本小节将介绍如何在多台本地机器上部署一个安全模式下多节点集群，并使用[HAProxy](http://www.haproxy.org/)均衡多种类型的客户端请求负载。

如果集群只是用做测试环境而非生产环境，对安全性没有要求，可以选择非安全模式。

要求

* 本地机器上需要[安装CockroachDB](http://doc.cockroachchina.baidu.com/#quick-start/install-cockorachdb/)。
* 本地机器能够[SSH访问](https://www.digitalocean.com/community/tutorials/how-to-set-up-ssh-keys--2)到集群的各个机器，以便传输和执行CockroachDB程序。
* 确保每台机器的以下端口能被CockroachDB使用：
* 26257用于集群节点之间和客户端-节点之间的通讯
* 8080用于登陆Admin界面

建议

* 如果打算部署在生产环境，请事先阅读[Production Checklist](http://doc.cockroachchina.baidu.com/#deploy/production-checklist/)
* 如果采用非安全模式，注意以下风险：
* 集群对客户端是开放的，通过IP地址可以访问集群任意节点。
* 任何非root用户能够不需要密码访问集群。
* 包括root用户在内的所有用户能够读写集群当中的所有数据。
* 集群节点间任何网络通讯都是非加密的，不需要任何验证。
* 确定Admin界面的访问级别

| **访问级别** | **简述** |
| --- | --- |
| 部分开放 | 设置防火墙规则，允许部分指定的ip地址访问到机器的8080端口 |
| 完全开放 | 设置防火墙规则，允许全部的ip地址访问到机器的8080端口 |
| 完全关闭 | 设置防火墙规则，阻止所有指向8080端口的连接请求。在这种情况下，Admin界面能够通过创建SSH通道去访问。 |

Step 1: 同步时钟

CockroachDB需要中等强度的[时间同步](http://doc.cockroachchina.baidu.com/#deploy/production-checklist/#clock-synchronization)机制以维持数据的一致性时间同步机制以维持数据的一致性。

当一个节点检测到自身的机器时间与集群当中至少50%的节点的机器时间之间的误差值超过集群最大允许时间误差值（默认500ms）的80%的时候，该节点会自动停止。这能够避免违反数据一致性导致读写旧数据的风险。通过在每个节点上运行NTP或其他时钟同步软件，来防止时钟漂移得太远是很重要的。

[ntpd](http://doc.ntp.org/)需要将误差保持在个数位的毫秒范围内，也可以考虑其他方式的时间同步机制。

对于集群当中的每一台机器，执行以下步骤：

* 关闭timesyncd服务，在一些linux发行版本里面是默认开启。

sudo timedatectl set-ntp no

*# 再次检查输出是否有 Network time on: no 或是 NTP*

timedatectl enabled: no

* 安装ntp包

sudo apt-get install ntp

* 关闭NTP后台进程

sudo service ntp stop

* 通过Google NTP服务同步机器时间

sudo ntpd -b time.google.com

为使得配置永久生效，需要配置/etc/ntp.conf文件，查找server和pool相关的注释并修改为如下内容：

server time1.google.com iburst

server time2.google.com iburst

server time3.google.com iburst

server time4.google.com iburst

然后重启ntp服务

sudo service ntp start

**TIPS:** 在多数情况下，我们推荐将NTP与Google的外部NTP服务配合使用，原因是Google提供了闰秒涂抹的技术。如果你采用了不提供闰秒涂抹的NTP服务，你需要在客户端手动地进行该操作。

* 验证服务

*#检查机器是否使用Google NTP服务，使用中的NTP服务器将标有星号。*

sudo ntpq -p

Step 2: 生成证书

如果采用非安全模式，请跳过此步骤。 如果采用安全模式，使用[cockroach cert](http://doc.cockroachchina.baidu.com/#deploy/create-security-certificates/)命令或是[openssl](https://www.cockroachlabs.com/docs/stable/create-security-certificates-openssl.html)工具，生成以下文件：

* CA密钥对（包括ca.crt 证书和密钥文件ca.key）
* 节点间通信密钥对，使用节点的IP、域名或者机器名签发
* root 用户密钥对，用于在实例本地执行管理命令

**TIPS:** 在生成证书的时候需要收集集群所有机器的内网或是外网IP地址。

本地安装CockroachDB

可查看：[Install CockroachDB](http://doc.cockroachchina.baidu.com/#quick-start/install-cockorachdb/)

准备CA证书和密钥

* 创建归档目录：
* certs用于存放生成的CA证书和所有节点和客户端的证书和密钥文件，其中部分的文件会传输到节点机器上。
* my-safe-directory用于存放生成的CA密钥文件，在之后为节点和用户创建证书和密钥的时候使用。

mkdir certs

mkdir my-safe-directory

* 创建CA证书和密钥

cockroach cert create-ca --certs-dir=certs --ca-key=my-safe-directory/ca.key

为每个节点创建证书和密钥

用于其他节点或是负载均衡器访问该节点。对于每个节点进行如下操作：

cockroach cert create-node \

<node1 internal IP address> \

<node1 external IP address> \

<node1 hostname> \

<other common names **for** node1> \

localhost \

127.0.0.1 \

<load balancer IP address> \

<load balancer hostname> \

<other common names **for** load balancer instances> \

--certs-dir=certs \

--ca-key=my-safe-directory/ca.key

将证书和密钥传输到对应节点

*# Create the certs directory:*

ssh <username>@<node address> "mkdir certs"

*# Upload the CA certificate and node certificate and key:*

scp certs/ca.crt certs/node.crt certs/node.key \

<username>@<node address>:~/certs

删除本地机器的证书和密钥缓存

rm certs/node.crt certs/node.key

**TIPS:** 考虑为所有节点生成的证书和密钥文件都是同样的文件名，删除证书和密钥文件是有必要的。在默认情况下，只有为cockroach cert create-node命令添加--overwrite标签才会覆盖已有的同名的证书和密钥文件。

为root用户创建证书和密钥

cockroach cert create-client root --certs-dir=certs --ca-key=my-safe-directory/ca.key

将证书和密钥传输到某个节点机器

*# Create the certs directory:*

ssh <username>@<workload address> "mkdir certs"

*# Upload the CA certificate and client certificate and key:*

scp certs/ca.crt certs/client.root.crt certs/client.root.key <username>@<workload address>:~/certs

拥有证书的机器能够使用root账户执行[cockroach](http://doc.cockroachchina.baidu.com/#deploy/cockroach-commands/)命令，通过该节点机器访问集群。

Step 3: 启动节点

可以选择手动方式或是使用systemd工具，对需要加入到集群当中的每个节点执行以下操作：

**NOTE:** 节点完成启动后需要相互之间构成集群才能被使用。

手动方式

环境搭建

* SSH到需要启动服务的节点机器
* 下载对应的[CockroachDB归档文件](https://binaries.cockroachdb.com/cockroach-v2.0.5.linux-amd64.tgz)，解压出可执行文件。

wget -qO- https://binaries.cockroachdb.com/cockroach-v2.0.5.linux-amd64.tgz | tar xvz

* 复制文件到PATH路径下

sudo cp -i cockroach-v2.0.5.linux-amd64/cockroach /usr/local/bin

分模式启动节点

*#非安全模式*

cockroach start \

--insecure \

--advertise-host=<node1 address> \

--join=<node1 address>:26257,<node2 address>:26257,<node3 address>:26257 \

--cache=.25 \

--max-sql-memory=.25 \

--background

*#安全模式*

cockroach start \

--certs-dir=certs \

--advertise-host=<node1 address> \

--join=<node1 address>:26257,<node2 address>:26257,<node3 address>:26257 \

--cache=.25 \

--max-sql-memory=.25 \

--background

涉及到的flags：

| **flag** | **简介** |
| --- | --- |
| --insecure | 指定以非安全模式建立连接 |
| --certs-dir | 指定存放ca.crt、node.crt、node.key文件的目录 |
| --advertise-host | 指定IP地址或是hostname供集群其他节点访问该节点。具体配置详见[Networking](http://doc.cockroachchina.baidu.com/#deploy/production-checklist/#networking) |
| --join | 指定集群3-5个初始化节点的地址和端口 |
| --cache --max-sql-memory | 增加节点缓存和临时SQL内存大小到操作系统内存的25%，用于优化读性能和内存中SQL执行，详见[Cache and SQL Memory Size](http://doc.cockroachchina.baidu.com/#deploy/production-checklist/#cache-and-sql-memory-size) |
| --background | 指定后台运行 |

如果集群部署在多个数据中心或是节点之间网络延迟较高，推荐使用--locality配置拓扑节点位置，这属于enterprise版本的功能，详见：[Locality](http://doc.cockroachchina.baidu.com/#deploy/start-a-node/#locality)

其他未指定的选项采用默认配置，例如默认--store即数据库文件，为当前目录下已有或新建的cockroach-data；默认--port即节点通信和客户端访问端口，为26257端；更多配置详见：[开启节点](http://doc.cockroachchina.baidu.com/#deploy/start-a-node/)

使用systemd工具

环境搭建

* 以root账户登陆SSH到需要启动服务的节点机器。
* 下载对应的[CockroachDB归档文件](https://binaries.cockroachdb.com/cockroach-v2.0.5.linux-amd64.tgz)，解压出可执行文件

wget -qO- https://binaries.cockroachdb.com/cockroach-v2.0.5.linux-amd64.tgz | tar xvz

* 执行操作

*#复制文件到`PATH`路径下*

sudo cp -i cockroach-v2.0.5.linux-amd64/cockroach /usr/local/bin

*#创建 `cockroach文件夹*

mkdir /var/lib/cockroach

*#创建用户*

useradd cockroach

*#移动`certs`文件夹到`cockroach`文件夹当中*

mv certs /var/lib/cockroach/

*#修改`cockroach`文件夹权限*

chown -R cockroach.cockroach /var/lib/cockroach

准备配置模版

* 分模式下载模版文件并保存到/etc/systemd/system文件夹下

*#非安全模式*

wget -qO- https://raw.githubusercontent.com/cockroachdb/docs/master/\_includes/v2.0/prod-deployment/insecurecockroachdb.service

*#安全模式*

wget -qO- https://raw.githubusercontent.com/cockroachdb/docs/master/\_includes/v2.0/prod-deployment/securecockroachdb.service

非安全模式的配置模版文件内容：

[Unit]

Description=Cockroach Database cluster node

Requires=network.target

[Service]

Type=notify

WorkingDirectory=/var/lib/cockroach

ExecStart=/usr/local/bin/cockroach start --insecure --advertise-host=<node1 address> --join=<node1 address>:26257,<node2 address>:26257,<node3 address>:26257 --cache=.25 --max-sql-memory=.25

TimeoutStopSec=60

Restart=always

RestartSec=10

StandardOutput=syslog

StandardError=syslog

SyslogIdentifier=cockroach

User=cockroach

[Install]

WantedBy=default.target

安全模式的配置模版文件内容：

[Unit]

Description=Cockroach Database cluster node

Requires=network.target

[Service]

Type=notify

WorkingDirectory=/var/lib/cockroach

ExecStart=/usr/local/bin/cockroach start --certs-dir=certs --advertise-host=<node1 address> --join=<node1 address>:26257,<node2 address>:26257,<node3 address>:26257 --cache=.25 --max-sql-memory=.25

TimeoutStopSec=60

Restart=always

RestartSec=10

StandardOutput=syslog

StandardError=syslog

SyslogIdentifier=cockroach

User=cockroach

[Install]

WantedBy=default.target

* 自定义配置模版

关于配置项ExecStart的命令，涉及到的flag：

| **flag** | **简介** |
| --- | --- |
| --advertise-host | 指定IP地址或是hostname供其他节点访问该节点。相关的flag有--host，具体配置详见[Networking](http://doc.cockroachchina.baidu.com/#deploy/production-checklist/#networking) |
| --join | 指定集群3-5个初始化节点的地址和端口 |

如果集群部署在多个数据中心或是节点之间网络延迟较高，推荐使用--locality配置拓扑节点位置，这属于enterprise版本的功能，详见：[Locality](http://doc.cockroachchina.baidu.com/#deploy/start-a-node/#locality)

其他未指定的选项采用默认配置，例如默认--store即数据库文件，为当前目录下已有或新建的cockroach-data；默认--port即节点通信和客户端访问端口，为26257端；更多配置详见：[开启节点](http://doc.cockroachchina.baidu.com/#deploy/start-a-node/)

启动CockroachDB集群

systemctl start securecockroachdb

**NOTE:** systemd在节点故障的时候能够重启节点. 如果不需要自动重启，执行run systemctl stop insecurecockroachdb

Step 4: 初始化集群

分模式在本地机器执行[cockroach init](http://doc.cockroachchina.baidu.com/#deploy/initialize-a-cluster/)命令

*#非安全模式*

cockroach init --insecure --host=<address of any node>

*#安全模式*

cockroach init --certs-dir=certs --host=<address of any node>

涉及到的flags：

| **flag** | **简介** |
| --- | --- |
| --certs-dir | 指定存放ca.crt、node.crt、node.key文件的目录 |
| --host | 指定集群当中任意节点的地址 |

执行命令后，节点会在[标准输出](http://doc.cockroachchina.baidu.com/#deploy/start-a-node/#standard-output)里打印详细信息，包括了CockroachDB版本信息、Admin界面的URl地址、供客户端访问的SQL地址。

Step 5: 测试集群

CockroachDB底层自动进行数据冗余和均衡分发，使用[Gossip协议](https://en.wikipedia.org/wiki/Gossip_protocol)能够使每个节点在集群当中能够定位到数据。下面就该功能进行测试：

创建nodetest数据库

* 开启交互式shell

*#非安全模式*

cockroach sql --insecure --host=<address of any node>

*#安全模式*

cockroach sql --certs-dir=certs --host=<address of any node>

涉及到的flags：

| **flag** | **简介** |
| --- | --- |
| --certs-dir | 指定存放ca.crt、node.crt、node.key文件的目录 |
| --host | 指定集群当中任意节点的地址 |

* 创建nodetest数据库

**CREATE** **DATABASE** nodetest;

* 使用\q或是CTRL+D退出交互式shell

连接另一个节点检查nodetest数据库

* 分模式开启交互式shell

*#非安全模式*

cockroach sql --insecure --host=<address of different node>

*#安全模式*

cockroach sql --certs-dir=certs --host=<address of different node>

* 检查数据库

**SHOW** **DATABASES**;

+*--------------------+*

| Database |

+*--------------------+*

| crdb\_internal |

| information\_schema |

| nodetest |

| pg\_catalog |

| system |

+*--------------------+*

(5 rows)

* 使用\q或是CTRL+D退出交互式shell

Step 6: 配置HAProxy负载均衡

在集群当中每个CockroachDB节点都是一个平等的SQL访问网关。在负载均衡的时候，需要考虑节点的性能和可靠性：

* 性能 均衡各个节点的客户端访问量，避免单个节点处理过多的访问请求以至于影响集群的性能。
* 可靠性 负载均衡器没有监控集群各个节点健康状况，因此为了避免客户端请求路由到故障或是未准备好处理请求的节点，负载均衡器需要使用CockroachDB ‘s readiness health check

**TIPS:** 单一负载均衡器能够处理节点故障的情况，但其本身对整个系统来说也是一个故障的风险点。因此需要设置多个负载均衡器，通过浮动IP或是DNS为客户端选择合适的负载均衡器。

[HAProxy](http://www.haproxy.org/)是一款比较流行的开源TCP负载均衡器，CockroachDB集成到cockroach命令当中用于生成HAProxy配置文件。

非安全模式下

可以设置多台机器作为网关，运行负载均衡器，对每一台的机器进行如下配置：

环境搭建

* SSH到配置机器
* 下载HAProxy

apt-get install haproxy

* 下载对应的[CockroachDB归档文件](https://binaries.cockroachdb.com/cockroach-v2.0.5.linux-amd64.tgz)，解压出可执行文件

wget -qO- https://binaries.cockroachdb.com/cockroach-v2.0.5.linux-amd64.tgz | tar xvz

* 复制文件到PATH路径下

sudo cp -i cockroach-v2.0.5.linux-amd64/cockroach /usr/local/bin

生成配置文件

执行cockroach gen haproxy命令

cockroach gen haproxy --insecure --host=<address of any node> --port=26257

默认情况下，生成配置文件为haproxy.cfg，内容如下：

global

maxconn 4096

defaults

mode tcp

*# Timeout values should be configured for your specific use.*

*# See: https://cbonte.github.io/haproxy-dconv/1.8/configuration.html#4-timeout%20connect*

timeout connect 10s

timeout client 1m

timeout server 1m

*# TCP keep-alive on client side. Server already enables them.*

option clitcpka

listen psql

bind **:26257**

mode tcp

balance roundrobin

option httpchk GET /health?ready=1

server cockroach1 <node1 address>:26257 check port 8080

server cockroach2 <node2 address>:26257 check port 8080

server cockroach3 <node3 address>:26257 check port 8080

涉及到配置项注解：

| **配置项** | **简介** |
| --- | --- |
| timeout connect timeout client timeout server | 集群各部分timeout的时间值 |
| bind | HAProxy的监听端口，注意修改系统相关网络配置，确保该端口能够开启并访问。  推荐将HAProxy部署到CockroachDB集群以外的单独机器上，若部署在集群节点上，需要注意端口占用问题。 |
| balance | 均衡算法，此处为roundrobin算法，以简单的轮询方式将连接均衡到各个节点，更多算法详见:[HAProxy Configuration Manual](http://cbonte.github.io/haproxy-dconv/1.7/configuration.html#4-balance) |
| option httpchk | HTTP端点供HAProxy用来检查节点的健康状况，此处/health?ready=1能够保证HAProxy不会将请求转发到存活但未准备好接受请求的节点上。 |
| server | 该配置指定集群当中节点的地址（例如节点启动时--host指定的地址）和用于确保各节点健康的监听端口。 |

**NOTE:** 更多介绍和配置，详见：[HAProxy Configuration Manual](http://cbonte.github.io/haproxy-dconv/1.7/configuration.html#4-balance)

启动HAProxy

haproxy -f haproxy.cfg // -f指定配置文件

安全模式下

本地机器生成配置文件

执行cockroach gen haproxy命令

cockroach gen haproxy --certs-dir=certs --host=<address of any node> --port=26257

其中--host为集群当中任意节点，--certs指向集群客户端访问的证书目录。默认情况下，生成配置文件为haproxy.cfg，内容如下：

global

maxconn 4096

defaults

mode tcp

*# Timeout values should be configured for your specific use.*

*# See: https://cbonte.github.io/haproxy-dconv/1.8/configuration.html#4-timeout%20connect*

timeout connect 10s

timeout client 1m

timeout server 1m

*# TCP keep-alive on client side. Server already enables them.*

option clitcpka

listen psql

bind **:26257**

mode tcp

balance roundrobin

option httpchk GET /health?ready=1

server cockroach1 <node1 address>:26257 check port 8080

server cockroach2 <node2 address>:26257 check port 8080

server cockroach3 <node3 address>:26257 check port 8080

涉及到配置项注解：

| **配置项** | **简介** |
| --- | --- |
| timeout connect timeout client timeout server | 各节点timeout的时间值 |
| bind | HAProxy的监听端口，注意修改系统相关网络配置，确保该端口能够开启并访问。  推荐将HAProxy部署到CockroachDB集群以外的单独机器上，若部署在集群节点上，需要注意端口占用问题。 |
| balance | 均衡算法，此处为roundrobin算法，以简单的轮询方式将连接均衡到各个节点，更多算法详见:[HAProxy Configuration Manual](http://cbonte.github.io/haproxy-dconv/1.7/configuration.html#4-balance) |
| option httpchk | HTTP端点供HAProxy用来检查节点的健康状况，此处/health?ready=1能够保证HAProxy不会将请求转发到存活但未准备好接受请求的节点上。 |
| server | 该配置指定集群当中节点的地址（例如节点启动时--host指定的地址）和用于确保各节点健康的监听端口。 |

**NOTE:** 更多介绍和配置，详见：[HAProxy Configuration Manual](http://cbonte.github.io/haproxy-dconv/1.7/configuration.html#4-balance)

启动HAProxy

可以设置多台机器作为网关，运行负载均衡器，对每一台的机器进行如下配置：

* 上传配置文件haproxy.cfg

scp haproxy.cfg <username>@<haproxy address>:~/

* SSH登陆机器
* 安装HAProxy并启动

apt-get install haproxy

haproxy -f haproxy.cfg // -f指定配置文件

Step 7: 生成测试负载

Linux下Cockroach提供了workload程序，能够对集群生成几种客户端负载，此处我们将模拟[TPC-C](http://www.tpc.org/tpcc/)负载。

**TIPS:** 全面了解CockroachDB生成的TPC-C负载，可阅读：[Performance Benchmarking white paper](https://www.cockroachlabs.com/guides/cockroachdb-performance/)

生成负载

* SSH到用于生成负载的机器。 如果是安全模式，需要确保该机器具有certs文件夹（包含ca.crt、client.root.crt、client.root.key文件）
* 下载负载生成器

wget https://edge-binaries.cockroachdb.com/cockroach/workload.LATEST | chmod 755 workload.LATEST

* 复制生成器到PATH文件夹下

cp -i workload.LATEST /usr/local/bin/workload

* 分模式运行负载生成器

#非安全模式

workload run tpcc \

--drop \

--init \

--duration=20m \

--tolerate-errors \

"postgresql://root@<IP ADDRESS OF LOAD BALANCER:26257/tpcc?sslmode=disable"

#安全模式

workload run tpcc \

--drop \

--init \

--duration=20m \

--tolerate-errors \

"postgresql://root@<IP ADDRESS OF LOAD BALANCER:26257/tpcc?sslmode=verify-full&sslrootcert=certs/ca.crt&sslcert=certs/client.root.crt&sslkey=certs/client.root.key"

该命令执行20分钟，初始化一个数据仓库，并使用同一个SQL连接有10个并发线程生成查询，总体上每分钟12个查询。

**TIPS:** 查看更多TPC-C配置，可以执行workload run tpcc --help 查看当前生成器支持的更多负载类型，可以执行workload run --help

查看集群状态

登陆集群的Admin界面，点击左方**Metrics**，选择**SQL**仪表盘，查看**SQL Connection**图，可以查看生成的负载通过负载均衡器均分到各个节点执行处理。

Step 8: 设置监控和报警

尽管CockroachDB有多种保障集群可用性的设计，对于集群健康和性能的持续监控仍然是有必要的，预先针对需要调查和干预的事件创建报警规则，以便出现问题的时候能够第一时间获知并及时处理。 详细的监控选项、监控事件以及警告指标，可以查看：[Monitoring and Alerting](http://doc.cockroachchina.baidu.com/#deploy/monitoring-and-alerting/overview/)

Step 9: 集群扩容

选择手动方式或是使用systemd工具拓展集群。

手动方式

根据业务需要可以拓展集群节点数量，对需要加入集群的节点重复以下步骤：

* SSH到集群新节点机器
* 下载对应的[CockroachDB归档文件](https://binaries.cockroachdb.com/cockroach-v2.0.5.linux-amd64.tgz)，解压出可执行文件。

wget -qO- https://binaries.cockroachdb.com/cockroach-v2.0.5.linux-amd64.tgz | tar xvz

* 复制文件到PATH路径下

sudo cp -i cockroach-v2.0.5.linux-amd64/cockroach /usr/local/bin

* 分模式执行cockroach start命令

*#非安全模式*

cockroach start --insecure \

--host=<node4 address> \

--locality=<key-value pairs> \

--cache=.25 \

--max-sql-memory=.25 \

--join=<node1 address>:26257,<node2 address>:26257,<node3 address>:26257 \

--background

*#安全模式*

cockroach start \

--certs-dir=certs \

--host=<node4 address> \

--locality=<key-value pairs> \

--cache=.25 \

--max-sql-memory=.25 \

--join=<node1 address>:26257,<node2 address>:26257,<node3 address>:26257 \

--background

* 更新负载均衡器的配置，添加新节点路由。

使用systemd工具

根据业务需要可以拓展集群节点数量，对需要加入集群的节点重复以下步骤：

环境搭建

* 以root账户SSH到需要启动服务的节点机器
* 下载对应的[CockroachDB归档文件](https://binaries.cockroachdb.com/cockroach-v2.0.5.linux-amd64.tgz)，解压出可执行文件。

wget -qO- https://binaries.cockroachdb.com/cockroach-v2.0.5.linux-amd64.tgz | tar xvz

* 执行操作

*#复制文件到`PATH`路径下*

sudo cp -i cockroach-v2.0.5.linux-amd64/cockroach /usr/local/bin

*#创建 `cockroach文件夹*

mkdir /var/lib/cockroach

*#创建用户*

useradd cockroach

*#移动`certs`文件夹到`cockroach`文件夹*

mv certs /var/lib/cockroach/

*#修改`cockroach`文件夹权限*

chown -R cockroach.cockroach /var/lib/cockroach

准备配置模版

* 分模式下载模版文件，将文件保存到/etc/systemd/system文件夹下。

*#非安全模式*

wget -qO- https://raw.githubusercontent.com/cockroachdb/docs/master/\_includes/v2.0/prod-deployment/insecurecockroachdb.service

*#安全模式*

wget -qO- https://raw.githubusercontent.com/cockroachdb/docs/master/\_includes/v2.0/prod-deployment/securecockroachdb.service

非安全模式的模版文件内容：

[Unit]

Description=Cockroach Database cluster node

Requires=network.target

[Service]

Type=notify

WorkingDirectory=/var/lib/cockroach

ExecStart=/usr/local/bin/cockroach start --insecure --advertise-host=<node1 address> --join=<node1 address>:26257,<node2 address>:26257,<node3 address>:26257 --cache=.25 --max-sql-memory=.25

TimeoutStopSec=60

Restart=always

RestartSec=10

StandardOutput=syslog

StandardError=syslog

SyslogIdentifier=cockroach

User=cockroach

[Install]

WantedBy=default.target

安全模式的模版文件内容：

[Unit]

Description=Cockroach Database cluster node

Requires=network.target

[Service]

Type=notify

WorkingDirectory=/var/lib/cockroach

ExecStart=/usr/local/bin/cockroach start --certs-dir=certs --advertise-host=<node1 address> --join=<node1 address>:26257,<node2 address>:26257,<node3 address>:26257 --cache=.25 --max-sql-memory=.25

TimeoutStopSec=60

Restart=always

RestartSec=10

StandardOutput=syslog

StandardError=syslog

SyslogIdentifier=cockroach

User=cockroach

[Install]

WantedBy=default.target

* 自定义配置模版

关于配置项ExecStart的命令，涉及到的flag：

| **flag** | **简介** |
| --- | --- |
| --host | 指定IP地址或是hostname供客户端或其他节点访问，同时作为Admin界面里该节点的标识。如果是hostname，需要注意能被所有节点解析，如果是IP地址，需要注意该地址能被所有节点访问到。  默认情况下--host为空是将使用该机器所有网卡的IP地址.  如果需要设置内网地址供集群节点访问，同时对外能够供客户端访问，则设置--host为空，再设置--advertise-host为内网地址。 |
| --join | 指定集群3-5个初始化节点的地址和端口 |

Step 10: 部署业务

集群已经完成部署与测试，可以：

* 创建用户自定义的[数据模型](http://doc.cockroachchina.baidu.com/#develop/sql-statements/overview/)
* [创建用户](http://doc.cockroachchina.baidu.com/#deploy/access-management/manage-users/)并进行[授权](http://doc.cockroachchina.baidu.com/#develop/sql-statements/GRANT-%3Cprivileges%3E/)
* 通过负载均衡器[访问集群](http://doc.cockroachchina.baidu.com/#develop/install-client-drivers/)

此外，根据业务情况可以调整集群数据冗余的副本数量，默认情况下冗余3份副本。在特定场景下可以针对数据库和表格配置不同的冗余副本数量，详见：[配置复制区域](http://doc.cockroachchina.baidu.com/#deploy/configure-replication-zones/)