文件名称：Kafka集群部署手册（Linux）

文件编号：

项目名称：

项目编号：

物料编码：

版 本 号：A

文件密级：秘密

文件状态：CFC

受控标识：受控

**修订页**

| 序号 | 版本号 | 修订内容简述 | 拟制/日期 | 审核 | 批准 |
| --- | --- | --- | --- | --- | --- |
|  | A | 创建 | 张毅 |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

1. **Zookeeper配置构建**
2. 创建文件

*在130、131、132服务器*

|  |
| --- |
| rm -rf /opt/deploy/data/zookeeper && mkdir -p /opt/deploy/data/zookeeper/data /opt/deploy/data/zookeeper/datalog /opt/deploy/data/zookeeper/conf /opt/deploy/data/zookeeper/logs  cd /opt/deploy/data/zookeeper |

1. 创建Zookeeper模板

|  |
| --- |
| cd /opt/deploy/data/zookeeper  vim zookeeper-cluster.tmpl |

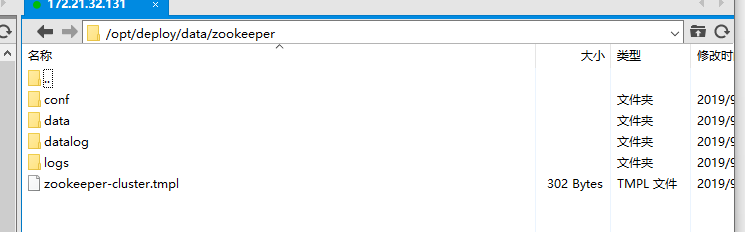
*在130、131、132服务器zookeeper-cluster.tmpl文件内容*

|  |
| --- |
| dataDir=/data  dataLogDir=/datalog  tickTime=2000  initLimit=5  syncLimit=2  autopurge.snapRetainCount=3  autopurge.purgeInterval=0  maxClientCnxns=100  standaloneEnabled=true  admin.enableServer=true  server.1=zookeeper-1:2888:3888;2181  server.2=zookeeper-2:2888:3888;2181  server.3=zookeeper-3:2888:3888;2181 |

1. 构建目录

*在130、131、132服务器*

|  |
| --- |
| cd /opt/deploy/data/zookeeper  mkdir -p ./conf \  && PORT=2181 envsubst < ./zookeeper-cluster.tmpl > ./conf/zoo.cfg \  && rm -rf data \  && mkdir -p ./data \  && rm -rf datalog \  && mkdir -p ./datalog \  && rm -rf logs \  && mkdir -p ./logs; |



1. **Kafka配置构建**
2. 创建文件

*在130、131、132服务器*

|  |
| --- |
| rm -rf /opt/deploy/data/kafka && mkdir -p /opt/deploy/data/kafka/data /opt/deploy/data/kafka/conf /opt/deploy/data/kafka/logs  cd /opt/deploy/data/kafka |

1. 创建Kafka配置模板

|  |
| --- |
| cd /opt/deploy/data/kafka  vim kafka-cluster.tmpl |

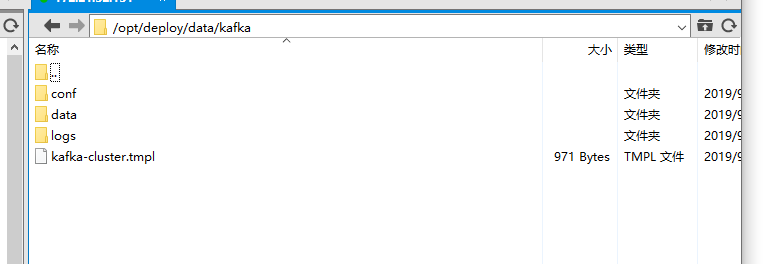
*在130、131、132服务器*kafka-cluster.tmpl*文件内容*

|  |
| --- |
| broker.id=${ID}  port=${PORT}  listeners=PLAINTEXT://:${PORT}  #advertised.listeners=PLAINTEXT://${IP}:${PORT}  #advertised.host.name=${IP}  #listener.security.protocol.map=PLAINTEXT:PLAINTEXT,SSL:SSL,SASL\_PLAINTEXT:SASL\_PLAINTEXT,SASL\_SSL:SASL\_SSL  num.network.threads=3  num.io.threads=8  socket.send.buffer.bytes=104857600  socket.receive.buffer.bytes=104857600  socket.request.max.bytes=104857600  replica.fetch.max.bytes=104857600  quota.producer.default=104857600  quota.consumer.default=104857600  log.dirs=/kafka/kafka-logs  num.partitions=1  num.recovery.threads.per.data.dir=1  offsets.topic.replication.factor=1  transaction.state.log.replication.factor=1  transaction.state.log.min.isr=1  #log.flush.interval.messages=10000  #log.flush.interval.ms=1000  log.retention.hours=168  #log.retention.bytes=1073741824  log.segment.bytes=1073741824  log.retention.check.interval.ms=300000  zookeeper.connect=${ZK\_HOSTS}  zookeeper.connection.timeout.ms=6000  group.initial.rebalance.delay.ms=0 |

1. 构建目录

*在130、131、132服务器*

|  |
| --- |
| cd /opt/deploy/data/kafka  mkdir -p ./conf \  && PORT=9092 ID=`date '+%S'` ZK\_HOSTS=zookeeper-1:2181,zookeeper-2:2181,zookeeper-3:2181 IP=`ifconfig -a|grep inet|grep -v 127.0.0.1|grep -v 0.0.0.0|grep -v inet6|grep 172.21.32|awk '{print $2}'|tr -d 'addr:'` envsubst < ./kafka-cluster.tmpl > ./conf/server.properties \  && rm -rf data \  && mkdir -p ./data \  && rm -rf logs \  && mkdir -p ./logs; |

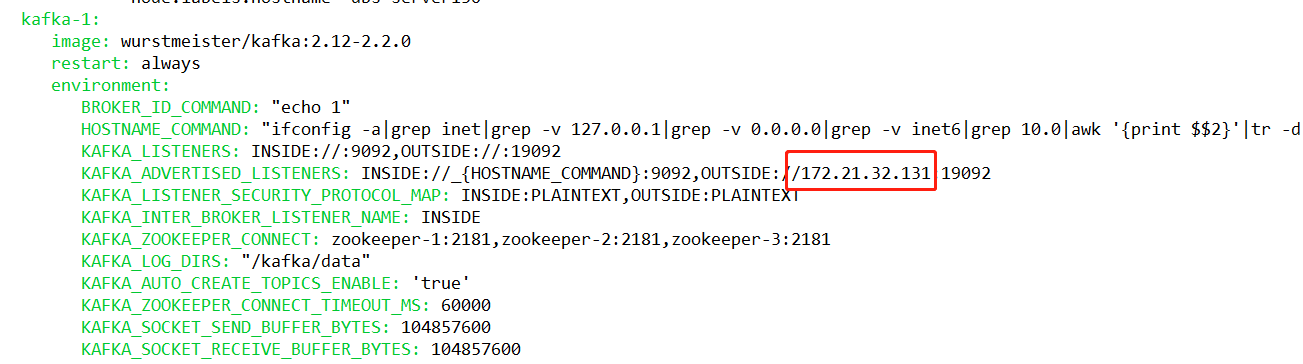
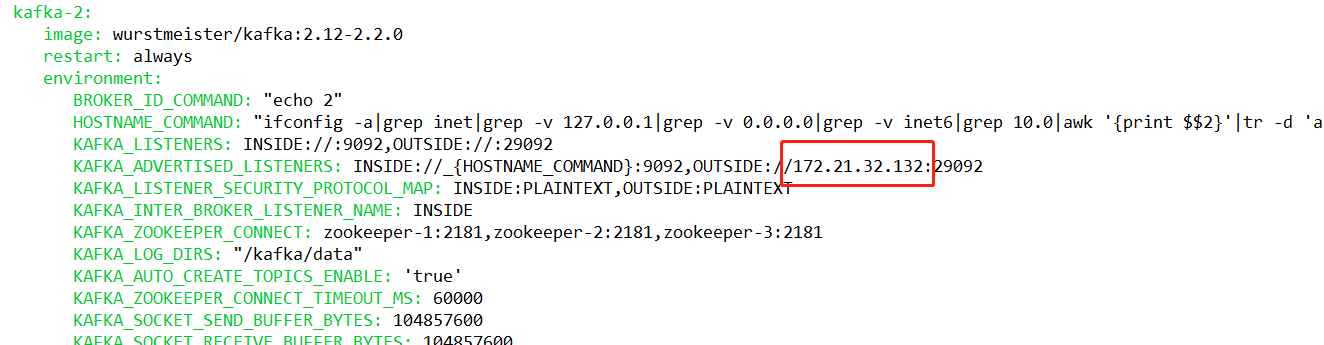
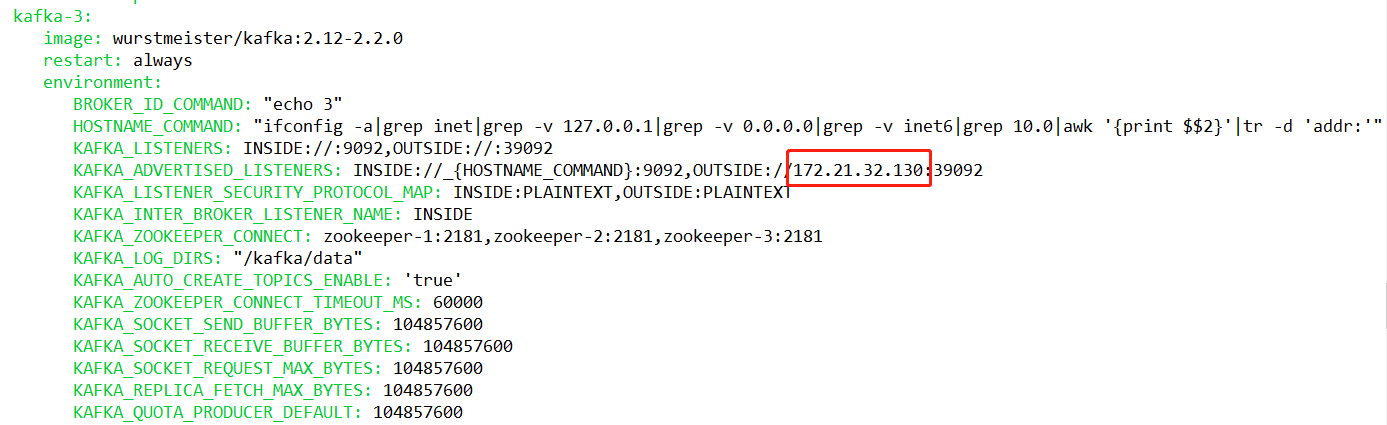


1. **Zookeeper以及Kafka集群部署**
2. 打docker标签

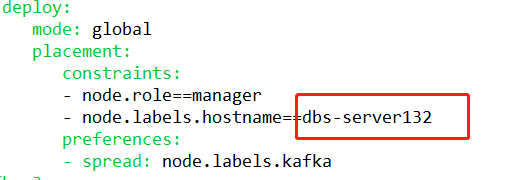
*在130、131、132服务器*

|  |
| --- |
| docker node update --label-add hostname=`hostname -s` `hostname -s`  docker node update --label-add kafka=`hostname -s` `hostname -s`  docker node inspect `hostname -s` |

1. Docker-Compose文件

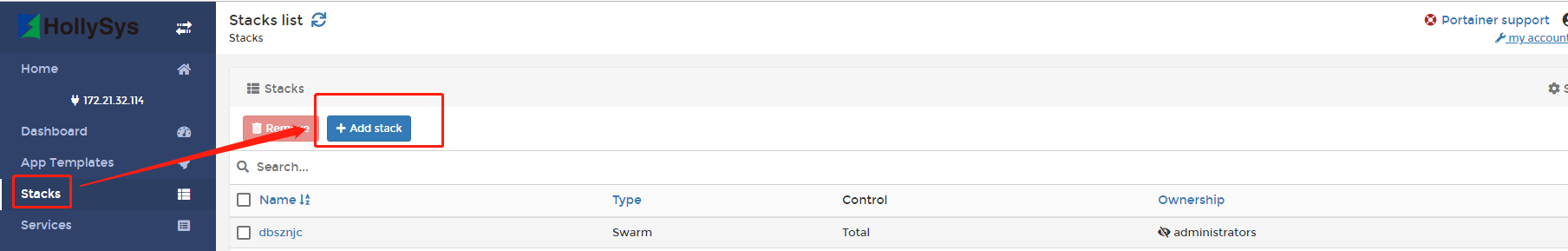
  

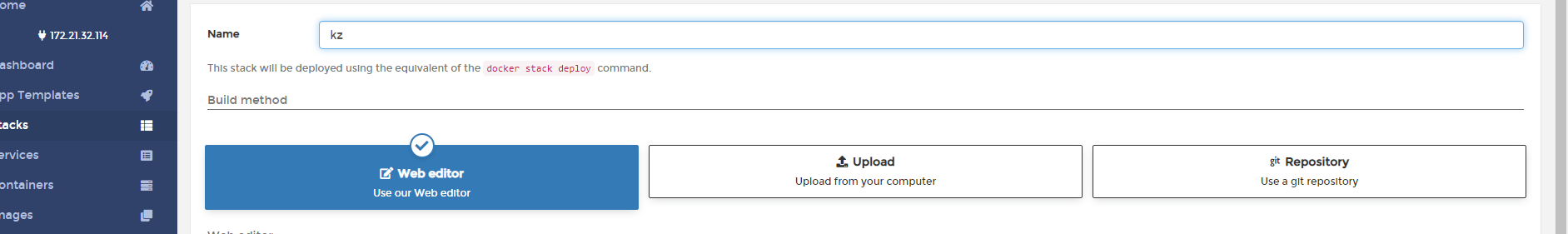
***注：标注部分必须虚拟机主机IP***

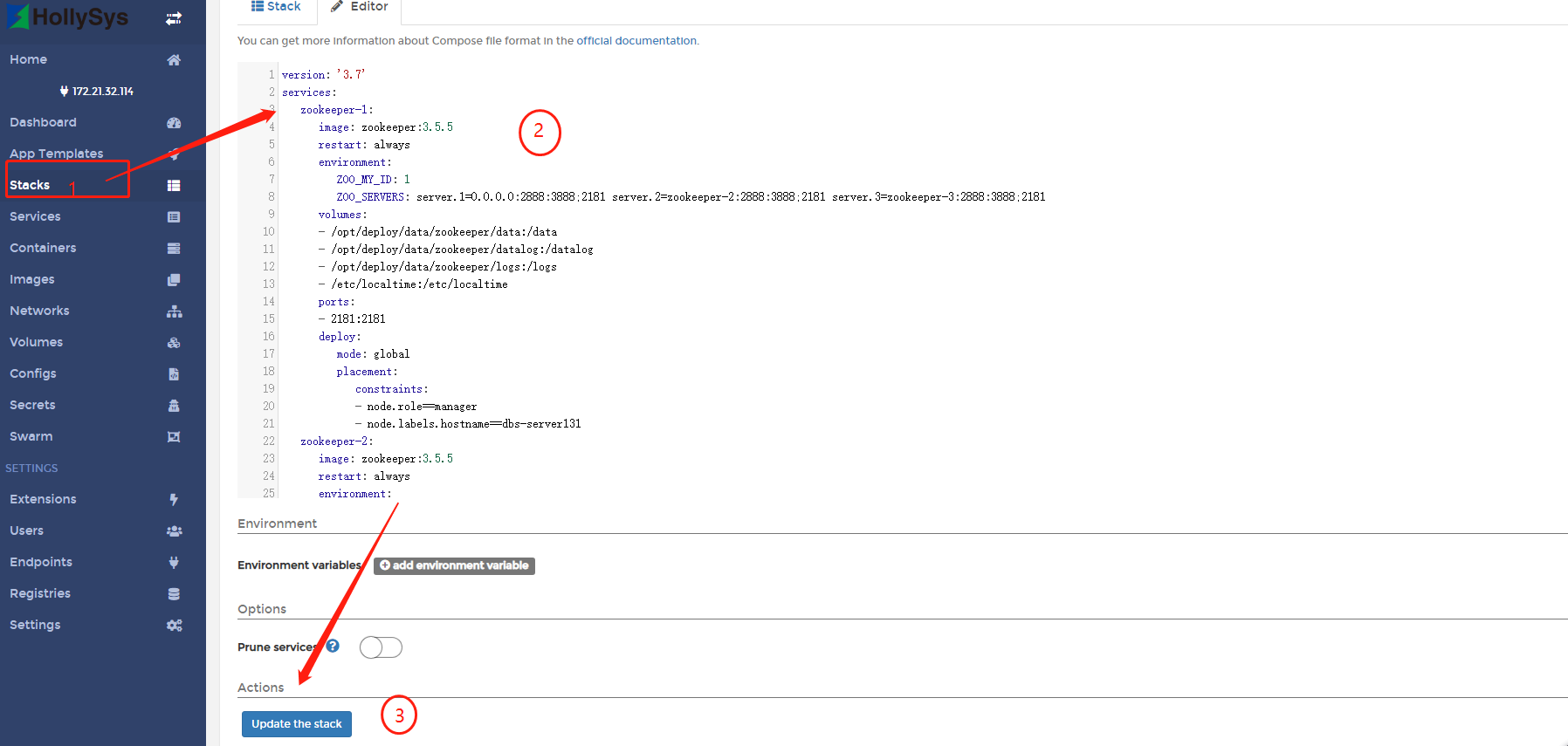


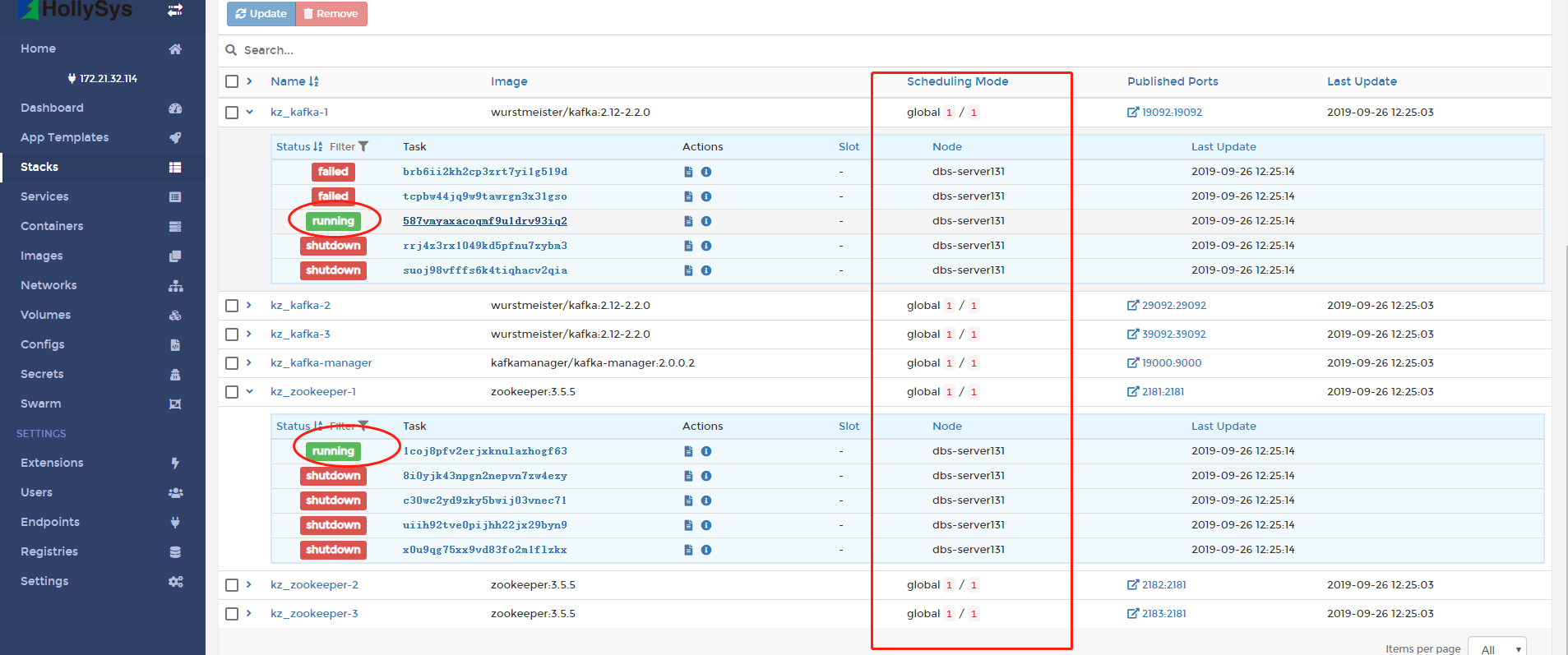
***注：主机名更改为虚拟机主机名称***

1. 发布集群



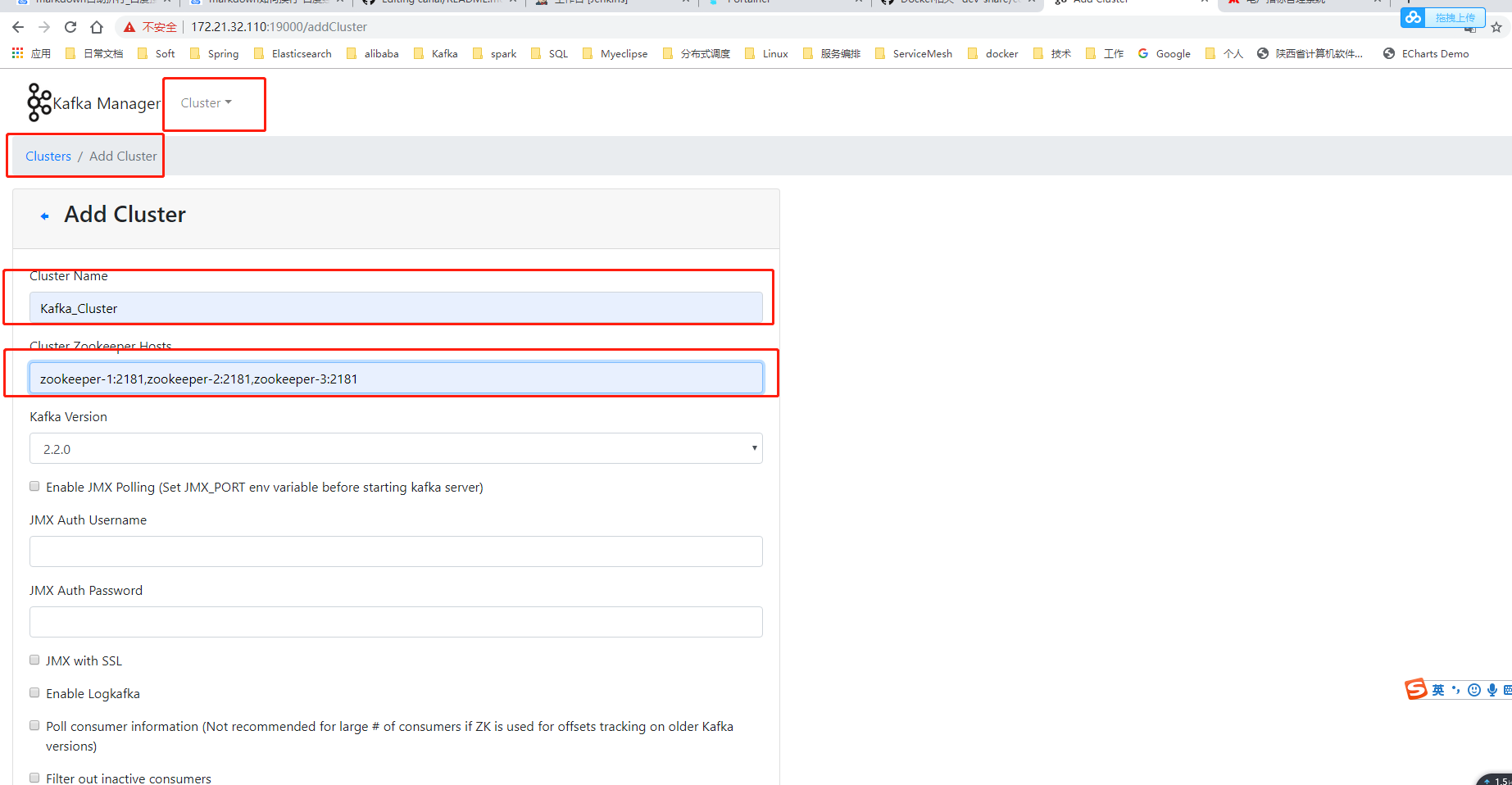


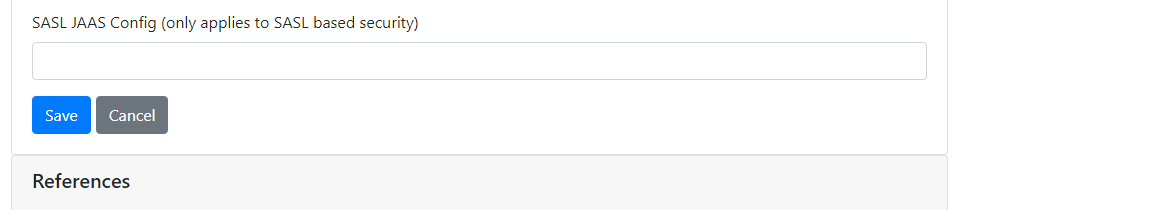


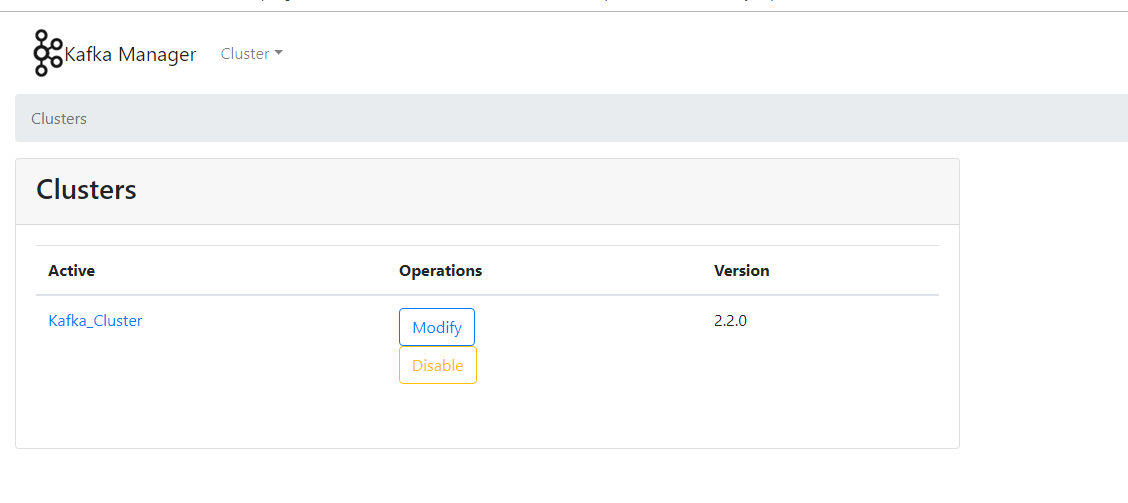


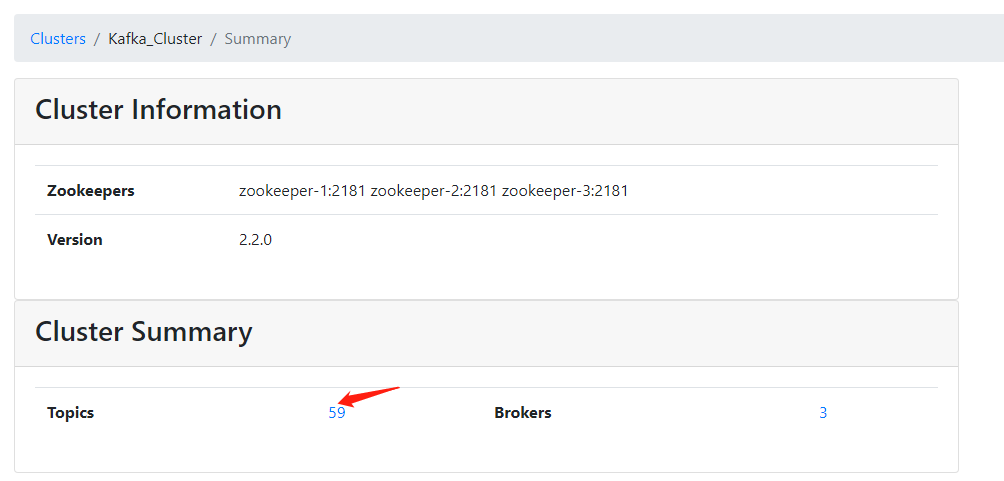
**注：集群分片为1,状态为running**

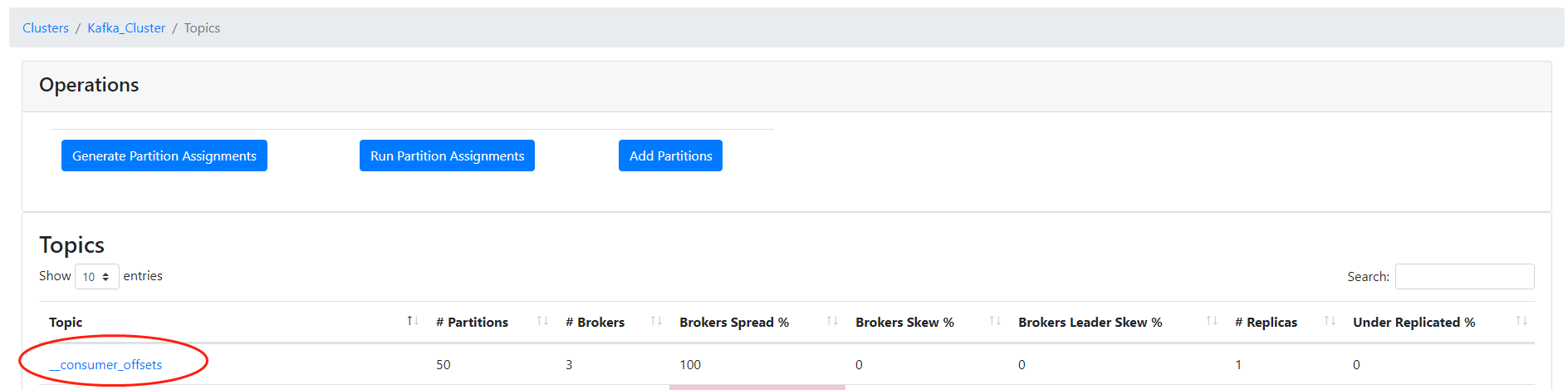
1. 查看集群











**注: 标注地方有值则为成功**