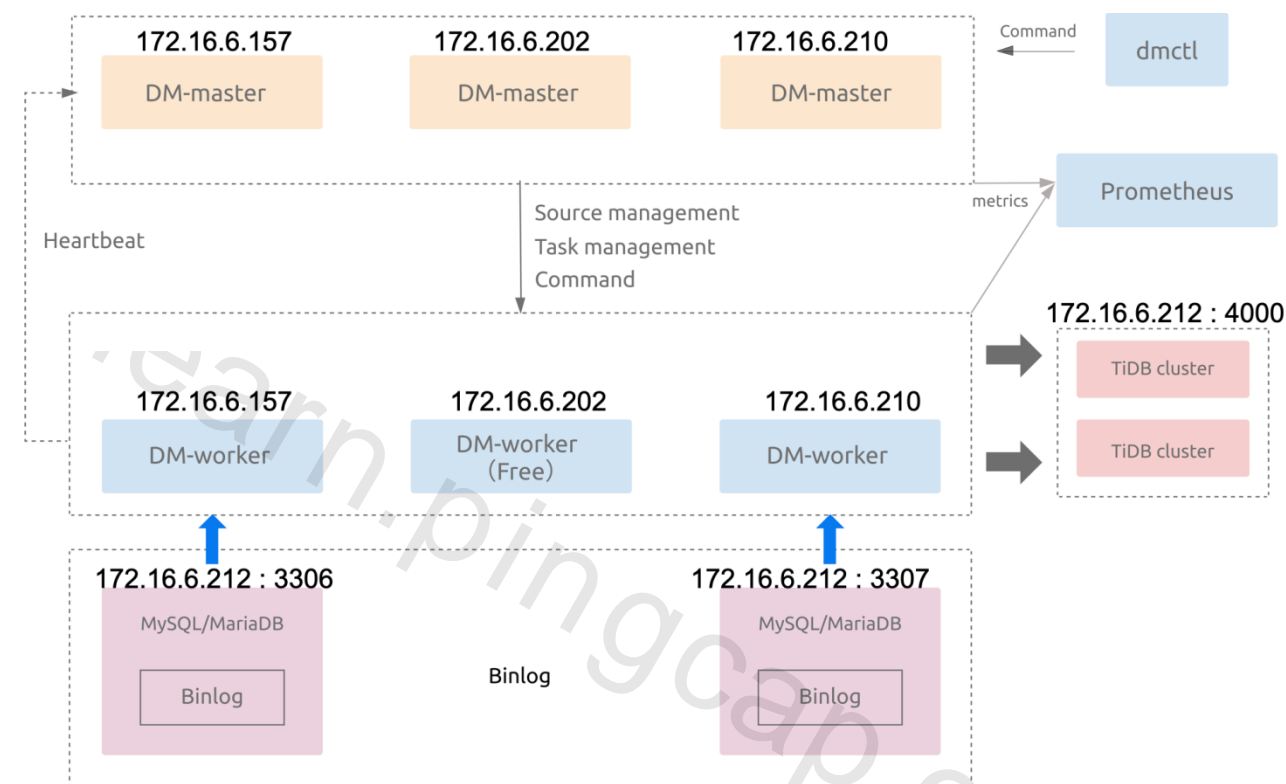
### 11. 获取集群控制工具 dmctl，用于后面的练习：

```
[root@centos76_vm ~]# tiup dmctl:v5.0.0-nightly-20210531
The component `dmctl` version v5.0.0-nightly-20210531 is not installed; downloading from repository.
download https://tiup-mirrors.pingcap.com/dmctl-v5.0.0-nightly-20210531-linux-amd64.tar.gz 26.50 MiB / 26.50 MiB 100.00% 10.49
MiB p/s
Starting component `dmctl`: /root/.tiup/components/dmctl/v5.0.0-nightly-20210531/dmctl/dmctl
Usage: dmctl [global options] command [command options] [arguments...]
... (省略中间内容)
Global Options:
--V Prints version and exit.
--config Path to config file.
--master-addr Master API server address, this parameter is required when interacting with the dm-master
--rpc-timeout RPC timeout, default is 10m.
--ssl-ca Path of file that contains list of trusted SSL CAs for connection.
--ssl-cert Path of file that contains X509 certificate in PEM format for connection.
--ssl-key Path of file that contains X509 key in PEM format for connection.
```

## Lesson 20 练习-2: 从 MySQL 同步数据到 TiDB

### 概述

在练习二中, 我们将 MySQL 数据库实例 3306 和 3307 中的 schema 和表同步到 TiDB 数据库中。拓扑结构如下图:



注意: IP 地址可能会和您的环境不同, 请注意调整。

同步规则为:

规则一:MySQL 数据库实例 3306 中的 user 库中所有的表同步到 TiDB 数据库的 user\_north 中去, 3307 中的 user 库中所有的表同步到 TiDB 数据库的 user\_east 中去。

规则二:MySQL 数据库实例 3306 和 3307 中的 store 库中的表原样同步到 TiDB 数据库中的 store 库中的表,但是 3307 中的 store 库中的表 store\_sz 会同步到 TiDB 的 store\_suzhou 表中。

规则三:MySQL 数据库实例 3306 和 3307 中的 salesdb 库中的表 sales 做了分表,它们会同步到 TiDB 中的,salesdb 库的 sales 表中。(分表分库规则)

规则四:MySQL 数据库实例 3306 和 3307 中的 user 库不会复制删除操作,user 库中的 trace 表不会复制 truncate ,drop 和 delete 操作,store 库不会复制删除操作,store 库的表不会复制 truncate ,drop 和 delete 操作。

规则五:MySQL 数据库实例 3306 和 3307 中的 log 库不会参与复制。

## Lesson 20 练习-2: 从 MySQL 同步数据到 TiDB

### 任务

#### 1. MySQL 数据库准备,为 MySQL 数据库开通用户权限,并初始化数据

##### 1.1. 分别连接到端口号 3306 和 3307 的 MySQL 数据库,创建

'root'@'172.16.6.157', 'root'@'172.16.6.210' 和 'root'@'172.16.6.202', 并赋予all privileges 权限(更详细权限请参考文档),这 3 个用户用于 dm-worker 连接 MySQL 数据库进行全量和增量数据的读取:

```
mysql> create user 'root'@'172.16.6.157' identified by 'mysql';
Query OK, 0 rows affected (0.00 sec)

mysql> grant all privileges on *.* to 'root'@'172.16.6.157';
Query OK, 0 rows affected (0.00 sec)

mysql> create user 'root'@'172.16.6.210' identified by 'mysql';
Query OK, 0 rows affected (0.00 sec)

mysql> grant all privileges on *.* to 'root'@'172.16.6.210';
Query OK, 0 rows affected (0.00 sec)

mysql> create user 'root'@'172.16.6.202' identified by 'mysql';
Query OK, 0 rows affected (0.00 sec)

mysql> grant all privileges on *.* to 'root'@'172.16.6.202';
Query OK, 0 rows affected (0.01 sec)
```

注意:上面操作需要在 3306 和 3307 两个实例中分别执行

##### 1.2. 分别连接到端口号 3306 和 3307 的 MySQL 数据库,导入数据库 user, store, log, salesdb。

```
[root@centos76_vm ~]# mysql -uroot -pmysql -S /data/mydb3306/mysql.sock < 3306db.sql
[root@centos76_vm ~]# mysql -uroot -pmysql -S /data/mydb3307/mysql.sock < 3307db.sql
```

注意: 3306db.sql 和 3307db.sql 文件在 script 目录下



## Lesson 20 练习-2: 从 MySQL 同步数据到 TiDB

---

18. TiDB 数据库准备, 为 TiDB 数据库开通权限, 并准备好数据。

2.1. 在已有 TiDB 数据库中, 创建用户 'root'@'172.16.6.157', 'root'@'172.16.6.210' 和 'root'@'172.16.6.202', 并赋予 all privileges 权限(更详细权限请参考文档), 这 3 个用户用于 dm-worker 连接 TiDB 数据库进行全量和增量数据的导入:

```
mysql> create user 'root'@'172.16.6.157' identified by 'tidb';
Query OK, 0 rows affected (0.07 sec)

mysql> grant all privileges on *.* to 'root'@'172.16.6.157';
Query OK, 0 rows affected (0.03 sec)

mysql> create user 'root'@'172.16.6.210' identified by 'tidb';
Query OK, 0 rows affected (0.03 sec)

mysql> grant all privileges on *.* to 'root'@'172.16.6.210';
Query OK, 0 rows affected (0.03 sec)

mysql> create user 'root'@'172.16.6.202' identified by 'tidb';
Query OK, 0 rows affected (0.02 sec)

mysql> grant all privileges on *.* to 'root'@'172.16.6.202';
Query OK, 0 rows affected (0.02 sec)
```

2.2. 创建数据库 user\_north, user\_east, store 和 salesdb, 创建相关表

user\_north.information, user\_north.trace, user\_east.information, user\_east.trace, store.store\_bj, store.store\_tj, store.store\_sh, store.store\_suzhou 和 salesdb.sales, 如下所示:

```
mysql> create database user_north;
Query OK, 0 rows affected (1.02 sec)

mysql> create database user_east;
Query OK, 0 rows affected (1.02 sec)

mysql> create database store;
Query OK, 0 rows affected (1.52 sec)

mysql> use user_north;
Database changed
```

## Lesson 20 练习-2: 从 MySQL 同步数据到 TiDB

```
mysql> create table information(id int primary key, info varchar(64));
Query OK, 0 rows affected (1.59 sec)

mysql> create table trace(id int primary key, content varchar(64));
Query OK, 0 rows affected (0.52 sec)

mysql> use user_east
Database changed

mysql> create table information(id int primary key, info varchar(64));
Query OK, 0 rows affected (0.53 sec)

mysql> create table trace(id int primary key, content varchar(64));
Query OK, 0 rows affected (0.53 sec)

mysql> use store;
Database changed

mysql> create table store_bj(id int primary key, pname varchar(64));
Query OK, 0 rows affected (1.03 sec)

mysql> create table store_tj(id int primary key, pname varchar(64));
Query OK, 0 rows affected (0.52 sec)

mysql> create table store_sh(id int primary key, pname varchar(64));
Query OK, 0 rows affected (1.03 sec)

mysql> create table store_suzhou(id int primary key, pname varchar(64));
Query OK, 0 rows affected (0.52 sec)

mysql> show tables;
+-----+
| Tables_in_store |
+-----+
| store_bj |
| store_sh |
| store_suzhou |
| store_tj |
+-----+
4 rows in set (0.00 sec)

mysql> create database salesdb;
Query OK, 0 rows affected (0.52 sec)

mysql> use salesdb;
Database changed

mysql> create table sales(id int primary key, pname varchar(20), cnt int);
Query OK, 0 rows affected (1.02 sec)
```

## Lesson 20 练习-2: 从 MySQL 同步数据到 TiDB

```
mysql> show databases;
+-----+
| Database |
+-----+
| INFORMATION_SCHEMA |
| METRICS_SCHEMA |
| PERFORMANCE_SCHEMA |
| mysql |
| salesdb |
| store |
| test |
| user_east |
| user_north |
+-----+
9 rows in set (0.00 sec)

mysql> create database log;
Query OK, 0 rows affected (1.03 sec)

mysql> use log;
Database changed

mysql> create table messages(id int primary key, msg varchar(64));
Query OK, 0 rows affected (1.03 sec)
```

注意:TiDB 数据库中的数据库和表也可以不预先创建,在 DM 全量同步数据之前会创建这些数据库和表

19. 数据准备完毕后,编辑数据源配置文件,如下:

19.1. 为端口号为 3306 的 MySQL 实例编辑数据源配置文件,如下:

```
[root@centos76_vm ~]# vi mysql-source-conf1.yaml
source-id: "mysql-replica-01"

from:
 host: "172.16.6.212"
 user: "root"
 password: "F1GJCYzuWx/8H4EJHRDWwkBJCrN+4A=="
 port: 3306
```

3.2. 为端口号为 3307 的 MySQL 实例编辑数据源配置文件,如下:

```
[root@centos76_vm ~]# vi mysql-source-conf2.yaml
source-id: "mysql-replica-02"

from:
 host: "172.16.6.212"
 user: "root"
 password: "F1GJCYzuWx/8H4EJHRDWwkBJCrN+4A=="
 port: 3307
```

注意:password: "F1GJCYzuWx/8H4EJHRDWwkBJCrN+4A==" 表示隐藏掉明文的密码,我们可以按照如下方法生成:

```
tiup dmctl -encrypt 'mysql'
F1GJCYzuWx/8H4EJHRDWwkBJCrN+4A==
```

## Lesson 20 练习-2: 从 MySQL 同步数据到 TiDB

20. 将数据源配置文件加载到 DM 中, 如下:

### 4.1. 加载 MySQL 数据库的 3306 端口配置文件:

```
[root@centos76_vm ~]# tiup dmctl --master-addr=172.16.6.202:8261 operate-source create mysql-source-conf1.yaml
Starting component 'dmctl': /root/.tiup/components/dmctl/v2.0.3/dmctl/dmctl --master-addr=172.16.6.202:8261 operate-source create mysql-source-conf1.yaml
{
 "result": true,
 "msg": "",
 "sources": [
 {
 "result": true,
 "msg": "",
 "source": "mysql-replica-01",
 "worker": "dm-172.16.6.210-8262"
 }
]
}
```

### 4.2. 加载 MySQL 数据库的 3307 端口配置文件:

```
[root@centos76_vm ~]# tiup dmctl --master-addr=172.16.6.202:8261 operate-source create mysql-source-conf2.yaml
Starting component 'dmctl': /root/.tiup/components/dmctl/v2.0.3/dmctl/dmctl --master-addr=172.16.6.202:8261 operate-source create mysql-source-conf2.yaml
{
 "result": true,
 "msg": "",
 "sources": [
 {
 "result": true,
 "msg": "",
 "source": "mysql-replica-02",
 "worker": "dm-172.16.6.157-8262"
 }
]
}
```

注意: --master-addr=172.16.6.202:8261 为 DM 集群中的任意一个 master 节点, 您的环境可能和手册不同。

## Lesson 20 练习-2: 从 MySQL 同步数据到 TiDB

### 4.3. 查看已经加载的数据源, 如下:

```
[root@centos76_vm ~]# tiup dmctl --master-addr=172.16.6.202:8261 get-config source
mysql-replica-01
Starting component `dmctl`: /root/.tiup/components/dmctl/v2.0.3/dmctl/dmctl --master-addr=172.16.6.202:8261 get-config source
mysql-replica-01
{
 "result": true,
 "msg": "",
 "cfg": "enable-gtid: false\nauto-fix-gtid: false\nrelay-dir: relay-dir\nmeta-dir: \"/\n\nflavor: mysql\ncharset: \"/\n\nenable-relay:
false\nrelay-binlog-name: \"/\n\nrelay-binlog-gtid: \"/\n\nsource-id: mysql-replica-02\nfrom:\n host: 172.16.6.212\n port: 3307\n user:
root\n password: '*****'\n max-allowed-packet: null\n session:\n time_zone: \"+00:00"\n security: null\npurge:\n interval: 3600\n
expires: 0\n remain-space: 15\nchecker:\n check-enable: true\n backoff-rollback: 5m0s\n backoff-max: 5m0s\n check-interval: 5s\n
backoff-min: 1s\n backoff-jitter: true\n backoff-factor: 2\nserver-id: 429523516\ntracer: {}\ncase-sensitive: false\nfilters: []\n"
}

[root@centos76_vm ~]# tiup dmctl --master-addr=172.16.6.202:8261 get-config source
mysql-replica-02
Starting component `dmctl`: /root/.tiup/components/dmctl/v2.0.3/dmctl/dmctl --master-addr=172.16.6.202:8261 get-config source
mysql-replica-02
{
 "result": true,
 "msg": "",
 "cfg": "enable-gtid: false\nauto-fix-gtid: false\nrelay-dir: relay-dir\nmeta-dir: \"/\n\nflavor: mysql\ncharset: \"/\n\nenable-relay:
false\nrelay-binlog-name: \"/\n\nrelay-binlog-gtid: \"/\n\nsource-id: mysql-replica-01\nfrom:\n host: 172.16.6.212\n port: 3306\n user:
root\n password: '*****'\n max-allowed-packet: null\n session:\n time_zone: \"+00:00"\n security: null\npurge:\n interval: 3600\n
expires: 0\n remain-space: 15\nchecker:\n check-enable: true\n backoff-rollback: 5m0s\n backoff-max: 5m0s\n check-interval: 5s\n
backoff-min: 1s\n backoff-jitter: true\n backoff-factor: 2\nserver-id: 429567054\ntracer: {}\ncase-sensitive: false\nfilters: []\n"
}
```

### 4.4. 查看数据源和 dm-worker 的对应关系, 如下:

```
[root@centos76_vm ~]# tiup dmctl --master-addr=172.16.6.202:8261 operate-
source show
Starting component `dmctl`: /root/.tiup/components/dmctl/v2.0.3/dmctl/dmctl --master-addr=172.16.6.202:8261 operate-source show
{
 "result": true,
 "msg": "",
 "sources": [
 {
 "result": true,
 "msg": "",
 "source": "mysql-replica-01",
 "worker": "dm-172.16.6.210-8262"
 },
 {
 "result": true,
 "msg": "",
 "source": "mysql-replica-02",
 "worker": "dm-172.16.6.157-8262"
 }
]
}
```

## Lesson 20 练习-2: 从 MySQL 同步数据到 TiDB

21. 按照规则, 配置 DM 任务配置文件 dm-task, 如下:

5.1. 任务信息如下:

```
name: "dm-taskX"
task-mode: all
ignore-checking-items: ["auto_increment_ID"]
```

任务名:dm-taskX, (X 代表任意字符) 复制方式:all(全量 + 增量), ignore-checking-items: ["auto\_increment\_ID"]:忽略自增主键检测。

5.2. 目标 TiDB 数据库配置信息如下:

```
target-database:
 host: "172.16.6.212"
 port: 4000
 user: "root"
 password: "tidb"
```

数据库地址:172.16.6.212, 端口为:4000, 用户名:root, 密码:tidb。

5.3. 对于规则一 : 规则五我们进行配置, 如下:

规则一:MySQL 数据库实例 3306 中的 user 库中所有的表同步到 TiDB 数据库的 user\_north 中去, 3307 中的 user 库中所有的表同步到 TiDB 数据库的 user\_east 中去。我们

```
routes:
 instance-1-user-rule:
 schema-pattern: "user"
 target-schema: "user_north"
 instance-2-user-rule:
 schema-pattern: "user"
 target-schema: "user_east"
```

使用 Table routings 实现, 如下:

规则二:MySQL 数据库实例 3306 和 3307 中的 store 库中的表原样同步到 TiDB 数据库中的 store 库中的表, 但是 3307 中的 store 库中的表 store\_sz 会同步到 TiDB 的

```
instance-2-store-rule:
 schema-pattern: "store"
 table-pattern: "store_sz"
 target-schema: "store"
 target-table: "store_suzhou"
```

store\_suzhou 表中。我们使用 Table routings 实现, 如下:

规则三:MySQL 数据库实例 3306 和 3307 中的 salesdb 库中的表 sales 做了分表, 它们会同步到 TiDB 中的, salesdb 库的 sales 表中。(分表分库规则)

```
sale-route-rule:
 schema-pattern: "salesdb"
 target-schema: "salesdb"
```

我们使用 Table routings 实现, 如下:

## Lesson 20 练习-2: 从 MySQL 同步数据到 TiDB

规则四:MySQL 数据库实例 3306 和 3307 中的 user 库不会复制删除操作, user 库中的 trace 表不会复制 truncate , drop 和 delete 操作, store 库不会复制删除操作, store 库的表不会复制 truncate , drop 和 delete 操作。

我们使用 Binlog event filter 实现, 如下:

```
filters:
 trace-filter-rule: # user 库中的 trace 表不会复制 truncate , drop 和 delete 操作
 schema-pattern: "user"
 table-pattern: "trace"
 events: ["truncate table", "drop table", "delete"]
 action: ignore
 user-filter-rule: # MySQL 数据库实例 3306 和 3307 中的 user 库不会复制删除操作
 schema-pattern: "user"
 events: ["drop database"]
 action: ignore
 store-filter-rule: # store 库不会复制删除操作, store 库的表不会复制 truncate , drop 和 delete 操作
 schema-pattern: "store"
 events: ["drop database", "truncate table", "drop table", "delete"]
 action: ignore
```

规则五:MySQL 数据库实例 3306 和 3307 中的 log 库不会参与复制。

我们使用 block allow list 实现, 如下:

```
block-allow-list:
 log-ignored:
 ignore-dbs: ["log"]
```

5.4. 我们将 MySQL 数据库实例 3306 和 3307 两个实例关联上述规则:

```
mysql-instances:
-
 source-id: "mysql-replica-01"
 route-rules: ["instance-1-user-rule", "sale-route-rule"]
 filter-rules: ["trace-filter-rule", "user-filter-rule", "store-filter-rule"]
 block-allow-list: "log-ignored"
 mydumper-config-name: "global"
 loader-config-name: "global"
 syncer-config-name: "global"
-
 source-id: "mysql-replica-02"
 route-rules: ["instance-2-user-rule", "instance-2-store-rule", "sale-route-rule"]
 filter-rules: ["trace-filter-rule", "user-filter-rule", "store-filter-rule"]
 block-allow-list: "log-ignored"
 mydumper-config-name: "global"
 loader-config-name: "global"
 syncer-config-name: "global"
```

5.5. 得出最终配置如下:

```
name: "dm-taskX"
task-mode: all
ignore-checking-items: ["auto_increment_ID"]

target-database:
 host: "172.16.6.212"
 port: 4000
 user: "root"
 password: "tidb"
```

## Lesson 20 练习-2: 从 MySQL 同步数据到 TiDB

learn.pingcap.cc



```
mysql-instances:
-
 source-id: "mysql-replica-01"
 route-rules: ["instance-1-user-rule", "sale-route-rule"]
 filter-rules: ["trace-filter-rule", "user-filter-rule", "store-filter-rule"]
 block-allow-list: "log-ignored"
 mydumper-config-name: "global"
 loader-config-name: "global"
 syncer-config-name: "global"
-
 source-id: "mysql-replica-02"
 route-rules: ["instance-2-user-rule", "instance-2-store-rule", "sale-route-rule"]
 filter-rules: ["trace-filter-rule", "user-filter-rule", "store-filter-rule"]
 block-allow-list: "log-ignored"
 mydumper-config-name: "global"
 loader-config-name: "global"
 syncer-config-name: "global"
```

# 所有实例的共有配置

```
routes:
instance-1-user-rule:
 schema-pattern: "user"
 target-schema: "user_north"
instance-2-user-rule:
 schema-pattern: "user"
 target-schema: "user_east"
instance-2-store-rule:
 schema-pattern: "store"
 table-pattern: "store_sz"
 target-schema: "store"
 target-table: "store_suzhou"
sale-route-rule:
 schema-pattern: "salesdb"
 target-schema: "salesdb"

filters:
trace-filter-rule:
 schema-pattern: "user"
 table-pattern: "trace"
 events: ["truncate table", "drop table", "delete"]
 action: Ignore
user-filter-rule:
 schema-pattern: "user"
 events: ["drop database"]
 action: Ignore
store-filter-rule:
 schema-pattern: "store"
 events: ["drop database", "truncate table", "drop table", "delete"]
 action: Ignore

block-allow-list:
log-ignored:
 ignore-dbs: ["log"]

mydumpers:
global:
 threads: 4
 chunk-filesize: 64
```

## Lesson 20 练习-2: 从 MySQL 同步数据到 TiDB

6. 对于上游 MySQL 源数据库进行检查, 得到期待结果, 如下:

```
[root@centos76_vm ~]# tiup dmctl --master-addr=172.16.6.202:8261 check-task dm-task.yaml
Starting component `dmctl`: /root/.tiup/components/dmctl/v2.0.3/dmctl/dmctl --master-addr=172.16.6.202:8261 check-task dm-task.yaml
{
 "result": true,
 "msg": "check pass!!!"
}
```

注意:开始任务之前会自动执行检查任务。

7. 创建复制任务, 并默认开始, 如下:

```
[root@centos76_vm ~]# tiup dmctl --master-addr=172.16.6.202:8261 start-task dm-task.yaml
Starting component `dmctl`: /root/.tiup/components/dmctl/v2.0.3/dmctl/dmctl --master-addr=172.16.6.202:8261 start-task dm-task.yaml
{
 "result": true,
 "msg": "",
 "sources": [
 {
 "result": true,
 "msg": "",
 "source": "mysql-replica-01",
 "worker": "dm-172.16.6.210-8262"
 },
 {
 "result": true,
 "msg": "",
 "source": "mysql-replica-02",
 "worker": "dm-172.16.6.157-8262"
 }
]
}
```

8. 查询任务状态, 查看是否正常, 如下:

```
[root@centos76_vm ~]# tiup dmctl --master-addr=172.16.6.202:8261 query-status dm-task.yaml
Starting component `dmctl`: /root/.tiup/components/dmctl/v2.0.3/dmctl/dmctl --master-addr=172.16.6.202:8261 query-status dm-task.yaml
{
 "result": true,
 "msg": "",
 "sources": [
 {
 "result": true,
 "msg": "",
 "sourceStatus": {
 "source": "mysql-replica-01",
 "worker": "dm-172.16.6.210-8262",
 "result": null,
 "relayStatus": null
 }
 },
],
}
```

## Lesson 20 练习-2: 从 MySQL 同步数据到 TiDB

```
"subTaskStatus": [
 {
 "name": "dm-task1",
 "stage": "Running",
 "unit": "Sync",
 "result": null,
 "unresolvedDDLLockID": "",
 "sync": {
 "totalEvents": "8",
 "totalTps": "0",
 "recentTps": "0",
 "masterBinlog": "(mysql-bin.000002, 8358)",
 "masterBinlogGtid": "",
 "syncerBinlog": "(mysql-bin.000002, 8023)",
 "syncerBinlogGtid": "",
 "blockingDDLs": [
],
 "unresolvedGroups": [
],
 "synced": false,
 "binlogType": "remote"
 }
 }
],
{
 "result": true,
 "msg": "",
 "sourceStatus": {
 "source": "mysql-replica-02",
 "worker": "dm-172.16.6.157-8262",
 "result": null,
 "relayStatus": null
 },
 "subTaskStatus": [
 {
 "name": "dm-task1",
 "stage": "Running",
 "unit": "Sync",
 "result": null,
 "unresolvedDDLLockID": "",
 "sync": {
 "totalEvents": "3",
 "totalTps": "0",
 "recentTps": "0",
 "masterBinlog": "(mysql-bin.000002, 5706)",
 "masterBinlogGtid": "",
 "syncerBinlog": "(mysql-bin.000002, 5706)",
 "syncerBinlogGtid": "",
 "blockingDDLs": [
],
 "unresolvedGroups": [
],
 "synced": true,
 "binlogType": "remote"
 }
 }
]
}
]
```

## Lesson 20 练习-2: 从 MySQL 同步数据到 TiDB

9. 检查目标 TiDB 数据库的数据, 查看是否数据同步正确, 如下:

```
mysql> use user_east
Database changed
mysql> show tables;
+-----+
| Tables_in_user_east |
+-----+
| information |
| trace |
+-----+
2 rows in set (0.00 sec)

mysql> select * from information;
+----+-----+
| id | info |
+----+-----+
| 1 | andy |
| 2 | candy|
+----+-----+
2 rows in set (0.00 sec)

mysql> select * from trace;
+----+-----+
| id | content |
+----+-----+
| 1 | error |
+----+-----+
1 row in set (0.01 sec)

mysql> use user_north;
Database changed
mysql> select * from information;
+----+-----+
| id | info |
+----+-----+
| 1 | tom |
| 2 | jack |
+----+-----+
2 rows in set (0.01 sec)

mysql> select * from trace;
+----+-----+
| id | content |
+----+-----+
| 1 | login |
+----+-----+
1 row in set (0.01 sec)

mysql> use store;
Database changed
mysql> show tables;
+-----+
| Tables_in_store |
+-----+
| store_bj |
| store_sh |
| store_suzhou |
| store_tj |
+-----+
4 rows in set (0.00 sec)
```

## Lesson 20 练习-2: 从 MySQL 同步数据到 TiDB

```
mysql> select * from store_bj;
+----+-----+
| id | pname |
+----+-----+
| 1 | book |
+----+-----+
1 row in set (0.00 sec)

mysql> select * from store_tj;
+----+-----+
| id | pname |
+----+-----+
| 1 | cup |
+----+-----+
1 row in set (0.00 sec)

mysql> select * from store_suzhou;
+----+-----+
| id | pname |
+----+-----+
| 1 | hat |
+----+-----+
1 row in set (0.00 sec)

mysql> select * from store_sh;
+----+-----+
| id | pname |
+----+-----+
| 1 | bike |
+----+-----+
1 row in set (0.01 sec)

mysql> use salesdb;
Database changed
mysql> select * from sales;
+----+-----+-----+
| id | pname | cnt |
+----+-----+-----+
| 1 | book | 100 |
| 2 | cup | 200 |
+----+-----+-----+
2 rows in set (0.01 sec)
```

10. 对于分表分库复制进行测试, 如下:

10.1 在 TiDB 数据库的 salesdb 中查询, 如下:

```
mysql> select * from sales;
+----+-----+-----+
| id | pname | cnt |
+----+-----+-----+
| 1 | book | 100 |
| 2 | cup | 200 |
+----+-----+-----+
2 rows in set (0.01 sec)
```

## Lesson 20 练习-2: 从 MySQL 同步数据到 TiDB

10.2 在 MySQL 数据库 3306 端口的 salesdb 中插入数据, 如下:

```
mysql> insert into sales values(3,'shose',120);
Query OK, 1 row affected (0.00 sec)

> select * from sales;
+----+-----+-----+
| id | pname | cnt |
+----+-----+-----+
| 1 | book | 100 |
| 3 | shose | 120 |
+----+-----+-----+
2 rows in set (0.00 sec)
```

10.3 在 TiDB 数据库的 salesdb 中查询, 如下:

```
mysql> select * from sales;
+----+-----+-----+
| id | pname | cnt |
+----+-----+-----+
1	book	100
2	cup	200
3	shose	120
+----+-----+-----+
3 rows in set (0.00 sec)
```

10.4 在 MySQL 数据库 3307 端口的 salesdb 中插入数据, 如下:

```
mysql> insert into sales values(4, 'paper',1000);
Query OK, 1 row affected (0.00 sec)

mysql> select * from sales;
+----+-----+-----+
| id | pname | cnt |
+----+-----+-----+
| 2 | cup | 200 |
| 4 | paper | 1000|
+----+-----+-----+
2 rows in set (0.00 sec)
```

10.5 在 TiDB 数据库的 salesdb 中查询, 如下:

```
mysql> select * from sales;
+----+-----+-----+
| id | pname | cnt |
+----+-----+-----+
1	book	100
2	cup	200
3	shose	120
4	paper	1000
+----+-----+-----+
4 rows in set (0.00 sec)
```

11. 对于单表复制的测试, 如下:

11.1 在 MySQL 数据库 3306 端口的 user 中插入数据, 如下:

```
mysql> use user;
Database changed
mysql> show tables;
+-----+
| Tables_in_user |
+-----+
| information |
| trace |
+-----+
2 rows in set (0.00 sec)
```

## Lesson 20 练习-2: 从 MySQL 同步数据到 TiDB

```
mysql> select * from information;
+----+-----+
| id | info |
+----+-----+
| 1 | tom |
| 2 | jack |
+----+-----+
2 rows in set (0.00 sec)

mysql> insert into information values(3,'frank');
Query OK, 1 row affected (0.01 sec)

mysql> select * from information;
+----+-----+
| id | info |
+----+-----+
1	tom
2	jack
3	frank
+----+-----+
3 rows in set (0.00 sec)
```

11.2 在 TiDB 数据库的 user\_north 中查询,如下:

```
mysql> use user_north
Database changed
mysql> select * from information;
+----+-----+
| id | info |
+----+-----+
1	tom
2	jack
3	frank
+----+-----+
3 rows in set (0.00 sec)
```

12. 对于“规则五: MySQL 数据库实例 3306 和 3307 中的 log 库不会参与复制。”的测试:

12.1 在 MySQL 数据库 3306 端口的 log 库中插入数据,如下:

```
mysql> select * from messages;
+----+-----+
| id | msg |
+----+-----+
| 1 | hello|
+----+-----+
1 row in set (0.00 sec)

mysql> insert into messages values(2,'world');
Query OK, 1 row affected (0.02 sec)
```

对于“规则四中:MySQL 数据库实例 3306 和 3307 中的 user 库中的 trace 表不会复制 truncate ,drop 和 delete 操作。”的测试

12.2 在 TiDB 数据库的 log 库中查询,如下:

```
mysql> select * from messages;
Empty set (0.00 sec)
```

13. 对于“规则四中:MySQL 数据库实例 3306 和 3307 中的 user 库中的 trace 表不会复制 truncate ,drop 和 delete 操作。”的测试

13.1 在 MySQL 数据库 3306 端口的 user 库中进行如下操作:

## Lesson 20 练习-2: 从 MySQL 同步数据到 TiDB

```
mysql> insert into trace values (2, 'query');
Query OK, 1 row affected (0.00 sec)

mysql> insert into trace values (3, 'log out');
Query OK, 1 row affected (0.00 sec)

mysql> select * from trace;
+----+-----+
| id | content |
+----+-----+
1	login
2	query
3	log out
+----+-----+
3 rows in set (0.00 sec)

mysql> delete from trace where id=3;
Query OK, 1 row affected (0.01 sec)

mysql> truncate table trace;
Query OK, 0 rows affected (0.03 sec)

mysql> drop table trace;
Query OK, 0 rows affected (0.00 sec)

mysql> show tables;
+-----+
| Tables_in_user |
+-----+
| information |
+-----+
1 row in set (0.00 sec)
```

### 13.2 检查 TiDB 数据库中 user\_north 库中的 trace 表是否存在:

```
mysql> select * from trace;
+----+-----+
| id | content |
+----+-----+
1	login
2	query
3	log out
+----+-----+
3 rows in set (0.00 sec)
```



## Lesson 20 练习-2: 从 MySQL 同步数据到 TiDB

---

14. 暂停复制任务, 如下:

```
[root@centos76_vm ~]# tiup dmctl --master-addr=172.16.6.202:8261
pause-task dm-task.yaml
Starting component `dmctl`: /root/.tiup/components/dmctl/v2.0.3/dmctl/dmctl --master-addr=172.16.6.202:8261 pause-task dm-task1
{
 "op": "Pause",
 "result": true,
 "msg": "",
 "sources": [
 {
 "result": true,
 "msg": "",
 "source": "mysql-replica-01",
 "worker": "dm-172.16.6.210-8262"
 },
 {
 "result": true,
 "msg": "",
 "source": "mysql-replica-02",
 "worker": "dm-172.16.6.157-8262"
 }
]
}
```

15. 在任务暂定的情况下, 连接 MySQL 数据库 3307 端口的 user 库中进行如下操作:

```
mysql> use user;

Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
Database changed

mysql> show tables;
+-----+
| Tables_in_user |
+-----+
| information |
| trace |
+-----+
2 rows in set (0.00 sec)

mysql> select * from information;
+----+-----+
| id | info |
+----+-----+
| 1 | andy |
| 2 | candy|
+----+-----+
2 rows in set (0.00 sec)

mysql> insert into information values(3,'joe');
Query OK, 1 row affected (0.01 sec)

mysql> select * from information;
+----+-----+
| id | info |
+----+-----+
1	andy
2	candy
3	joe
+----+-----+
3 rows in set (0.00 sec)
```

## Lesson 20 练习-2: 从 MySQL 同步数据到 TiDB

16. 连接 TiDB 数据库, 校验 user\_east 库中的 information 表数据是否被复制操作, 如下:

```
mysql> show tables;
+-----+
| Tables_in_user_east |
+-----+
| information |
| trace |
+-----+
2 rows in set (0.00 sec)

mysql> select * from information;
+----+-----+
| id | info |
+----+-----+
| 1 | andy |
| 2 | candy|
+----+-----+
1 row in set (0.00 sec)
```

我们发现暂停复制后并没有操作被复制。

17. 恢复复制任务, 如下:

```
[root@centos76_vm ~]# tiup dmctl --master-addr=172.16.6.202:8261
resume-task dm-task.yaml
Starting component `dmctl`: /root/.tiup/components/dmctl/v2.0.3/dmctl/dmctl --master-addr=172.16.6.202:8261 resume-task dm-task1
{
 "op": "Resume",
 "result": true,
 "msg": "",
 "sources": [
 {
 "result": true,
 "msg": "",
 "source": "mysql-replica-01",
 "worker": "dm-172.16.6.210-8262"
 },
 {
 "result": true,
 "msg": "",
 "source": "mysql-replica-02",
 "worker": "dm-172.16.6.157-8262"
 }
]
}
```

18. 连接 TiDB 数据库, 校验 user\_east 库中的 information 表数据是否被复制操作, 如下:

```
mysql> select * from information;
+----+-----+
| id | info |
+----+-----+
1	andy
2	candy
3	joe
+----+-----+
2 rows in set (0.01 sec)
```

## Lesson 20 练习-2: 从 MySQL 同步数据到 TiDB

19. 停止 DM 的复制任务, 如下:

```
[root@centos76_vm ~]# tiup dmctl --master-addr=172.16.6.202:8261 stop-task dm-task.yaml
Starting component `dmctl`: /root/.tiup/components/dmctl/v2.0.3/dmctl/dmctl --master-addr=172.16.6.202:8261 stop-task dm-task1
{
 "op": "Stop",
 "result": true,
 "msg": "",
 "sources": [
 {
 "result": true,
 "msg": "",
 "source": "mysql-replica-01",
 "worker": "dm-172.16.6.210-8262"
 },
 {
 "result": true,
 "msg": "",
 "source": "mysql-replica-02",
 "worker": "dm-172.16.6.157-8262"
 }
]
}
```

20. 查询 DM 复制操作是否停止, 如下:

```
[root@centos76_vm ~] # tiup dmctl --master-addr=172.16.6.202:8261 query-status dm-task.yaml
Starting component `dmctl`: /root/.tiup/components/dmctl/v2.0.3/dmctl/dmctl --master-addr=172.16.6.202:8261 query-status dm-task.yaml
{
 "result": false,
 "msg": "task dm-task1 has no source or not exist",
 "sources": [
]
}
```

## Lesson 20 练习-3: Data Migration(DM)的管理

### 概述

在练习三中,我们将练习对于现有正在运行的 DM 集群进行扩容、缩容和状态查询管理。

注意: IP 地址可能会和您的环境不同, 请注意调整。

## 任务

### 1. 对于现有的 DM 集群进行扩容操作:

#### 1.1. 我们编辑扩容配置文件 dm-scale.yaml ,加入一个 worker 节点, 如下:

```
[root@centos76_vm ~] # vi dm-scale.yaml
```

加入如下内容:

```
worker_servers:
- host: 172.16.6.202
```

注意:您的环境可能和文档中有差别。

#### 1.2. 使用扩容配置文件, 扩容现有 DM 集群, 如下:

```
[root@centos76_vm ~] # tiup dm scale-out dm-test dm-scale.yaml -uroot -p
Starting component `dm`: /root/.tiup/components/dm/v1.4.4/tiup-dm scale-out dm-test dm-scale.yaml -uroot -p
Please confirm your topology:
Cluster type: dm
Cluster name: dm-test
Cluster version: v5.0.0-nightly-20210531
Role Host Ports OS/Arch Directories

dm-worker 172.16.6.202 8262 linux/x86_64 /home/tidb/dm/deploy/dm-worker-8262,/home/tidb/dm/data/dm-worker-8262
Attention:
 1. If the topology is not what you expected, check your yaml file.
 2. Please confirm there is no port/directory conflicts in same host.
Do you want to continue? [y/N]: (default=N) y
Input SSH password:
+ [Serial] - SSHKeySet: privateKey=/root/.tiup/storage/dm/clusters/dm-test/ssh/id_rsa, publicKey=/root/.tiup/storage/dm/clusters/dm-test/ssh/id_rsa.pub
- Download dm-worker:v5.0.0-nightly-20210531 (linux/amd64) ... Done
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.157
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.157
... (省略中间内容)
+ [Serial] - SystemCtl: host=172.16.6.157 action=reload prometheus-9090.service
Scaled cluster `dm-test` out successfully
```

注意:您的环境可能和文档中有差别。

## Lesson 20 练习-3: Data Migration(DM)的管理

### 1.3. 查询现有 DM 集群, 是否新的 worker 节点加入, 如下:

```
[root@centos76_vm ~] # tiup dm display dm-test
Starting component `dm`: /root/.tiup/components/dm/v1.4.4/tiup-dm display dm-test
Cluster type: dm
Cluster name: dm-test
Cluster version: v5.0.0-nightly-20210531
SSH type: builtin
```

| ID                | Role         | Host         | Ports     | OS/Arch      | Status  | Data Dir                             | Deploy Dir                             |
|-------------------|--------------|--------------|-----------|--------------|---------|--------------------------------------|----------------------------------------|
| 172.16.6.157:9093 | alertmanager | 172.16.6.157 | 9093/9094 | linux/x86_64 | Up      | /home/tidb/dm/data/alertmanager-9093 | /home/tidb/dm/deploy/alertmanager-9093 |
| 172.16.6.157:8261 | dm-master    | 172.16.6.157 | 8261/8291 | linux/x86_64 | Healthy | /home/tidb/dm/data/dm-master-8261    | /home/tidb/dm/deploy/dm-master-8261    |
| 172.16.6.202:8261 | dm-master    | 172.16.6.202 | 8261/8291 | linux/x86_64 | Healthy | /home/tidb/dm/data/dm-master-8261    | /home/tidb/dm/deploy/dm-master-8261    |
| 172.16.6.210:8261 | dm-master    | 172.16.6.210 | 8261/8291 | linux/x86_64 | Healthy | /home/tidb/dm/data/dm-master-8261    | /home/tidb/dm/deploy/dm-master-8261    |
| 172.16.6.157:8262 | dm-worker    | 172.16.6.157 | 8262      | linux/x86_64 | Bound   | /home/tidb/dm/data/dm-worker-8262    | /home/tidb/dm/deploy/dm-worker-8262    |
| 172.16.6.202:8262 | dm-worker    | 172.16.6.202 | 8262      | linux/x86_64 | Free    | /home/tidb/dm/data/dm-worker-8262    | /home/tidb/dm/deploy/dm-worker-8262    |
| 172.16.6.210:8262 | dm-worker    | 172.16.6.210 | 8262      | linux/x86_64 | Bound   | /home/tidb/dm/data/dm-worker-8262    | /home/tidb/dm/deploy/dm-worker-8262    |
| 172.16.6.157:3000 | grafana      | 172.16.6.157 | 3000      | linux/x86_64 | Up      | -                                    | /home/tidb/dm/deploy/grafana-3000      |
| 172.16.6.157:9090 | prometheus   | 172.16.6.157 | 9090      | linux/x86_64 | Up      | -                                    | /home/tidb/dm/data/prometheus-9090     |

Total nodes: 9

注意:您的环境可能和文档中有差别。

## 2. 缩容现有 DM 集群, 将 worker 节点进行下线, 如下:

```
[root@centos76_vm ~]# tiup dm scale-in dm-test -N 172.16.6.202:8262
Starting component `dm`: /root/.tiup/components/dm/v1.4.4/tiup-dm scale-in dm-test -N 172.16.6.202:8262
This operation will delete the 172.16.6.202:8262 nodes in `dm-test` and all their data.
Do you want to continue? [y/N]:(default=N) y
Scale-in nodes...
+ [Serial] - SSHKeySet: privateKey=/root/.tiup/storage/dm/clusters/dm-test/ssh/id_rsa, publicKey=/root/.tiup/storage/dm/clusters/dm-test/ssh/id_rsa.pub
... (省略中间内容)
- Regenerate config grafana -> 172.16.6.157:3000 ... Done
- Regenerate config alertmanager -> 172.16.6.157:9093 ... Done
+ [Serial] - SystemCtl: host=172.16.6.157 action=reload prometheus-9090.service
Scaled cluster `dm-test` in successfully
```

注意:您的环境可能和文档中有差别。

## Lesson 20 练习-3: Data Migration(DM)的管理

### 3. 查询现有 DM 集群, 是否指定的 worker 节点已经下线, 如下:

```
[root@centos76_vm ~]# tiup dm display dm-test
Starting component `dm`: /root/.tiup/components/dm/v1.4.4/tiup-dm display dm-test
Cluster type: dm
Cluster name: dm-test
Cluster version: v5.0.0-nightly-20210531
SSH type: builtin
```

| ID                | Role         | Host         | Ports     | OS/Arch      | Status  | Data Dir                             | Deploy Dir                             |
|-------------------|--------------|--------------|-----------|--------------|---------|--------------------------------------|----------------------------------------|
| 172.16.6.157:9093 | alertmanager | 172.16.6.157 | 9093/9094 | linux/x86_64 | Up      | /home/tidb/dm/data/alertmanager-9093 | /home/tidb/dm/deploy/alertmanager-9093 |
| 172.16.6.157:8261 | dm-master    | 172.16.6.157 | 8261/8291 | linux/x86_64 | Healthy | /home/tidb/dm/data/dm-master-8261    | /home/tidb/dm/deploy/dm-master-8261    |
| 172.16.6.202:8261 | dm-master    | 172.16.6.202 | 8261/8291 | linux/x86_64 | Healthy | /home/tidb/dm/data/dm-master-8261    | /home/tidb/dm/deploy/dm-master-8261    |
| 172.16.6.210:8261 | dm-master    | 172.16.6.210 | 8261/8291 | linux/x86_64 | Healthy | /home/tidb/dm/data/dm-master-8261    | /home/tidb/dm/deploy/dm-master-8261    |
| 172.16.6.157:8262 | dm-worker    | 172.16.6.157 | 8262      | linux/x86_64 | Bound   | /home/tidb/dm/data/dm-worker-8262    | /home/tidb/dm/deploy/dm-worker-8262    |
| 172.16.6.210:8262 | dm-worker    | 172.16.6.210 | 8262      | linux/x86_64 | Bound   | /home/tidb/dm/data/dm-worker-8262    | /home/tidb/dm/deploy/dm-worker-8262    |
| 172.16.6.157:3000 | grafana      | 172.16.6.157 | 3000      | linux/x86_64 | Up      | -                                    | /home/tidb/dm/deploy/grafana-3000      |
| 172.16.6.157:9090 | prometheus   | 172.16.6.157 | 9090      | linux/x86_64 | Up      | -                                    | /home/tidb/dm/data/prometheus-9090     |

Total nodes: 8

注意:您的环境可能和文档中有差别。

### 4. 关闭所有窗口。

## Lesson 21 练习: 数据同步工具 TiDB Binlog

© 北京平凯星辰科技发展有限公司，2021。北京平凯星辰科技发展有限公司及其关联公司（“PingCAP 集团”）保留所有权利。未经 PingCAP 集团事先书面许可，严禁复制、改编、汇编、传播或翻译本文全部或部分内容。

### Lesson 21 练习: 概述

---



## 概述

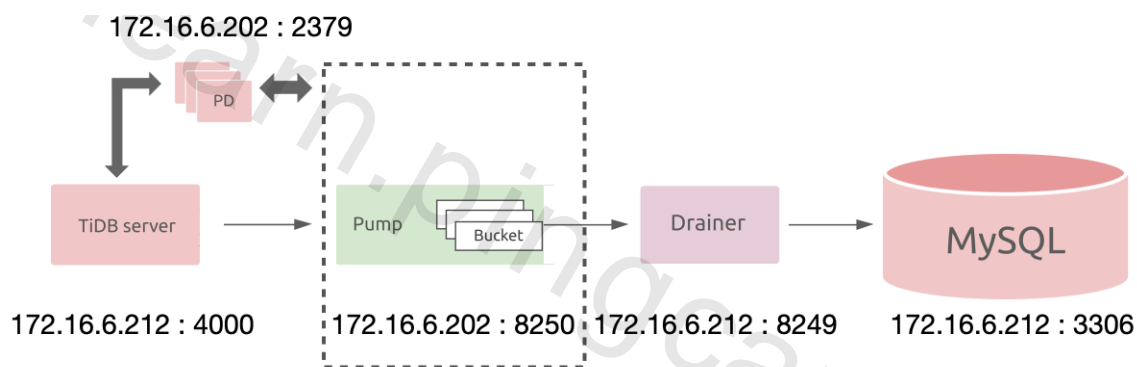
在本课练习中,我们将为已经部署好的 TiDB 数据库集群加入 TiDB Binlog 组件,用来验证数据的同步复制功能。

## 实验环境要求

1. 已经部署好一套 TiDB 数据库集群。
2. 已经部署好一套 MySQL 数据库。
3. 外部网络畅通。
4. 请清理之前实验数据。

注意:您的环境可能和实验手册不同,包括 IP 和端口号等。

拓扑结构:



## Lesson 21 练习-1: 部署 TiDB Binlog

## 概述

在本练习中，您将用 TiUP 工具为现有的 TiDB 数据库集群扩容出一个 pump 节点和 drainer 节点，为后面的数据同步复制做准备。

## 任务

22. 编辑扩容文件, 如下（注意：在 mysql 数据库中需要创建用户）：

```
mysql> create user 'root'@'172.16.6.212' identified by 'mysql';
Query OK, 0 rows affected (0.00 sec)

mysql> grant all privileges on *.* to 'root'@'172.16.6.212';
Query OK, 0 rows affected (0.00 sec)

[root@centos76_vm ~]# vi scale-out-binlog.yaml
内容如下:
pump_servers:
- host: 172.16.6.202
drainer_servers:
- host: 172.16.6.212
config:
 syncer.db-type: "mysql"
 syncer.to.host: "172.16.6.212"
 syncer.to.user: "root"
 syncer.to.password: "mysql"
 syncer.to.port: 3306
```

参数解释:

```
pump_servers:
- host: 172.16.6.202
代表 pump 节点为 172.16.6.202。
drainer_servers:
- host: 172.16.6.212
config:
 syncer.db-type: "mysql"
 syncer.to.host: "172.16.6.212"
 syncer.to.user: "root"
 syncer.to.password: "mysql"
 syncer.to.port: 3306
```

代表 drainer 节点为172.16.6.212,同时配置目标数据库(下游)的 MySQL 数据源的 IP 为 172.16.6.212 ,端口为 3306,MySQL 的复制用户为 root,密码为 mysql。

## Lesson 21 练习-1: 部署 TiDB Binlog

## 2. 准备数据, 为验证做准备。

### 2.1. 登录到 TiDB 数据库, 在 test 数据库中创建表 T, 如下:

```
mysql> create table T(id int primary key, name varchar(20));
Query OK, 0 rows affected (0.10 sec)

mysql> insert into T values(1, 'Tom');
Query OK, 1 row affected (0.01 sec)

mysql> insert into T values(2, 'Jack');
Query OK, 1 row affected (0.01 sec)

mysql> insert into T values(3, 'Frank');
Query OK, 1 row affected (0.01 sec)

mysql> select * from T;
+----+-----+
| id | name |
+----+-----+
1	Tom
2	Jack
3	Frank
+----+-----+
3 rows in set (0.00 sec)
```

### 2.2. 登录到 MySQL 数据库, 在 test 数据库中创建表 T, 如下:

```
mysql> create table T(id int primary key, name varchar(20));
Query OK, 0 rows affected (0.01 sec)

mysql> insert into T values(1, 'Tom');
Query OK, 1 row affected (0.00 sec)

mysql> insert into T values(2, 'Jack');
Query OK, 1 row affected (0.00 sec)

mysql> insert into T values(3, 'Frank');
Query OK, 1 row affected (0.00 sec)

mysql> select * from T;
+----+-----+
| id | name |
+----+-----+
1	Tom
2	Jack
3	Frank
+----+-----+
3 rows in set (0.00 sec)
```

## Lesson 21 练习-1: 部署 TiDB Binlog

### 3. 使用 tiup 组件对现有 tidb 数据库进行扩容, 增加 pump 和 drainer 节点, 如下:

```
[root@centos76_vm ~]# tiup cluster scale-out tidb-test
scale-out-binlog.yaml -uroot -p

Starting component `cluster`: /root/.tiup/components/cluster/v1.4.3/tiup-cluster scale-out tidb-test scale-out-binlog.yaml -uroot -p
Please confirm your topology:
Cluster type: tidb
Cluster name: tidb-test
Cluster version: v5.0.0
Role Host Ports OS/Arch Directories

pump 172.16.6.202 8250 linux/x86_64 /tidb-deploy/pump-8250,/tidb-data/pump-8250
drainer 172.16.6.212 8249 linux/x86_64 /tidb-deploy/drainer-8249,/tidb-data/drainer-8249
Attention:
 1. If the topology is not what you expected, check your yaml file.
 2. Please confirm there is no port/directory conflicts in same host.
Do you want to continue? [y/N]: (default=N) y
Input SSH password:
+ [Serial] - SSHKeySet: privateKey=/root/.tiup/storage/cluster/clusters/tidb-test/ssh/id_rsa,
publicKey=/root/.tiup/storage/cluster/clusters/tidb-test/ssh/id_rsa.pub

- Download pump:v5.0.0 (linux/amd64) ... Done
... (省略中间内容)
+ [Serial] - UpdateTopology: cluster=tidb-test
Scaled cluster `tidb-test` out successfully
```

## Lesson 21 练习-1: 部署 TiDB Binlog

#### 4. 扩容结束后, 查看 TiDB 数据库集群, 如下所示:

```
[root@centos76_vm ~]# tiup cluster display tidb-test
```

Starting component `cluster`: /root/.tiup/components/cluster/v1.4.3/tiup-cluster display tidb-test

Cluster type: tidb

Cluster name: tidb-test

Cluster version: v5.0.0

SSH type: builtin

Dashboard URL: http://172.16.6.210:2379/dashboard

| ID                 | Role         | Host         | Ports       | OS/Arch      | Status | Data Dir                     | Deploy Dir                     |
|--------------------|--------------|--------------|-------------|--------------|--------|------------------------------|--------------------------------|
| 172.16.6.157:9093  | alertmanager | 172.16.6.157 | 9093/9094   | linux/x86_64 | Up     | /tidb-data/alertmanager-9093 | /tidb-deploy/alertmanager-9093 |
| 172.16.6.212:8249  | drainer      | 172.16.6.212 | 8249        | linux/x86_64 | Up     | /tidb-data/drainer-8249      | /tidb-deploy/drainer-8249      |
| 172.16.6.157:3000  | grafana      | 172.16.6.157 | 3000        | linux/x86_64 | Up     | -                            | /tidb-deploy/grafana-3000      |
| 172.16.6.157:2379  | pd           | 172.16.6.157 | 2379/2380   | linux/x86_64 | Up     | /tidb-data/pd-2379           | /tidb-deploy/pd-2379           |
| 172.16.6.202:2379  | pd           | 172.16.6.202 | 2379/2380   | linux/x86_64 | Up     | /tidb-data/pd-2379           | /tidb-deploy/pd-2379           |
| 172.16.6.210:2379  | pd           | 172.16.6.210 | 2379/2380   | linux/x86_64 | Up     | /tidb-data/pd-2379           | /tidb-deploy/pd-2379           |
| 172.16.6.157:9090  | prometheus   | 172.16.6.157 | 9090        | linux/x86_64 | Up     | /tidb-data/prometheus-9090   | /tidb-deploy/prometheus-9090   |
| 172.16.6.202:8250  | pump         | 172.16.6.202 | 8250        | linux/x86_64 | Up     | /tidb-data/pump-8250         | /tidb-deploy/pump-8250         |
| 172.16.6.212:4000  | tidb         | 172.16.6.212 | 4000/10080  | linux/x86_64 | Up     | -                            | /tidb-deploy/tidb-4000         |
| 172.16.6.162:20160 | tikv         | 172.16.6.162 | 20160/20180 | linux/x86_64 | Up     | /tidb-data/tikv-20160        | /tidb-deploy/tikv-20160        |
| 172.16.6.163:20160 | tikv         | 172.16.6.163 | 20160/20180 | linux/x86_64 | Up     | /tidb-data/tikv-20160        | /tidb-deploy/tikv-20160        |
| 172.16.6.164:20160 | tikv         | 172.16.6.164 | 20160/20180 | linux/x86_64 | Up     | /tidb-data/tikv-20160        | /tidb-deploy/tikv-20160        |

Total nodes: 12

可以看到, 当前的 TiDB 数据库集群增加了 pump 节点 172.16.6.202:8250 和 drainer 节点 172.16.6.212:8249, 并且状态都为 UP。

#### 5. 启动 pump 节点和 drainer 节点后, 我们需要开启 TiDB 数据库的 binlog 日志, 我们使用 tiup cluster edit-config 命令来编辑系统变量 binlog.enable: true 和 binlog.ignore-error: true, 如下所示:

```
[root@centos76_vm ~]# tiup cluster edit-config tidb-test
```

... (省略上面内容)

```
server_configs:
 tidb:
 binlog.enable: true
 binlog.ignore-error: true
... (省略中间内容)
```

Please check change highlight above, do you want to apply the change? [y/N]:(default=N) y

Applying changes...

Applied successfully, please use `tiup cluster reload tidb-test [-N <nodes>] [-R <roles>]` to reload config.

## Lesson 21 练习-1: 部署 TiDB Binlog

6. 使用命令 `tiup cluster reload` 来载入新的配置, 如下:

```
[root@centos76_vm ~]# tiup cluster reload tidb-test
Starting component `cluster`: /root/.tiup/components/cluster/v1.4.3/tiup-cluster reload tidb-test
+ [Serial] - SSHKeySet: privateKey=/root/.tiup/storage/cluster/clusters/tidb-test/ssh/id_rsa,
publicKey=/root/.tiup/storage/cluster/clusters/tidb-test/ssh/id_rsa.pub
... (省略中间内容)
Upgrading component alertmanager
 Restarting instance 172.16.6.157
 Restart 172.16.6.157 success
Reloaded cluster `tidb-test` successfully
```

7. 登录到 TiDB 数据库, 查看 `binlog` 是否已经开启, 如下:

```
mysql> show variables like "log_bin";
+-----+-----+
| Variable_name | Value |
+-----+-----+
| log_bin | ON |
+-----+-----+
1 row in set (0.65 sec)
```

8. 登录到 TiDB 数据库, 查看 `pump` 节点和 `drainer` 节点是否正常, 如下:

```
mysql> show pump status;
+-----+-----+-----+-----+-----+
| NodeID | Address | State | Max_Commit_Ts | Update_Time |
+-----+-----+-----+-----+-----+
| 172.16.6.202:8250 | 172.16.6.202:8250 | online | 425318669436321793 | 2021-05-31 19:52:46 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> show drainer status;
+-----+-----+-----+-----+-----+
| NodeID | Address | State | Max_Commit_Ts | Update_Time |
+-----+-----+-----+-----+-----+
| 172.16.6.212:8249 | 172.16.6.212:8249 | online | 425318672582049793 | 2021-05-31 19:52:57 |
+-----+-----+-----+-----+-----+
1 row in set (0.01 sec)
```

在上面结果中, 我们会看到 `pump` 和 `drainer` 的状态都为 `online`。

## Lesson 21 练习-2: 使用 TiDB Binlog 进行数据同步复制

### 概述

在这个练习中, 您将使用 TiDB Binlog 进行 TiDB 数据库到 MySQL 数据库的同步复制。

### 任务

#### 1. 确认复制数据准备完毕

1.1. 登录到 TiDB 数据库, 查询 test 数据库下表 T, 如下:

```
mysql> select * from T;
```

| id | name  |
|----|-------|
| 1  | Tom   |
| 2  | Jack  |
| 3  | Frank |

3 rows in set (0.00 sec)

1.2. 登录到 MySQL 数据库, 查询 test 数据库下表 T, 如下:

```
mysql> select * from T;
```

| id | name  |
|----|-------|
| 1  | Tom   |
| 2  | Jack  |
| 3  | Frank |

3 rows in set (0.00 sec)

#### 2. 进行复制数据确认

2.1. 登录到 TiDB 数据库, 并插入一行数据, 如下:

```
mysql> insert into T select 4, 'Tony';
```

Query OK, 1 row affected (0.01 sec)

Records: 1 Duplicates: 0 Warnings: 0

```
mysql> select * from T;
```

| id | name  |
|----|-------|
| 1  | Tom   |
| 2  | Jack  |
| 3  | Frank |
| 4  | Tony  |

4 rows in set (0.00 sec)

## Lesson 21 练习-2: 使用 TiDB Binlog 进行数据同步复制

---

2.2. 登录到 MySQL 数据库, 确认数据行是否插入, 如下:

```
mysql> select * from T;
```

| id | name  |
|----|-------|
| 1  | Tom   |
| 2  | Jack  |
| 3  | Frank |
| 4  | Tony  |

4 rows in set (0.00 sec)

通过实验, 我们验证了 TiDB 数据库的数据变更已经复制到了 MySQL 数据库中。



## Lesson 21 练习-3: 管理 TiDB Binlog 复制

---

### 概述

在这个练习中, 您将使用 `binlogctl` 工具来管理 TiDB Binlog 的复制, 包括暂停复制和恢复复制。

### 任务

1. 您将使用 `binlogctl` 来管理 TiDB Binlog 的复制, 所以需要预先下载安装 `binlogctl` 工具, 方法如下

1.1. 下载 `tidb-v5.0.0-linux-amd64.tar.gz` 安装包, `binlogctl` 工具在里面, 如下:

```
[root@centos76_vm ~]# wget
https://download.pingcap.org/tidb-v5.0.0-linux-amd64.tar.gz
--2021-06-04 15:07:41-- https://download.pingcap.org/tidb-v5.0.0-linux-amd64.tar.gz
Resolving download.pingcap.org (download.pingcap.org)... 42.81.118.41
Connecting to download.pingcap.org (download.pingcap.org)|42.81.118.41|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 517009154 (493M) [application/x-compressed]
Saving to: 'tidb-v5.0.0-linux-amd64.tar.gz.1'

100%[=====] 517,009,154 18.6MB/s in 25s

2021-06-04 15:08:07 (19.6 MB/s) - 'tidb-v5.0.0-linux-amd64.tar.gz.1' saved [517009154/517009154]
```

## Lesson 21 练习-3: 管理 TiDB Binlog 复制

2. 解压 tidb-v5.0.0-linux-amd64.tar.gz, 获取二进制文件, 如下:

```
[root@centos76_vm ~] # tar xvf tidb-v5.0.0-linux-amd64.tar.gz
tidb-v5.0.0-linux-amd64/
tidb-v5.0.0-linux-amd64/PingCAP Community Software Agreement(Chinese Version).pdf
tidb-v5.0.0-linux-amd64/bin/
tidb-v5.0.0-linux-amd64/bin/tidb-ctl
tidb-v5.0.0-linux-amd64/bin/binlogctl
tidb-v5.0.0-linux-amd64/bin/etcctl
tidb-v5.0.0-linux-amd64/bin/pd-server
tidb-v5.0.0-linux-amd64/bin/pd-recover
tidb-v5.0.0-linux-amd64/bin/tikv-server
tidb-v5.0.0-linux-amd64/bin/arbitrator
tidb-v5.0.0-linux-amd64/bin/drain
tidb-v5.0.0-linux-amd64/bin/reparo
tidb-v5.0.0-linux-amd64/bin/pump
tidb-v5.0.0-linux-amd64/bin/tidb-server
tidb-v5.0.0-linux-amd64/bin/tikv-ctl
tidb-v5.0.0-linux-amd64/bin/pd-ctl
tidb-v5.0.0-linux-amd64/PingCAP Community Software Agreement(English Version).pdf
```

3. 检查 binlogctl 是否存在, 如下:

```
[root@centos76_vm ~] # cd tidb-v5.0.0-linux-amd64/bin/
[root@centos76_vm bin]# ls
arbitrator binlogctl drain etcdctl pd-ctl pd-recover pd-server pump reparo tidb-ctl tidb-server tikv-ctl tikv-server
```

4. 查看 pump 节点和 drainer 节点当前的状态, 如下:

4.1. 查看 pump 节点的状态, 如下:

```
[root@centos76_vm bin]# ./binlogctl -pd-urls=http://172.16.6.202:2379 -cmd pumps
[2021/05/31 20:15:26.417 +08:00] [INFO] [nodes.go:53] ["query node"] [type=pump] [node="{NodeID: 172.16.6.202:8250, Addr: 172.16.6.202:8250, State: online, MaxCommitTS: 425319025886625793, UpdateTime: 2021-05-31 20:15:25 +0800 CST}"]
```

4.2. 查看 drainer 节点的状态, 如下:

```
[root@centos76_vm bin]# ./binlogctl -pd-urls=http://172.16.6.202:2379 -cmd drainers
[2021/05/31 20:15:41.963 +08:00] [INFO] [nodes.go:53] ["query node"] [type=drainer] [node="{NodeID: 172.16.6.212:8249, Addr: 172.16.6.212:8249, State: online, MaxCommitTS: 425319029818785793, UpdateTime: 2021-05-31 20:15:40 +0800 CST}"]
```

## Lesson 21 练习-3: 管理 TiDB Binlog 复制

5. 暂停 drainer 节点, 如下:

```
[root@centos76_vm bin]# ./binlogctl -pd-urls=http://172.16.6.202:2379
-cmd pause-drainer -node-id 172.16.6.212:8249
[2021/05/31 21:06:34.684 +08:00] [INFO] [nodes.go:123] ["Apply action on node success"] [action=pause]
[NodeID=172.16.6.212:8249]
```

6. 确认暂停的 drainer 节点状态, 如下:

```
[root@centos76_vm bin]# ./binlogctl -pd-urls=http://172.16.6.202:2379
-cmd drainers
[2021/05/31 21:06:41.154 +08:00] [INFO] [nodes.go:53] ["query node"] [type=drainer] [node="{NodeID: 172.16.6.212:8249, Addr:
172.16.6.212:8249, State: paused, MaxCommitTS: 425319396492705793, UpdateTime: 2021-05-31 20:39:00 +0800 CST}"]]
```

我们发现 drainer 节点 172.16.6.212:8249 的 State 已经为 paused。

7. 确认复制是否还可以继续。

7.1. 登录到 TiDB 数据库, 向 test 库的表 T 中插入一行数据, 如下:

```
mysql> insert into T values(6, 'Andy');
```

Query OK, 1 row affected (0.01 sec)

```
mysql> select * from T;
```

| id | name  |
|----|-------|
| 1  | Tom   |
| 2  | Jack  |
| 3  | Frank |
| 4  | Tony  |
| 6  | Andy  |

5 rows in set (0.00 sec)

7.2. 登录到 MySQL 数据库, 查询 test 库的表 T, 如下:

```
mysql> select * from T;
```

| id | name  |
|----|-------|
| 1  | Tom   |
| 2  | Jack  |
| 3  | Frank |
| 4  | Tony  |

4 rows in set (0.00 sec)

从结果上看, drainer 节点停止后, 复制已经停止了。

8. 重新启动 drainer 节点, 如下:

```
[root@centos76_vm ~] cd /tidb-deploy/drainder-8249/bin/
[root@centos76_vm bin]# ./drainer -pd-urls=http://172.16.6.202:2379
-config ../conf/drainder.toml
```

## Lesson 21 练习-3: 管理 TiDB Binlog 复制

---

9. 查看 MySQL 数据库, 查看复制是否继续, 如下:

```
mysql> select * from T;
```

| id | name  |
|----|-------|
| 1  | Tom   |
| 2  | Jack  |
| 3  | Frank |
| 4  | Tony  |
| 6  | Andy  |

5 rows in set (0.00 sec)

您可以看到, 数据复制已经恢复了。

## Lesson 21 练习-4: 缩容 TiDB Binlog 节点

---

### 概述

在这个练习中, 您将对现有的 TiDB Binlog 进行缩容。

### 任务

#### 1. 先关闭 tidb 数据库的 binlog 功能。

1.1.使用 tiup cluster edit-config 设置 binlog.enable 和 binlog.ignore-error 为 false,如下:

```
[root@centos76_vm ~]# tiup cluster edit-config tidb-test
Starting component `cluster`: /root/.tiup/components/cluster/v1.4.3/tiup-cluster edit-config tide-test
... (省略中间内容)
server_configs:
 tidb:
 binlog.enable: false
 binlog.ignore-error: false {}
... (省略中间内容)
Please check change highlight above, do you want to apply the change? [y/N]:(default=N) y
Applying changes...
Applied successfully, please use `tiup cluster reload tidb-test [-N <nodes>] [-R <roles>]` to reload config.
```

1.2.使用 tiup cluster edit-config 设置 binlog.enable 和 binlog.ignore-error 为 false,如下:

```
[root@centos76_vm ~]# tiup cluster reload tidb-test
Starting component `cluster`: /root/.tiup/components/cluster/v1.4.3/tiup-cluster reload tidb-test
+ [Serial] - SSHKeySet: privateKey=/root/.tiup/storage/cluster/clusters/tidb-test/ssh/id_rsa,
publicKey=/root/.tiup/storage/cluster/clusters/tidb-test/ssh/id_rsa.pub
... (省略中间内容)
Upgrading component alertmanager
 Restarting instance 172.16.6.157
 Restart 172.16.6.157 success
Reloaded cluster `tidb-test` successfully
```

## Lesson 21 练习-4: 缩容 TiDB Binlog 节点

2. 查看缩容 drainer 和 pump 节点后的 TiDB 数据库集群状态, 如下:

```
[root@centos76_vm bin]# tiup cluster display tidb-test
Starting component `cluster`: /root/.tiup/components/cluster/v1.4.3/tiup-cluster display tidb-test
Cluster type: tidb
Cluster name: tidb-test
Cluster version: v5.0.0
SSH type: builtin
Dashboard URL: http://172.16.6.210:2379/dashboard
```

| ID                 | Role         | Host         | Ports       | OS/Arch      | Status | Data Dir                     | Deploy Dir                     |
|--------------------|--------------|--------------|-------------|--------------|--------|------------------------------|--------------------------------|
| 172.16.6.157:9093  | alertmanager | 172.16.6.157 | 9093/9094   | linux/x86_64 | Up     | /tidb-data/alertmanager-9093 | /tidb-deploy/alertmanager-9093 |
| 172.16.6.212:8249  | drainer      | 172.16.6.212 | 8249        | linux/x86_64 | Down   | /tidb-data/drainer-8249      | /tidb-deploy/drainer-8249      |
| 172.16.6.157:3000  | grafana      | 172.16.6.157 | 3000        | linux/x86_64 | Up     | -                            | /tidb-deploy/grafana-3000      |
| 172.16.6.157:2379  | pd           | 172.16.6.157 | 2379/2380   | linux/x86_64 | Up     | /tidb-data/pd-2379           | /tidb-deploy/pd-2379           |
| 172.16.6.202:2379  | pd           | 172.16.6.202 | 2379/2380   | linux/x86_64 | Up     | /tidb-data/pd-2379           | /tidb-deploy/pd-2379           |
| 172.16.6.210:2379  | pd           | 172.16.6.210 | 2379/2380   | linux/x86_64 | Up     | /tidb-data/pd-2379           | /tidb-deploy/pd-2379           |
| 172.16.6.157:9090  | prometheus   | 172.16.6.157 | 9090        | linux/x86_64 | Up     | /tidb-data/prometheus-9090   | /tidb-deploy/prometheus-9090   |
| 172.16.6.202:8250  | pump         | 172.16.6.202 | 8250        | linux/x86_64 | Down   | /tidb-data/pump-8250         | /tidb-deploy/pump-8250         |
| 172.16.6.212:4000  | tidb         | 172.16.6.212 | 4000/10080  | linux/x86_64 | Up     | -                            | /tidb-deploy/tidb-4000         |
| 172.16.6.162:20160 | tikv         | 172.16.6.162 | 20160/20180 | linux/x86_64 | Up     | /tidb-data/tikv-20160        | /tidb-deploy/tikv-20160        |
| 172.16.6.163:20160 | tikv         | 172.16.6.163 | 20160/20180 | linux/x86_64 | Up     | /tidb-data/tikv-20160        | /tidb-deploy/tikv-20160        |
| 172.16.6.164:20160 | tikv         | 172.16.6.164 | 20160/20180 | linux/x86_64 | Up     | /tidb-data/tikv-20160        | /tidb-deploy/tikv-20160        |

Total nodes: 12

## Lesson 21 练习-4: 缩容 TiDB Binlog 节点

---

3. 缩容当前集群中的 drainer 节点 172.16.6.212 8249, 如下:

learn.pingcap.cc

```
[root@centos76_vm bin]# tiup cluster scale-in tidb-test --node 172.16.6.212:8249
Starting component `cluster`: /root/.tiup/components/cluster/v1.4.3/tiup-cluster scale-in tidb-test --node 172.16.6.212:8249
This operation will delete the 172.16.6.212:8249 nodes in `tidb-test` and all their data.
Do you want to continue? [y/N]:(default=N) y
Scale-in nodes...
+ [Serial] - SSHKeySet: privateKey=/root/.tiup/storage/cluster/clusters/tidb-test/ssh/id_rsa,
publicKey=/root/.tiup/storage/cluster/clusters/tidb-test/ssh/id_rsa.pub
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.157
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.164
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.212
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.202
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.157
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.212
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.157
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.210
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.157
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.162
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.202
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.163
+ [Serial] - ClusterOperate: operation=ScaleInOperation, options={Roles:[] Nodes:[172.16.6.212:8249] Force:false SSHTimeout:5
OptTimeout:120 APITimeout:300 IgnoreConfigCheck:false NativeSSH:false SSHType: CleanupData:false CleanupLog:false
RetainDataRoles:[] RetainDataNodes:[] Operation:StartOperation}
The component `drainer` will become tombstone, maybe exists in several minutes or hours, after that you can use the prune
command to clean it
+ [Serial] - UpdateMeta: cluster=tidb-test, deleted=""
+ [Serial] - UpdateTopology: cluster=tidb-test
+ Refresh instance configs
- Regenerate config pd -> 172.16.6.202:2379 ... Done
- Regenerate config pd -> 172.16.6.157:2379 ... Done
- Regenerate config pd -> 172.16.6.210:2379 ... Done
- Regenerate config tikv -> 172.16.6.162:20160 ... Done
- Regenerate config tikv -> 172.16.6.163:20160 ... Done
- Regenerate config tikv -> 172.16.6.164:20160 ... Done
- Regenerate config pump -> 172.16.6.202:8250 ... Done
- Regenerate config tidb -> 172.16.6.212:4000 ... Done
- Regenerate config prometheus -> 172.16.6.157:9090 ... Done
- Regenerate config grafana -> 172.16.6.157:3000 ... Done
- Regenerate config alertmanager -> 172.16.6.157:9093 ... Done
+ [Serial] - SystemCtl: host=172.16.6.157 action=reload prometheus-9090.service
Scaled cluster `tidb-test` in successfully
```

## Lesson 21 练习-4: 缩容 TiDB Binlog 节点

### 4. 查看缩容后的集群状态, 如下:



```
[root@centos76_vm bin]# tiup cluster display tidb-test
```

Starting component `cluster`: /root/.tiup/components/cluster/v1.4.3/tiup-cluster display tidb-test

Cluster type: tidb

Cluster name: tidb-test

Cluster version: v5.0.0

SSH type: builtin

Dashboard URL: http://172.16.6.210:2379/dashboard

| ID                 | Role         | Host         | Ports       | OS/Arch      | Status | Data Dir                     | Deploy Dir                     |
|--------------------|--------------|--------------|-------------|--------------|--------|------------------------------|--------------------------------|
| 172.16.6.157:9093  | alertmanager | 172.16.6.157 | 9093/9094   | linux/x86_64 | Up     | /tidb-data/alertmanager-9093 | /tidb-deploy/alertmanager-9093 |
| 172.16.6.212:8249  | drainer      | 172.16.6.212 | 8249        | linux/x86_64 | Down   | /tidb-data/drainer-8249      | /tidb-deploy/drainer-8249      |
| 172.16.6.157:3000  | grafana      | 172.16.6.157 | 3000        | linux/x86_64 | Up     | -                            | /tidb-deploy/grafana-3000      |
| 172.16.6.157:2379  | pd           | 172.16.6.157 | 2379/2380   | linux/x86_64 | Up     | /tidb-data/pd-2379           | /tidb-deploy/pd-2379           |
| 172.16.6.202:2379  | pd           | 172.16.6.202 | 2379/2380   | linux/x86_64 | Up     | /tidb-data/pd-2379           | /tidb-deploy/pd-2379           |
| 172.16.6.210:2379  | pd           | 172.16.6.210 | 2379/2380   | linux/x86_64 | Up     | /tidb-data/pd-2379           | /tidb-deploy/pd-2379           |
| 172.16.6.157:9090  | prometheus   | 172.16.6.157 | 9090        | linux/x86_64 | Up     | /tidb-data/prometheus-9090   | /tidb-deploy/prometheus-9090   |
| 172.16.6.202:8250  | pump         | 172.16.6.202 | 8250        | linux/x86_64 | Up     | /tidb-data/pump-8250         | /tidb-deploy/pump-8250         |
| 172.16.6.212:4000  | tidb         | 172.16.6.212 | 4000/10080  | linux/x86_64 | Up     | -                            | /tidb-deploy/tidb-4000         |
| 172.16.6.162:20160 | tikv         | 172.16.6.162 | 20160/20180 | linux/x86_64 | Up     | /tidb-data/tikv-20160        | /tidb-deploy/tikv-20160        |
| 172.16.6.163:20160 | tikv         | 172.16.6.163 | 20160/20180 | linux/x86_64 | Up     | /tidb-data/tikv-20160        | /tidb-deploy/tikv-20160        |
| 172.16.6.164:20160 | tikv         | 172.16.6.164 | 20160/20180 | linux/x86_64 | Up     | /tidb-data/tikv-20160        | /tidb-deploy/tikv-20160        |

Total nodes: 12

There are some nodes can be pruned:

Nodes: [172.16.6.212:8249]

You can destroy them with the command: `tiup cluster prune tidb-test`

## Lesson 21 练习-4: 缩容 TiDB Binlog 节点

5. 缩容当前集群中的 pump 节点 172.16.6.202 8250，如下：

```
[root@centos76_vm bin]# tiup cluster scale-in tidb-test --node 172.16.6.202:8250
Found cluster newer version:

The latest version: v1.4.4
Local installed version: v1.4.3
Update current component: tiup update cluster
Update all components: tiup update --all

Starting component `cluster`: /root/.tiup/components/cluster/v1.4.3/tiup-cluster scale-in tidb-test --node 172.16.6.202:8250
This operation will delete the 172.16.6.202:8250 nodes in `tidb-test` and all their data.
Do you want to continue? [y/N]:(default=N) y
Scale-in nodes...
+ [Serial] - SSHKeySet: privateKey=/root/.tiup/storage/cluster/clusters/tidb-test/ssh/id_rsa,
publicKey=/root/.tiup/storage/cluster/clusters/tidb-test/ssh/id_rsa.pub
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.157
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.210
... (省略中间内容)
- Regenerate config grafana -> 172.16.6.157:3000 ... Done
- Regenerate config alertmanager -> 172.16.6.157:9093 ... Done
+ [Serial] - SystemCtl: host=172.16.6.157 action=reload prometheus-9090.service
Scaled cluster `tidb-test` in successfully
```

## Lesson 21 练习-4: 缩容 TiDB Binlog 节点

6. 查看缩容 drainer 和 pump 节点后的 TiDB 数据库集群状态, 如下:

```
[root@centos76_vm bin]# tiup cluster display tidb-test
```

Starting component `cluster`: /root/.tiup/components/cluster/v1.4.3/tiup-cluster display tidb-test

Cluster type: tidb

Cluster name: tidb-test

Cluster version: v5.0.0

SSH type: builtin

Dashboard URL: http://172.16.6.210:2379/dashboard

| ID                 | Role         | Host         | Ports       | OS/Arch      | Status | Data Dir                     | Deploy Dir                     |
|--------------------|--------------|--------------|-------------|--------------|--------|------------------------------|--------------------------------|
| 172.16.6.157:9093  | alertmanager | 172.16.6.157 | 9093/9094   | linux/x86_64 | Up     | /tidb-data/alertmanager-9093 | /tidb-deploy/alertmanager-9093 |
| 172.16.6.212:8249  | drainer      | 172.16.6.212 | 8249        | linux/x86_64 | Down   | /tidb-data/drainer-8249      | /tidb-deploy/drainer-8249      |
| 172.16.6.157:3000  | grafana      | 172.16.6.157 | 3000        | linux/x86_64 | Up     | -                            | /tidb-deploy/grafana-3000      |
| 172.16.6.157:2379  | pd           | 172.16.6.157 | 2379/2380   | linux/x86_64 | Up     | /tidb-data/pd-2379           | /tidb-deploy/pd-2379           |
| 172.16.6.202:2379  | pd           | 172.16.6.202 | 2379/2380   | linux/x86_64 | Up     | /tidb-data/pd-2379           | /tidb-deploy/pd-2379           |
| 172.16.6.210:2379  | pd           | 172.16.6.210 | 2379/2380   | linux/x86_64 | Up     | /tidb-data/pd-2379           | /tidb-deploy/pd-2379           |
| 172.16.6.157:9090  | prometheus   | 172.16.6.157 | 9090        | linux/x86_64 | Up     | /tidb-data/prometheus-9090   | /tidb-deploy/prometheus-9090   |
| 172.16.6.202:8250  | pump         | 172.16.6.202 | 8250        | linux/x86_64 | Down   | /tidb-data/pump-8250         | /tidb-deploy/pump-8250         |
| 172.16.6.212:4000  | tidb         | 172.16.6.212 | 4000/10080  | linux/x86_64 | Up     | -                            | /tidb-deploy/tidb-4000         |
| 172.16.6.162:20160 | tikv         | 172.16.6.162 | 20160/20180 | linux/x86_64 | Up     | /tidb-data/tikv-20160        | /tidb-deploy/tikv-20160        |
| 172.16.6.163:20160 | tikv         | 172.16.6.163 | 20160/20180 | linux/x86_64 | Up     | /tidb-data/tikv-20160        | /tidb-deploy/tikv-20160        |
| 172.16.6.164:20160 | tikv         | 172.16.6.164 | 20160/20180 | linux/x86_64 | Up     | /tidb-data/tikv-20160        | /tidb-deploy/tikv-20160        |

Total nodes: 12

There are some nodes can be pruned:

Nodes: [172.16.6.202:8250 172.16.6.212:8249]

You can destroy them with the command: `tiup cluster prune tidb-test`

## 7. 使用 tiup cluster prune 命令清理节点。:

```
[root@centos76_vm bin]# tiup cluster prune tidb-test
```

Starting component `cluster`: /root/.tiup/components/cluster/v1.4.3/tiup-cluster prune tidb-test

+ [ Serial ] - SSHKeySet: privateKey=/root/.tiup/storage/cluster/clusters/tidb-test/ssh/id\_rsa, publicKey=/root/.tiup/storage/cluster/clusters/tidb-test/ssh/id\_rsa.pub

+ [Parallel] - UserSSH: user=tidb, host=172.16.6.157

+ [Parallel] - UserSSH: user=tidb, host=172.16.6.164

... (省略中间内容)

- Regenerate config grafana -> 172.16.6.157:3000 ... Done

- Regenerate config alertmanager -> 172.16.6.157:9093 ... Done

+ [ Serial ] - SystemCtl: host=172.16.6.157 action=reload prometheus-9090.service

Destroy success

## Lesson 21 练习-4: 缩容 TiDB Binlog 节点

### 8. 查看集群状态, 发现 pump 和 drainer 节点已经下线成功, 如下:

```
[root@centos76_vm bin]# tiup cluster display tidb-test
```

Starting component `cluster`: /root/.tiup/components/cluster/v1.4.3/tiup-cluster display tidb-test

Cluster type: tidb

Cluster name: tidb-test

Cluster version: v5.0.0

SSH type: builtin

Dashboard URL: http://172.16.6.210:2379/dashboard

| ID                 | Role         | Host         | Ports       | OS/Arch      | Status | Data Dir                     | Deploy Dir                     |
|--------------------|--------------|--------------|-------------|--------------|--------|------------------------------|--------------------------------|
| 172.16.6.157:9093  | alertmanager | 172.16.6.157 | 9093/9094   | linux/x86_64 | Up     | /tidb-data/alertmanager-9093 | /tidb-deploy/alertmanager-9093 |
| 172.16.6.157:3000  | grafana      | 172.16.6.157 | 3000        | linux/x86_64 | Up     | -                            | /tidb-deploy/grafana-3000      |
| 172.16.6.157:2379  | pd           | 172.16.6.157 | 2379/2380   | linux/x86_64 | Up     | /tidb-data/pd-2379           | /tidb-deploy/pd-2379           |
| 172.16.6.202:2379  | pd           | 172.16.6.202 | 2379/2380   | linux/x86_64 | Up L   | /tidb-data/pd-2379           | /tidb-deploy/pd-2379           |
| 172.16.6.210:2379  | pd           | 172.16.6.210 | 2379/2380   | linux/x86_64 | Up UI  | /tidb-data/pd-2379           | /tidb-deploy/pd-2379           |
| 172.16.6.157:9090  | prometheus   | 172.16.6.157 | 9090        | linux/x86_64 | Up     | /tidb-data/prometheus-9090   | /tidb-deploy/prometheus-9090   |
| 172.16.6.212:4000  | tidb         | 172.16.6.212 | 4000/10080  | linux/x86_64 | Up     | -                            | /tidb-deploy/tidb-4000         |
| 172.16.6.162:20160 | tikv         | 172.16.6.162 | 20160/20180 | linux/x86_64 | Up     | /tidb-data/tikv-20160        | /tidb-deploy/tikv-20160        |
| 172.16.6.163:20160 | tikv         | 172.16.6.163 | 20160/20180 | linux/x86_64 | Up     | /tidb-data/tikv-20160        | /tidb-deploy/tikv-20160        |
| 172.16.6.164:20160 | tikv         | 172.16.6.164 | 20160/20180 | linux/x86_64 | Up     | /tidb-data/tikv-20160        | /tidb-deploy/tikv-20160        |

Total nodes: 10

9. 关闭所有窗口。

## Lesson 22 练习: 数据同步工具 TiCDC

© 北京平凯星辰科技发展有限公司，2021。北京平凯星辰科技发展有限公司及其关联公司（“PingCAP 集团”）保留所有权利。未经 PingCAP 集团事先书面许可，严禁复制、改编、汇编、传播或翻译本文全部或部分内容。

### Lesson 22 练习: 概述

---

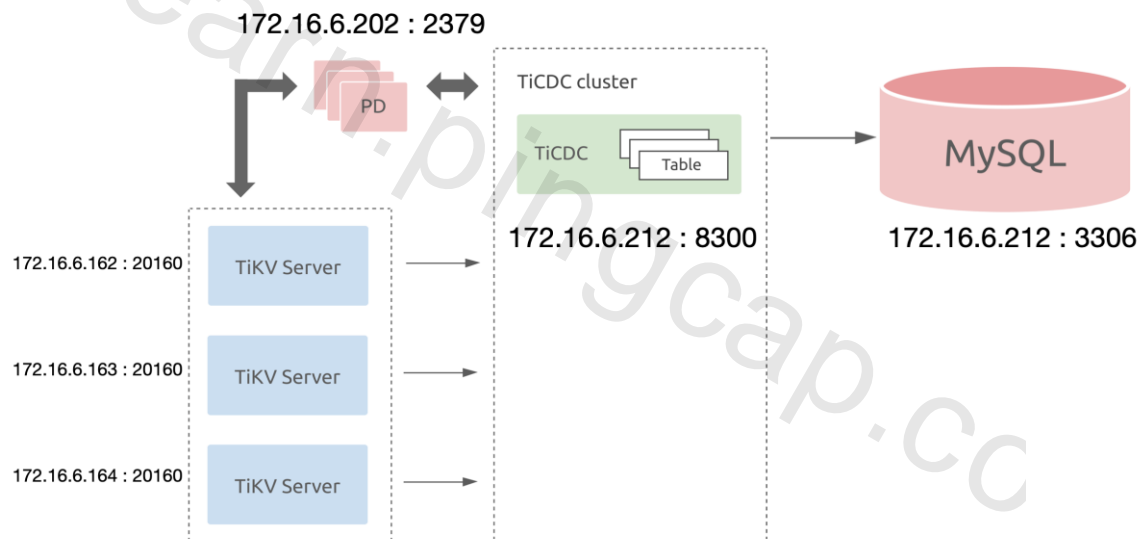
## 概述：

本课练习中，您会在已有的 TiDB 数据库集群中增加 TiCDC 节点，之后进行数据的同步工作，之后将添加的 TiCDC 节点进行缩容。

## 实验环境要求：

1. 部署好 1 个 TiDB 数据库集群，集群名为 tidb-test。
2. 部署好 1 个 MySQL 数据库实例。
3. tiup 组件部署完毕。
4. 可以连接到互联网。
5. 清理掉之前的实验数据。

## 拓扑结构图如下：



## Lesson 22 练习-1: 为原有 TiDB 数据库集群部署 TiCDC

### 概述

在这个练习中, 我们将在已有 TiDB 数据库集群中增加 TiCDC 节点, 为下面的数据同步练习做准备。

## 任务

23. 查看当前已有 TiDB 数据库集群状态, 如下所示:

```
[root@centos76_vm ~]# tiup cluster display tidb-test
Starting component `cluster`: /root/.tiup/components/cluster/v1.4.4/tiup-cluster display tidb-test
Cluster type: tidb
Cluster name: tidb-test
Cluster version: v5.0.0
SSH type: builtin
Dashboard URL: http://172.16.6.157:2379/dashboard
```

| ID                 | Role         | Host         | Ports       | OS/Arch      | Status  | Data Dir                     | Deploy Dir                     |
|--------------------|--------------|--------------|-------------|--------------|---------|------------------------------|--------------------------------|
| 172.16.6.157:9093  | alertmanager | 172.16.6.157 | 9093/9094   | linux/x86_64 | Up      | /tidb-data/alertmanager-9093 | /tidb-deploy/alertmanager-9093 |
| 172.16.6.157:3000  | grafana      | 172.16.6.157 | 3000        | linux/x86_64 | Up      | -                            | /tidb-deploy/grafana-3000      |
| 172.16.6.157:2379  | pd           | 172.16.6.157 | 2379/2380   | linux/x86_64 | Up L UI | /tidb-data/pd-2379           | /tidb-deploy/pd-2379           |
| 172.16.6.202:2379  | pd           | 172.16.6.202 | 2379/2380   | linux/x86_64 | Up      | /tidb-data/pd-2379           | /tidb-deploy/pd-2379           |
| 172.16.6.210:2379  | pd           | 172.16.6.210 | 2379/2380   | linux/x86_64 | Up      | /tidb-data/pd-2379           | /tidb-deploy/pd-2379           |
| 172.16.6.157:9090  | prometheus   | 172.16.6.157 | 9090        | linux/x86_64 | Up      | /tidb-data/prometheus-9090   | /tidb-deploy/prometheus-9090   |
| 172.16.6.212:4000  | tidb         | 172.16.6.212 | 4000/10080  | linux/x86_64 | Up      | -                            | /tidb-deploy/tidb-4000         |
| 172.16.6.162:20160 | tikv         | 172.16.6.162 | 20160/20180 | linux/x86_64 | Up      | /tidb-data/tikv-20160        | /tidb-deploy/tikv-20160        |
| 172.16.6.163:20160 | tikv         | 172.16.6.163 | 20160/20180 | linux/x86_64 | Up      | /tidb-data/tikv-20160        | /tidb-deploy/tikv-20160        |
| 172.16.6.164:20160 | tikv         | 172.16.6.164 | 20160/20180 | linux/x86_64 | Up      | /tidb-data/tikv-20160        | /tidb-deploy/tikv-20160        |

Total nodes: 10

## Lesson 22 练习-1: 为原有 TiDB 数据库集群部署 TiCDC

24. 编辑扩容配置文件, 准备将 TiCDC 节点 172.16.6.212 加入到集群中去。如下:

```
[root@centos76_vm ~]# vi scale-out.yaml
```

内容:

```
cdc_servers:
- host: 172.16.6.212
 port: 8300
 deploy_dir: "/tidb-deploy/cdc-8300"
 log_dir: "/tidb-deploy/cdc-8300/log"
```

注意:

1. 您的环境可能和实验手册不一致。
2. 加入 1 个 TiCDC 节点, IP 为 172.16.6.212, 端口为 8300, 软件部署在 /tidb-deploy/cdc-8300 中, 日志部署在 /tidb-deploy/cdc-8300/log 中。
3. 因为只有一个集群节点, 所以这个 TiCDC 集群不具备高可用性。

25. 使用 tiup 为原有 TiDB 数据库集群扩容 TiCDC 节点。如下:

```
[root@centos76_vm ~]# tiup cluster scale-out tidb-test scale-out.yaml -uroot -p
```

```
Starting component `cluster`: /root/.tiup/components/cluster/v1.4.4/tiup-cluster scale-out tidb-test scale-out.yaml -uroot -p
```

```
Please confirm your topology:
```

```
Cluster type: tidb
```

```
Cluster name: tidb-test
```

```
Cluster version: v5.0.0
```

```
Role Host Ports OS/Arch Directories
```

```

```

```
cdc 172.16.6.212 8300 linux/x86_64 /tidb-deploy/cdc-8300
```

```
Attention:
```

1. If the topology is not what you expected, check your yaml file.
2. Please confirm there is no port/directory conflicts in same host.

```
Do you want to continue? [y/N]: (default=N) y
```

```
Input SSH password:
```

```
+ [Serial] - SSHKeySet: privateKey=/root/.tiup/storage/cluster/clusters/tidb-test/ssh/id_rsa,
publicKey=/root/.tiup/storage/cluster/clusters/tidb-test/ssh/id_rsa.pub
```

```
- Download cdc:v5.0.0 (linux/amd64) ... Done
```

```
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.157
```

```
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.212
```

```
... (省略中间内容)
```

```
cache_dir=/root/.tiup/storage/cluster/clusters/tidb-test/config-cache
```

```
+ [Serial] - SystemCtl: host=172.16.6.157 action=reload prometheus-9090.service
```

```
+ [Serial] - UpdateTopology: cluster=tidb-test
```

```
Scaled cluster `tidb-test` out successfully
```

## Lesson 22 练习-1: 为原有 TiDB 数据库集群部署 TiCDC



26. 扩容完毕后, 检查 TiDB 数据库集群状态, 检查 TiCDC 集群状态。如下:

```
[root@centos76_vm ~]# tiup cluster display tidb-test
Starting component `cluster`: /root/.tiup/components/cluster/v1.4.4/tiup-cluster display tidb-test
Cluster type: tidb
Cluster name: tidb-test
Cluster version: v5.0.0
SSH type: builtin
Dashboard URL: http://172.16.6.157:2379/dashboard
```

| ID                 | Role         | Host         | Ports       | OS/Arch      | Status | Data Dir                     | Deploy Dir                     |
|--------------------|--------------|--------------|-------------|--------------|--------|------------------------------|--------------------------------|
| 172.16.6.157:9093  | alertmanager | 172.16.6.157 | 9093/9094   | linux/x86_64 | Up     | /tidb-data/alertmanager-9093 | /tidb-deploy/alertmanager-9093 |
| 172.16.6.212:8300  | cdc          | 172.16.6.212 | 8300        | linux/x86_64 | Up     | -                            | /tidb-deploy/cdc-8300          |
| 172.16.6.157:3000  | grafana      | 172.16.6.157 | 3000        | linux/x86_64 | Up     | -                            | /tidb-deploy/grafana-3000      |
| 172.16.6.157:2379  | pd           | 172.16.6.157 | 2379/2380   | linux/x86_64 | Up     | /tidb-data/pd-2379           | /tidb-deploy/pd-2379           |
| 172.16.6.202:2379  | pd           | 172.16.6.202 | 2379/2380   | linux/x86_64 | Up     | /tidb-data/pd-2379           | /tidb-deploy/pd-2379           |
| 172.16.6.210:2379  | pd           | 172.16.6.210 | 2379/2380   | linux/x86_64 | Up     | /tidb-data/pd-2379           | /tidb-deploy/pd-2379           |
| 172.16.6.157:9090  | prometheus   | 172.16.6.157 | 9090        | linux/x86_64 | Up     | /tidb-data/prometheus-9090   | /tidb-deploy/prometheus-9090   |
| 172.16.6.212:4000  | tidb         | 172.16.6.212 | 4000/10080  | linux/x86_64 | Up     | -                            | /tidb-deploy/tidb-4000         |
| 172.16.6.162:20160 | tikv         | 172.16.6.162 | 20160/20180 | linux/x86_64 | Up     | /tidb-data/tikv-20160        | /tidb-deploy/tikv-20160        |
| 172.16.6.163:20160 | tikv         | 172.16.6.163 | 20160/20180 | linux/x86_64 | Up     | /tidb-data/tikv-20160        | /tidb-deploy/tikv-20160        |
| 172.16.6.164:20160 | tikv         | 172.16.6.164 | 20160/20180 | linux/x86_64 | Up     | /tidb-data/tikv-20160        | /tidb-deploy/tikv-20160        |

```
Total nodes: 11
```

我们看到 TiCDC 集群的 ID 为 172.16.6.212:8300, Status(状态)为 UP, 表示 TiCDC 部署成功。可以进行下面的实验。

27. 使用 tiup ctl:v5.0.0 cdc 检查 TiCDC 的状态, 如下:

```
[root@centos76_vm ~]# tiup ctl:v5.0.0 cdc capture list --pd=http://172.16.6.202:2379
Starting component `ctl`: /root/.tiup/components/ctl/v5.0.0/ctl cdc capture list --pd=http://172.16.6.202:2379
[
 {
 "id": "df74147f-f3cc-4789-83be-c1f9ffc31369",
 "is-owner": true,
 "address": "172.16.6.212:8300"
 }
]
```

注意:

1. 命令中 --pd=http://172.16.6.202:2379, 可以是任何一个 PD 节点。
2. "is-owner": true 代表当 TiCDC 节点为 owner 节点。

## Lesson 22 练习-2: 创建 TiDB 数据库到 MySQL 数据库的同步任务, 并检查任务状态

## 概述

在这个练习中, 您将为已经为源库(上游) TiDB 数据库到 MySQL 数据库(下游)创建同步任务, 开启数据同步, 并检查任务状态。

## 任务

1. 为 MySQL 数据库(端口号为 3306)加入时区信息, 创建数据库 test, 并创建表 T1, 注意不插入数据, 如下操作:

```
[root@centos76_vm ~]# mysql_tzinfo_to_sql /usr/share/zoneinfo | mysql
-u root -p mysql -S /data/mydb3306/mysql.sock
Enter password:
Warning: Unable to load '/usr/share/zoneinfo/iso3166.tab' as time zone. Skipping it.
Warning: Unable to load '/usr/share/zoneinfo/leapseconds' as time zone. Skipping it.
Warning: Unable to load '/usr/share/zoneinfo/tzdata.zi' as time zone. Skipping it.
Warning: Unable to load '/usr/share/zoneinfo/zone.tab' as time zone. Skipping it.
Warning: Unable to load '/usr/share/zoneinfo/zone1970.tab' as time zone. Skipping it.
```

登录 MySQL 数据库后执行如下命令:

```
[root@centos76_vm ~]# mysql -uroot -p'mysql' -S /data/mydb3306/mysql.sock
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 98
Server version: 5.7.34-log MySQL Community Server (GPL)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

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> create database test;
Query OK, 1 row affected (0.00 sec)
```

## Lesson 22 练习-2: 创建 TiDB 数据库到 MySQL 数据库的同步任务, 并检查任务状态

```
mysql> use test;
Database changed
mysql> create table T1(id int primary key, name varchar(20));
Query OK, 0 rows affected (0.02 sec)

mysql> select * from T1;
Empty set (0.01 sec)
```

2. 准备 TiDB 数据库, 创建数据库 test, 并创建表 T1, 注意不插入数据, 如下操作:

```
mysql> create database test;
Query OK, 0 rows affected (0.09 sec)

mysql> use test;
Database changed
mysql> create table T1(id int primary key, name varchar(20));
Query OK, 0 rows affected (0.11 sec)

mysql> select * from T1;
Empty set (0.01 sec)
```

3. 进入到刚刚部署的 TiCDC 节点 172.16.6.212:8300, 开启数据同步任务, 如下:

```
[root@centos76_vm ~]# cd /tidb-deploy/cdc-8300/bin/
[root@centos76_vm bin]# ls
cdc
[root@centos76_vm bin]# ./cdc cli changefeed create
--pd=http://172.16.6.202:2379
--sink-uri="mysql://root:mysql@172.16.6.212:3306/"
--changefeed-id="replication-task-1" --sort-engine="unified"
[WARN] some tables are not eligible to replicate, []model.TableName{model.TableName{Schema:"test", Table:"T1", TableID:0,
IsPartition:false}}
Could you agree to ignore those tables, and continue to replicate [Y/N]
y
Create changefeed successfully!
ID: replication-task-1

Info: {"sink-uri":"mysql://root:mysql@172.16.6.212:3306/", "opts":{"_changefeed_id":"cli-verify", "create-time":"2021-05-31T13:17:49.507258259+08:00", "start-ts":425312457708535810, "target-ts":0, "admin-job-type":0, "sort-engine":"unified", "sort-dir":"","config":{"case-sensitive":true, "enable-old-value":true, "force-replicate":false, "check-gc-safe-point":true, "filter":{"rules":["*.*"], "ignore-txn-start-ts":null, "ddl-allow-list":null}, "mounter":{"worker-num":16}, "sink":{"dispatchers":null, "protocol":"default"}, "cyclic-replication":{"enable":false, "replica-id":0, "filter-replica-ids":null, "id-buckets":0, "sync-ddl":false}, "scheduler":{"type":"table-number", "polling-time":-1}}, "state":"normal", "history":null, "error":null, "sync-point-enabled":false, "sync-point-interval":600000000000, "creator-version":"v5.0.0"}
```

## Lesson 22 练习-2: 创建 TiDB 数据库到 MySQL 数据库的同步任务, 并检查任务状态

参数解释如下:

`./cdc cli changefeed create --pd=http://172.16.6.202:2379` :为任意一个 PD 节点。  
`--sink-uri="mysql://root:mysql@127.0.0.1:3306/"` : 目标库(下游)MySQL 数据库的 IP 地址为:172.16.6.212,端口号为:3306。  
`--changefeed-id="replication-task-1" --sort-engine="unified"` :开启的数据同步任务 ID 是 replication-task-1,不定义引擎。

#### 4. 对于刚刚创建的同步任务进行查询,如下:

```
[root@centos76_vm bin]# ./cdc cli changefeed list --pd=http://172.16.6.202:2379
[
 {
 "id": "replication-task-1",
 "summary": {
 "state": "normal",
 "tso": 425312468718583809,
 "checkpoint": "2021-05-31 13:18:31.457",
 "error": null
 }
 }
]
```

注意:

"state": "normal" : 表示任务状态正常。

"tso": 425312468718583809 : 表示同步任务的时间戳信息。

"checkpoint": "2021-05-31 13:18:31.457" :表示同步任务的时间。

#### 5. 详细查询复制任务信息,如下:

```
[root@centos76_vm bin]# ./cdc cli changefeed query
--pd=http://172.16.6.202:2379
--changefeed-id=replication-task-1
{
 "info": {
 "sink-uri": "mysql://root:mysql@172.16.6.212:3306/",
 "opts": {
 "_changefeed_id": "cli-verify"
 },
 },
 "create-time": "2021-05-31T13:17:49.507258259+08:00",
 "start-ts": 425312457708535810,
 "target-ts": 0,
 "admin-job-type": 0,
 "sort-engine": "unified",
 "sort-dir": "",
 "config": {
 "case-sensitive": true,
 }
}
```

## Lesson 22 练习-2: 创建 TiDB 数据库到 MySQL 数据库的同步任务, 并检查任务状态

```
"enable-old-value": true,
 "force-replicate": false,
 "check-gc-safe-point": true,
 "filter": {
 "rules": [
 "*"
],
 "ignore-txn-start-ts": null,
 "ddl-allow-list": null
 },
 "mounter": {
 "worker-num": 16
 },
 "sink": {
 "dispatchers": null,
 "protocol": "default"
 },
 "cyclic-replication": {
 "enable": false,
 "replica-id": 0,
 "filter-replica-ids": null,
 "id-buckets": 0,
 "sync-ddl": false
 },
 "scheduler": {
 "type": "table-number",
 "polling-time": -1
 }
},
"state": "normal",
"history": null,
"error": null,
"sync-point-enabled": false,
"sync-point-interval": 600000000000,
"creator-version": "v5.0.0"
},
"status": {
 "resolved-ts": 425312478168875009,
 "checkpoint-ts": 425312477906731009,
 "admin-job-type": 0
},
```

## Lesson 22 练习-2: 创建 TiDB 数据库到 MySQL 数据库的同步任务, 并检查任务状态

```
"count": 0,
"task-status": [
 {
 "capture-id": "9d18ce94-fa16-4814-8e80-45d1115b0f22",
 "status": {
 "tables": {
 "377": {
 "start-ts": 425312457708535810,
 "mark-table-id": 0
 }
 },
 "operation": null,
 "admin-job-type": 0
 }
 }
]
```

28. 对于同步任务进行验证, 如下:

6.1. 登录 TiDB 数据库, 查询刚刚创建的 test 数据库下面的表 T1, 并且插入一行数据, 如下所示:

```
mysql> select * from T1;
Empty set (0.00 sec)

mysql> insert into T1 values(1, 'Tom');
Query OK, 1 row affected (0.01 sec)

mysql> select * from T1;
+-----+
| id | name |
+-----+
| 1 | Tom |
+-----+
1 row in set (0.01 sec)
```

6.2. 登录 MySQL 数据库, 查询 test 数据库下面的表 T1, 发现数据库已经同步过去, 如下所示:

```
mysql> select * from T1;
+-----+
| id | name |
+-----+
| 1 | Tom |
+-----+
1 row in set (0.00 sec)
```

## Lesson 22 练习-3: 缩容当前 TiCDC 节点

### 概述

在这个练习中, 您将对当前的 TiCDC节点进行缩容。

### 任务

1. 如果您已经完成了练习二, 那么可以使用如下命令停止同步任务并删除同步任务。

- 1.1. 停止同步任务, 如下:

```
[root@centos76_vm bin]# ./cdc cli changefeed pause --pd=http://172.16.6.202:2379
--changefeed-id replication-task-1
```

- 1.2. 删除同步任务, 如下:

```
[root@centos76_vm bin]# ./cdc cli changefeed remove
--pd=http://172.16.6.202:2379 --changefeed-id replication-task-1
```

2. 使用如下命令缩容 TiCDC 集群, 如下所示:

```
[root@centos76_vm ~]# tiup cluster scale-in tidb-test --node 172.16.6.212:8300
Starting component `cluster`: /root/.tiup/components/cluster/v1.4.4/tiup-cluster scale-in tidb-test --node 172.16.6.212:8300
This operation will delete the 172.16.6.212:8300 nodes in `tidb-test` and all their data.
Do you want to continue? [y/N]:(default=N) y
Scale-in nodes...
+ [Serial] - SSHKeySet: privateKey=/root/.tiup/storage/cluster/clusters/tidb-test/ssh/id_rsa,
publicKey=/root/.tiup/storage/cluster/clusters/tidb-test/ssh/id_rsa.pub
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.157
+ [Parallel] - UserSSH: user=tidb, host=172.16.6.163
... (省略中间内容)
- Regenerate config prometheus -> 172.16.6.157:9090 ... Done
- Regenerate config grafana -> 172.16.6.157:3000 ... Done
- Regenerate config alertmanager -> 172.16.6.157:9093 ... Done
+ [Serial] - SystemCtl: host=172.16.6.157 action=reload prometheus-9090.service
Scaled cluster `tidb-test` in successfully
```

3. 关闭所有窗口。