Linux 中间件安装

> 以下操作基于 `CentOS 7.6`，其他发行版本的命令可能有所不同，操作系统内置的组件也可能有所不同
>
> 下方的链接具有时效性，失效后需到对应中间件的官网查找最新的地址
>
> 在下面的示例中，`$` 符号表示命令提示符，之后的文本是用户在命令行中输入的命令。

了解 systemd 服务

systemd 是一个用于初始化系统和管理系统服务的系统和服务管理器，主要用于 Linux 操作系统。

以 `/usr/lib/systemd/system/firewalld.service` 这个防火墙服务做简单介绍

```
[Unit] # 描述服务的元数据和依赖关系。
Description=firewalld - dynamic firewall daemon
Before=network-pre.target
Wants=network-pre.target
After=dbus.service
After=polkit.service
Conflicts=iptables.service ip6tables.service ebtables.service ipset.service
Documentation=man:firewalld(1)

[Service] # 定义服务的行为，例如启动命令、重启策略等。
EnvironmentFile=-/etc/sysconfig/firewalld
ExecStart=/usr/sbin/firewalld --nofork --nopid $FIREWALLD_ARGS
ExecReload=/bin/kill -HUP $MAINPID
# supress to log debug and error output also to /var/log/messages
StandardOutput=null
StandardError=null
Type=dbus
BusName=org.fedoraproject.FirewallD1
KillMode=mixed

[Install] # 定义安装相关的信息，例如目标。
WantedBy=multi-user.target
Alias=dbus-org.fedoraproject.FirewallD1.service
```

后续安装的中间件会使用 systemd 服务，方便中间件的启停、开机自启

每添加一个 systemd 的 service 需要重新加载

```
$ systemctl daemon-reload
```

可配置服务开机自启 systemctl enable {服务名}

```
$ systemctl enable firewalld
```

配置防火墙

为避免一些访问不通的情况，可先关闭防火墙，有需要时再打开

```
$ systemctl stop firewalld # 停止防火墙
$ systemctl disable firewalld # 禁止开机启动
$ systemctl status firewalld # 查看防火墙状态
```

JDK

我们开发过程中使用 `Eclipse Temurin` 的发行版本

卸载可能预装的 JDK

在安装之前需检查是否已有 JDK （CentOS 一些版本中可能会带有 OpenJDK），可通过执行命令查看

```
$ java -version
openjdk version "1.8.0_181"
OpenJDK Runtime Environment (build 1.8.0_181-b13)
OpenJDK 64-Bit Server VM (build 25.181-b13, mixed mode)
```

执行命令后能正常返回则说明存在了其他 JDK，如该版本的 JDK 不是必须的可卸载避免后面使用中引起一些不可预知的错误
。对于 rpm 安装的软件，可执行命令查看已有的 JDK 信息

```
$ rpm -qa | grep java
java-1.7.0-openjdk-headless-1.7.0.191-2.6.15.5.el7.x86_64
java-1.8.0-openjdk-headless-1.8.0.181-7.b13.el7.x86_64
python-javapackages-3.4.1-11.el7.noarch
java-1.8.0-openjdk-1.8.0.181-7.b13.el7.x86_64
tzdata-java-2018e-3.el7.noarch
javapackages-tools-3.4.1-11.el7.noarch
java-1.7.0-openjdk-1.7.0.191-2.6.15.5.el7.x86_64
```

执行命令卸载

```
$ sudo rpm -e `rpm -qa | grep java` --allmatches --nodeps
```

安装 JDK

进入Eclipse Temurin JDK 下载页 根据操作系统和系统架构下载对应的 JDK

下载后的文件名为 `OpenJDK11U-jdk_x64_linux_hotspot_11.0.24_8.tar.gz`，将文件上传至服务器 `/opt` 目录下，然后解压文件

```
$ cd /opt/
$ tar -zxvf OpenJDK11U-jdk_x64_linux_hotspot_11.0.24_8.tar.gz
```

可创建软链接，方便访问及后续的升级切换

```
$ ln -s jdk-11.0.24+8 jdk
```

配置环境变量

```
$ sudo vim /etc/profile
```

在这个文件末尾加上

```
export JAVA_HOME=/opt/jdk
export PATH=$PATH:$JAVA_HOME/bin
```

> vim基本命令：i插入模式，esc键退出插入模式，:wq保存退出。

利用下面命令使配置生效，并且查看 JDK 版本

```
$ sudo source /etc/profile
$ java -version
openjdk version "11.0.24" 2024-07-16
OpenJDK Runtime Environment Temurin-11.0.24+8 (build 11.0.24+8)
OpenJDK 64-Bit Server VM Temurin-11.0.24+8 (build 11.0.24+8, mixed mode)
```

Tomcat

进入Tomcat 下载页下载安装包

下载后的文件名为 `apache-tomcat-9.0.91.tar.gz`，将文件上传至服务器 `/opt` 目录下，然后解压

```
$ cd /opt/
$ tar zxf apache-tomcat-9.0.91.tar.gz
```

可创建软链接，方便访问及后续的升级切换

```
$ ln -s apache-tomcat-9.0.91 tomcat9
```

禁用selinux

修改config配置文件：vi /etc/selinux/config

```
SELINUX=disabled
```

配置服务自动启动

修改 setclasspath.sh

修改 `tomcat9/bin/setclasspath.sh`，开头增加：

```
CATALINA_PID=/opt/tomcat9/temp/tomcat.pid
JAVA_HOME=/opt/jdk
CATALINA_HOME=/opt/tomcat9
CATALINA_OPTS="-server -Xms1024M -Xmx2048M"
```

注册 systemd service

在 `/usr/lib/systemd/system` 目录下增加 `tomcat9.service`，目录必须是绝对目录。

```
[Unit]
Description=Tomcat9
After=syslog.target network.target remote-fs.target nss-lookup.target

[Service]
```

```
Type=forking
PIDFile=/opt/tomcat9/temp/tomcat.pid
ExecStart=/opt/tomcat9/bin/startup.sh
ExecReload=/bin/kill -s HUP $MAINPID
ExecStop=/bin/kill -s QUIT $MAINPID
PrivateTmp=true

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

重载所有修改过的配置文件：

```
$ sudo systemctl daemon-reload
```

```
$ systemctl enable tomcat9.service # 配置开机启动
$ systemctl start tomcat9.service # 启动 tomcat
$ systemctl stop tomcat9.service # 停止 tomcat
$ systemctl restart tomcat9.service # 重启 tomcat
```

MySQL

RPM 包安装

卸载 CentOS 中可能有的 MariaDB，防止安装 MySQL 时报错

```
$ rpm -qa | grep mariadb
mariadb-libs-5.5.60-1.el7_5.x86_64
$ rpm -e mariadb-libs-5.5.60-1.el7_5.x86_64 --nodeps
```

MySQL RPM 包下载页下载 rpm 包安装，根据操作系统和系统架构下载对应的 RPM Bundle 下载后的文件名为 `mysql-8.0.39-1.el7.x86_64.rpm-bundle.tar`，将文件上传至服务器 `/opt` 目录下，然后创建 `mysql-8.0.39` 目录并将压缩包解压到该目录下

```
$ mkdir mysql-8.0.39 && tar -xvf mysql-8.0.39-1.el7.x86_64.rpm-bundle.tar -C
mysql-8.0.39
```

进入目录并使用 rpm 安装 MySQL

```
$ cd mysql-8.0.39 && rpm -ivh mysql-community-{server,client,client-plugins,icu-
data-files,common,libs}-*
```

执行无误即可启动 MySQL

```
$ systemctl start mysqld
```

启动后初始化的密码会输出到日志中，通过查看日志可以得到初始化密码

```
$ grep 'temporary password' /var/log/mysqld.log
2024-07-30T10:11:15.557169Z 6 [Note] [MY-010454] [Server] A temporary password
is generated for root@localhost: y0--fSCsBurl
```

登录 MySQL

```
$ mysql -uroot -p
```

修改 MySQL 密码

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'MyNewPass4!';
```

NGINX

源码编译安装

在源代码编译安装 NGINX 之前，需要安装一些依赖库

- PCRE – NGINX 核心和 Rewrite 模块需要，用于支持正则表达式。点击下载 下载后的文件名为 `pcre2-10.42.tar.gz`，将文件上传至服务器 `/opt` 目录下，然后解压

```
$ tar -zxf pcre2-10.42.tar.gz
$ cd pcre2-10.42
$ ./configure
$ make && sudo make install
```

- zlib – NGINX Gzip 模块需要，用于支持请求头压缩。点击下载 下载后的文件名为 `zlib-1.2.13.tar.gz`，将文件上传至服务器 `/opt` 目录下，然后解压

```
$ tar -zxf zlib-1.2.13.tar.gz
$ cd zlib-1.2.13
$ ./configure
$ make && sudo make install
```

- OpenSSL – NGINX SSL 模块和其他模块需要，用于支持 HTTPS 协议。点击下载 下载后的文件名为 `openssl-1.1.1v.tar.gz`，将文件上传至服务器 `/opt` 目录下，然后解压

```
$ tar -zxf openssl-1.1.1v.tar.gz
$ cd openssl-1.1.1v
$ ./Configure linux-x86_64 --prefix=/usr
$ make && sudo make install
```

进入NGINX下载页下载源码

下载后的文件名为 `nginx-1.26.1.tar.gz`，将文件上传至服务器 `/opt` 目录下，然后解压

```
$ tar -zxvf  nginx-1.26.1.tar.gz
```

NGINX 的编译安装

```
$ cd nginx-1.26.1
$ ./configure --prefix=/usr/local/nginx --with-pcre=../pcre2-10.42 --with-
zlib=../zlib-1.2.13 --with-http_ssl_module
$ make && sudo make install
```

修改nginx.conf并启动

```
$ ln -s /usr/local/nginx/sbin/nginx /usr/sbin/nginx
$ nginx -c /usr/local/nginx/conf/nginx.conf
```

注册 systemd service

在`/usr/lib/systemd/system`目录下增加`nginx.service`，目录必须是绝对目录。

```
[Unit]
Description=nginx - high performance web server
Documentation=http://nginx.org/en/docs/
After=network-online.target remote-fs.target nss-lookup.target
Wants=network-online.target

[Service]
Type=forking
PIDFile=/usr/local/nginx/logs/nginx.pid
ExecStart=/usr/local/nginx/sbin/nginx -c /usr/local/nginx/conf/nginx.conf
ExecReload=/bin/sh -c "/bin/kill -s HUP $(/bin/cat
/usr/local/nginx/logs/nginx.pid)"
ExecStop=/bin/sh -c "/bin/kill -s TERM $(/bin/cat
/usr/local/nginx/logs/nginx.pid)"
PrivateTmp=true

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

Kafka (Zookeeper)

压缩包安装

下载 **kafka 2.6.0** tgz压缩包并上传至服务器 kafka 官方下载地址，或直接使用 wget 下载。 以 **CentOS** 为例，详情可参考kafka 官方文档.

将文件上传至服务器/opt目录下，然后解压

```
$ wget https://archive.apache.org/dist/kafka/2.6.0/kafka_2.13-2.6.0.tgz
# 解压
$ tar -xvf kafka_2.13-2.6.0.tgz
```

修改config/server.properties

```
broker.id=0
listeners=PLAINTEXT://IP(需要根据实际情况填写):9092
advertised.listeners=PLAINTEXT://IP:9092
log.dirs=/opt/kafka/data    //根据实际情况填写
zookeeper.connect=localhost:2181
```

后台启动 zookeeper，将 /usr/local 替换成自己安装的路径

```
$ /opt/kafka_2.13-2.6.0/bin/zookeeper-server-start.sh /opt/kafka_2.13-
2.6.0/config/server.properties &
```

后台启动 kafka，将 /usr/local 替换成自己安装的路径

```
$ /opt/kafka_2.13-2.6.0/bin/kafka-server-start.sh /opt/kafka_2.13-
2.6.0/config/server.properties
```

Redis

源码安装

下载最新的稳定版源码，下载后的文件名为 `redis-stable.tar.gz`，将文件上传至服务器/opt目录下，然后解压

```
$ tar -xzvf redis-stable.tar.gz
$ cd redis-stable
$ make
$ sudo make install
```

其中，对于要启用 TLS 支持的

```
$ make BUILD_TLS=yes
```

创建单独 `/usr/local/redis` 目录，将相关安装的二进制文件链接到此，配置文件复制到此

```
$ mkdir -p /usr/local/redis
$ ln -s /usr/local/bin/redis-* /usr/local/redis
$ cp redis.conf  /usr/local/redis/
```

注册 systemd service

在 `/usr/lib/systemd/system` 目录下增加 `redis.service`，目录必须是绝对目录。

```
[Unit]
Description=Redis
After=syslog.target network.target remote-fs.target nnS-lookup.target

[Service]
Type=forking
PIDFile=/var/run/redis.pid
ExecStart=/usr/local/redis/redis-server /usr/local/redis/redis.conf
ExecReload=/bin/kill -s HUP $MAINPID
ExecStop=/bin/kill -s QUIT $MAINPID

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

ElasticSearch

RPM包安装

安装

Elasticsearch RPM 包下载页下载 rpm 包安装，根据操作系统和系统架构下载对应的 RPM Bundle，执行安装命令

```
$ rpm -ivh  elasticsearch-7.17.22-x86_64.rpm
```

修改配置： /etc/elasticsearch/elasticsearch.yml

```
$ cat >> ./elasticsearch.yml << EOF
# 集群名称
cluster.name: y9elasticsearch
# 存放数据的路径
path.data: /software/elasticsearch/data
# 存放日志的路径
path.logs: /software/elasticsearch/logs
# 节点名称
node.name: y9Node
# 监听地址（默认为0.0.0.0）
network.host: 192.168.3.38
# 设置对外服务的http端口，默认为9200。
http.port: 9200
# 默认情况下端口是9300，这里添加修改，指定为其他端口
#transport.tcp.port: 9300

#浏览器插件访问配置
http.cors.enabled: true
http.cors.allow-origin: "*"
http.cors.allow-headers: Authorization
EOF
```

查看修改结果：

```
$ grep -n '^[a-z]' elasticsearch.yml
17:cluster.name: y9elasticsearch
23:node.name: y9Node
33:path.data: /software/elasticsearch/data
37:path.logs: /software/elasticsearch/logs
55:network.host: 192.168.3.38
59:http.port: 9200
94:http.cors.enabled: true
95:http.cors.allow-origin: "*"
96:http.cors.allow-headers: Authorization
```

启动Elasticsearch

```
$ systemctl  enable  elasticsearch.service
$ systemctl  start   elasticsearch.service
```

通过压缩包安装

下载地址

Elasticsearch 包下载页下载 压缩包安装，根据操作系统和系统架构下载对应的

联网的服务器可以直接下载

```
$ wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-7.17.22-linux-x86_64.tar.gz
$ wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-7.17.22-linux-x86_64.tar.gz.sha512
```

解压文件

```
$ shasum -a 512 -c elasticsearch-7.17.22-linux-x86_64.tar.gz.sha512
elasticsearch-7.17.22-linux-x86_64.tar.gz: OK

$ tar -xzf elasticsearch-7.17.22-linux-x86_64.tar.gz
$ cd elasticsearch-7.17.22/
```

修改系统参数

```
$ cat >> /etc/security/limits.conf << EOF
* soft nproc 65536
* hard nproc 65536
* soft nofile 65536
* hard nofile 65536
EOF
```

```
$ cat >> /etc/sysctl.conf<< EOF
vm.max_map_count = 655300
EOF
```

载入sysctl配置文件：

```
$ sysctl -p
vm.max_map_count = 655300
```

创建用户以及相关的文件夹

创建用户：

```
$ groupadd elastic

$ useradd elastic -g elastic -s /bin/bash
```

创建elasticsearch数据及日志目录

```
$ chown elastic.elastic -R  /software/elasticsearch-7.17.22
$ ls -l
total 668
-rw-r--r--  1 elastic elastic   3860 Jun  6 15:34 LICENSE.txt
-rw-r--r--  1 elastic elastic 640930 Jun  6 15:36 NOTICE.txt
-rw-r--r--  1 elastic elastic   2710 Jun  6 15:34 README.asciidoc
drwxr-xr-x  2 elastic elastic   4096 Aug 23 15:56 bin
drwxr-xr-x  3 elastic elastic   4096 Aug 23 15:59 config
drwxr-xr-x  3 elastic elastic   4096 Aug 23 15:49 data
drwxr-xr-x  8 elastic elastic   4096 Jun  6 15:39 jdk
drwxr-xr-x  3 elastic elastic   4096 Jun  6 15:39 lib
```

```
drwxr-xr-x  2 elastic elastic   4096 Aug 26 01:03 logs
drwxr-xr-x 61 elastic elastic   4096 Jun  6 15:39 modules
drwxr-xr-x  2 elastic elastic   4096 Jun  6 15:36 plugins
```

- **修改Elasticsearch配置文件**

修改 **./config/elasticsearch.yml** 的文件配置，RPM包安装里面的配置设置。

启动Elasticsearch

```
$ su elastic

$ nohup ./elasticsearch > ./elastic.log 2>&1 &

#查看日志
$ tail -f elastic.log -n 500

#查询进程
$ ps -ef | grep elastic
```

测试Elasticsearch状态

```
$ curl 192.168.3.31:9200
{
  "name" : "elasticsearch-node1",
  "cluster_name" : "y9elasticsearch",
  "cluster_uuid" : "4gAWR1QIRSudSMc18YHZ2g",
  "version" : {
    "number" : "7.17.16",
    "build_flavor" : "default",
    "build_type" : "docker",
    "build_hash" : "2b23fa076334f8d4651aeebe458a955a2ae23218",
    "build_date" : "2023-12-08T10:06:54.672540567Z",
    "build_snapshot" : false,
    "lucene_version" : "8.11.1",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
```

开启验证

第一步 切换到elastsearch的目录下，使用下列命令生成证书

```
$ cd /software/elasticsearch-7.17.22/
$ bin/elasticsearch-certutil cert -out config/elastic-certificates.p12 -pass ""
```

注意，要让新生成的证书文件能够被普通用户读取到，即，需要重新更改一次权限。

```
$ chown elastic.elastic -R  /software/elasticsearch-7.17.22
```

第二步 打开 config/elasticsearch.yaml ，在尾部添加下面的配置代码：

```
$ cat >> config/elasticsearch.yml << EOF
xpack.security.enabled: true
xpack.security.transport.ssl.enabled: true
xpack.security.transport.ssl.verification_mode: certificate
xpack.security.transport.ssl.keystore.path: elastic-certificates.p12
xpack.security.transport.ssl.truststore.path: elastic-certificates.p12

EOF
```

启动 elasticsearch 服务，使用cd命令切换到 elasticsearch 目录。使用一下命令设置密码，然后一一输入账号的密码

```
$ ./elasticsearch-setup-passwords interactive
```

重新登录就会发现需要进行验证了，输入密码，即可显示安装信息

```
$ curl 192.168.3.31:9200 -u elastic
Enter host password for user 'elastic':
{
  "name" : "elasticsearch-node1",
  "cluster_name" : "y9elasticsearch",
  "cluster_uuid" : "4gAWR1QIRSudSMc18YHZ2g",
  "version" : {
    "number" : "7.17.16",
    "build_flavor" : "default",
    "build_type" : "docker",
    "build_hash" : "2b23fa076334f8d4651aeebe458a955a2ae23218",
    "build_date" : "2023-12-08T10:06:54.672540567Z",
    "build_snapshot" : false,
    "lucene_version" : "8.11.1",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
```

升级 （示例：7.9.3 升级至 7.17.22，有开启验证）

1、解压 elasticsearch-7.17.22.tar.gz

2、停止 elasticsearch 服务，拷贝7.9.3版本的配置到7.17.22版本。

- 拷贝 elasticsearch.yml，elasticsearch.keystore，jvm.options，elastic-certificates.p12 文件到 config 路径下

  注：拷贝 jvm.options 文件（官方专门强调的）

3、拷贝 path.data（data数据目录）的内容至新版本

拷贝完，更新一下用户和用户组权限，例如：

```
chown -R elastic:elastic elasticsearch-7.17.22
```

4、修改ES配置文件，设置密码

第一步、修改 elasticsearch.yml 配置，将身份验证相关配置屏蔽掉；

```
$ vim config/elasticsearch.yml

#将验证关闭
xpack.security.enabled: false
```

第二步、切换 elastic 用户，启动 elasticsearch 服务

```
$ systemctl restart elasticsearch

或者
$ nohup ./elasticsearch > ./elastic.log 2>&1 &
```

删除多余的.security-7索引

```
$  curl -X DELETE "http://192.168.3.38:9206/.security-*"
```

到此就回到 ES 没有设置密码的阶段了，重新将验证打开，重启服务，执行设置密码的操作

第三步、设置 elasticsearch 密码

先将验证开启

```
xpack.security.enabled: true
```

然后重启服务执行下面设置命令

```
$ bin/elasticsearch-setup-passwords interactive
```

因为需要设置 elastic，apm_system，kibana，kibana_system，logstash_system，beats_system，remote_monitoring_user
这些用户的密码，故这个过程比较漫长，耐心设置；

MongoDB

手动下载安装包安装

下载装包

https://fastdl.mongodb.org/linux/mongodb-linux-x86_64-rhel70-5.0.28.tgz

其他版本和平台安装包可前往 MongoDB Community Downloads 下载中心下载

详情可参考官网安装手册

解压到/usr/local目录下

```
$ wget https://fastdl.mongodb.org/linux/mongodb-linux-x86_64-rhel70-5.0.28.tgz
$ tar -xvf mongodb-linux-x86_64-rhel70-5.0.28.tgz -C /usr/local
$ cd /usr/local/
$ mv mongodb-linux-x86_64-rhel70-5.0.28 mongodb
```

创建目录及文件

```
$ mkdir -p /usr/local/mongodb/{data,logs,conf}
$ touch /usr/local/mongodb/conf/mongodb.conf
```

在mongodb.conf添加配置内容

```
$ cat >> /usr/local/mongodb/conf/mongodb.conf <<EOF
# mongod.conf
# for documentation of all options, see:
#   http://docs.mongodb.org/manual/reference/configuration-options/
# where to write logging data.
systemLog:
  destination: file
  logAppend: true
  path: /usr/local/mongodb/logs/mongod.log
# Where and how to store data.
storage:
  dbPath: /usr/local/mongodb/data
  journal:
    enabled: true
#  engine:
#  mmapv1:
#  wiredTiger:
# how the process runs
processManagement:
  fork: true  # fork and run in background
  pidFilePath: /usr/local/mongodb/mongod.pid  # location of pidfile
# network interfaces
net:
  port: 27017
  bindIp: 0.0.0.0  # Listen to local interface only, comment to listen on all
interfaces.
security:
 authorization: enabled
#operationProfiling:
#replication:
#sharding:
## Enterprise-Only Options
#auditLog:
#snmp:
EOF
```

详细的mongoDB配置参考

https://docs.mongodb.com/manual/reference/configuration-options

禁用 selinux

修改 /etc/selinux/config

```
SELINUX=disabled
```

启动 mongodb

```
$ nohup /usr/local/mongodb/bin/mongod --config
/usr/local/mongodb/conf/mongodb.conf &
```

注册 systemd service

在 `/usr/lib/systemd/system` 目录下增加 `mongodb.service`，目录必须是绝对目录。

```
$ cat >>/usr/lib/systemd/system/mongodb.service << EOF
[Unit]
Description=mongodb
After=network.target remote-fs.target nss-lookup.target
[Service]
Type=forking
ExecStart=/usr/local/mongodb/bin/mongod --config
/usr/local/mongodb/conf/mongodb.conf
ExecReload=/bin/kill -s HUP $MAINPID
ExecStop=/usr/local/mongodb/bin/mongod --shutdown
/usr/local/mongodb/conf/mongodb.conf
PrivateTmp=true
[Install]
WantedBy=multi-user.target
EOF
```

加载服务，设置开机自启动，然后启动：

```
$ systemctl daemon-reload
$ systemctl enable mongodb.service
$ systemctl start mongodb.service
```

MongoDB设置用户验证

设置需要先关闭验证，不然无法登录。

1、进入mongodb

```
$ ./mongo 127.0.0.1:27017 或者 mongo直接进入
```

2、设置用户，db.auth验证返回1表示设置成功：

```
use admin
db.createUser({user:"y9admin",pwd:"83204585",roles:["root"],mechanisms :
["SCRAM-SHA-1"] })

use y9database
db.createUser({user: "y9admin",pwd: "83204585",roles:[{ role: "readWrite", db:
"y9database" }],mechanisms : ["SCRAM-SHA-1"]})

db.auth("y9admin", "83204585");
exit
```

3、修改mongodb.conf

```
security:
  authorization: enabled
```

4、重启服务MongoDB。

5、验证是否需要开启验证：

```
$ ./mongo localhost:27017
> use admin
> show dbs
2017-12-28T11:45:43.112+0800 E QUERY    [thread1] Error: listDatabases failed:{
        "ok" : 0,
        "errmsg" : "not authorized on admin to execute command { listDatabases:
1.0 }",
        "code" : 13,
        "codeName" : "Unauthorized"
} :
 _getErrorWithCode@src/mongo/shell/utils.js:25:13
Mongo.prototype.getDBs@src/mongo/shell/mongo.js:62:1
shellHelper.show@src/mongo/shell/utils.js:769:19
shellHelper@src/mongo/shell/utils.js:659:15
@(shellhelp2):1:1
> exit

$ ./mongo localhost:27017/admin -u y9admin -p 83204585
show dbs
admin       0.000GB
local       0.000GB
y9database  0.001GB
```

4.2升级到5.0版本

mongodb 的稳定版本全部为偶数，例如：4.0、4.2、4.4、5.0。不能跨大版本升级，需要先由4.2升级到4.4再升级到5.0。

第一步 版本参数修改

```
$ ./mongo localhost:27017
> use admin
> db.auth("y9admin", "83204585");
> db.adminCommand( { setFeatureCompatibilityVersion: "4.2" } )
> db.shutdownServer();
> exit
```

第二步 首先停止数据库，然后官网下载要升级的版本,先升级至4.4

```
$ wget https://fastdl.mongodb.org/linux/mongodb-linux-x86_64-rhel70-4.4.18.tgz
$ tar -xvf mongodb-linux-x86_64-rhel70-4.4.18.tgz -C /usr/local
$ cd /usr/local/
$ mv mongodb-linux-x86_64-rhel70-4.4.18.tgz mongodb-4.4.18
$ mkdir -p /usr/local/mongodb-4.4.18/{data,logs,conf}
$ cp /usr/local/mongodb/conf/mongodb.conf /usr/local/mongodb-4.4.18/conf
```

修改mongodb.conf里面的数据目录，日志目录和pid存放目录，

然后复制旧版数据至新版本中

```
$ cd /usr/local/mongodb/
$ cp data -R /usr/local/mongodb-4.4.18/
$ cd /usr/local/mongodb-4.4.18/bin
```

第三步 启动数据库，使用mongo命令连上数据库设置featureCompatibilityVersion版本号为5.0，即完成了4.2升级到4.4的工作。

```
$ ./mongod --config /usr/local/mongodb-4.4.18/mongodb.conf -fork
$ ./mongo localhost:27017
> use admin
> db.auth("y9admin", "83204585");
> db.adminCommand( { setFeatureCompatibilityVersion: "4.4" } )
> db.shutdownServer();
> exit
```

然后修改 /usr/lib/systemd/system/mongodb.service 里面的目录，指向新版本目录地址，将原来的服务停止，重新加载服务，并启动：

```
$ systemctl disable mongodb.service
$ systemctl daemon-reload
$ systemctl enable mongodb.service
$ systemctl start mongodb.service
```

4.4 升级 5.0 的步骤如上，设置版本为5.0

```
db.adminCommand( { setFeatureCompatibilityVersion: "5.0" } )
```

Consul

下载安装包，地址：

https://releases.hashicorp.com/consul/1.4.3/consul_1.4.3_linux_amd64.zip

安装配置

```
$ unzip consul_1.4.3_linux_amd64.zip
$ mkdir -p /opt/consul/{data,conf}
$ cp consul /opt/consul/

$ cat > /opt/consul/conf/config.json <<EOF
{
  "bind_addr": "192.168.33.131",
  "client_addr": "0.0.0.0",
  "bootstrap_expect": 1,
  "ports": {
    "dns": -1,
    "http": 8500,
    "server": 8300
  },
  "log_level": "INFO",
  "server": true
}
EOF
```

创建服务

```
$ cat >> /usr/lib/systemd/system/consul.service << EOF
[Unit] Description=consul
[Service]
ExecStart=/opt/consul/consul agent -config-dir /opt/consul/conf -data-dir
/opt/consul/data --ui KillSignal=SIGINT
ExecReload=/bin/kill -s HUP $MAINPID
ExecStop=/bin/kill -s QUIT $MAINPID

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

启动Consul服务

```
$ systemctl daemon-reload
$ systemctl enable consul.service
$ systemctl start consul.service
```

Nacos

下载编译后压缩包方式

您可以从 最新稳定版本 下载 `nacos-server-$version.zip` 包。

```
$ wget https://github.com/alibaba/nacos/releases/download/2.2.1/nacos-server-
2.2.1.tar.gz
$ tar -xvf nacos-server-2.2.1.tar.gz
```

修改配置文件 conf/application.properties :

```
#*************** Config Module Related Configurations ***************#
spring.datasource.platform=mysql

db.num=1
db.url.0=jdbc:mysql://192.168.0.1:3306/nacos?
serverTimezone=GMT%2B8&nullCatalogMeansCurrent=true&useUnicode=true&characterEnc
oding=utf-
8&rewriteBatchedStatements=true&useCompression=true&useSSL=false&allowPublicKeyR
etrieval=true
db.user=root
db.password=111111

server.tomcat.basedir=/software/nacos/tomcat
nacos.core.auth.enabled=true
nacos.core.auth.server.identity.key=serverIdentity
nacos.core.auth.server.identity.value=security

nacos.core.auth.plugin.nacos.token.secret.key=SecretKey012345678901234567890123
567890123456789012345678901234567890123456789
```

启动命令(standalone代表着单机模式运行，非集群模式):

```
$ cd nacos/bin
$ sh startup.sh -m standalone
```

如果您使用的是ubuntu系统，或者运行脚本报错提示[[符号找不到，可尝试如下运行：

```
bash startup.sh -m standalone
```

关闭命令

```
sh shutdown.sh
```

Traefik

下载二进制包

https://github.com/containous/traefik/releases

配置（示例）

```
$ mkdir /traefik
$ vi /traefik/traefik.toml
debug = false
logLevel = "INFO"
[traefikLog]
filePath = "/traefik/traefik.log"
format = "json"

[accessLog]
filePath = "/traefik/access.log"
format = "json"
defaultEntryPoints = ["http"]
[entryPoints]
[entryPoints.http]
address = ":80"
[file]
[backends]
[backends.backend1]
[backends.backend1.servers]
[backends.backend1.servers.server0]
url = "http://10.161.56.145:7055"
weight = 1
[backends.backend1.loadBalancer]
method = "drr"
[frontends]
[frontends.frontend1]
entryPoints = ["http"]
backend = "backend1"
passHostHeader = true
[frontends.frontend1.routes]
[frontends.frontend1.routes.route1]
rule = "PathPrefix:/admin-dev;Host:192.168.175.128"
```

创建服务

```
$ cat >> /usr/lib/systemd/system/traefik.service << EOF
[Unit]
Description=traefik
[Service]
ExecStart=/traefik/traefik -c /traefik/traefik.toml
KillSignal=SIGINT
ExecReload=/bin/kill -s HUP $MAINPID
ExecStop=/bin/kill -s Quit $MAINPID
[Install]
WantedBy=multi-user.target
EOF
```

启动 traefik 服务

```
$ systemctl daemon-reload
$ systemctl enable traefik.service
$ systemctl start traefik.service
```