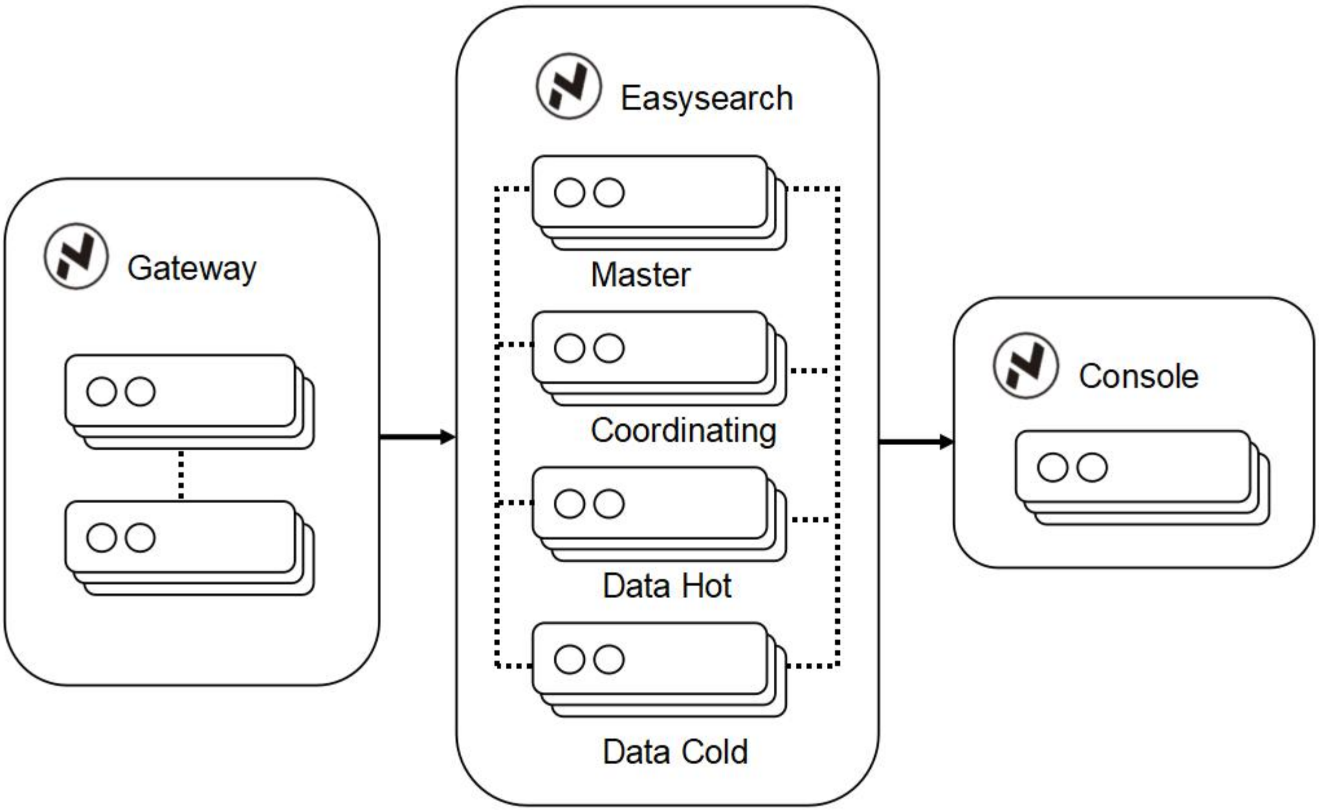
极限数据（北京）科技有限公司

产品安装手册

目录

一、产品架构



下面是生产环境推荐配置。如果是测试环境安装，可进行适当调整。

产品	cpu	内存	磁盘	说明
console	8	16gb	50gb	数据主要存放到 ES 中
Gateway	8	16gb	200gb	硬盘依照网关功能调整大小
Easysearch	16	64gb	1tb	依数据保留时长而定

二、产品安装

INFINI Easysearch

介绍

INFINI Easysearch 是一个分布式的近实时搜索与分析引擎，核心引擎基于开源的 Apache Lucene。Easysearch 的目标是提供一个自主可控的轻量级的 Elasticsearch 可替代版本，并继续完善和支持更多的企业级功能。 与 Elasticsearch 相比，Easysearch 更关注在搜索业务场景的优化和继续保持其产品的简洁与易用性。

Easysearch 的主要特点：

- 兼容 Elasticsearch 语法，业务代码不需要做任何调整；开发团队无缝衔接；
- 兼容 Elasticsearch 现有 API 及索引存储；
- 轻量级（安装包大小仅 50M，部署安装非常简单）；
- 稳定可靠（解决内核泄露、集群卡顿等问题）；
- 企业级安全（身份认证及细颗粒度权限管控）；
- 完善的容灾能力（同时支持基于 CDC 的单向主从复制和基于网关的异地容灾）；
- 企业级管理后台（同时管控多套搜索集群，实现运营标准化、自动化）；
- 信创适配（经过主流国产 CPU/OS 厂家认证）；

Easysearch 核心能力是分布式实时全文数据搜索及分析，以及周边的数据摄入、ETL 以及分析结果的 BI 展示。应用场景 包括日志分析、系统指标分析、安全分析、企业搜索、网站搜索、应用搜索、应用性能管理(APM)等。

安装过程

1. 系统调优

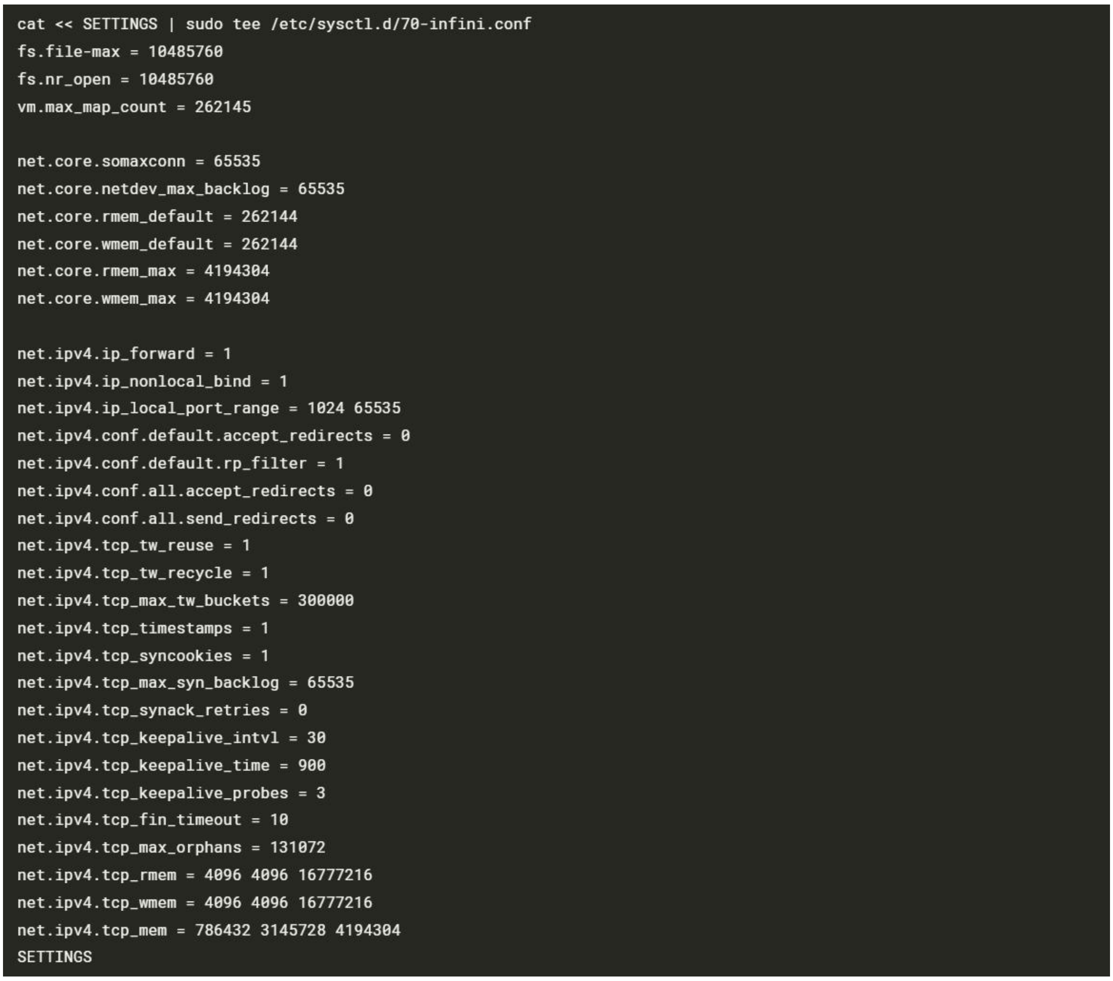
要保证 Easysearch 运行在最佳状态，其所在服务器的操作系统也需要进行相应的调优，以 Linux 为例。

```
sudo tee /etc/security/limits.d/21-infini.conf <<-'EOF'
*                 soft    nofile         1048576
*                 hard    nofile         1048576
*                 soft    memlock        unlimited
*                 hard    memlock        unlimited
root              soft    nofile         1048576
root              hard    nofile         1048576
root              soft    memlock        unlimited
root              hard    memlock        unlimited
EOF
```

```
sudo tee /etc/security/limits.d/21-infini.conf <<-’EOF’
*                 soft    nofile         1048576
*                 hard    nofile         1048576
*                 soft    memlock        unlimited
*                 hard    memlock        unlimited
root              soft    nofile         1048576
root              hard    nofile         1048576
root              soft    memlock        unlimited
root              hard    memlock        unlimited
EOF
```

2. 内核调优

```
cat << SETTINGS | sudo tee /etc/sysctl.d/70-infini.conf
fs.file-max = 10485760
fs.nr_open = 10485760
vm.max_map_count = 262145

net.core.somaxconn = 65535
net.core.netdev_max_backlog = 65535
net.core.rmem_default = 262144
net.core.wmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_max = 4194304

net.ipv4.ip_forward = 1
net.ipv4.ip_nonlocal_bind = 1
net.ipv4.ip_local_port_range = 1024 65535
net.ipv4.conf.default.accept_redirects = 0
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.all.send_redirects = 0
net.ipv4.tcp_tw_reuse = 1
net.ipv4.tcp_tw_recycle = 1
net.ipv4.tcp_max_tw_buckets = 300000
net.ipv4.tcp_timestamps = 1
net.ipv4.tcp_syncookies = 1
net.ipv4.tcp_max_syn_backlog = 65535
net.ipv4.tcp_synack_retries = 0
net.ipv4.tcp_keepalive_intvl = 30
net.ipv4.tcp_keepalive_time = 900
net.ipv4.tcp_keepalive_probes = 3
net.ipv4.tcp_fin_timeout = 10
net.ipv4.tcp_max_orphans = 131072
net.ipv4.tcp_rmem = 4096 4096 16777216
net.ipv4.tcp_wmem = 4096 4096 16777216
net.ipv4.tcp_mem = 786432 3145728 4194304
SETTINGS
#执行下面的命令验证配置参数是否合法
sysctl -p /etc/sysctl.d/70-infini.conf
```



```
cat << SETTINGS | sudo tee /etc/sysctl.d/70-infini.conf
fs.file-max = 10485760
fs.nr_open = 10485760
vm.max_map_count = 262145

net.core.somaxconn = 65535
net.core.netdev_max_backlog = 65535
net.core.rmem_default = 262144
net.core.wmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_max = 4194304

net.ipv4.ip_forward = 1
net.ipv4.ip_nonlocal_bind = 1
net.ipv4.ip_local_port_range = 1024 65535
net.ipv4.conf.default.accept_redirects = 0
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.all.send_redirects = 0
net.ipv4.tcp_tw_reuse = 1
net.ipv4.tcp_tw_recycle = 1
net.ipv4.tcp_max_tw_buckets = 300000
net.ipv4.tcp_timestamps = 1
net.ipv4.tcp_syncookies = 1
net.ipv4.tcp_max_syn_backlog = 65535
net.ipv4.tcp_synack_retries = 0
net.ipv4.tcp_keepalive_intvl = 30
net.ipv4.tcp_keepalive_time = 900
net.ipv4.tcp_keepalive_probes = 3
net.ipv4.tcp_fin_timeout = 10
net.ipv4.tcp_max_orphans = 131072
net.ipv4.tcp_rmem = 4096 4096 16777216
net.ipv4.tcp_wmem = 4096 4096 16777216
net.ipv4.tcp_mem = 786432 3145728 4194304
SETTINGS
```

3. 用户配置

```
#在各个节点上分别操作
groupadd -g 602 infini
useradd -u 602 -g infini -m -d /home/infini -c "Easysearch user" -s /bin/bash infini
```

4. 部署 Easysearch

```
# 在各个节点上分别操作
# 下载主机平台最新的安装包
https://release.infinilabs.com/easysearch/stable/bundle/

# 解压
mkdir -p /data/easysearch
tar -zxf /usr/src/easysearch*.tar.gz -C /data/easysearch
```

```
# 初始化，生成证书和初始密码
# 初始密码在 log 目录下 initialize.log

cd /data/easysearch
./bin/initialize.sh

# 多节点部署，将证书及用户密码配置文件同步至其他节点上
cd config
scp *.crt *.key ip:/data/easysearch/config
```

5. Easysearch 配置文件

```
# 修改参数
# 集群名称，随意修改
cluster.name: infini-easysearch
# 节点名称，各节点唯一
node.name: node-1
# 监听地址
network.host: 0.0.0.0
# 发现配置
discovery.seed_hosts: ["ip1:9300", "ip2:9300"]
cluster.initial_master_nodes: ["ip1:9300"]
# elastic 兼容
elasticsearch.api_compatibility: true
```

6. 启动服务

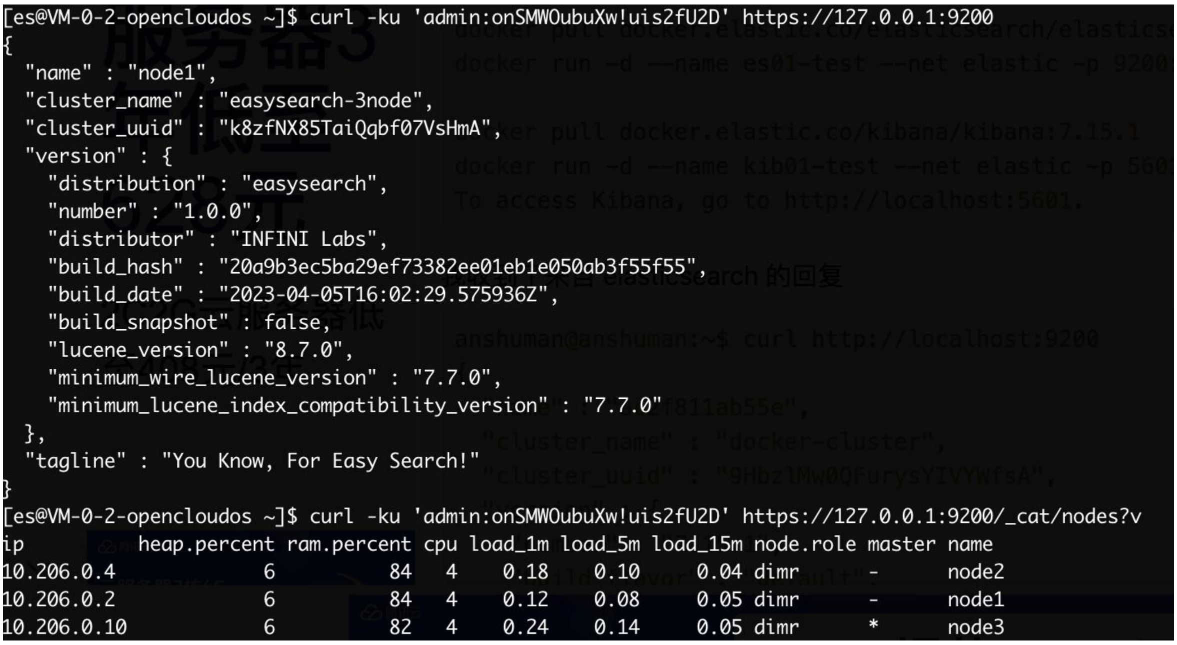
```
#在各个节点上分别操作
#根据实际机器内存的大小进行配置，推荐配置为机器内存一半，且不超过 31G
sed -i "s/1g/4g/g" $ES_HOME/config/jvm.options

chown -R infini. /data/easysearch

su - infini
grep -wq easysearch ~/.bashrc || cat <<EOF >> ~/.bashrc
export ES_HOME=/data/easysearch
```

```
EOF
source ~/.bashrc

$ES_HOME/bin/easysearch -d
```

7. 集群验证

验证命令如下：

```
curl -ku 'admin:changMe' https://10.206.0.2:9200
curl -ku 'admin:changMe' https://10.206.0.2:9200/_cat/nodes?v
```



```
[es@VM-0-2-opencloudos ~]$ curl -ku 'admin:onSMWOubuXw!uis2fU2D' https://127.0.0.1:9200
{
  "name" : "node1",
  "cluster_name" : "easysearch-3node",
  "cluster_uuid" : "k8zfNX85TaiQqbf07VsHmA",
  "version" : {
    "distribution" : "easysearch",
    "number" : "1.0.0",
    "distributor" : "INFINI Labs",
    "build_hash" : "20a9b3ec5ba29ef73382ee01eb1e050ab3f55f55",
    "build_date" : "2023-04-05T16:02:29.575936Z",
    "build_snapshot" : false,
    "lucene_version" : "8.7.0",
    "minimum_wire_lucene_version" : "7.7.0",
    "minimum_lucene_index_compatibility_version" : "7.7.0"
  },
  "tagline" : "You Know, For Easy Search!"
}
[es@VM-0-2-opencloudos ~]$ curl -ku 'admin:onSMWOubuXw!uis2fU2D' https://127.0.0.1:9200/_cat/nodes?v
ip          heap.percent ram.percent cpu load_1m load_5m load_15m node.role master name
10.206.0.4             6          84   4    0.18    0.10     0.04 dimr      -      node2
10.206.0.2             6          84   4    0.12    0.08     0.05 dimr      -      node1
10.206.0.10            6          82   4    0.24    0.14     0.05 dimr      *      node3
```

所有节点都成功加入集群，则安装完毕。

INFINI Console

介绍

INFINI Console 是一款非常轻量级的多集群、跨版本的搜索基础设施统一管控平台。通过对流行的搜索引擎基础设施进行跨版本、多集群的集中纳管， 企业可以快速方便的统一管理企业内部的不同版本的多套搜索集群。 INFINI

Console 还可以对集群内的索引及数据进行操作管理，可以配置灵活的告警规则，可以指定统一的安全策略，可以查看各个维度的日志和审计信息，真正实现企业级的搜索服务平台化建设和运营。

Console 的主要特点：

- 支持多集群管理，可以在一个平台内同时纳管任意多套集群；
- 多版本 Elasticsearch 支持，支持 1.x、2.x、5.x、6.x、7.x、8.x；
- 支持以项目为单位来分组管理集群的元数据信息、支持标签；
- 支持动态注册添加集群，目标集群无需任何变动即可被接入管理；
- 支持统一的多集群层面、索引和 API 接口粒度的权限控制；
- 支持统一的跨集群的告警引擎，灵活配置基于阈值的告警规则；
- 支持查看集群元数据的历史变更信息，用于审计、追踪集群变化；
- 开发者工具支持多个工作区快速切换，支持智能提示，支持常用命令保存和快捷加载；
- 支持任意版本的集群监控，包括集群、节点、索引等详细维度的指标查看和分析；
- 支持索引的管理操作，支持索引的快速查看浏览，支持索引内文档的更新、删除；
- 支持创建索引数据视图，可以修改字段的展示格式，支持时序索引数据的快速查看；

1. 系统调优

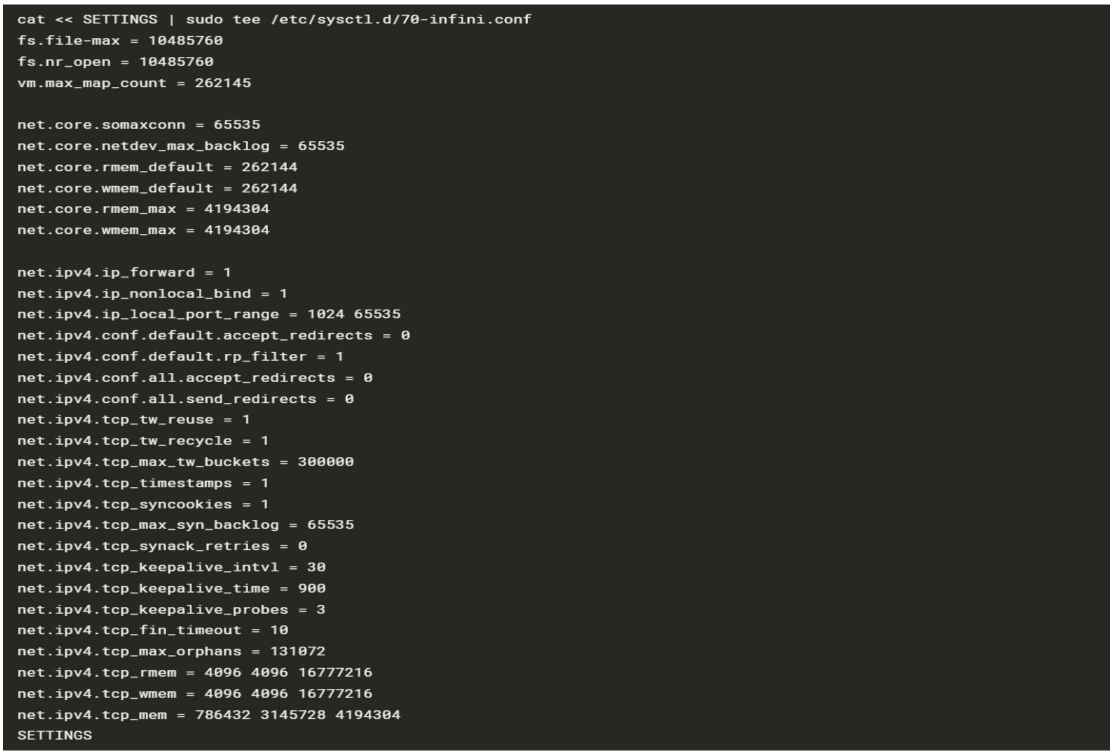
要保证 Console 运行在最佳状态，其所在服务器的操作系统也需要进行相应的调优，以 Linux 为例。

```
sudo tee /etc/security/limits.d/21-infini.conf <<-'EOF'
*                soft    nofile         1048576
*                hard    nofile         1048576
*                soft    memlock        unlimited
*                hard    memlock        unlimited
root             soft    nofile         1048576
root             hard    nofile         1048576
root             soft    memlock        unlimited
root             hard    memlock        unlimited
EOF
```

```
sudo tee /etc/security/limits.d/21-infini.conf <<-’EOF’
*                soft    nofile         1048576
*                hard    nofile         1048576
*                soft    memlock        unlimited
*                hard    memlock        unlimited
root             soft    nofile         1048576
root             hard    nofile         1048576
root             soft    memlock        unlimited
root             hard    memlock        unlimited
EOF
```

2. 内核调优

```
cat << SETTINGS | sudo tee /etc/sysctl.d/70-infini.conf
fs.file-max = 10485760
fs.nr_open = 10485760
vm.max_map_count = 262145

net.core.somaxconn = 65535
net.core.netdev_max_backlog = 65535
net.core.rmem_default = 262144
net.core.wmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_max = 4194304

net.ipv4.ip_forward = 1
net.ipv4.ip_nonlocal_bind = 1
net.ipv4.ip_local_port_range = 1024 65535
net.ipv4.conf.default.accept_redirects = 0
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.all.send_redirects = 0
net.ipv4.tcp_tw_reuse = 1
net.ipv4.tcp_tw_recycle = 1
net.ipv4.tcp_max_tw_buckets = 300000
net.ipv4.tcp_timestamps = 1
net.ipv4.tcp_syncookies = 1
net.ipv4.tcp_max_syn_backlog = 65535
net.ipv4.tcp_synack_retries = 0
net.ipv4.tcp_keepalive_intvl = 30
net.ipv4.tcp_keepalive_time = 900
net.ipv4.tcp_keepalive_probes = 3
net.ipv4.tcp_fin_timeout = 10
net.ipv4.tcp_max_orphans = 131072
net.ipv4.tcp_rmem = 4096 4096 16777216
net.ipv4.tcp_wmem = 4096 4096 16777216
net.ipv4.tcp_mem = 786432 3145728 4194304
SETTINGS

#执行下面的命令验证配置参数是否合法

sysctl -p /etc/sysctl.d/70-infini.conf
```



```
cat << SETTINGS | sudo tee /etc/sysctl.d/70-infini.conf
fs.file-max = 10485760
fs.nr_open = 10485760
vm.max_map_count = 262145

net.core.somaxconn = 65535
net.core.netdev_max_backlog = 65535
net.core.rmem_default = 262144
net.core.wmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_max = 4194304

net.ipv4.ip_forward = 1
net.ipv4.ip_nonlocal_bind = 1
net.ipv4.ip_local_port_range = 1024 65535
net.ipv4.conf.default.accept_redirects = 0
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.all.send_redirects = 0
net.ipv4.tcp_tw_reuse = 1
net.ipv4.tcp_tw_recycle = 1
net.ipv4.tcp_max_tw_buckets = 300000
net.ipv4.tcp_timestamps = 1
net.ipv4.tcp_syncookies = 1
net.ipv4.tcp_max_syn_backlog = 65535
net.ipv4.tcp_synack_retries = 0
net.ipv4.tcp_keepalive_intvl = 30
net.ipv4.tcp_keepalive_time = 900
net.ipv4.tcp_keepalive_probes = 3
net.ipv4.tcp_fin_timeout = 10
net.ipv4.tcp_max_orphans = 131072
net.ipv4.tcp_rmem = 4096 4096 16777216
net.ipv4.tcp_wmem = 4096 4096 16777216
net.ipv4.tcp_mem = 786432 3145728 4194304
SETTINGS
```

3. 用户配置

```
#在各个节点上分别操作
roupadd -g 602 infini
```

```
useradd -u 602 -g infini -m -d /home/infini -c "Easysearch user" -s /bin/bash infini
```

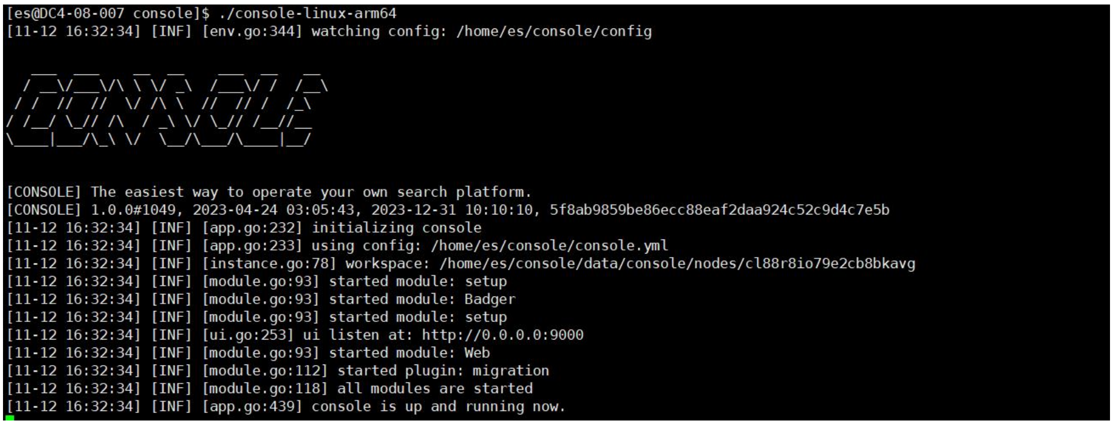
4. 部署 Console

建立一个目录，将软件包解压到指定目录

```
mkdir console-1.10.0
tar -xzf console-1.10.0-1301-linux-amd64.tar.gz -C console-1.10.0
```

前台运行

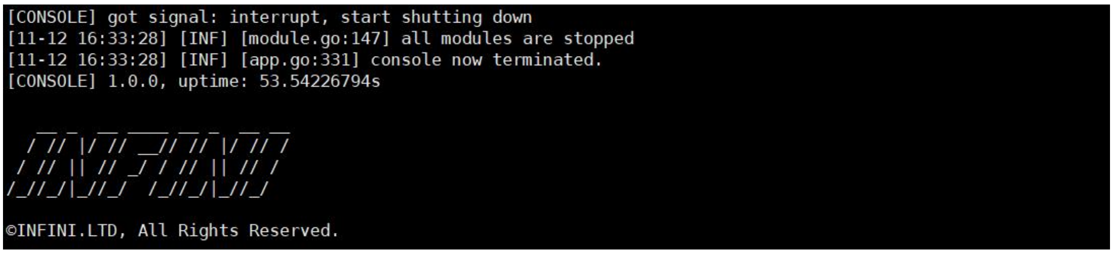
解压完成，直接运行程序即可启动 INFINI Console 了(这里使用的是 linux 版本的，不同平台的程序文件名称略有不同)，如下：



```
[es@DC4-08-007 console]$ ./console-linux-arm64
[11-12 16:32:34] [INF] [env.go:344] watching config: /home/es/console/config

   ___  ___   __  __   ___  __    __
  / __\/___\/\ \ \/ _\  /___\/ /   /__\
 / /  //  //  \/ /\ \  //  // /   /_\
/ /__/ \_// /\  / _\ \/ \_// /___//__
\____|___/\_\ \/  \__/\___/\____|__/

[CONSOLE] The easiest way to operate your own search platform.
[CONSOLE] 1.0.0#1049, 2023-04-24 03:05:43, 2023-12-31 10:10:10, 5f8ab9859be86ecc88eaf2daa924c52c9d4c7e5b
[11-12 16:32:34] [INF] [app.go:232] initializing console
[11-12 16:32:34] [INF] [app.go:233] using config: /home/es/console/console.yml
[11-12 16:32:34] [INF] [instance.go:78] workspace: /home/es/console/data/console/nodes/cl88r8io79e2cb8bkavg
[11-12 16:32:34] [INF] [module.go:93] started module: setup
[11-12 16:32:34] [INF] [module.go:93] started module: Badger
[11-12 16:32:34] [INF] [module.go:93] started module: setup
[11-12 16:32:34] [INF] [ui.go:253] ui listen at: http://0.0.0.0:9000
[11-12 16:32:34] [INF] [module.go:93] started module: Web
[11-12 16:32:34] [INF] [module.go:112] started plugin: migration
[11-12 16:32:34] [INF] [module.go:118] all modules are started
[11-12 16:32:34] [INF] [app.go:439] console is up and running now.
```

停止 INFINI Console

如果需要停止 INFINI Console，按 Ctrl+C 即可停止 INFINI Console 平台。

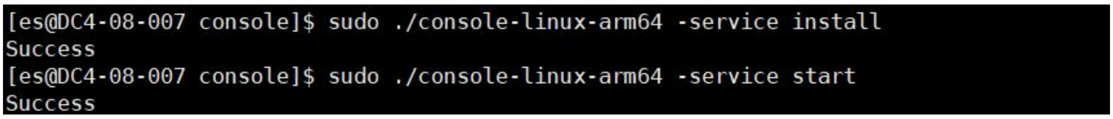


```
[CONSOLE] got signal: interrupt, start shutting down
[11-12 16:33:28] [INF] [module.go:147] all modules are stopped
[11-12 16:33:28] [INF] [app.go:331] console now terminated.
[CONSOLE] 1.0.0, uptime: 53.54226794s

   __ _  __ _____ __ _  __ __
  / // |/ // __// // |/ // /
 / // || // _/ / // || // /
/_//_/|_//_/  /_//_/|_//_/

©INFINI.LTD, All Rights Reserved.
```

配置服务后台运行

如果希望将 INFINI Console 以后台服务任务的方式运行，如下：

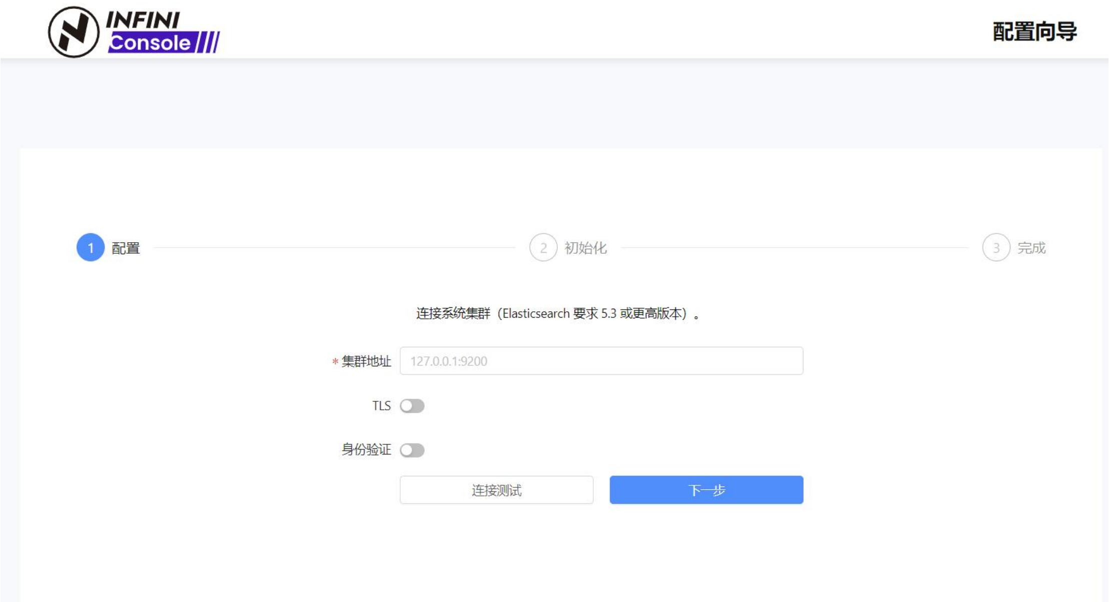
INFINI Console 成功运行后，会监听 9000 端口。



```
[es@DC4-08-007 console]$ sudo ./console-linux-arm64 -service install
Success
[es@DC4-08-007 console]$ sudo ./console-linux-arm64 -service start
Success
```

初始化

访问 http://ip:9000 进行初始化



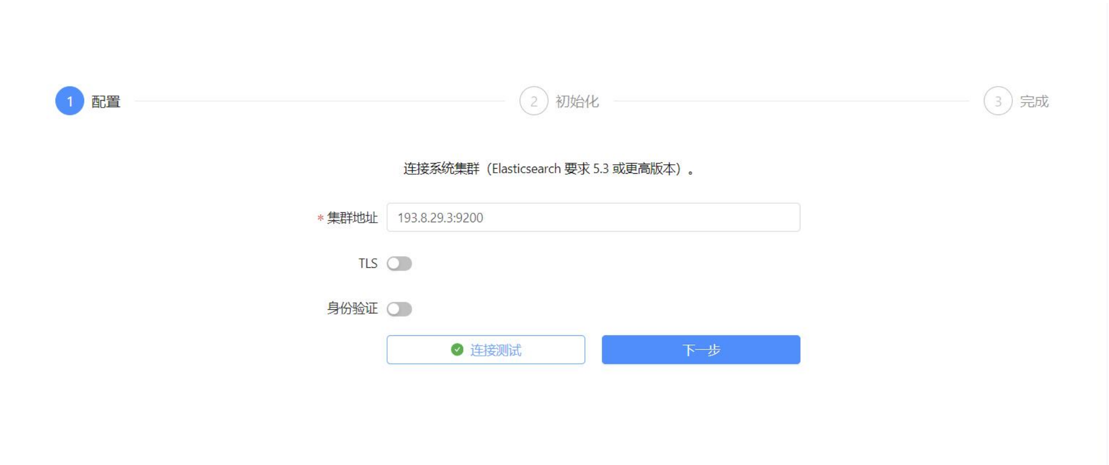
连接系统集群（Elasticsearch 要求 5.3 或更高版本）。

* 集群地址 127.0.0.1:9200

TLS ○

身份验证 ○

填写后端 ip:port，根据实际情况处理 TLS 及身份验证选择选项。



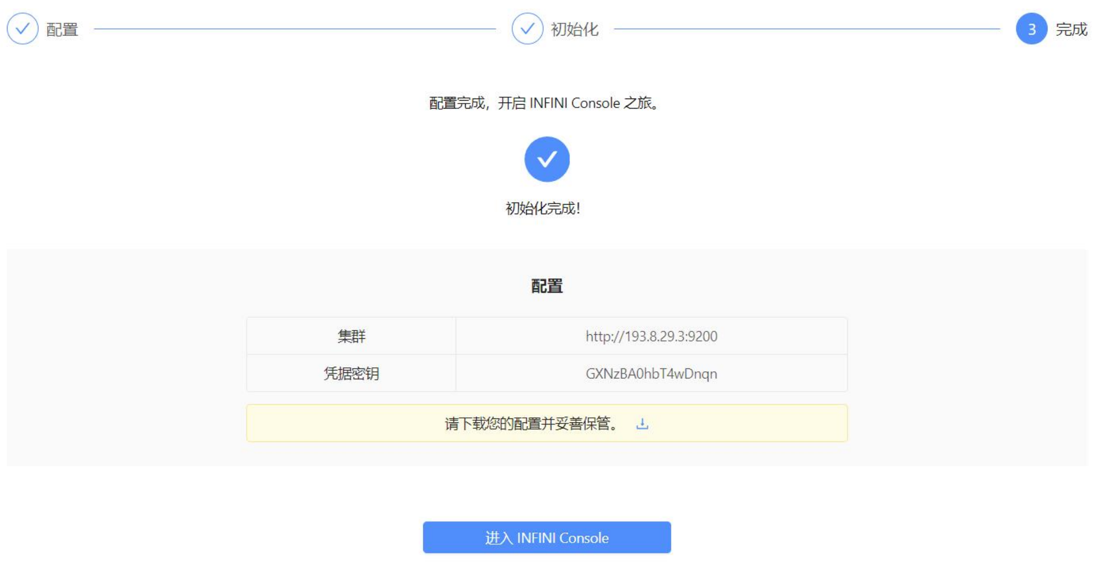
连接系统集群（Elasticsearch 要求 5.3 或更高版本）。

* 集群地址 193.8.29.3:9200

TLS ○

身份验证 ○

下一步



配置完成，开启 INFINI Console 之旅。

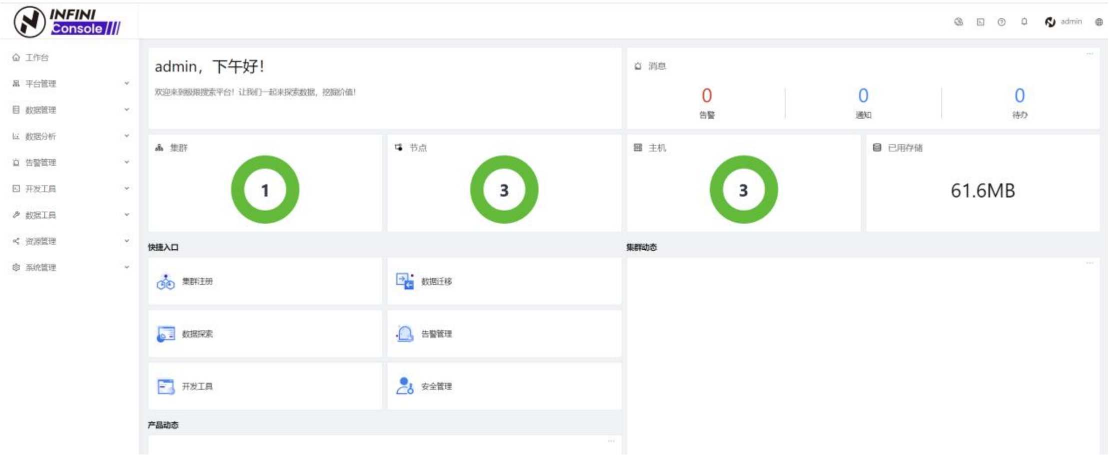
初始化完成!

配置

集群	http://193.8.29.3:9200
凭据密钥	GXNzBA0hbT4wDnqn

请下载您的配置并妥善保管。

初始化完毕



admin，下午好!

欢迎来到极限搜索平台！让我们一起来探索数据，挖掘价值！

INFINI Gateway

介绍

INFINI Gateway 是一个面向搜索场景的高性能数据网关，所有请求都经过网关处理后再转发到后端的搜索业务集群。基于 INFINI Gateway， 可以实现索引级别的限速限流、常见查询的缓存加速、查询请求的审计、查询结果的动态修改等等。

Gateway 特性

- 高可用，不停机索引，自动处理后端 Elasticsearch 的故障，不影响数据的正常摄取
- 写入加速，可自动合并独立的索引请求为批量请求，降低后端压力，提高索引效率
- 查询加速，可配置查询缓存，Kibana 分析仪表板的无缝智能加速，全面提升搜索体验
- 透明重试，自动处理后端 Elasticsearch 节点故障和对查询请求进行迁移重试
- 流量克隆，支持复制流量到多个不同的后端 Elasticsearch 集群，支持流量灰度迁移
- 一键重建，优化过的高速重建和增量数据的自动处理，支持新旧索引的透明无缝切换
- 安全传输，自动支持 TLS/HTTPS，可动态生成自签证书，也可指定自签可信证书
- 精准路由，多种算法的负载均衡模式，索引和查询可分别配置负载路由策略，动态灵活
- 限速限流，支持多种限速和限流测规则，可以实现索引级别的限速，保障后端集群的稳定性
- 并发控制，支持集群和节点级别的 TCP 并发连接数控制，保障后端集群和节点稳定性

- 无单点故障，内置基于虚拟 IP 的高可用解决方案，双机热备，故障自动迁移，避免单点故障

1. 系统调优

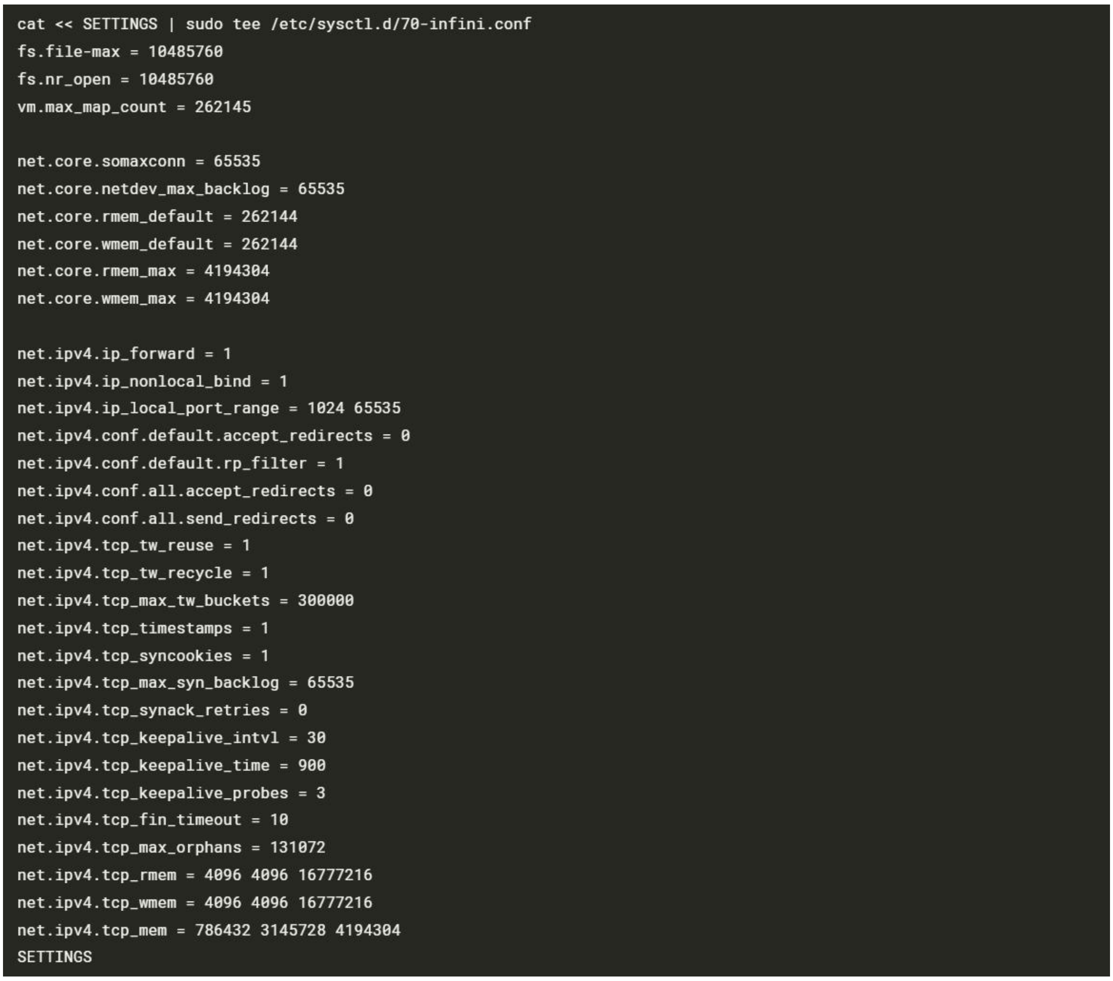
要保证 Gateway 运行在最佳状态，其所在服务器的操作系统也需要进行相应的调优，以 Linux 为例。

```
sudo tee /etc/security/limits.d/21-infini.conf <<-'EOF'
*                soft    nofile         1048576
*                hard    nofile         1048576
*                soft    memlock        unlimited
*                hard    memlock        unlimited
root             soft    nofile         1048576
root             hard    nofile         1048576
root             soft    memlock        unlimited
root             hard    memlock        unlimited
EOF
```

```
sudo tee /etc/security/limits.d/21-infini.conf <<-’EOF’
*                soft    nofile         1048576
*                hard    nofile         1048576
*                soft    memlock        unlimited
*                hard    memlock        unlimited
root             soft    nofile         1048576
root             hard    nofile         1048576
root             soft    memlock        unlimited
root             hard    memlock        unlimited
EOF
```

2. 内核调优

```
cat << SETTINGS | sudo tee /etc/sysctl.d/70-infini.conf
fs.file-max = 10485760
fs.nr_open = 10485760
vm.max_map_count = 262145

net.core.somaxconn = 65535
net.core.netdev_max_backlog = 65535
net.core.rmem_default = 262144
net.core.wmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_max = 4194304

net.ipv4.ip_forward = 1
net.ipv4.ip_nonlocal_bind = 1
net.ipv4.ip_local_port_range = 1024 65535
net.ipv4.conf.default.accept_redirects = 0
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.all.send_redirects = 0
net.ipv4.tcp_tw_reuse = 1
net.ipv4.tcp_tw_recycle = 1
net.ipv4.tcp_max_tw_buckets = 300000
net.ipv4.tcp_timestamps = 1
net.ipv4.tcp_syncookies = 1
net.ipv4.tcp_max_syn_backlog = 65535
net.ipv4.tcp_synack_retries = 0
net.ipv4.tcp_keepalive_intvl = 30
net.ipv4.tcp_keepalive_time = 900
net.ipv4.tcp_keepalive_probes = 3
net.ipv4.tcp_fin_timeout = 10
net.ipv4.tcp_max_orphans = 131072
net.ipv4.tcp_rmem = 4096 4096 16777216
net.ipv4.tcp_wmem = 4096 4096 16777216
net.ipv4.tcp_mem = 786432 3145728 4194304
SETTINGS

#执行下面的命令验证配置参数是否合法

sysctl -p /etc/sysctl.d/70-infini.conf
```



```
cat << SETTINGS | sudo tee /etc/sysctl.d/70-infini.conf
fs.file-max = 10485760
fs.nr_open = 10485760
vm.max_map_count = 262145

net.core.somaxconn = 65535
net.core.netdev_max_backlog = 65535
net.core.rmem_default = 262144
net.core.wmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_max = 4194304

net.ipv4.ip_forward = 1
net.ipv4.ip_nonlocal_bind = 1
net.ipv4.ip_local_port_range = 1024 65535
net.ipv4.conf.default.accept_redirects = 0
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.all.send_redirects = 0
net.ipv4.tcp_tw_reuse = 1
net.ipv4.tcp_tw_recycle = 1
net.ipv4.tcp_max_tw_buckets = 300000
net.ipv4.tcp_timestamps = 1
net.ipv4.tcp_syncookies = 1
net.ipv4.tcp_max_syn_backlog = 65535
net.ipv4.tcp_synack_retries = 0
net.ipv4.tcp_keepalive_intvl = 30
net.ipv4.tcp_keepalive_time = 900
net.ipv4.tcp_keepalive_probes = 3
net.ipv4.tcp_fin_timeout = 10
net.ipv4.tcp_max_orphans = 131072
net.ipv4.tcp_rmem = 4096 4096 16777216
net.ipv4.tcp_wmem = 4096 4096 16777216
net.ipv4.tcp_mem = 786432 3145728 4194304
SETTINGS
```

3. 用户配置

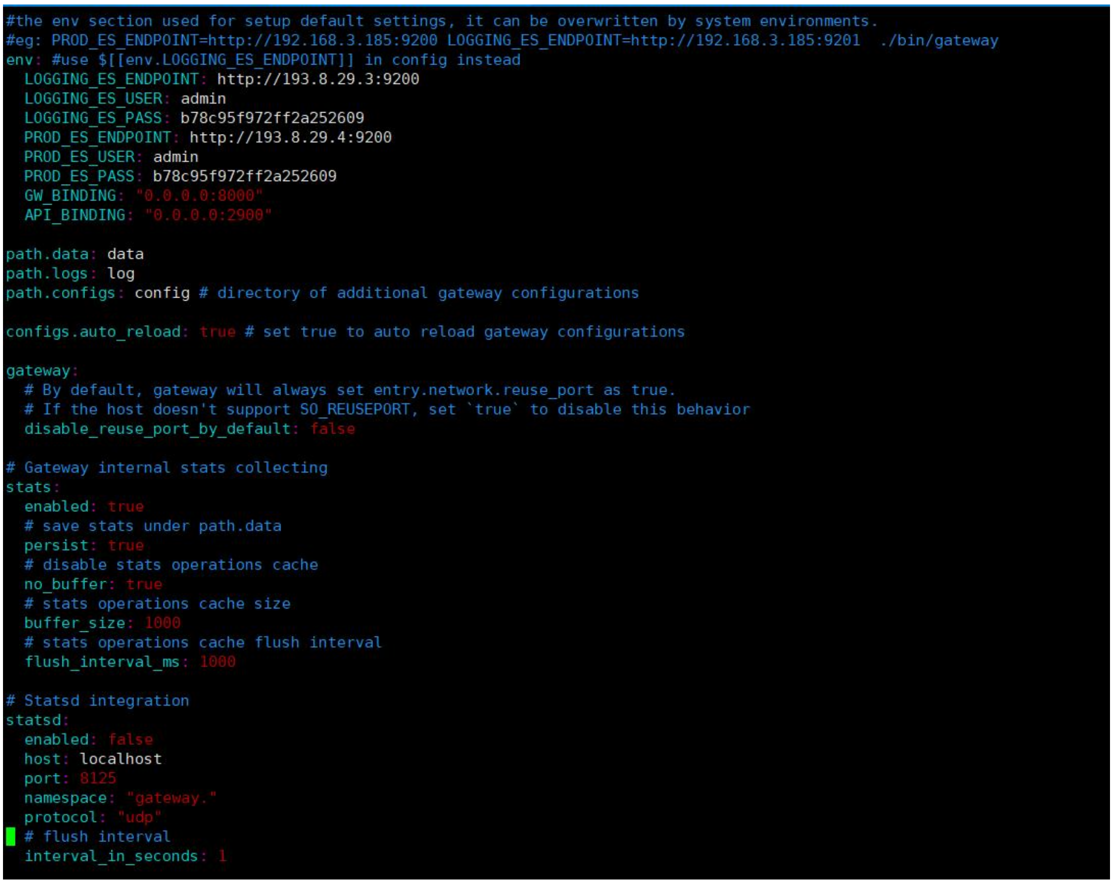
```
#在各个节点上分别操作
roupadd -g 602 infini
useradd -u 602 -g infini -m -d /home/infini -c "Easysearch user" -s /bin/bash infini
```

4. 部署 Gateway

建立一个目录，将软件包解压到指定目录

```
mkdir gateway-1.18.0
tar -xzf gateway-1.18.0-1301-linux-amd64.tar.gz -C gateway-1.18.0
```

修改配置文件



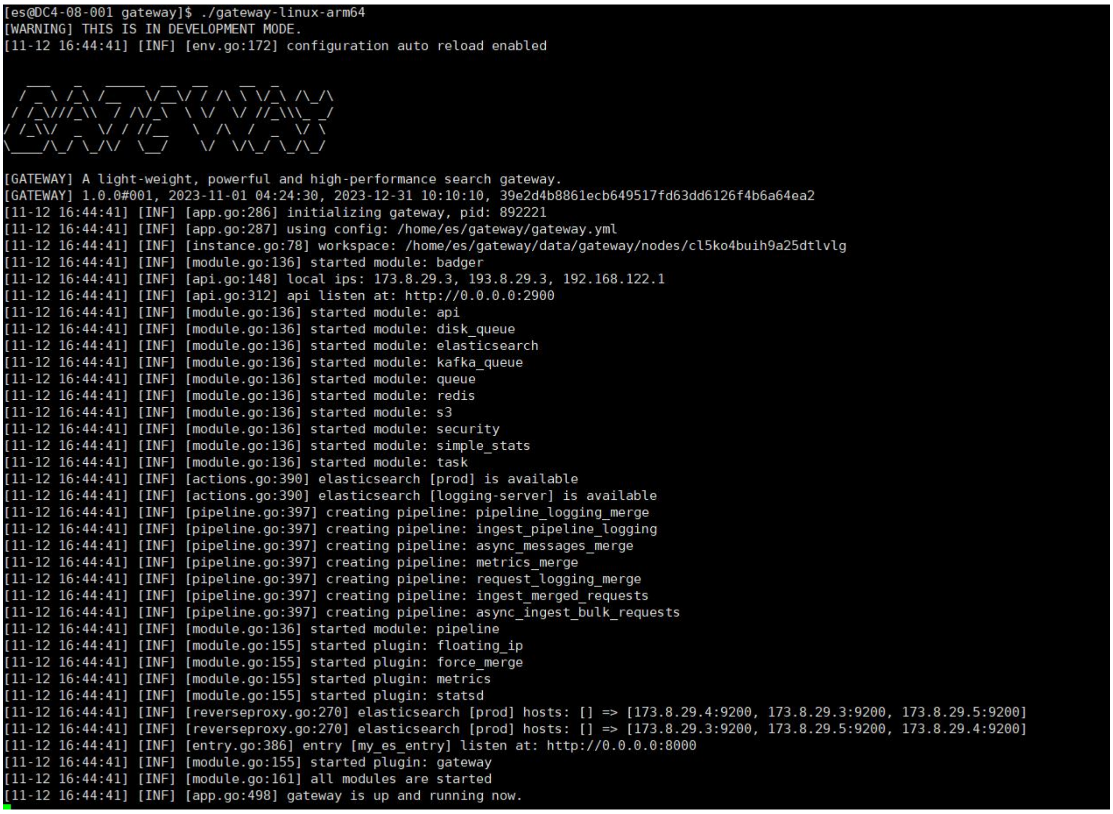
```
#the env section used for setup default settings, it can be overwritten by system environments.
#eg: PROD_ES_ENDPOINT=http://192.168.3.185:9200 LOGGING_ES_ENDPOINT=http://192.168.3.185:9201  ./bin/gateway
env: #use $[[env.LOGGING_ES_ENDPOINT]] in config instead
  LOGGING_ES_ENDPOINT: http://193.8.29.3:9200
  LOGGING_ES_USER: admin
  LOGGING_ES_PASS: b78c95f972ff2a252609
  PROD_ES_ENDPOINT: http://193.8.29.4:9200
  PROD_ES_USER: admin
  PROD_ES_PASS: b78c95f972ff2a252609
  GW_BINDING: "0.0.0.0:8000"
  API_BINDING: "0.0.0.0:2900"

path.data: data
path.logs: log
path.configs: config # directory of additional gateway configurations

configs.auto_reload: true # set true to auto reload gateway configurations

gateway:
  # By default, gateway will always set entry.network.reuse_port as true.
  # If the host doesn't support SO_REUSEPORT, set `true` to disable this behavior
  disable_reuse_port_by_default: false

# Gateway internal stats collecting
stats:
  enabled: true
  # save stats under path.data
  persist: true
  # disable stats operations cache
  no_buffer: true
  # stats operations cache size
  buffer_size: 1000
  # stats operations cache flush interval
  flush_interval_ms: 1000

# Statsd integration
statsd:
  enabled: false
  host: localhost
  port: 8125
  namespace: "gateway."
  protocol: "udp"
  # flush interval
  interval_in_seconds: 1
```

根据实际情况修改以下参数：

- LOGGING_ES_ENDPOINT 定义日志集群的访问信息，所有请求记录将写入该集群。
- PROD_ES_ENDPOINT 定义生产集群的访问信息，网关将代理此集群。
- *_ES_USER 和*_ES_PASS 定义集群的认证信息。
- API_BINDING 定义网关 API 服务监听的地址和端口。
- GW_BINDING 定义网关代理服务监听的地址和端口。

前台运行

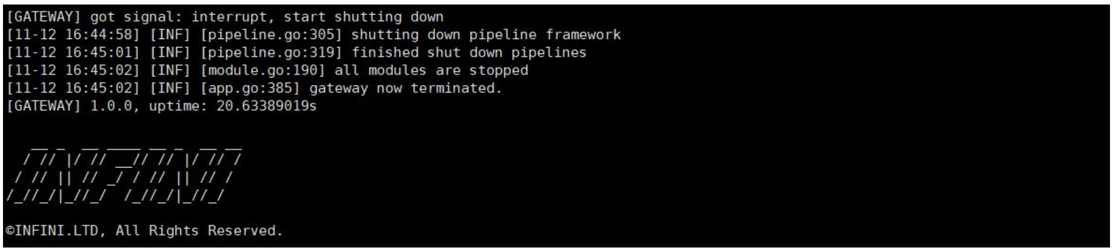
解压完成，直接运行程序即可启动 INFINI Gateway 了(这里使用的是 linux 版本的，不同平台的程序文件名称略有不同)，如下：



```
[es@DC4-08-001 gateway]$ ./gateway-linux-arm64
[WARNING] THIS IS IN DEVELOPMENT MODE.
[11-12 16:44:41] [INF] [env.go:172] configuration auto reload enabled

   ___   _   _____  __  __    __  _
  / _ \ /_\ /__   \/__\/ / /\ \ \/_\ /\_/\
 / /_\///_\\  / /\/_\  \ \/  \/ //_\\\_ _/
/ /_\\/  _  \/ / //__   \  /\  /  _  \/ \
\____/\_/ \_/\/  \__/    \/  \/\_/ \_/\_/

[GATEWAY] A light-weight, powerful and high-performance search gateway.
[GATEWAY] 1.0.0#001, 2023-11-01 04:24:30, 2023-12-31 10:10:10, 39e2d4b8861ecb649517fd63dd6126f4b6a64ea2
[11-12 16:44:41] [INF] [app.go:286] initializing gateway, pid: 892221
[11-12 16:44:41] [INF] [app.go:287] using config: /home/es/gateway/gateway.yml
[11-12 16:44:41] [INF] [instance.go:78] workspace: /home/es/gateway/data/gateway/nodes/cl5ko4buih9a25dtlvlg
[11-12 16:44:41] [INF] [module.go:136] started module: badger
[11-12 16:44:41] [INF] [api.go:148] local ips: 173.8.29.3, 193.8.29.3, 192.168.122.1
[11-12 16:44:41] [INF] [api.go:312] api listen at: http://0.0.0.0:2900
[11-12 16:44:41] [INF] [module.go:136] started module: api
[11-12 16:44:41] [INF] [module.go:136] started module: disk_queue
[11-12 16:44:41] [INF] [module.go:136] started module: elasticsearch
[11-12 16:44:41] [INF] [module.go:136] started module: kafka_queue
[11-12 16:44:41] [INF] [module.go:136] started module: queue
[11-12 16:44:41] [INF] [module.go:136] started module: redis
[11-12 16:44:41] [INF] [module.go:136] started module: s3
[11-12 16:44:41] [INF] [module.go:136] started module: security
[11-12 16:44:41] [INF] [module.go:136] started module: simple_stats
[11-12 16:44:41] [INF] [module.go:136] started module: task
[11-12 16:44:41] [INF] [actions.go:390] elasticsearch [prod] is available
[11-12 16:44:41] [INF] [actions.go:390] elasticsearch [logging-server] is available
[11-12 16:44:41] [INF] [pipeline.go:397] creating pipeline: pipeline_logging_merge
[11-12 16:44:41] [INF] [pipeline.go:397] creating pipeline: ingest_pipeline_logging
[11-12 16:44:41] [INF] [pipeline.go:397] creating pipeline: async_messages_merge
[11-12 16:44:41] [INF] [pipeline.go:397] creating pipeline: metrics_merge
[11-12 16:44:41] [INF] [pipeline.go:397] creating pipeline: request_logging_merge
[11-12 16:44:41] [INF] [pipeline.go:397] creating pipeline: ingest_merged_requests
[11-12 16:44:41] [INF] [pipeline.go:397] creating pipeline: async_ingest_bulk_requests
[11-12 16:44:41] [INF] [module.go:136] started module: pipeline
[11-12 16:44:41] [INF] [module.go:155] started plugin: floating_ip
[11-12 16:44:41] [INF] [module.go:155] started plugin: force_merge
[11-12 16:44:41] [INF] [module.go:155] started plugin: metrics
[11-12 16:44:41] [INF] [module.go:155] started plugin: statsd
[11-12 16:44:41] [INF] [reverseproxy.go:270] elasticsearch [prod] hosts: [] => [173.8.29.4:9200, 173.8.29.3:9200, 173.8.29.5:9200]
[11-12 16:44:41] [INF] [reverseproxy.go:270] elasticsearch [prod] hosts: [] => [173.8.29.3:9200, 173.8.29.5:9200, 173.8.29.4:9200]
[11-12 16:44:41] [INF] [entry.go:386] entry [my_es_entry] listen at: http://0.0.0.0:8000
[11-12 16:44:41] [INF] [module.go:155] started plugin: gateway
[11-12 16:44:41] [INF] [module.go:161] all modules are started
[11-12 16:44:41] [INF] [app.go:498] gateway is up and running now.
```

停止 INFINI Gateway

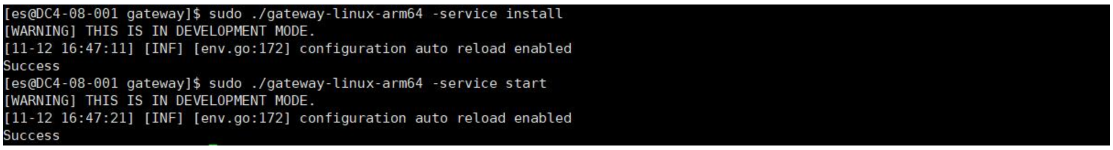
如果需要停止 INFINI Gateway，按 Ctrl+C 即可停止 INFINI Gateway。



```
[GATEWAY] got signal: interrupt, start shutting down
[11-12 16:44:58] [INF] [pipeline.go:305] shutting down pipeline framework
[11-12 16:45:01] [INF] [pipeline.go:319] finished shut down pipelines
[11-12 16:45:02] [INF] [module.go:190] all modules are stopped
[11-12 16:45:02] [INF] [app.go:385] gateway now terminated.
[GATEWAY] 1.0.0, uptime: 20.63389019s

   __ _  __ ____ __ _  __ __
  / // |/ // __// // |/ // /
 / // || // _/ / // || // /
/_//_/|_//_/  /_//_/|_//_/

©INFINI.LTD, All Rights Reserved.
```

配置服务后台运行

如果希望将 INFINI Gateway 以后台服务任务的方式运行，如下：

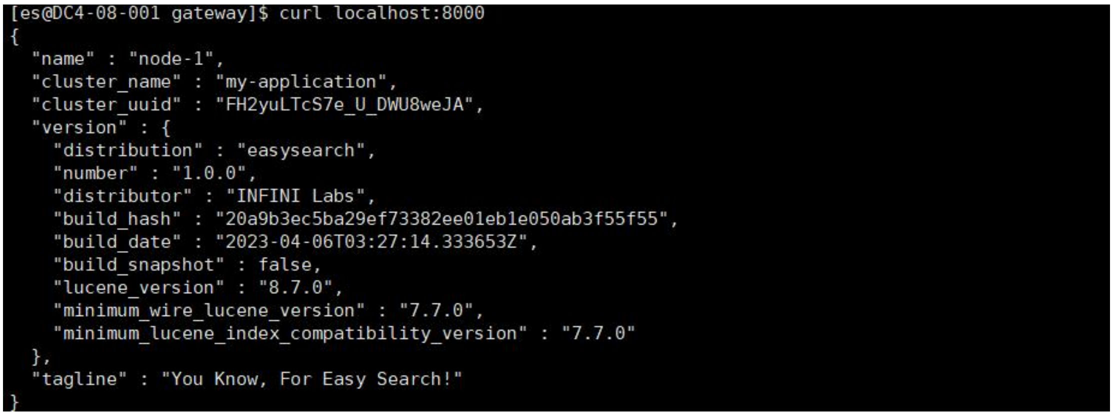


```
[es@DC4-08-001 gateway]$ sudo ./gateway-linux-arm64 -service install
[WARNING] THIS IS IN DEVELOPMENT MODE.
[11-12 16:47:11] [INF] [env.go:172] configuration auto reload enabled
Success
[es@DC4-08-001 gateway]$ sudo ./gateway-linux-arm64 -service start
[WARNING] THIS IS IN DEVELOPMENT MODE.
[11-12 16:47:21] [INF] [env.go:172] configuration auto reload enabled
Success
```

INFINI Gateway 成功运行后，会监听 8000 端口。

验证

访问 IP:8000 端口，能返回代理的后端信息。



```
[es@DC4-08-001 gateway]$ curl localhost:8000
{
  "name" : "node-1",
  "cluster_name" : "my-application",
  "cluster_uuid" : "FH2yuLTcS7e_U_DWU8weJA",
  "version" : {
    "distribution" : "easysearch",
    "number" : "1.0.0",
    "distributor" : "INFINI Labs",
    "build_hash" : "20a9b3ec5ba29ef73382ee01eb1e050ab3f55f55",
    "build_date" : "2023-04-06T03:27:14.333653Z",
    "build_snapshot" : false,
    "lucene_version" : "8.7.0",
    "minimum_wire_lucene_version" : "7.7.0",
    "minimum_lucene_index_compatibility_version" : "7.7.0"
  },
  "tagline" : "You Know, For Easy Search!"
}
```