

Kafka

[零 参考资料 2](#_Toc66112799)

[一 概述 2](#_Toc66112800)

[1 体系结构 2](#_Toc66112801)

[1.1 概念 2](#_Toc66112802)

[2 时间轮 4](#_Toc66112803)

[3 ISR & HW & LEO 4](#_Toc66112804)

[二 Producer 6](#_Toc66112805)

[1 概述 6](#_Toc66112806)

[2 配置 6](#_Toc66112807)

[3 API 8](#_Toc66112808)

[三 Consumer 8](#_Toc66112809)

[1 概述 8](#_Toc66112810)

[2 配置 8](#_Toc66112811)

[3 API 12](#_Toc66112812)

[四 Topic&Partition 12](#_Toc66112813)

[1 概述 12](#_Toc66112814)

[2 Topic配置 12](#_Toc66112815)

[五 Log Storage 13](#_Toc66112816)

[1 概述 13](#_Toc66112817)

[2 日志V0 13](#_Toc66112818)

[3 日志V1 13](#_Toc66112819)

[4 日志V3 13](#_Toc66112820)

[六 Server 13](#_Toc66112821)

[1 概述 13](#_Toc66112822)

[2 Broker配置 13](#_Toc66112823)

[七 监控 20](#_Toc66112824)

[八 Connect 20](#_Toc66112825)

[1 概述 20](#_Toc66112826)

[2 配置 20](#_Toc66112827)

[九 应用 20](#_Toc66112828)

# 零 参考资料

https://kafka.apache.org/documentation/

https://kafka.apache.org/26/javadoc/index.html

《深入理解Kafka 核心设计与实践原理》朱忠华

PS：1 文档是2.6版本，书是2.0版本，参数配置以文档为准，具体差异不在本文体现

2 红色配置为关键配置

# 概述

## 体系结构

### 概念

* + - 1. Producer：生产者，即消息发送方，负责产生消息，投递到Kafka中
      2. Consumer：消费者，即消息接收方，连接到kafka上并接收消息，然后根据消息处理业务逻辑
      3. Broker：服务代理节点
         1. 可以简单的理解为一个单独的kafka服务节点或Kafka服务实例，
         2. 但如果一台服务器上只部署了一个kafka实例，才可以将broker看作一台Kafka服务器。
         3. 一个或多个Broker组成了kafka集群
      4. Topic：主题，逻辑上的概念，kafka消息按topic进行分类
         1. 生产者负责将消息发送到特定Topic
         2. 消费者通过订阅topic消费消息
         3. 一个Topic可以横跨多个Broker
      5. Partition: Topic的分区
         1. 一个Partition只属于单个Topic，因此也被称为Topic-Partition
         2. 同个Topic下，不同Partition包含的消息不同
         3. 在存储层面，Partition可以看作是一个可追加的Log文件
      6. Offset：偏移量，从0开始
         1. 消息被追加到Partition的log文件时，会分配到一个特定的Offset
         2. offset是消息在存储其的Partition中唯一标识。
         3. offset不跨分区，因此kafka通过offset保证消息在当前分区有序，但不保证消息在Topic中有序
      7. Replica：副本
         1. kafka基于Partition实现多Replica机制，提高容灾能力
         2. 同一个Partition的不同Replica中保存的消息最终相同，即同一时刻，同一个Partition的多Replica之间并非一定相同
         3. Replica之间是一主多从关系：

Leader 负责处理消息读写请求，follower只负责与Leader消息同步

当Leader出现故障时，从Follower中重新选举新的Leader对外服务

大部分时候follower的消息相对Leader有一定延后

* + - 1. Assigned Replica：Partition所有replica的统称
         1. ISR(In-Sync Replica)：由所有与Leader保持一定程度同步的replica组成，是AR的子集
         2. OSR(Out-of-Sync Replica)：由与Leader同步滞后过多的Replica组成，是AR的子集
         3. AR = ISR+OSR,正常情况下，AR=ISR，OSR为空
         4. Leader负责维护&跟踪所有follower的同步情况

当ISR中的follower滞后太多或失效时，leader会把其移出ISR到OSR中

当OSR中的follower同步消息追上leader时，将追上的follower从OSR移到ISR

* + - 1. HW(High Watermark)：高水位，标识了一个特定的消息offset，consumer只能获取到这个offset之前的消息
      2. LEO(Log End Offset)：标识当前log下一条待写入消息offset，相当于当前log最后条消息offset+1







## 时间轮

## ISR & HW & LEO

* + - 1. 假设某个分区的ISR集合中有3个副本，即1个leader和两个follow
         1. 此时LEO和HW都为3
         2. 生产者发出消息3和消息4，会被先存入leader



* + - 1. 消息3&4写入Leader后，Follower会请求拉取消息3&4以进行同步



* + - 1. 同步过程中，不同Follower同步效率不同，下图情况：
         1. Leader的LEO=5，Follower1 LEO=5，Follower2 LEO=4， HW=4
         2. Consumer可以获取0-3消息



* + - 1. 当所有Replica都同步到消息4，则所有HW和LEO都为5，Consumer可以获得到消息4了



# Producer

## 概述



## 配置

|  |  |
| --- | --- |
| 配置 | 说明 |
| key.serializer |  |
| value.serializer |  |
| acks |  |
| bootstrap.servers |  |
| buffer.memory |  |
| compression.type |  |
| retries |  |
| ssl.key.password |  |
| ssl.keystore.location |  |
| ssl.keystore.password |  |
| ssl.truststore.location |  |
| ssl.truststore.password |  |
| batch.size |  |
| client.dns.lookup |  |
| client.id |  |
| connections.max.idle.ms |  |
| delivery.timeout.ms |  |
| linger.ms |  |
| max.block.ms |  |
| max.request.size |  |
| partitioner.class |  |
| receive.buffer.bytes |  |
| request.timeout.ms |  |
| sasl.client.callback.handler.class |  |
| sasl.jaas.config |  |
| sasl.kerberos.service.name |  |
| sasl.login.callback.handler.class |  |
| sasl.login.class |  |
| sasl.mechanism |  |
| security.protocol |  |
| send.buffer.bytes |  |
| ssl.enabled.protocols |  |
| ssl.keystore.type |  |
| ssl.protocol |  |
| ssl.provider |  |
| ssl.truststore.type |  |
| enable.idempotence |  |
| interceptor.classes |  |
| max.in.flight.requests.per.connection |  |
| metadata.max.age.ms |  |
| metadata.max.idle.ms |  |
| metric.reporters |  |
| metrics.num.samples |  |
| metrics.recording.level |  |
| metrics.sample.window.ms |  |
| reconnect.backoff.max.ms |  |
| reconnect.backoff.ms |  |
| retry.backoff.ms |  |
| sasl.kerberos.kinit.cmd |  |
| sasl.kerberos.min.time.before.relogin |  |
| sasl.kerberos.ticket.renew.jitter |  |
| sasl.kerberos.ticket.renew.window.factor |  |
| sasl.login.refresh.buffer.seconds |  |
| sasl.login.refresh.min.period.seconds |  |
| sasl.login.refresh.window.factor |  |
| sasl.login.refresh.window.jitter |  |
| security.providers |  |
| ssl.cipher.suites |  |
| ssl.endpoint.identification.algorithm |  |
| ssl.engine.factory.class |  |
| ssl.keymanager.algorithm |  |
| ssl.secure.random.implementation |  |
| ssl.trustmanager.algorithm |  |
| transaction.timeout.ms |  |
| transactional.id |  |

## API

# Consumer

## 概述

## 配置

|  |  |
| --- | --- |
| 配置项 | 说明 |
| key.deserializer | 指定反序列化key值的反序列化工具  继承org.apache.kafka.common.serialization.Deserializer |
| value.deserializer | 指定反序列化消息value值的反序列化工具  继承org.apache.kafka.common.serialization.Deserializer |
| bootstrap.servers | 指定broker服务地址，用于消费者获取集群信息  可以只指定集群一个broker，或全部broker  多个地址用逗号隔开host1:port1,host2:port2,... |
| group.id | 设置消费者所属组  如果消费者采用subscribe(topic) 或 the Kafka-based offset管理策略作为组管理方法，则必须配置，且必须唯一 |
| fetch.max.bytes | 每次批量拉取消息最大数据大小  为保障消费者正常处理，如果第一次拉取消息数据大小大于此值，仍继续正常拉取  值受message.max.bytes (broker config) or max.message.bytes (topic config) |
| fetch.min.bytes | 指定消费者抓取数据最小大小，如果没有足够数据，服务器将等待积累数据，直到数据足够时才返回数据给消费者  默认1byte，即服务器及时响应请求  适当设置高点，可以提高服务器吞吐量 |
| fetch.max.wait.ms | 拉取数据没达到fetch.min.bytes指定数据量时，最大等待时间 |
| heartbeat.interval.ms | 心跳检查时间间隔  值必须小于session.timeout.ms配置，推荐不超过1/3 session.timeout.ms  默认3000ms |
| max.partition.fetch.bytes | 每个分区批量获取消息最大数据量  为保障消费者正常执行，第一次批量拉取消息时超过了限制，则也会拉取消息  此配置受message.max.bytes (broker config) 或max.message.bytes(topic config)影响  默认1048576(1mb) |
| session.timeout.ms | 心跳检测超时时间  如果超时，broker会移除当前消费者，并重新负载  默认10000  值必须在group.min.session.timeout.ms配置和group.max.session.timeout.ms配置之间 |
| allow.auto.create.topics | 是否自动创建订阅的topic  仅当broker配置auto.create.topics.enable为tru时有效  链接0.11.0之前版本的broker，必须为false  默认为true |
| auto.offset.reset | 当消费者没有指定初始offset或指定的offset不存在时，指定定位offset策略  取值[latest, earliest, none]  earliest：定位到当前分区第一条记录  latest：定位到当前分区最后一条记录  none：如果没有指定初始offset或指定的offset不存在时，抛异常  其他值都抛异常 |
| client.dns.lookup | 设置DNS查找域名对应的ip列表缓存规则  默认use\_all\_dns\_ips  取值[default, use\_all\_dns\_ips, resolve\_canonical\_bootstrap\_servers\_only]  use\_all\_dns\_ips：依次访问DNS返回的IP 列表，直到ip可访问；如果断开链接，则尝试下个ip；如果所有ip都无法连接，则重新请求DNS获取ip列表  resolve\_canonical\_bootstrap\_servers\_only：解析bootstrap address为规范名称列表，然后按照use\_all\_dns\_ips策略访问  default：仅尝试请求第一个ip地址，即使配置了多个broker地址 |
| connections.max.idle.ms | 链接空闲最大时间  默认540000s (9m) |
| default.api.timeout.ms | api请求超时时间，每个超时相关配置没配置时，会取此配置值  默认60000ms(1m) |
| enable.auto.commit | 是否自动提交消费事务  默认true |
| auto.commit.interval.ms | 定时提交offset，当enable.auto.commit为true时  默认5000 ms |
| exclude.internal.topics | 当使用正则表达式订阅topic时，是否排除内部topic  默认为true |
| group.instance.id | 消费者在群组中唯一标识  默认为不设置  如果不设置，则作为一个动态成员加入组  如果设置，则作为一个静态成员加入组，可以优化长时间session |
| isolation.level | 配置读取写事务中消息的策略  默认值read\_uncommitted  取值[read\_committed, read\_uncommitted]  read\_committed：只读取事务提交后的消息，  read\_uncommitted：读取所有消息 |
| max.poll.interval.ms | customer每次poll等待最大时间  当customer没有在此时间内拉取数据，broker会认为此customer失效，会将此customer负责的分区重新分配给组内其他成员  如果此customer指定了group.instance.id，则不会立即重新分配分区，直到customer的session断开时间超过session.timeout.ms配置，才会在组内重新分配分区  默认300000ms |
| max.poll.records | 每次poll最大数据量  默认500 |
| partition.assignment.strategy | 分区获取策略，默认是依次轮询  默认org.apache.kafka.clients.consumer.RoundRobinAssignor  可以通过实现org.apache.kafka.clients.consumer.ConsumerPartitionAssignor接口自定义策略 |
| send.buffer.bytes | 配置发送套接字缓存  如果设置为-1，则以OS配置为准  默认131072 B(128KB) |
| receive.buffer.bytes | 套接字(SO\_RCVBUF)缓冲大小  如果设置为-1，则以OS配置为准  默认65536 B(64KB) |
| request.timeout.ms | customer等待broker响应时间  默认30000ms |
| security.protocol | 配置安全协议  默认PLAINTEXT  取值PLAINTEXT, SSL, SASL\_PLAINTEXT, SASL\_SSL. |
| security.providers |  |
| socket.connection.setup.timeout.max.ms | 配置套接字链接超时最大时间  每次套接字链接超时，都会以0.2为因子，随机比例增加或减少超时时间,但最大超时时间不会超过此值 |
| socket.connection.setup.timeout.ms | 配置套接字链接超时时间 |
| check.crcs | 是否检查消息的CRC32值，保证消息完整  默认为true  会有一定的性能消耗 |
| client.id | 配置客户端ID，用于broker日志中追踪请求源  默认为空字符串 |
| client.rack | 主机所在位置标识，用于标识customer物理位置  默认为空字符串 |
| interceptor.classes | 配置拦截器  通过org.apache.kafka.clients.consumer.ConsumerInterceptor接口实现自定义拦截器，  默认空 |
| metadata.max.age.ms | 定时强制更新元数据信息，保证发现broker或分区变动  默认300000s(5min) |
| metric.reporters |  |
| metrics.num.samples |  |
| metrics.recording.level |  |
| metrics.sample.window.ms |  |
| reconnect.backoff.max.ms | 重新链接前最大等待时间  默认1000ms |
| reconnect.backoff.ms | 重新链接前等待时间  默认50ms |
| retry.backoff.ms | 重试间隔等时间  默认100ms |
| ssl.key.password |  |
| ssl.keystore.location |  |
| ssl.keystore.password |  |
| ssl.truststore.location |  |
| ssl.truststore.password |  |
| ssl.enabled.protocols |  |
| ssl.keystore.type |  |
| ssl.protocol |  |
| ssl.provider |  |
| ssl.truststore.type |  |
| ssl.cipher.suites |  |
| ssl.endpoint.identification.algorithm |  |
| ssl.engine.factory.class |  |
| ssl.keymanager.algorithm |  |
| ssl.secure.random.implementation |  |
| ssl.trustmanager.algorithm |  |
| sasl.client.callback.handler.class |  |
| sasl.jaas.config |  |
| sasl.kerberos.service.name |  |
| sasl.login.callback.handler.class |  |
| sasl.login.class |  |
| sasl.mechanism |  |
| sasl.kerberos.kinit.cmd |  |
| sasl.kerberos.min.time.before.relogin |  |
| sasl.kerberos.ticket.renew.jitter |  |
| sasl.kerberos.ticket.renew.window.factor |  |
| sasl.login.refresh.buffer.seconds |  |
| sasl.login.refresh.min.period.seconds |  |
| sasl.login.refresh.window.factor |  |
| sasl.login.refresh.window.jitter |  |

## API

# Topic&Partition

## 概述

## Topic配置

|  |  |
| --- | --- |
| 配置项 | 说明 |
| cleanup.policy |  |
| compression.type |  |
| delete.retention.ms |  |
| file.delete.delay.ms |  |
| flush.messages |  |
| flush.ms |  |
| follower.replication.throttled.replicas |  |
| index.interval.bytes |  |
| leader.replication.throttled.replicas |  |
| max.compaction.lag.ms |  |
| max.message.bytes |  |
| message.format.version |  |
| message.timestamp.difference.max.ms |  |
| message.timestamp.type |  |
| min.cleanable.dirty.ratio |  |
| min.compaction.lag.ms |  |
| min.insync.replicas |  |
| preallocate |  |
| retention.bytes |  |
| retention.ms |  |
| segment.bytes |  |
| segment.index.bytes |  |
| segment.jitter.ms |  |
| segment.ms |  |
| unclean.leader.election.enable |  |
| message.downconversion.enable |  |

# Log Storage

## 概述

## 日志V0

## 日志V1

## 日志V3

# Server

## 概述

## Broker配置

|  |  |
| --- | --- |
| 配置项 | 说明 |
| zookeeper.connect | 指定zk地址hostname:port  多个用英文逗号隔开  也可以指定zk路径  hostname1:port1,hostname2:port2,hostname3:port3/chroot/path |
| advertised.host.name | 不推荐使用  仅当advertised.listeners或listeners配置为空时有效  建议用advertised.listeners替代  将指定外网地址推送到zk  如果没有配置则取host.name值  如果也没有配置host.name则取java.net.InetAddress.getCanonicalHostName()值  在IaaS等外网地址需要特指的环境中使用 |
| advertised.listeners | 将指定外网地址推送到zk  如果没有配置，则推送listeners配置  不能配置为0.0.0.0 |
| advertised.port | 不推荐使用  仅当advertised.listeners或listeners配置为空时有效  建议用advertised.listeners替代  将指定端口推送到zk  如果没有配置，则推送broker绑定端口 |
| auto.create.topics.enable | 是否允许自动创建topic  默认true  如果为true，则producer发送消息到一个未存在的topic，broker会自动创建一个topic |
| auto.leader.rebalance.enable | 是否允许leader重新负载  默认为true 当为true时，后台线程会根据leader.imbalance.check.interval.seconds定期检查当前leader，如果当前leader负载超过了leader.imbalance.per.broker.percentage百分比，则自动重新负载分配leader |
| background.threads | 配置后台处理辅助任务的线程数  默认10，取值为大于0的正整数 |
| broker.id | 设置broker唯一标识  默认值-1， 取值int型  如果不设置，则自动生成一个  为保证同个zk下broker.id唯一，自动生成的id从reserved.broker.max.id+1开始 |
| compression.type | 指定topic消息默认压缩方式  默认值 producer  取值gzip, snappy, lz4, zstd，producer，uncompressed  如果设置为uncompressed，则为不压缩  如果设置为producer，则根据producer配置决定压缩方式 |
| control.plane.listener.name |  |
| delete.topic.enable | 是否允许非admin权限删除topic  默认为true  如果设置为false，不影响admin tool删除topic |
| host.name |  |
| leader.imbalance.check.interval.seconds | 定期检查leader负载情况  默认300s |
| leader.imbalance.per.broker.percentage | 每个leader最高负载率  默认10 |
| listeners |  |
| log.dir | 指定log目录  当log.dirs未设置时，生效 |
| log.dirs | 指定多个log目录  如果未设置则取log.dir配置 |
| log.flush.interval.messages | log partation积累的消息flush到disk的阈值  默认9223372036854775807  取值为大于0的整数 |
| log.flush.interval.ms | 定时将消息从memory flush to disk  默认不设置  如果不设置，则取log.flush.scheduler.interval.ms值 |
| log.flush.offset.checkpoint.interval.ms | 设置更新最后flush to disk的记录信息的频率，相当于定时更新恢复点  默认60000ms  取值大于等于0整数 |
| log.flush.scheduler.interval.ms | 定时检查日志是否flush to disk  默认9223372036854775807ms  取值大于等于0整数 |
| log.flush.start.offset.checkpoint.interval.ms | 定时更新当前检查点记录对应的起始offset  默认60000ms (1分钟) |
| log.retention.bytes | 日志保留最大大小  默认 -1 |
| log.retention.hours | 消息日志保留时间小时，  默认168 h（7d） |
| log.retention.minutes | 消息日志保留时间分钟，  如果不设置则按log.retention.hours配置处理  默认null |
| log.retention.ms | 消息日志保留时间毫秒  如果不设置则按log.retention.minutes配置处理  默认null  如果设置为-1，则永久保留消息 |
| log.roll.hours | 当前日志文件创建时间距离当前时间的小时差大于指定值，则切分日志文件  默认168 h（7d）  仅当log.roll.ms没配置时生效 |
| log.roll.jitter.hours |  |
| log.roll.jitter.ms |  |
| log.roll.ms | 当前日志文件创建时间距离当前时间的毫秒差大于指定值，则切分日志文件  默认不设置，采用log.roll.hours配置 |
| log.segment.bytes |  |
| log.segment.delete.delay.ms |  |
| message.max.bytes |  |
| min.insync.replicas |  |
| num.io.threads |  |
| num.network.threads |  |
| num.recovery.threads.per.data.dir |  |
| num.replica.alter.log.dirs.threads |  |
| num.replica.fetchers |  |
| offset.metadata.max.bytes |  |
| offsets.commit.required.acks |  |
| offsets.commit.timeout.ms |  |
| offsets.load.buffer.size |  |
| offsets.retention.check.interval.ms |  |
| offsets.retention.minutes |  |
| offsets.topic.compression.codec |  |
| offsets.topic.num.partitions |  |
| offsets.topic.replication.factor |  |
| offsets.topic.segment.bytes |  |
| port |  |
| queued.max.requests |  |
| quota.consumer.default |  |
| quota.producer.default |  |
| replica.fetch.min.bytes |  |
| replica.fetch.wait.max.ms |  |
| replica.high.watermark.checkpoint.interval.ms |  |
| replica.lag.time.max.ms |  |
| replica.socket.receive.buffer.bytes |  |
| replica.socket.timeout.ms |  |
| request.timeout.ms |  |
| socket.receive.buffer.bytes |  |
| socket.request.max.bytes |  |
| socket.send.buffer.bytes |  |
| transaction.max.timeout.ms |  |
| transaction.state.log.load.buffer.size |  |
| transaction.state.log.min.isr |  |
| transaction.state.log.num.partitions |  |
| transaction.state.log.replication.factor |  |
| transaction.state.log.segment.bytes |  |
| transactional.id.expiration.ms |  |
| unclean.leader.election.enable |  |
| zookeeper.connection.timeout.ms |  |
| zookeeper.max.in.flight.requests |  |
| zookeeper.session.timeout.ms |  |
| zookeeper.set.acl |  |
| broker.id.generation.enable |  |
| broker.rack |  |
| connections.max.idle.ms |  |
| connections.max.reauth.ms |  |
| controlled.shutdown.enable |  |
| controlled.shutdown.max.retries |  |
| controlled.shutdown.retry.backoff.ms |  |
| controller.socket.timeout.ms |  |
| default.replication.factor |  |
| delegation.token.expiry.time.ms |  |
| delegation.token.master.key |  |
| delegation.token.max.lifetime.ms |  |
| delete.records.purgatory.purge.interval.requests |  |
| fetch.max.bytes |  |
| fetch.purgatory.purge.interval.requests |  |
| group.initial.rebalance.delay.ms |  |
| group.max.session.timeout.ms |  |
| group.max.size |  |
| group.min.session.timeout.ms |  |
| inter.broker.listener.name |  |
| inter.broker.protocol.version |  |
| log.cleaner.backoff.ms |  |
| log.cleaner.dedupe.buffer.size |  |
| log.cleaner.delete.retention.ms |  |
| log.cleaner.enable |  |
| log.cleaner.io.buffer.load.factor |  |
| log.cleaner.io.buffer.size |  |
| log.cleaner.io.max.bytes.per.second |  |
| log.cleaner.max.compaction.lag.ms |  |
| log.cleaner.min.cleanable.ratio |  |
| log.cleaner.min.compaction.lag.ms |  |
| log.cleaner.threads |  |
| log.cleanup.policy |  |
| log.index.interval.bytes |  |
| log.index.size.max.bytes |  |
| log.message.format.version |  |
| log.message.timestamp.difference.max.ms |  |
| log.message.timestamp.type |  |
| log.preallocate |  |
| log.retention.check.interval.ms |  |
| max.connections |  |
| max.connections.per.ip |  |
| max.connections.per.ip.overrides |  |
| max.incremental.fetch.session.cache.slots |  |
| num.partitions |  |
| password.encoder.old.secret |  |
| password.encoder.secret |  |
| principal.builder.class |  |
| producer.purgatory.purge.interval.requests |  |
| queued.max.request.bytes |  |
| replica.fetch.backoff.ms |  |
| replica.fetch.max.bytes |  |
| replica.fetch.response.max.bytes |  |
| replica.selector.class |  |
| reserved.broker.max.id |  |
| sasl.client.callback.handler.class |  |
| sasl.enabled.mechanisms |  |
| sasl.jaas.config |  |
| sasl.kerberos.kinit.cmd |  |
| sasl.kerberos.min.time.before.relogin |  |
| sasl.kerberos.principal.to.local.rules |  |
| sasl.kerberos.service.name |  |
| sasl.kerberos.ticket.renew.jitter |  |
| sasl.kerberos.ticket.renew.window.factor |  |
| sasl.login.callback.handler.class |  |
| sasl.login.class |  |
| sasl.login.refresh.buffer.seconds |  |
| sasl.login.refresh.min.period.seconds |  |
| sasl.login.refresh.window.factor |  |
| sasl.login.refresh.window.jitter |  |
| sasl.mechanism.inter.broker.protocol |  |
| sasl.server.callback.handler.class |  |
| security.inter.broker.protocol |  |
| ssl.cipher.suites |  |
| ssl.client.auth |  |
| ssl.enabled.protocols |  |
| ssl.key.password |  |
| ssl.keymanager.algorithm |  |
| ssl.keystore.location |  |
| ssl.keystore.password |  |
| ssl.keystore.type |  |
| ssl.protocol |  |
| ssl.provider |  |
| ssl.trustmanager.algorithm |  |
| ssl.truststore.location |  |
| ssl.truststore.password |  |
| ssl.truststore.type |  |
| zookeeper.clientCnxnSocket |  |
| zookeeper.ssl.client.enable |  |
| zookeeper.ssl.keystore.location |  |
| zookeeper.ssl.keystore.password |  |
| zookeeper.ssl.keystore.type |  |
| zookeeper.ssl.truststore.location |  |
| zookeeper.ssl.truststore.password |  |
| zookeeper.ssl.truststore.type |  |
| alter.config.policy.class.name |  |
| alter.log.dirs.replication.quota.window.num |  |
| alter.log.dirs.replication.quota.window.size.seconds |  |
| authorizer.class.name |  |
| client.quota.callback.class |  |
| connection.failed.authentication.delay.ms |  |
| create.topic.policy.class.name |  |
| delegation.token.expiry.check.interval.ms |  |
| kafka.metrics.polling.interval.secs |  |
| kafka.metrics.reporters |  |
| listener.security.protocol.map |  |
| log.message.downconversion.enable |  |
| metric.reporters |  |
| metrics.num.samples |  |
| metrics.recording.level |  |
| metrics.sample.window.ms |  |
| password.encoder.cipher.algorithm |  |
| password.encoder.iterations |  |
| password.encoder.key.length |  |
| password.encoder.keyfactory.algorithm |  |
| quota.window.num |  |
| quota.window.size.seconds |  |
| replication.quota.window.num |  |
| replication.quota.window.size.seconds |  |
| security.providers |  |
| ssl.endpoint.identification.algorithm |  |
| ssl.engine.factory.class |  |
| ssl.principal.mapping.rules |  |
| ssl.secure.random.implementation |  |
| transaction.abort.timed.out.transaction.cleanup.interval.ms |  |
| transaction.remove.expired.transaction.cleanup.interval.ms |  |
| zookeeper.ssl.cipher.suites |  |
| zookeeper.ssl.crl.enable |  |
| zookeeper.ssl.enabled.protocols |  |
| zookeeper.ssl.endpoint.identification.algorithm |  |
| zookeeper.ssl.ocsp.enable |  |
| zookeeper.ssl.protocol |  |
| zookeeper.sync.time.ms |  |

# 监控

# Connect

## 概述

## 配置

|  |  |
| --- | --- |
| 配置项 | 说明 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# 应用