

## ## machine

lab = 10.1.1.151	fqdn=bflab.sc.local
dev = 10.1.1.152	fqdn=bfdev.sc.local
bf-qat=35.201.169.59	fqdn=bf-qat.sc.gcp
bf-uat / bf-prod	

[https://docs.google.com/spreadsheets/d/1Yn0O3ttDlr5ZZdtT96bbf64bJUI-fkoTQNQvCTpSvfE/edit?ouid=104417807377034085472&usp=sheets\\_home&ths=true](https://docs.google.com/spreadsheets/d/1Yn0O3ttDlr5ZZdtT96bbf64bJUI-fkoTQNQvCTpSvfE/edit?ouid=104417807377034085472&usp=sheets_home&ths=true)

## ## rd modules

7011	user-1.0.jar	
7012	pay-1.0.jar	
7014	admin-1.0.jar	
7015	activity-1.0.jar	
7017	auth-1.0.jar	
7021 (current port is 7013)	hierarchy-1.0.jar	
8011	user-1.0.jar	cd /opt/gm/user ./start.sh nohup java -Dspring.profiles.active=bf_dev -jar <b>user-1.0.jar</b> &>user.log &
8012	pay-1.0.jar	cd /opt/gm/pay ./start.sh nohup java -Dspring.profiles.active=bf_dev -jar <b>pay-1.0.jar</b> &>pay.log &
8014	admin-1.0.jar	cd /opt/gm/admin ./start.sh nohup java -Dspring.profiles.active=bf_dev -jar <b>admin-1.0.jar</b> &>admin.log &
8015	activity-1.0.jar	cd /opt/gm/activity ./start.sh nohup java -Dspring.profiles.active=bf_dev -jar <b>activity-1.0.jar</b> &>activity.log &
8016	gateway-1.0.jar	cd /opt/gm/gateway <b>./admin.sh</b> > long script  running in ps /bin/java -Dspring.profiles.active=bf_dev -jar <b>gateway-1.0.jar</b> --logging.file=../logs/gateway-admin.log --server.port=8016 --clientSource=2
8017	auth-1.0.jar	cd /opt/gm/auth ./start.sh nohup java -Dspring.profiles.active=bf_dev -jar <b>auth-1.0.jar</b> &>auth.log &
8018	gateway-1.0.jar	cd /opt/gm/gateway <b>./member.sh</b> > long scrip  running in ps

		/bin/java -Dspring.profiles.active=bf_dev -jar <b>gateway-1.0.jar</b> --logging.file=./logs/gateway-member.log --server.port=8018 --clientSource=1
8019	file-manager-1.0.jar	cd /opt/gm/file-manager .start.sh nohup java -Dspring.profiles.active=bf_dev -jar <b>file-manager-1.0.jar</b> &>file-manager.log &
8020	job-1.0.jar	cd /opt/gm/job .start.sh nohup java -Dspring.profiles.active=bf_dev -jar <b>job-1.0.jar</b> &>job.log &
8021	hierarchy-1.0.jar	cd /opt/gm/hierarchy; ./start.sh  nohup java -Dspring.profiles.active=bf_dev -jar <b>hierarchy-1.0.jar</b> >hierarchy.log 2>&1 &
unknow	dataAnalyze-1.0.jar	<p>deploy to storm-nimbus /opt/apache-storm-1.2.2/dataAnalyze-1.0.jar</p> <p>./bin/storm jar /opt/apache-storm-1.2.2/dataAnalyze-1.0.jar com.gm.topology.gamerecord.GameRecordTopo bf_dev .bin/storm jar /opt/apache-storm-1.2.2/dataAnalyze-1.0.jar com.gm.topology.activity.ActivityStatTopo bf_dev .bin/storm jar /opt/apache-storm-1.2.2/dataAnalyze-1.0.jar com.gm.topology.finance.StatementTopo bf_dev .bin/storm jar /opt/apache-storm-1.2.2/dataAnalyze-1.0.jar com.gm.topology.platform.PlatformStatTopo <b>bf_dev</b></p> <p><b>bf_dev bf_qat bf_uat bf_prod</b></p> <p>未來要提供資料給RD, 他們會包入每個環境需要連線DB等的設定 這些bf_dev 等參數傳入後才會有效</p> <p>上傳 environment_{env}.properties &amp; environment.properties 以qat為例就是environment.properties &amp; environment_qat.properties</p> <p>cd /opt/apache-storm-1.2.2 .bin/storm jar /topo/dataAnalyze-1.0.jar com.gm.topology.gamerecord.GameRecordTopo .bin/storm jar /topo/dataAnalyze-1.0.jar com.gm.topology.activity.ActivityStatTopo .bin/storm jar /topo/dataAnalyze-1.0.jar com.gm.topology.finance.StatementTopo .bin/storm jar /topo/dataAnalyze-1.0.jar com.gm.topology.platform.PlatformStatTopo</p> <p>storm kill</p> <p>storm kill gameRecordAnalyze -w 5 storm kill ActivityStatement -w 5 storm kill FinanceStatement -w 5 storm kill PlatformStatement -w 5</p>

-Dspring.profiles.active=bf\_dev 上到qat 變成 bf\_qat

和youda確認後 這個 log的邏輯是這樣的

- 1) nohup 起程式, 導stdout/stderr到指定檔案,
- 2) 大小到一個程度後, 程式會把檔案移到logs目錄存放 (並改名與壓縮 i.g. pay.log.2020-04-28.0.gz)
- 3) 程式如果重啟, 會先備份log檔到logs目錄, 再導stdout/stderr到檔案中 (不然事發當時的log會被蓋掉)
- 4) 運維可以加強的做為是排程刪除舊的log (上次說保留30天)

```
var_APPNAME="activity-1.0.jar"
kill -9 $(pgrep -f ${var_APPNAME})
```

程式吃RAM有點多~

```
> root 2074 1.8 3.7 6069156 985.426M ? Sl Jul27 109:23 java -Xmx1G -Xms1G -server
-XX:+UseG1GC -XX:MaxGCPauseMillis=20 -XX:InitiatingHeapOccupancyPercent=35
-XX:+ExplicitGCInvokesConcurrent
root 12984 0.9 6.5 10806708 1708.15M ? Sl Jul30 11:24 java -Dspring.profiles.active=bf_qat
-jar job-1.0.jar
root 13041 3.0 6.6 10721124 1722.22M ? Sl Jul30 36:11 java -Dspring.profiles.active=bf_qat
-jar auth-1.0.jar
root 12990 11.7 7.1 10795648 1843.57M pts/0 Sl 10:00 2:16 java
-Dspring.profiles.active=bf_qat -jar activity-1.0.jar
root 13242 2.9 9.4 10715692 2452.48M ? Sl Jul30 34:59 java -Dspring.profiles.active=bf_qat
-jar pay-1.0.jar
root 13093 3.5 9.5 10823348 2479.89M ? Sl Jul30 42:55 java -Dspring.profiles.active=bf_qat
-jar admin-1.0.jar
root 13249 3.1 9.8 10747552 2566.38M ? Sl Jul30 38:01 java -Dspring.profiles.active=bf_qat
-jar user-1.0.jar
[root@bf-qat log]# date
Fri Jul 31 10:19:59 CST 2020
[root@bf-qat log]#
```

## ## qat

hostname:

- bf-qat.sc.gcp

website:

- <http://bf-qat-h5.sc.gcp>
- <http://bf-qat-pc.sc.gcp>
- <http://bf-qat-admin.gc.gcp>
- <http://bf-qat-www.sc.gcp>

## ## uat

server\*12

fe in domain below (admin with whitelist)

- <http://bf-uat-www.sc.gcp>
- <http://bf-uat-admin.sc.gcp>

```
## prod  
server*12  
fe in  
• http://bf-prod-www.sc.gcp  
• http://bf-prod-admin.sc.gcp
```

## ## info

bfh5.sc.local 10.1.1.152 root=/data/html/h5  
bfpc.sc.local 10.1.1.152 root=/data/html/pc  
bfadmin.sc.local 10.1.1.152 root=/data/html/admin

現在的h5 / web / pc的nginx做cache control時是使用private，他會造成無法使用cdn加速的效果  
和 howard確認後，我們 未來上到prod (會用cdn) 的那一套，在 cache-control部份會改成public哦~

all data store in folder /data

fdbs 本來用80 PORT的，改成用 8880(fdbs改docker network hostmode, 現查起來沒有聽80  
redis 本來用remi redis6, 改成ius redis 5

storm 系列中 docker compsed本來有放行 22，現做了關閉

storm的zookeeper 改成固定PORT 2181時和 kafka的zookeeper，相衝，故改成2182

storm中的supervisord本來要auto scale =2，現直接在composed file中 做兩次，故scale就不用了

storm 本來用了 4萬多的high port, 現在改成8千多的PORT

sentinel 的port本來用8019，和filemanger相沖了，現改成9019

frontend用的目錄 現在改放在/data/html中，按功能分開，使用時 要用指定的fqdn登入

mysql升成8後，有password policy，youda說就改密碼，新密碼 =  
6gyQmfv4ithRZuQLr1wH@?

log rotate待處理，現暫定要保留1個月的log

redis有改sysctl，對其它程序的影響

## ## issue / checklist in lab

user > dev (db/java/backend/...) machine \*1

os	
base os	<input checked="" type="checkbox"/> <u>centos_7</u> (EOL: full update to 2020Q4, maintenance update to 2024.6.30 <input type="checkbox"/> centos 8 (EOL: full update to "May 2024", maintenance update to 2029.3.31, not support i386 Systems [hw] ref: <a href="https://wiki.centos.org/About/Product">https://wiki.centos.org/About/Product</a>
timezone	timedatectl set-timezone Asia/Taipei
yum-cron	<input type="checkbox"/> enable (gcp default <input checked="" type="checkbox"/> <u>disable</u>
selinux	<input checked="" type="checkbox"/> <u>enable</u> (default <input type="checkbox"/> disable
firewalld	<input type="checkbox"/> enable <input checked="" type="checkbox"/> <u>disable</u> (gcp default  --- <u>service port list:</u> redis: 6379 monogodb: 27017 mysql: 3306 kafka: 9092 kafka-zookeeper: 2181 nacos: 8886 storm-nimbus: 49773:3773,49772:3772,49627:6627( 3773:3773,3772:3772,6627:6627 storm-supervisor: 8000:8000 storm-supervisor1: 8001:8000  storm-zookeeper: 49181:2181 (2182:2181 storm-ui:49080:8080 (8080:8080 fdfs: 8888,22122,23000 sentinel: 9019, 8719  net.ipv4.ip_local_port_range = 32768 60999
bastion	
jumpserver	<input type="checkbox"/> <a href="https://jumpserver.org/">https://jumpserver.org/</a> bastion server has public ip for we can login gcp servers with lan ip office > wan > jumpserver > lan > gcp servers <input checked="" type="checkbox"/> no bastion server. use public ip, direct connecto gcp servers
hw fw	
sshd	allow from management node (two office and so on...)
httpd	<input type="checkbox"/> allow from everywhere <input type="checkbox"/> block with country code (need solution)
backend/admin	allow from whitelist (customer office and so on...)

n for customer	
others	[ ] ???
software version	
java	[x] <u>openjdk 8 LTS</u> [ ] openjdk 11 LTS
redis	[ ] centos 7 = 3.2.12 (lastest) [ ] centos 8 = 5.0.3-2 (lastest) [x] ius redis5 [ ] remi = 6.0.5 (lastest) [ ] zf = redis-4.0.2.tar.gz
docker	[ ] docker-ce-17.12.1 [x] docker-ce-19.03.12 (lastest)
docker-composed	[x] docker-compose 1.18.0
mongodb	[ ] zf = 4.0 [x] <u>official</u> = 4.2 (lastest)
mysql	[ ] zf = mysql5.6 [x] official = <u>mysql80</u> [ ] mariadb centos 7 = 5.5.65 (lastest) [ ] mariadb centos 8 = 10.3.17 (lastest)
kafka	[ ] zf = kafka_2.12 , ? <a href="https://kafka.apache.org/downloads">https://kafka.apache.org/downloads</a> , no 2.12 in offical website [x] current 2.5
fdfs	[x] zf = ( <u>fastdfs-5.11.tar.gz</u> [yum, lab docker] [ ] 6.06 (lastest)
nacos	[x] zf = nacos-server-1.1.4.tar.gz [ ] <u>nacos-server-1.3.0</u> (lastest)
sentinel	[ ] zf = sentinel-dashboard-1.7.0 [x] <u>sentinel-dashboard-1.7.2</u> (lastest)
server group	
usage	qat_redis (c=1,r=7.5,d=200; idle=99,ava=6.5,d=2% qat_mongo (c-1,r=7.1,d-200; idle=99,ava=6.6,d=3% qat_mysql (c=2,r=12,d=200; idle=99,ava=11,d=3%  qat_kafka (c=2,r=14,d=200; idle=75,ava=5.9,d=11% qat_storm (qat-resource , same as above qat_nacos (qat-resource , same as above qat_sentinel (qat-resource , same as above

	qat_fdfs (c=1,ram=3.6,d=400; idle=99,ava=2.8,d=1%  backend-job
db	mysql redis mongo
kafka	kafka zookeeper
docker	storm nacos sentinel zookeeper supervisord
backend	backend *n (java)
web	<p><u>h5.domain:80 (h5,</u>  <u>web.domain:80</u>  <u>admin.domain:80</u></p> <pre> <u>ngx</u> <u>server</u> { <u>listen 80</u> <u>server_name h5.sc.local;</u> <u>location / {</u> <u>root /var/www/w1</u>  <u>server</u> { <u>listen 80</u> <u>server_name admin.sc.local</u> <u>location /</u> { <u>root /var/www/w2</u> </pre> <p><u>x fdfs.domain</u></p> <p>h5 (nginx)</p> <p>---</p> <p>filemanager 10.140.0.22:8019 [backend-01, java filemanager]  ~~gateway 10.140.0.22:8016 [ qat-backend-01, java gateway]  ~~gatewaysh5 10.140.0.22:8018 [backend-01 , java gateway]  image 10.140.0.23:80 [qat-fdfs]</p>

	web --- pc version (gateway) admin --- ?? alias location / to /opt/gm/admin_view/dist/ ( <a href="http://nginx.org/en/docs/http/ngx_http_core_module.html#alias">http://nginx.org/en/docs/http/ngx_http_core_module.html#alias</a> it is <u>better to use the root directive instead:</u> ) gateway 10.140.0.22:8016 (backend-01 group1 10.140.0.23:80 (fd
storage	
db	redis dir /var/lib/redis monogo: dbPath: /var/lib/mongo mysql : datadir=/var/lib/mysql  /data gcp gce= /root 20g disk = /data 100g
others	fdfs: store_path0=/fastdfs/storage

---

firewalld stop  
selinux stop

---

note  
**need modify**  
**notice**  
**different**

tw esxi  
centos 7 template=10.1.1.150  
lab=10.1.1.151

###

jumpserver

```
##  
log rotate??  
###  
  
###  
qat_h5  
>  
  
qat_admin  
> proxy_pass  
  
qat_backend (all in one but job  
> java (gateway/ user / admin/auth...  
  
qat_backendjob (loading...  
> java job (api to others  
  
###  
oracle jdk to openjdk 8  
oracle jdk to openjdk 11  
  
###  
timezone to asia/taipei  
  
###  
os  
centos 7.8= 3.10.0-1127.8.2.el7  
centos 8.1= 4.18.0-147.el8.x86_64  
  
###  
redis ( redis-4.0.2.tar.gz  
> tar to yum ( centos 7 yum=3.2.12 , centos 8=5.0.3-2  
> tar to cloud (3.2, 4.0, 5.0  
  
mongo (mongodb-org-4.0.repo  
> current version is 4.2yum  
  
mysql(mysql-5.6.46  
> current mysql80  
> mariadb (centos 8 =10.3.17 , centos 7 =5.5.65
```

kafka (kafka\_2.12 , ? https://kafka.apache.org/downloads , no 2.12 in offical website  
> current version 2.5

fastdfs (fastdfs-5.11.tar.gz  
> current version 6.06  
> move to gcp cloud filestore (<https://console.cloud.google.com/filestore/instances>)

storm  
> docker 17.12.1 ( current version 19.03  
> storm (<https://github.com/ziyunhx/storm-mono-docker> 1.2.2 (current version is 2.2.0  
>> service= nimbus, supervisord, zookeeper

nacos (nacos-server-1.1.4.tar.gz  
> last stable 1.3.0 (<https://github.com/alibaba/nacos/releases>  
> 1.1.4 (<https://github.com/alibaba/nacos/releases/tag/1.1.4>)

sentinel (sentinel-dashboard-1.7.0.jar  
> current version 1.7.2 (<https://github.com/alibaba/Sentinel>)

---

## ## redis:6379

### remi

```
yum -y install http://rpms.remirepo.net/enterprise/remi-release-7.rpm
yum --enablerepo=remi install redis #default is redis6### ius
https://ius.io/setup
yum install https://repo.ius.io/ius-release-el7.rpm -y
```

yum install -y redis5

```
dir /var/lib/redis link to /data/redis
drwxr-x---. redis redis system_u:object_r:redis_var_lib_t:s0 redis
```

```
mkdir /data/redis
chown redis:redis /data/redis -R
```

base warning message on log : /var/log/redis.log  
WARNING: The TCP backlog setting of 511 cannot be enforced because  
/proc/sys/net/core/somaxconn is set to the lower value of 128.  
WARNING overcommit\_memory is set to 0! Background save may fail under low memory  
condition. To fix this issue add 'vm.overcommit\_memory = 1'  
WARNING you have Transparent Huge Pages (THP) support enabled in your kernel. This  
will create latency and memory usage issues with Redis.

solution> <https://www.twblogs.net/a/5d6d556bb9eee541c33b102>

```
vim /etc/sysctl.d/99-sysctl.conf (centos 7
net.core.somaxconn=1024
(bigger than 511 will work
vm.overcommit_memory=1
rc.local
echo never > /sys/kernel/mm/transparent_hugepage/enabled
#default is always
```

```
#thp solve, ref= https://docs.mongodb.com/manual/tutorial/transparent-huge-pages/
cat > /etc/systemd/system/disable-transparent-huge-pages.service
```

```
[Unit]
Description=Disable Transparent Huge Pages (THP)
DefaultDependencies=no
After=sysinit.target local-fs.target
Before=mongod.service

[Service]
Type=oneshot
ExecStart=/bin/sh -c 'echo never | tee /sys/kernel/mm/transparent_hugepage/enabled > /dev/null'

[Install]
WantedBy=basic.target
```

```
systemctl daemon-reload
systemctl enable --now disable-transparent-huge-pages
```

```
systemctl enable --now redis #after modify config
```

mongo sample <https://docs.mongodb.com/manual/tutorial/transparent-huge-pages/>  
make it as systemd unit file

```
maxclients??
maxmemory??
vi /etc/redis.conf
```

redis6 cat redis.conf   grep -v '^#'   grep -v '^\$'	redis 5	redis4
bind 127.0.0.1 protected-mode yes port 6379 tcp-backlog 511 timeout 0	#bind 127.0.0.1 >>將原本檔案中的註解掉 #dir /var/lib/redis >>將原本檔案中的註解掉 bind 0.0.0.0	protected-mode yes port 6379 tcp-backlog 511 timeout 0

<pre> tcp-keepalive 300 daemonize no supervised no pidfile /var/run/redis_6379.pid loglevel notice logfile /var/log/redis/redis.log databases 16 <b>always-show-logo yes</b> save 900 1 save 300 10 save 60 10000 stop-writes-on-bgsave-error yes rdbcompression yes rdbchecksum yes dbfilename dump.rdb <b>rdb-del-sync-files no</b> <b>dir /var/lib/redis</b> <b>replica-serve-stale-data yes</b> <b>replica-read-only yes</b> <b>repl-diskless-sync no</b> <b>repl-diskless-sync-delay 5</b> <b>repl-diskless-load disabled</b> <b>repl-disable-tcp-nodelay no</b> replica-priority 100 aclog-max-len 128 lazyfree-lazy-eviction no lazyfree-lazy-expire no lazyfree-lazy-server-del no <b>replica-lazy-flush no</b> <b>lazyfree-lazy-user-del no</b> appendonly no appendfilename "appendonly.aof" appendfsync everysec no-appendfsync-on-rewrite no auto-aof-rewrite-percentage 100 auto-aof-rewrite-min-size 64mb aof-load-truncated yes <b>aof-use-rdb-preamble yes</b> lua-time-limit 5000 slowlog-log-slower-than 10000 slowlog-max-len 128 latency-monitor-threshold 0 notify-keyspace-events "" hash-max-ziplist-entries 512 hash-max-ziplist-value 64 list-max-ziplist-size -2 list-compress-depth 0 set-max-intset-entries 512 zset-max-ziplist-entries 128 zset-max-ziplist-value 64 hll-sparse-max-bytes 3000 <b>stream-node-max-bytes 4096</b> <b>stream-node-max-entries 100</b> activerehashing yes client-output-buffer-limit normal 0 0 0 client-output-buffer-limit replica 256mb 64mb 60 client-output-buffer-limit pubsub 32mb 8mb 60 hz 10 <b>dynamic-hz yes</b> aof-rewrite-incremental-fsync yes <b>rdb-save-incremental-fsync yes</b> jemalloc-bg-thread yes </pre>	<pre> <b>dir /data/redis</b> <b>requirepass r07vDcruTTuNQ4J2vP92</b>  protected-mode yes port 6379 tcp-backlog 511 timeout 0 tcp-keepalive 300 daemonize no supervised no pidfile /var/run/redis_6379.pid loglevel notice logfile /var/log/redis/redis.log databases 16 always-show-logo yes save 900 1 save 300 10 save 60 10000 stop-writes-on-bgsave-error yes rdbcompression yes rdbchecksum yes dbfilename dump.rdb <b>dir /var/lib/redis</b> <b>slave-serve-stale-data yes</b> <b>slave-read-only yes</b> <b>repl-diskless-sync no</b> <b>repl-diskless-sync-delay 5</b> <b>repl-disable-tcp-nodelay no</b> <b>slave-priority 100</b> <b>requirepass r07vDcruTTuNQ4J2vP92</b> </pre>	<pre> tcp-keepalive 300 daemonize no supervised no pidfile /var/run/redis_6379.pid loglevel notice logfile /var/log/redis/redis.log databases 16 save 900 1 save 300 10 save 60 10000 stop-writes-on-bgsave-error yes rdbcompression yes rdbchecksum yes dbfilename dump.rdb <b>dir /var/lib/redis</b> <b>slave-serve-stale-data yes</b> <b>slave-read-only yes</b> <b>repl-diskless-sync no</b> <b>repl-diskless-sync-delay 5</b> <b>repl-disable-tcp-nodelay no</b> <b>slave-priority 100</b> <b>requirepass r07vDcruTTuNQ4J2vP92</b> </pre>
--	--	--

systemctl enable --now redis

[root@lab-bby-151 etc]# redis-cli -a r07vDcruTTuNQ4J2vP92 ping

Warning: Using a password with '-a' or '-u' option on the command line interface may not be safe.

PONG

```
[root@bf-qat ~]# redis-cli  
127.0.0.1:6379> auth r07vDcruTTuNQ4J2vP92  
OK  
127.0.0.1:6379> ping  
PONG
```

## ## mongodb:27017

```
cat > /etc/yum.repos.d/mongodb-org-4.2.repo
```

```
[mongodb-org-4.2]  
name=MongoDB Repository  
baseurl=https://repo.mongodb.org/yum/redhat/$releasever/mongodb-org/4.2/x86_64/  
gpgcheck=1  
enabled=1  
gpgkey=https://www.mongodb.org/static/pgp/server-4.2.asc
```

```
mkdir /data/mongo
```

```
yum install -y mongodb-org
```

```
vim /etc/mongod.conf
```

```
path: /var/log/mongodb/mongod.log  
dbPath: /var/lib/mongo  
dbPath: /data/mongo  
port: 27017  
bindIp: 127.0.0.1 #uat/prod to 0.0.0.0  
  
#disable ftdc  
setParameter:  
diagnosticDataCollectionEnabled: false
```

<https://docs.mongodb.com/manual/tutorial/install-mongodb-enterprise-on-red-hat/>

```
yum install checkpolicy -y  
cat > mongodb_cgroup_memory.te <<EOF
```

```
module mongodb_cgroup_memory 1.0;  
  
require {
```

```

type cgroup_t;
type mongod_t;
class dir search;
class file { getattr open read };
}

===== mongod_t =====
allow mongod_t cgroup_t:dir search;
allow mongod_t cgroup_t:file { getattr open read };
EOF

###

checkmodule -M -m -o mongodb_cgroup_memory.mod
mongodb_cgroup_memory.te
semodule_package -o mongodb_cgroup_memory.pp -m
mongodb_cgroup_memory.mod
sudo semodule -i mongodb_cgroup_memory.pp

```

chown mongod:mongod /data/mongo/ -R

systemctl enable --now mongod

createMongodbUser.sh

```

mongo --eval "db.getSiblingDB('admin').createUser({user:'root',pwd:'XpXuXCi1qx4MRAh35yNL',roles:['root']});"
mongo --eval "db.getSiblingDB('admin').auth('root','XpXuXCi1qx4MRAh35yNL')"

mongo --eval "db.getSiblingDB('admin').createUser({user:'gm_root',pwd:'xXiE94ktmi418vF9njrB',roles:
[{"role:'readWrite',db:'gm'}]});"
mongo --eval "db.getSiblingDB('admin').auth('gm_root','xXiE94ktmi418vF9njrB');"

```

disable ftdc

<https://docs.mongodb.com/manual/administration/analyzing-mongodb-performance/>

selinux

```

> type=AVC msg=audit(1593680241.000:843): avc: denied { open } for pid=20929
comm="ftdc" path="/proc/20929/net/netstat"

```

## ## mysql:3306

```
mysql -u root -p  
Pw : 6gyQmfv4ithRZuQLr1wH@?
```

<https://dev.mysql.com/doc/mysql-repo-excerpt/5.6/en/linux-installation-yum-repo.html>

/etc/yum.repos.d/mysql56.repo

```
# Enable to use MySQL 5.6  
[mysql56-community]  
name=MySQL 5.6 Community Server  
baseurl=http://repo.mysql.com/yum/mysql-5.6-community/el/6/$basearch/  
enabled=1  
gpgcheck=0  
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-mysql
```

yum install -y <https://dev.mysql.com/get/mysql80-community-release-el7-3.noarch.rpm>

mysql80.repo

```
[mysql80-community]  
name=MySQL 8.0 Community Server  
baseurl=http://repo.mysql.com/yum/mysql-8.0-community/el/7/$basearch/  
enabled=1  
gpgcheck=1  
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-mysql
```

gpgkey = <https://dev.mysql.com/doc/refman/8.0/en/checking-gpg-signature.html>

yum repolist enabled | grep "mysql.\*-community.\*"

yum install -y mysql-community-server.x86\_64

```
mkdir /data/mysql  
chown mysql:mysql /data/mysql -R
```

vi /etc/my.cnf

```
datadir=/data/mysql  
#datadir=/var/lib/mysql  
  
#slow query  
long_query_time = 2  
slow_query_log = 1
```

```
#ray> try to find who connection error
log_error_verbosity=3
skip-name-resolve=on

mysqlx=0 #disable x plug
```

systemctl enable --now mysqld

```
cat /var/log/mysqld.log | grep password
> show "A temporary password is generated for root@localhost:"
>> ALTER USER 'root'@'localhost' IDENTIFIED BY '6gyQmfv4ithRZuQLr1wH@?';
>>> mysql -u root -p --connect-expired-password -e " ALTER USER 'root'@'localhost'
IDENTIFIED BY '6gyQmfv4ithRZuQLr1wH@?';"
```

```
mysql -u root -e "SET PASSWORD FOR root@'localhost' =
PASSWORD('6gyQmfv4ithRZuQLr1wH@?');"
> ERROR 1819 (HY000): Your password does not satisfy the current policy requirements
>> UNINSTALL COMPONENT 'file://component_validate_password';
(https://dev.mysql.com/doc/refman/8.0/en/validate-password-installation.html
```

```
allow mysql user = root , connect from anyhost
mysql>
use mysql;
select user,host from mysql.user;
update user set host='%' where user="root" and host="localhost";
GRANT ALL PRIVILEGES ON *.* TO 'root'@'%'
```

```
mys
GRANT ALL PRIVILEGES ON *.* TO      'root'@'%'      IDENTIFIED BY
'6gyQmfv4ithRZuQLr1wH@?';
GRANT ALL PRIVILEGES ON mydb.* TO 'myuser'@'%' IDENTIFIED
CREATE USER 'root'@'%' IDENTIFIED BY '6gyQmfv4ithRZuQLr1wH@?';
GRANT ALL PRIVILEGES ON *.* TO 'root'@'%';
```

my.cnf

ori 5.6	ori 8	youda
[mysqld] <b>datadir=/var/lib/mysql</b> socket=/var/lib/mysql/mysql.sock symbolic-links=0 sql_mode=NO_ENGINE_SUBSTITUTION,STRICT_TRANS_TABLES	[mysqld] datadir=/var/lib/mysql <b>datadir=/data/mysql</b> socket=/var/lib/mysql/mysql.sock log-error=/var/log/mysqld.log pid-file=/var/run/mysqld/mysqld.pid	[client] port = 3306 socket = /data/mysql/mysql.sock default-character-set=utf8  [mysqld] port = 3306

[mysqld_safe] log-error=/var/log/mysqld.log pid-file=/var/run/mysqld/mysqld.pid	??mysqlx 33060 port	basedir = /usr/local/mysql character-set-server=utf8 collation-server=utf8_general_ci datadir = /data/mysql socket = /data/mysql/mysql.sock tmpdir = /tmp user = mysql key_buffer_size = 64M max_allowed_packet = 10G table_open_cache = 256 sort_buffer_size = 1M net_buffer_length = 8K read_buffer_size = 1M read_rnd_buffer_size = 512K myisam_sort_buffer_size = 16M thread_cache_size = 32 query_cache_size = 32M tmp_table_size = 64M skip-grant-tables symbolic-links=0  [mysqld_safe] log-error = /data/mysql/error.log pid-file = /data/mysql/mysql.pid  [mysql] no-auto-rehash default-character-set=utf8  [mysqldump] quick max_allowed_packet = 16M !includedir /etc/my.cnf.d
---	---------------------	--

## init schema

```
create database gm character set utf8 collate utf8_general_ci;
create database GM default character set utf8mb4 collate utf8mb4_unicode_ci;
```

```
mysql -u root -p GM < gm.sql
```

```
error> java.sql.SQLException: null, message from server: "Host '10.1.1.152' is blocked because of many connection errors; unblock with 'mysqladmin flush-hosts'"
```

```
my.cnf
> log_error_verbosity=3
```

```
mysql
show variables like "log_error_verbosity"; (default is 2
SET GLOBAL log_error_verbosity=3;
```

```
mysql -u root -p (20200810, bfdev; 20200811, bfqat
```

```
> SET GLOBAL max_connect_errors=10000;
> show variables like "max_connect_errors";
```

## kafka:9092 , zookeeper:2181

yum install java -y #openjdk 1.8.0 >>安裝  
<https://kafka.apache.org/quickstart>

```
cd /opt
curl -O http://apache.stu.edu.tw/kafka/2.5.0/kafka_2.12-2.5.0.tgz
tar xvf kafka_2.12-2.5.0.tgz
ln -s kafka_2.12-2.5.0 kafka
```

>>至此目錄底下  
>> 下載此http協定檔案  
>>壓縮下載後此檔案  
>>建立連結

```
#set logdir (origin=/tmp
mkdir -p /data/kafka/kafka-logs /data/kafka/zookeeper
```

```
server.properties:log.dirs=/data/kafka/kafka-logs
zookeeper.properties:dataDir=/data/kafka/zookeeper
```

```
#disable kafka bind on hostname (ONLY FOR DEV
vim config/server.properties      >>編輯此檔案
listeners=PLAINTEXT://ip:9092 (assign ip here 0.0.0.0 will receive cannot use the
nonroutable meta-address 0.0.0.0
```

```
#disable random high port
vim /opt/kafka/bin/kafka-run-class.sh
KAFKA_JMX_OPTS="-Dcom.sun.management.jmxremote.enable=false
```

OpenJDK 64-Bit Server VM warning: If the number of processors is expected to increase from one, then you should configure the number of parallel GC threads appropriately using -XX:ParallelGCThreads=N

ori	daemon bin/zookeeper-server-start.sh -daemon config/zookeeper.properties
cd /opt/kafka bin/zookeeper-server-start.sh config/zookeeper.properties & bin/kafka-server-start.sh config/server.properties &	cd /opt/kafka bin/kafka-server-start.sh -daemon config/server.properties  #change to daemon

```
# kafka rc.local, change to daemon
cd /opt/kafka
bin/zookeeper-server-start.sh -daemon config/zookeeper.properties
sleep 10
bin/kafka-server-start.sh -daemon config/server.properties
```

#### #### kafka service

<https://gist.github.com/dyoung522/6f7aab567f70f67030ae4ee0191933c0>

```
cat > /usr/lib/systemd/system/kafka-zookeeper.service
```

```
[Unit]
Description=Apache Zookeeper server (Kafka)
Documentation=http://zookeeper.apache.org
Requires=network.target remote-fs.target
After=network.target remote-fs.target

[Service]
Type=simple
User=root
Group=root
Environment=JAVA_HOME=/usr/lib/jvm/jre
ExecStart=/opt/kafka/bin/zookeeper-server-start.sh
/opt/kafka/config/zookeeper.properties
ExecStop=/opt/kafka/bin/zookeeper-server-stop.sh

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

```
cat > /usr/lib/systemd/system/kafka.service
```

```
[Unit]
Description=Apache Kafka server (broker)
```

```

Documentation=http://kafka.apache.org/documentation.html
Requires=network.target remote-fs.target
After=network.target remote-fs.target kafka-zookeeper.service

[Service]
Type=simple
User=root
Group=root
Environment=JAVA_HOME=/usr/lib/jvm/jre
ExecStart=/opt/kafka/bin/kafka-server-start.sh
/opt/kafka/config/server.properties
ExecStop=/opt/kafka/bin/kafka-server-stop.sh

[Install]
WantedBy=multi-user.target

```

```

systemctl daemon-reload
systemctl enable --now kafka-zookeeper kafka

```

log rotate  
<https://dzone.com/articles/kafka-setup>

### topic check (要到 cd /opt/kafka 目錄底下)  
cd /opt/kafka  
bin/kafka-topics.sh --list --zookeeper localhost:2181

**## nacos:8848 to 8886;**  
need java  
cd /opt  
curl -OL <https://github.com/alibaba/nacos/releases/download/1.1.4/nacos-server-1.1.4.tar.gz>  
tar xvf nacos-server-1.1.4.tar.gz

youda	new
JAVA_HOME=/usr/local/java/jdk1.8.0_231	/opt/nacos/bin/startup.sh add

	JAVA_HOME=/usr/lib/jvm/jre
--	----------------------------

vim /opt/nacos/conf/application.properties

youda (use this)	ori
server.contextPath=/nacos server.servlet.contextPath=/nacos <b>server.port=8886</b> management.metrics.export.elastic.enabled=false management.metrics.export.influx.enabled=false <b>server.tomcat.accesslog.enabled=true</b> server.tomcat.accesslog.pattern=%h %l %u %t "%r" %s %b %D %{User-Agent}i server.tomcat.basedir= nacos.security.ignore.urls=/,/*/*.css,//*/*.js,//*/*.html,//*/*.ma p,//*/*.svg,//*/*.png,//*/*.ico,/console-fe/public/*,/v1/auth/logi n,/v1/console/health/**,/v1/cs/**,/v1/ns/**,/v1/cmdb/**,/actuat or/**,/v1/console/server/** nacos.istio.mcp.server.enabled=false <b>server.tomcat.accesslog.enabled=false</b>	server.contextPath=/nacos server.servlet.contextPath=/nacos <b>server.port=8848</b> management.metrics.export.elastic.enabled=false management.metrics.export.influx.enabled=false server.tomcat.accesslog.enabled=true server.tomcat.accesslog.pattern=%h %l %u %t "%r" %s %b %D %{User-Agent}i server.tomcat.basedir= nacos.security.ignore.urls=/,/*/*.css,//*/*.js,//*/*.html,//*/*.ma p,//*/*.svg,//*/*.png,//*/*.ico,/console-fe/public/*,/v1/auth/logi n,/v1/console/health/**,/v1/cs/**,/v1/ns/**,/v1/cmdb/**,/actuat or/**,/v1/console/server/** nacos.istio.mcp.server.enabled=false

sed -i 's/accesslog.enabled=true/accesslog.enabled=false/g' application.properties

### nacos.systemd

```
cat > /usr/lib/systemd/system/nacos.service

[Unit]
Description=Nacos
Documentation=https://nacos.io
Requires=network.target remote-fs.target
After=network.target remote-fs.target

[Service]
Type=simple
User=root
Group=root
Environment=JAVA_HOME=/usr/lib/jvm/jre
ExecStart=/opt/nacos/bin/startup.sh -m standalone
ExecStop=/opt/nacos/bin/shutdown.sh

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

```
#rc.local
cd /opt/nacos/bin
./startup.sh -m standalone
```

<http://ip:port/nacos>

nacos / nacos

## ## docker

```
https://docs.docker.com/engine/install/centos/  
yum install -y yum-utils  
yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo  
yum install docker-ce docker-ce-cli containerd.io -y
```

change docker root folder (default is /var/lib/docker)  
<https://linuxconfig.org/how-to-move-docker-s-default-var-lib-docker-to-another-directory-on-ubuntu-debian-linux>

```
mkdir /data/docker
```

```
vi /lib/systemd/system/docker.service  
FROM:  
ExecStart=/usr/bin/docker daemon -H fd://  
TO:  
ExecStart=/usr/bin/docker daemon -g /data/docker -H fd://
```

```
systemctl enable --now docker  
docker info | grep -i root  
systemctl start docker
```

```
#for storm  
yum install docker-compose -y
```

## ## storm:2182, 3373,3372,6627, 8000,8001,8080

storm也許可以獨立出來做一套

```
yum install -y unzip  
cd /opt  
curl -L https://github.com/ziyunhx/storm-mono-docker/archive/master.zip -o  
storm-mono-docker.zip
```

```
unzip storm-mono-docker.zip  
(ubuntu 14.04, java 1.7)  
cd storm-mono-docker-master
```

```
cd /opt/storm-mono-docker-master/base  
put jdk-8u231-linux-x64.tar.gz  
modify Dockerfile
```

docker file note:

[https://docs.google.com/document/d/10hNeXSIVTfkZMc9eOVIPz4n78dYFE\\_OXNGCpQNid7NE/edit](https://docs.google.com/document/d/10hNeXSIVTfkZMc9eOVIPz4n78dYFE_OXNGCpQNid7NE/edit)

```
#base  
FROM ubuntu:trusty  
  
#RUN apt-get update; apt-get install -y unzip openjdk-7-jre-headless wget supervisor docker.io openssh-server  
#RUN apt-get update;apt-get install wget supervisor docker.io openssh-server --assume-yes  
  
ADD jdk-8u231-linux-x64.tar.gz /usr/local/  
RUN ln -s /usr/local/jdk1.8.0_231 /usr/local/jdk \  
    && apt-get update \  
    && apt-get install wget supervisor docker.io openssh-server --assume-yes \  
    && cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime \  
    && echo "Asia/Shanghai" > /etc/timezone \  
    && rm -rf /var/lib/apt/lists/*  
  
#RUN cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime && echo "/Shanghai" > /etc/timezone  
  
ENV JAVA_HOME="/usr/local/jdk"  
ENV JRE_HOME="$JAVA_HOME/jre" \  
CLASSPATH=".:$JAVA_HOME/lib/tools.jar:$JAVA_HOME/lib/dt.jar" \  
PATH="$JAVA_HOME/bin:$JRE_HOME/bin:$PATH"  
  
#ENV JRE_HOME $JAVA_HOME/jre  
#ENV CLASSPATH .:$JAVA_HOME/lib/tools.jar:$JAVA_HOME/lib/dt.jar  
#ENV PATH $JAVA_HOME/bin:$JRE_HOME/bin:$PATH  
  
#ENV JAVA_HOME /usr/lib/jvm/java-7-openjdk-amd64/  
  
??/etc/timezone改 Asia/Shanghai  
  
#RUN echo 'root:ziyunhxpass' | chpasswd  
#RUN mkdir /var/run/sshd  
#RUN sed -i 's/PermitRootLogin without-password/PermitRootLogin yes/' /etc/ssh/sshd_config  
#EXPOSE 22  
  
#base  
cd /opt/storm-mono-docker-master/base  
cp Dockerfile Dockerfile_ori  
(then modify  
docker build -t localstorm/base .  
  
#zookeeper  
cd /opt/storm-mono-docker-master/zookeeper  
cp Dockerfile Dockerfile_ori  
FROM localstorm/base  
wget change https://archive.apache.org/dist/zookeeper/zookeeper-3.4.9/zookeeper-3.4.9.tar.gz  
docker build -t localstorm/zookeeper .
```

```

#storm

cd /opt/storm-mono-docker-master/storm
cp Dockerfile Dockerfile_ori
change FROM localstorm/base
wget change https://archive.apache.org/dist/storm/apache-storm-1.2.2/apache-storm-1.2.2.tar.gz

docker build -t localstorm/storm .

#nimbus
cd /opt/storm-mono-docker-master/storm-nimbus
cp Dockerfile Dockerfile_ori
FROM localstorm/storm

docker build -t localstorm/storm-nimbus .

# supervisor:
cd /opt/storm-mono-docker-master/storm-supervisor
cp Dockerfile Dockerfile_ori
FROM localstorm/storm

docker build -t localstorm/storm-supervisor .

# ui:
cd /opt/storm-mono-docker-master/storm-ui
cp Dockerfile Dockerfile_ori
FROM localstorm/storm

docker build -t localstorm/storm-ui .

```

net.ipv4.ip\_local\_port\_range = 32768 60999

function	ori	change docker-compose.yml
ziyunhx/zookeeper	ports: - "49181:2181" - "22"	#change to localimage image: localstorm/zookeeper image: localstorm/storm-nimbus image: localstorm/storm-supervisor image: localstorm/storm-supervisor image: localstorm/storm-ui  container_name: storm_zookeeper #conflict with kafka zookeeper ports: # - "2182:2181" (這沒調用，可以拿掉試試 #- "22"
ziyunhx/storm-nimbus:1.2.2	ports: - "49773:3773" - "49772:3772" - "49627:6627" - "22"	container_name: storm_nimbus ports: - "3773:3773" - "3772:3772" - "6627:6627"

		#- "22"
ziyunhx/storm-supervisor:1.2.2	ports: - "8000" - "22"	container_name: storm_supervisor1 container_name: storm_supervisor2 ports: - "8000" #- "22"
ziyunhx/storm-ui:1.2.2	ports: - "49080" 8080" - "22"	container_name: storm_ui ports: - "8080:8080" #- "22" #disable ssh port"

docker-compose.yml

```

zookeeper:
  image: localstorm/zookeeper
  container_name: storm_zookeeper
  #ports:
  #- "2182:2181"
  #- "22"
nimbus:
  image: localstorm/storm-nimbus
  container_name: storm_nimbus
  ports:
    - "3773:3773"
    - "3772:3772"
    - "6627:6627"
  #- "22"
  links:
    - zookeeper:zk
supervisor:
  image: localstorm/storm-supervisor
  container_name: storm_supervisor1
  # ports:
  # - "8000:8000"
  #- "22"
  links:
    - nimbus:nimbus
    - zookeeper:zk
supervisor1:
  image: localstorm/storm-supervisor
  container_name: storm_supervisor2
  ports:
    - "8001:8000"
  #- "22"
  links:
    - nimbus:nimbus
    - zookeeper:zk
ui:
  image: localstorm/storm-ui
  container_name: storm_ui

```

```
ports:  
  - "8080:8080"  
  #- "22"  
links:  
  - nimbus:nimbus  
  - zookeeper:zk
```

```
#rc.local  
cd /opt/storm-mono-docker-master  
docker-compose up -d  
docker-compose scale supervisor=2 #docker file add second supervisor
```

### nimbus need java 8

```
apt-get install software-properties-common  
add-apt-repository ppa:openjdk-r/ppa  
apt-get update  
  
apt-get install openjdk-8-jdk (apt-get install openjdk-8-jre  
update-java-alternatives --list  
update-alternatives --config java
```

```
????  
youda ansible  
/root/configcollection/storm/storm-mono-docker/storm-nimbus
```

```
RUN /usr/bin/config-supervisord.sh nimbus  
>> supervisord create nimbus
```

## fastdfs: 8888, 22122, 23000

<https://github.com/MrSorrow/docker-fastdfs>  
?? fastdfs 5.11 <https://hub.docker.com/r/moocu/fastdfs> with more stars ?? no nginx

```
store_path0=/fastdfs/storage
```

```
mkdir -p /data/fastdfs/storage  
mkdir -p /data/fastdfs/tracker  
mkdir -p /data/fastdfs/client
```

<pre>docker run -itd --name taotao-fastdfs \ -p 80:80 -p 8888:8888 -p 22122:22122 \ -p 23000:23000 coffeecoder/fastdfs:1.0 \ /bin/bash</pre>	<pre>docker run -itd --name fastdfs \ -p 8880:80 -p 8888:8888 -p 22122:22122 \ -p 23000:23000 -v /data/fastdfs:/fastdfs \ coffeecoder/fastdfs:1.0 /bin/bash</pre>
	<pre>docker run --net=host -itd --name fastdfs \ -v /data/fastdfs:/fastdfs \ coffeecoder/fastdfs:1.0 /bin/bash</pre>
	<pre>docker run -itd --net=host --name fastdfs \ -v /data/fastdfs:/fastdfs \ coffeecoder/fastdfs:1.0 /bin/bash  # change instance storage.conf  docker run -itd --name ngxfastdfs \ -p 8880:80 -p 8888:8888 \ coffeecoder/fastdfs:1.0 /bin/bash #only nginx</pre>

```
docker exec -it fastdfs /bin/bash
cd /etc/fdfs
cp storage.conf storage.conf_ori
> modify tracker's ip (cannot be 127.0.01)
tracker_server=192.168.2.107:22122
> sed -i 's/192.168.2.107/10.140.0.2/g' storage.conf
```

```
cp client.conf client.conf_ori
> sed -i 's/192.168.2.107/10.140.0.2/g' client.conf
```

```
#rc.local
docker start fastdfs
docker exec -i fastdfs /bin/bash /usr/local/src/start.sh
```

```
docker exec -it taotao-fastdfs /bin/bash
./start.sh (/usr/local/src/start.sh)
```

```
docker ex h
> it works
>> ngx_http_fastdfs_set pid=45
```

?docker exec -it taotao-fastdfs /usr/local/src/start.sh

```
> OCI runtime exec failed: exec failed: container_linux.go:349: starting container process caused "exec format error": unknown
```

```
## fastdfs
https://hub.docker.com/r/qbanxiaoli/fastdfs
cd /opt
git clone https://github.com/qbanxiaoli/fastdfs.git
cd fastdfs
cp docker-compose.yml win-docker-compose.yml
mv docker-compose-linux.yml docker-compose.yml
vi docker-compose.yml (modify 80 to 8880, vol mapping /data/fdfs:/var/local)
docker-compose up -d
```

## ## sentinel: 9019, 8719

```
8019 ? 8719
need java
```

```
cd /opt
mkdir sentinel-dashboard
cd sentinel-dashboard
curl -OL
https://github.com/alibaba/Sentinel/releases/download/1.7.2/sentinel-dashboard-1.7.2.jar
```

```
cat > start.sh
```

<pre>#!/bin/bash nohup java -Dserver.port=8019 \ -Dcsp.sentinel.dashboard.server=localhost:8019 \ -Dproject.name=sentinel-dashboard \ -jar sentinel-dashboard-1.7.2.jar &amp;&gt;sentinel-dashboard.out &amp;</pre>	<pre>#!/bin/bash nohup java -Dserver.port=9019 \ -Dcsp.sentinel.dashboard.server=localhost:9019 \ -Dproject.name=sentinel-dashboard \ -Dcsp.sentinel.log.dir=/opt/sentinel-dashboard/logs/csp \ -jar sentinel-dashboard-1.7.2.jar &amp;&gt;/dev/null &amp;</pre>
---	--

log配置說明

<https://github.com/alibaba/Sentinel/wiki/%E5%90%AF%E5%8A%A8%E9%85%8D%E7%BD%AE%E9%A1%B9>

```
chmod u+x start.sh
```

```
#rc.local
cd /opt/sentinel-dashboard
./start.sh
```

### ### sentinel.systemd

```
cat > /usr/lib/systemd/system/sentinel.service

[Unit]
Description=Sentinel Dashboard
Documentation=https://github.com/alibaba/Sentinel
Requires=network.target remote-fs.target
After=network.target remote-fs.target

[Service]
Type=simple
User=root
Group=root
Environment=JAVA_HOME=/usr/lib/jvm/jre
ExecStart=/opt/sentinel-dashboard/start.sh
ExecStop=???

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

[root@lab-bf-151 sentinel-dashboard]# ss -tlnp | grep 7424

```
LISTEN 0 100 [::]:8719 [::]:* users:(("java",pid=7424,fd=11))
#"sentinel.api.port" LISTEN 0 100 [::]:0019 [::]:* users:(("java",pid=7424,fd=42)) #service port
```

### ## rc.local

```
systemctl enable rc-local
```

```
http://ip:port (http://10.1.1.152:9019)
default username/passwd
sentinel/sentinel
```

### ## nginx:80

```
yum install nginx (centos 7 = 1.16
```

```
/etc/nginx
```

```
Ls
```

```
[root@bf-lab-153 etc]# cd nginx/
[root@bf-lab-153 nginx]# ls
conf.d      fastcgi.conf.default    koi-utf      mime.types.default  scgi_params          uwsgi_params.default
default.d    fastcgi_params         koi-win      nginx.conf        scgi_params.default  win-utf
fastcgi.conf fastcgi_params.default mime.types  nginx.conf.default uwsgi_params
[root@bf-lab-153 nginx]# pwd
/etc/nginx
[root@bf-lab-153 nginx]# pwd
```

URL = bfh5.sc.local = 10.1.1.152 root=/data/html/h5

URL = bfpc.sc.local = 10.1.1.152 root=/data/html/pc

### Nginx.conf

disable default 80

add

cd /etc/nginx/conf.d

add config files

- bfaadmin.conf
- bfwww.conf (default\_server with automatic pc/h5 root switch)
- bfh5.conf
- bfpc.conf

proxy\_pass 用內網IP (127.0.0.1 在module互調中會有問題, 取消upstream名也會有問題)

#### website 調用

```
location /filemanager/ { proxy_pass http://filemanager/; }
location /group1/ { proxy_pass http://fastdfs/group1/; }
location /gateway/ { proxy_pass http://gateway/; }
location /gatewayh5/ { proxy_pass http://gatewayh5/; }
```

#### upstream.conf

```
upstream gateway {server 10.1.1.152:8016;}
upstream gatewayh5 {server 10.1.1.152:8018;}
upstream filemanager{server 10.1.1.152:8019;}
upstream fastdfs {server 10.1.1.152:8888;}
```

## ## selinux

```
nginx to upstream need  
setsebool -P httpd_can_network_connect 1
```

```
ls -alZ /data
```

```
drwxr-xr-x. mongod mongod system_u:object_r:default_t:s0 mongo  
drwxr-x--x. mysql mysql system_u:object_r:default_t:s0 mysql  
drwxr-x---. redis redis system_u:object_r:default_t:s0 redis  
drwxr-xr-x. root root unconfined_u:object_r:unlabeled_t:s0 html
```

```
cd /data
```

```
mkdir mysql  
mkdir redis  
mkdir mongo  
mkdir html
```

```
chown mongod:mongod mongo  
chown redis:redis redis  
chown mysql:mysql mysql
```

```
chcon -R -t mysqld_db_t mysql  
chcon -R -t redis_var_lib_t redis  
chcon -R -t mongod_var_lib_t mongo  
chown nginx:nginx /data/html -R
```

```
chcon -R -t httpd_sys_content_t /data/html  
chcon -R -t httpd_sys_content_t /data/gitlab
```

```
semanage fcontext -a -t mysqld_db_t "/data/mysql(-files|-keyring)?(/.*)?"  
restorecon -R /data/mysql/  
semanage fcontext -a -t mysqld_var_run_t "/data/mysql/mysql.sock"
```

```
semanage fcontext -a -t mongod_var_lib_t "/data/mongo.*"  
restorecon -R /data/mongo/
```

```
semanage fcontext -a -t redis_var_lib_t "/data/redis(/.*)?"  
restorecon -R /data/redis
```

## ## logclearn

```
bfdev 10.1.1.152  
cd /opt/sre/
```

```
[root@bf-dev-152 sre]# cat clear-old-log.sh
```

```
#!/bin/bash  
# ray  
# 2020-7-23 init version: delete old logs in assign folders  
  
array_log_dir=("/opt/kafka/logs"  
               "/opt/nacos/bin/logs"  
               "/root/logs"  
               "/root/nacos"  
               "/opt/nacos/logs/")  
  
var_keep_days=14  
var_log_pattern="*.log.*"  
  
for var_log_dir in "${array_log_dir[@]}"; do  
    find ${var_log_dir} -type f -mtime +${var_keep_days} -name ${var_log_pattern} | xargs -r ls -al  
done  
  
#bigsize check  
#find / -type f -size +100M  
# kafka log  
# /tmp/kafka-logs  
# /tmp/zookeeper/  
  
  
# docker log clean  
##fastdfs  
# dfs log (already volume mapping to host) /data/fastdfs/storage/logs  
# dfs log (already volume mapping to host) /data/fastdfs/tracker/logs  
# ngx log /usr/local/nginx/logs  
#storm , check supervisord , check /opt/apache-storm-1.2.2/logs/workers-artifacts/  
# he is worker  
  
[root@bf-dev-152 sre]#
```

## ## frontend

前端的帳密用  
fe  
12345  
需要Sudo su 取得權限

Ray>

12345 太簡單了 我們改一下  
3vflHmAnB2

Howard Liao>  
對了 在做QAT的時候  
前端在gitlab 會做自動build 自動deploy 用的 scp 和ssh  
到時候新的QAT 也要有這個功能

H5

```
- tar -czvf h5.tgz dist
- scp -i ~/.ssh/gb_fe h5.tgz gb_fe@10.140.0.43:/home/gb_fe/
- ssh -t -i ~/.ssh/gb_fe gb_fe@10.140.0.43 "sudo rm -rf /opt/gm/h5_view/dist && sudo tar -xzvf /home/gb_fe/h5.tgz --directory /opt/gm/h5_view/ && exit"
```

admin

```
- tar -czvf admin.tgz dist
- scp -i ~/.ssh/gb_fe admin.tgz gb_fe@10.140.0.44:/home/gb_fe/
- ssh -t -i ~/.ssh/gb_fe gb_fe@10.140.0.44 "sudo rm -rf /opt/gm/admin_view/dist && sudo tar -xzvf /home/gb_fe/admin.tgz --directory /opt/gm/admin_view/ && exit"
```

ray>

10.140.0.43 - 44 在gcp上掛的是 dev-admin dev-h5 (game專案中的dev就是qa環境

frontend gitlab

<https://gitlab.sincheng.net/GroupProduct/frontend/h5.gm.sincheng.net/-/pipelines>  
runner to cd

cannot cd to twoffice and also other project...