

常用的 docker-compose

作者: [QForever](#)

原文链接: <https://ld246.com/article/1592323164120>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

[点击下载docker-compose](#)

一、 Gitlab

```
version: '3'
services:
  gitlab:
    restart: always
    image: gitlab/gitlab-ee:latest
    container_name: gitlab
    privileged: true
    ports:
      - '80:80'
      - '443:443'
    volumes:
      - ./gitlab:/etc/gitlab
      - ./opt:/var/opt/gitlab
      - ./log:/var/log/gitlab
```

二、 Jenkins

```
version: '3'
services:
  jenkins:
    restart: always
    image: jenkins/jenkins:its
    container_name: jenkins
    ports:
      - '8080:8080'
      - '50000:50000'
    volumes:
      - ./data:/var/jenkins_home
      - ./apache-maven-3.6.3:/var/apache-maven-3.6.3
      - ./gradle-4.0.1:/var/gradle-4.0.1
      - ./jdk1.8.0_181:/var/jdk1.8.0_181
      - ./git-2.25.0:/var/git-2.25.0
```

三、 MySQL

```
# docker-compose.yaml的书写格式版本
version: "3.0"
# 定义服务,可以有很多服务,说白了就是多个容器,这个值是固定
services:
# 自定义服务
  mysql:
    # mysql依赖的基础镜像
    image: mysql:5.7
    # 端口
```

```
ports:
  - "3306:3306"
# 容器名
container_name: mysql
# 数据卷
volumes:
  # 将mysql的数据持久化到宿主机
  - "./data:/var/lib/mysql"
  # 让容器中mysql的配置使用我们自定义的配置文件,主要是编码问题
  - "./conf/my.cnf:/etc/mysql/my.cnf"
environment:
  # mysql的root用户密码
  - MYSQL_ROOT_PASSWORD=123456
```

my.cnf

```
[mysqld]
character_set_server=utf8
pid-file = /var/run/mysqld/mysqld.pid
socket = /var/run/mysqld/mysqld.sock
datadir = /var/lib/mysql
```

```
version: "3.0"
services:
  mysql:
    image: mysql:5.7.26
    ports:
      - "3366:3306"
    container_name: mysql
    restart: always
    privileged: true
    command:
      --character-set-server=utf8
      --lower_case_table_names=1
    volumes:
      - "./data:/var/lib/mysql"
    environment:
      - MYSQL_ROOT_PASSWORD=123456
```

四、Nexus

```
version: '3'
services:
  nexus:
    restart: always
    image: sonatype/nexus3:latest
    container_name: nexus3
    ports:
      - '8081:8081'
      - '8082:8082'
      - '8083:8083'
      - '8084:8084'
      - '8085:8085'
    volumes:
```

- ./data:/nexus-data

五、 Redis

version: '3'

services:

redis:

image: redis:4.0.6

container_name: redis

restart: always

command: --appendonly yes

ports:

- 6379:6379

volumes:

- ./data:/data

六、 RocketMQ

version: '3'

services:

rmqnamesrv:

image: rocketmqinc/rocketmq:4.4.0

container_name: rmqnamesrv

ports:

- 9876:9876

volumes:

- ./namesrv/logs:/root/logs

- ./namesrv/store:/root/store

environment:

- MAX_POSSIBLE_HEAP=100000000

command: sh mqnamesrv

rmqbroker:

image: rocketmqinc/rocketmq:4.4.0

container_name: rmqbroker

ports:

- 10909:10909

- 10911:10911

volumes:

- ./broker/conf/broker.conf:/opt/rocketmq-4.4.0/conf/broker.conf

- ./broker/logs:/root/logs

- ./broker/store:/root/store

depends_on:

- rmqnamesrv

links:

- rmqnamesrv:namesrv

command: sh mqbroker -n 192.168.17.101:9876 -c ../conf/broker.conf

environment:

- NAMESRV_ADDR=192.168.17.101:9876

- MAX_POSSIBLE_HEAP=200000000

console:

image: styletang/rocketmq-console-ng:latest

container_name: rocketmq-console-ng

ports:

```
- 8087:8080
depends_on:
- rmqnamesrv
- rmqbroker
environment:
- JAVA_OPTS=-Drocketmq.namesrv.addr=192.168.17.101:9876
- Dcom.rocketmq.sendMessageWithVIPChannel=false
```

七、Nginx

```
version: "3"
services:
  nginx:
    image: docker.io/nginx:latest
    restart: always
    privileged: true
    ports:
      - "80:80"
      - "443:443"
    container_name: nginx
    volumes:
      - "./nginx/conf/nginx.conf:/etc/nginx/nginx.conf"
      - "./nginx/conf/conf.d:/etc/nginx/conf.d"
      - "./nginx/www:/usr/share/nginx/html"
      - "./nginx/logs:/var/log/nginx"
      - "./nginx/ssl:/ssl"
```

nginx.conf

```
user nginx;
worker_processes 1;

error_log /var/log/nginx/error.log warn;
pid /var/run/nginx.pid;

events {
    worker_connections 1024;
}

http {
    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    log_format main '$remote_addr - $remote_user [$time_local] "$request" '
        '$status $body_bytes_sent "$http_referer" '
        '"$http_user_agent" "$http_x_forwarded_for"';

    access_log /var/log/nginx/access.log main;

    sendfile on;
    #tcp_nopush on;
```

```
    keepalive_timeout 65;

    #gzip on;

    include /etc/nginx/conf.d/*.conf;
}
```

八、 ZipKin

```
version: '3.0'
services:
  zipkin:
    image: openzipkin/zipkin
    restart: always
    container_name: zipkin
    ports:
      - 9411:9411
```

九、 Elasticsearch

```
version: '3'
services:
  elasticsearch:
    image: elasticsearch:6.4.3
    container_name: elasticsearch
    restart: always
    volumes:
      - ./data:/usr/share/elasticsearch/data
    environment:
      - "ES_JAVA_OPTS=-Xms512m -Xmx512m"
      - "cluster.name=elasticsearch"
    ports:
      - "9200:9200"
      - "9300:9300"
```

十、 Nacos

```
version: "3"
services:
  nacos:
    image: nacos/nacos-server:latest
    container_name: nacos
    volumes:
      - ./logs:/home/nacos/logs
      - ./conf/custom.properties:/home/nacos/init.d/custom.properties
    ports:
      - "8848:8848"
      - "9555:9555"
    privileged: true
    restart: always
    environment:
      - "PREFER_HOST_MODE=hostname"
```

- "MODE=standalone"
- "SPRING_DATASOURCE_PLATFORM=mysql"
- "MYSQL_SERVICE_HOST=192.168.17.199"
- "MYSQL_SERVICE_DB_NAME=nacos"
- "MYSQL_SERVICE_PORT=3306"
- "MYSQL_SERVICE_USER=root"
- "MYSQL_SERVICE_PASSWORD=123456"

custom.properties

```
#spring.security.enabled=false
#management.security=false
#security.basic.enabled=false
#nacos.security.ignore.urls=/**
#management.metrics.export.elastic.host=http://localhost:9200
# metrics for prometheus
management.endpoints.web.exposure.include=*
```

```
# metrics for elastic search
#management.metrics.export.elastic.enabled=false
#management.metrics.export.elastic.host=http://localhost:9200
```

```
# metrics for influx
#management.metrics.export.influx.enabled=false
#management.metrics.export.influx.db=springboot
#management.metrics.export.influx.uri=http://localhost:8086
#management.metrics.export.influx.auto-create-db=true
#management.metrics.export.influx.consistency=one
#management.metrics.export.influx.compressed=true
```

sql

```
/*
*****
/* 数据库全名 = nacos_config */
/* 表名称 = config_info */
*****
CREATE TABLE `config_info` (
  `id` BIGINT(20) NOT NULL AUTO_INCREMENT COMMENT 'id',
  `data_id` VARCHAR(255) NOT NULL COMMENT 'data_id',
  `group_id` VARCHAR(255) DEFAULT NULL,
  `content` LONGTEXT NOT NULL COMMENT 'content',
  `md5` VARCHAR(32) DEFAULT NULL COMMENT 'md5',
  `gmt_create` DATETIME NOT NULL DEFAULT '2010-05-05 00:00:00' COMMENT '创建时间',
  `gmt_modified` DATETIME NOT NULL DEFAULT '2010-05-05 00:00:00' COMMENT '修改时间',
  `src_user` TEXT COMMENT 'source user',
  `src_ip` VARCHAR(20) DEFAULT NULL COMMENT 'source ip',
  `app_name` VARCHAR(128) DEFAULT NULL,
  `tenant_id` VARCHAR(128) DEFAULT '' COMMENT '租户字段',
  `c_desc` VARCHAR(256) DEFAULT NULL,
  `c_use` VARCHAR(64) DEFAULT NULL,
  `effect` VARCHAR(64) DEFAULT NULL,
  `type` VARCHAR(64) DEFAULT NULL,
  `c_schema` TEXT,
  PRIMARY KEY (`id`),
```

```

UNIQUE KEY `uk_configinfo_datagrouptenant` (`data_id`,`group_id`,`tenant_id`)
) ENGINE=INNODB DEFAULT CHARSET=utf8 COLLATE=utf8_bin COMMENT='config_info';

/*****/
/* 数据库全名 = nacos_config */
/* 表名称 = config_info_aggr */
/*****/
CREATE TABLE `config_info_aggr` (
  `id` BIGINT(20) NOT NULL AUTO_INCREMENT COMMENT 'id',
  `data_id` VARCHAR(255) NOT NULL COMMENT 'data_id',
  `group_id` VARCHAR(255) NOT NULL COMMENT 'group_id',
  `datum_id` VARCHAR(255) NOT NULL COMMENT 'datum_id',
  `content` LONGTEXT NOT NULL COMMENT '内容',
  `gmt_modified` DATETIME NOT NULL COMMENT '修改时间',
  `app_name` VARCHAR(128) DEFAULT NULL,
  `tenant_id` VARCHAR(128) DEFAULT "" COMMENT '租户字段',
  PRIMARY KEY (`id`),
  UNIQUE KEY `uk_configinfoaggr_datagrouptenantdatum` (`data_id`,`group_id`,`tenant_id`,`datum_id`)
) ENGINE=INNODB DEFAULT CHARSET=utf8 COLLATE=utf8_bin COMMENT='增加租户字段';

/*****/
/* 数据库全名 = nacos_config */
/* 表名称 = config_info_beta */
/*****/
CREATE TABLE `config_info_beta` (
  `id` BIGINT(20) NOT NULL AUTO_INCREMENT COMMENT 'id',
  `data_id` VARCHAR(255) NOT NULL COMMENT 'data_id',
  `group_id` VARCHAR(128) NOT NULL COMMENT 'group_id',
  `app_name` VARCHAR(128) DEFAULT NULL COMMENT 'app_name',
  `content` LONGTEXT NOT NULL COMMENT 'content',
  `beta_ips` VARCHAR(1024) DEFAULT NULL COMMENT 'betaIps',
  `md5` VARCHAR(32) DEFAULT NULL COMMENT 'md5',
  `gmt_create` DATETIME NOT NULL DEFAULT '2010-05-05 00:00:00' COMMENT '创建时间',
  `gmt_modified` DATETIME NOT NULL DEFAULT '2010-05-05 00:00:00' COMMENT '修改时间',
  `src_user` TEXT COMMENT 'source user',
  `src_ip` VARCHAR(20) DEFAULT NULL COMMENT 'source ip',
  `tenant_id` VARCHAR(128) DEFAULT "" COMMENT '租户字段',
  PRIMARY KEY (`id`),
  UNIQUE KEY `uk_configinfobeta_datagrouptenant` (`data_id`,`group_id`,`tenant_id`)
) ENGINE=INNODB DEFAULT CHARSET=utf8 COLLATE=utf8_bin COMMENT='config_info_beta';

/*****/
/* 数据库全名 = nacos_config */
/* 表名称 = config_info_tag */
/*****/
CREATE TABLE `config_info_tag` (
  `id` BIGINT(20) NOT NULL AUTO_INCREMENT COMMENT 'id',
  `data_id` VARCHAR(255) NOT NULL COMMENT 'data_id',
  `group_id` VARCHAR(128) NOT NULL COMMENT 'group_id',
  `tenant_id` VARCHAR(128) DEFAULT "" COMMENT 'tenant_id',
  `tag_id` VARCHAR(128) NOT NULL COMMENT 'tag_id',

```



```

`app_name` VARCHAR(128) DEFAULT NULL COMMENT 'app_name',
`content` LONGTEXT NOT NULL COMMENT 'content',
`md5` VARCHAR(32) DEFAULT NULL COMMENT 'md5',
`gmt_create` DATETIME NOT NULL DEFAULT '2010-05-05 00:00:00' COMMENT '创建时间',
`gmt_modified` DATETIME NOT NULL DEFAULT '2010-05-05 00:00:00' COMMENT '修改时间',
`src_user` TEXT COMMENT 'source user',
`src_ip` VARCHAR(20) DEFAULT NULL COMMENT 'source ip',
PRIMARY KEY (`id`),
UNIQUE KEY `uk_configinfotag_datagrouptenanttag` (`data_id`,`group_id`,`tenant_id`,`tag_id`)
) ENGINE=INNODB DEFAULT CHARSET=utf8 COLLATE=utf8_bin COMMENT='config_info_tag'

```

```

/*****/
/* 数据库全名 = nacos_config */
/* 表名称 = config_tags_relation */
/*****/
CREATE TABLE `config_tags_relation` (
  `id` BIGINT(20) NOT NULL COMMENT 'id',
  `tag_name` VARCHAR(128) NOT NULL COMMENT 'tag_name',
  `tag_type` VARCHAR(64) DEFAULT NULL COMMENT 'tag_type',
  `data_id` VARCHAR(255) NOT NULL COMMENT 'data_id',
  `group_id` VARCHAR(128) NOT NULL COMMENT 'group_id',
  `tenant_id` VARCHAR(128) DEFAULT '' COMMENT 'tenant_id',
  `nid` BIGINT(20) NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (`nid`),
  UNIQUE KEY `uk_configtagrelation_configidtag` (`id`,`tag_name`,`tag_type`),
  KEY `idx_tenant_id` (`tenant_id`)
) ENGINE=INNODB DEFAULT CHARSET=utf8 COLLATE=utf8_bin COMMENT='config_tag_relation';

```

```

/*****/
/* 数据库全名 = nacos_config */
/* 表名称 = group_capacity */
/*****/
CREATE TABLE `group_capacity` (
  `id` BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT COMMENT '主键ID',
  `group_id` VARCHAR(128) NOT NULL DEFAULT '' COMMENT 'Group ID, 空字符表示整个集群',
  `quota` INT(10) UNSIGNED NOT NULL DEFAULT '0' COMMENT '配额, 0表示使用默认值',
  `usage` INT(10) UNSIGNED NOT NULL DEFAULT '0' COMMENT '使用量',
  `max_size` INT(10) UNSIGNED NOT NULL DEFAULT '0' COMMENT '单个配置大小上限, 单位字节, 0表示使用默认值',
  `max_aggr_count` INT(10) UNSIGNED NOT NULL DEFAULT '0' COMMENT '聚合子配置最大个数, 0表示使用默认值',
  `max_aggr_size` INT(10) UNSIGNED NOT NULL DEFAULT '0' COMMENT '单个聚合数据的子配置大小上限, 单位为字节, 0表示使用默认值',
  `max_history_count` INT(10) UNSIGNED NOT NULL DEFAULT '0' COMMENT '最大变更历史数',
  `gmt_create` DATETIME NOT NULL DEFAULT '2010-05-05 00:00:00' COMMENT '创建时间',
  `gmt_modified` DATETIME NOT NULL DEFAULT '2010-05-05 00:00:00' COMMENT '修改时间',
  PRIMARY KEY (`id`),
  UNIQUE KEY `uk_group_id` (`group_id`)
) ENGINE=INNODB DEFAULT CHARSET=utf8 COLLATE=utf8_bin COMMENT='集群、各Group量信息表';

```

```

/*****/
/* 数据库全名 = nacos_config */
/* 表名称 = his_config_info */
/*****/
CREATE TABLE `his_config_info` (
  `id` BIGINT(64) UNSIGNED NOT NULL,
  `nid` BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT,
  `data_id` VARCHAR(255) NOT NULL,
  `group_id` VARCHAR(128) NOT NULL,
  `app_name` VARCHAR(128) DEFAULT NULL COMMENT 'app_name',
  `content` LONGTEXT NOT NULL,
  `md5` VARCHAR(32) DEFAULT NULL,
  `gmt_create` DATETIME NOT NULL DEFAULT '2010-05-05 00:00:00',
  `gmt_modified` DATETIME NOT NULL DEFAULT '2010-05-05 00:00:00',
  `src_user` TEXT,
  `src_ip` VARCHAR(20) DEFAULT NULL,
  `op_type` CHAR(10) DEFAULT NULL,
  `tenant_id` VARCHAR(128) DEFAULT '' COMMENT '租户字段',
  PRIMARY KEY (`nid`),
  KEY `idx_gmt_create` (`gmt_create`),
  KEY `idx_gmt_modified` (`gmt_modified`),
  KEY `idx_data_id` (`data_id`)
) ENGINE=INNODB DEFAULT CHARSET=utf8 COLLATE=utf8_bin COMMENT='多租户改造';

```

```

/*****/
/* 数据库全名 = nacos_config */
/* 表名称 = tenant_capacity */
/*****/
CREATE TABLE `tenant_capacity` (
  `id` BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT COMMENT '主键ID',
  `tenant_id` VARCHAR(128) NOT NULL DEFAULT '' COMMENT 'Tenant ID',
  `quota` INT(10) UNSIGNED NOT NULL DEFAULT '0' COMMENT '配额, 0表示使用默认值',
  `usage` INT(10) UNSIGNED NOT NULL DEFAULT '0' COMMENT '使用量',
  `max_size` INT(10) UNSIGNED NOT NULL DEFAULT '0' COMMENT '单个配置大小上限, 单位字节, 0表示使用默认值',
  `max_aggr_count` INT(10) UNSIGNED NOT NULL DEFAULT '0' COMMENT '聚合子配置最大个数',
  `max_aggr_size` INT(10) UNSIGNED NOT NULL DEFAULT '0' COMMENT '单个聚合数据的子配置大小上限, 单位为字节, 0表示使用默认值',
  `max_history_count` INT(10) UNSIGNED NOT NULL DEFAULT '0' COMMENT '最大变更历史数',
  `gmt_create` DATETIME NOT NULL DEFAULT '2010-05-05 00:00:00' COMMENT '创建时间',
  `gmt_modified` DATETIME NOT NULL DEFAULT '2010-05-05 00:00:00' COMMENT '修改时间',
  PRIMARY KEY (`id`),
  UNIQUE KEY `uk_tenant_id` (`tenant_id`)
) ENGINE=INNODB DEFAULT CHARSET=utf8 COLLATE=utf8_bin COMMENT='租户容量信息表';

```

```

CREATE TABLE `tenant_info` (
  `id` BIGINT(20) NOT NULL AUTO_INCREMENT COMMENT 'id',
  `kp` VARCHAR(128) NOT NULL COMMENT 'kp',
  `tenant_id` VARCHAR(128) DEFAULT '' COMMENT 'tenant_id',
  `tenant_name` VARCHAR(128) DEFAULT '' COMMENT 'tenant_name',

```

```
`tenant_desc` VARCHAR(256) DEFAULT NULL COMMENT 'tenant_desc',
`create_source` VARCHAR(32) DEFAULT NULL COMMENT 'create_source',
`gmt_create` BIGINT(20) NOT NULL COMMENT '创建时间',
`gmt_modified` BIGINT(20) NOT NULL COMMENT '修改时间',
PRIMARY KEY (`id`),
UNIQUE KEY `uk_tenant_info_kptenantid` (`kp`,`tenant_id`),
KEY `idx_tenant_id` (`tenant_id`)
) ENGINE=INNODB DEFAULT CHARSET=utf8 COLLATE=utf8_bin COMMENT='tenant_info';
```

```
CREATE TABLE users (
  username VARCHAR(50) NOT NULL PRIMARY KEY,
  PASSWORD VARCHAR(500) NOT NULL,
  enabled BOOLEAN NOT NULL
);
```

```
CREATE TABLE roles (
  username VARCHAR(50) NOT NULL,
  role VARCHAR(50) NOT NULL,
  CONSTRAINT uk_username_role UNIQUE (username,role)
);
```

```
CREATE TABLE permissions (
  role VARCHAR(50) NOT NULL,
  resource VARCHAR(512) NOT NULL,
  ACTION VARCHAR(8) NOT NULL,
  CONSTRAINT uk_role_permission UNIQUE (role,resource,ACTION)
);
```

```
INSERT INTO users (username, PASSWORD, enabled) VALUES ('nacos', '$2a$10$EuWPZHzz32
JN7jexM34MOeYirDdFAZm2kuWj7VEOJhhZkDrxfvUu', TRUE);
```

```
INSERT INTO roles (username, role) VALUES ('nacos', 'ROLE_ADMIN');
```

十一、Sentinel

```
version: '3.0'
services:
  sentinel:
    image: bladex/sentinel-dashboard:latest
    container_name: sentinel
    restart: always
    privileged: true
    ports:
      - '8858:8858'
```