

安裝

安裝說明書

安裝

OmniMessage SMPP 服務器 安裝說明書 SMPP 服務器 OmniMessage 安裝說明書
安裝說明書 OmniMessage Core 安裝 - 安裝

1. 安裝 安裝說明書 SMPP PDU
2. 安裝 REST API 安裝 OmniMessage 安裝
3. 安裝 OmniMessage 安裝說明書
4. 安裝 SMPP PDU 安裝
5. 安裝 安裝 OmniMessage

安裝 OmniMessage 安裝DiameterMAPIMS安裝說明書 - 安裝說明書
OmniMessage Core

安裝

```
/opt/omnimessage-smpp/config/runtime.exs
```

安裝說明書

```
sudo systemctl restart omnimessage-smpp
```

安裝

安裝 Elixir 安裝說明書

```
import Config

# 設定
config :omnimessage_smpp,
  setting_name: value

# SMPP 設定
config :omnimessage_smpp, :binds, [
  %{
    name: "bind_name",
    # ... 設定
  }
]
```

設定

API_BASE_URL

OmniMessage Core 設定 URL

```
config :omnimessage_smpp,
  api_base_url: "https://omnimessage-core.example.com:8443"
```

項目	項目	項目	項目
api_base_url	URL (URL)		-

OmniMessage Core 設定 URLはREST API の OmniMessage 設定に依存する

- 設定された SMPP 設定で OmniMessage 設定
- 設定された SMPP 設定
- 設定された OmniMessage
- 設定

設定された“”OmniMessage 設定

- ✓ 配置消息队列
- ✓ 配置消息队列的消费者
- ✓ 配置消息队列的发送者
- ✓ 配置消息队列的接收者
- ✓ 配置消息队列的发送者
- ✓ 配置消息队列的接收者
- ✓ 配置消息队列的发送者

配置消息队列 SMPP ↔ OmniMessage 配置消息队列

配置

```
# 配置 IP 和 HTTPS
api_base_url: "https://192.168.1.100:8443"

# 配置消息队列 HTTPS
api_base_url: "https://omnimessage-core.company.com:8443"

# HTTP配置消息队列
api_base_url: "http://192.168.1.100:8080"
```

配置消息队列

- 配置消息队列 OmniMessage Core
- 配置消息队列 HTTPS配置 `verify_ssl_peer`
- 配置消息队列消息队列 HTTPS

SMPP_POLL_INTERVAL

配置消息队列

```
config :omnimessage_smpp,
  smpp_poll_interval: 100
```


SMSC_NAME

□□□□□□□

```
config :omnimessage_smpp,
  smsc_name: "smpp_gateway"
```

消息	消息	消息	消息
smmc_name	消息	消息	"smpp_gateway"

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

SMSC_NAME

SMPP

ESME **SMSC**

配置参数

```
config :omnimessage_smpp, :binds, [
  %{
    # 名称
    name: "vodafone_uk",

    # 模式
    mode: :client,

    # SMPP 绑定
    bind_type: :transceiver,

    # SMPP 主机
    host: "smpp.vodafone.co.uk",
    port: 2775,

    # 系统 ID
    system_id: "your_username",
    password: "your_password",

    # TPS 限制
    tps_limit: 100,

    # 队列检查频率
    queue_check_frequency: 1000
  }
]
```

名称

name

名称

名称	值	说明
名称	字符串	"vodafone_uk"

配置参数 SMPP 绑定

host

Carrier SMPP Carrier IP

Carrier	IP	Host
Carrier	IP	"smpp.carrier.com" IP "10.5.1.100"

Carrier SMPP Carrier

Carrier

```
host: "smpp.vodafone.co.uk"  
host: "10.20.30.40"  
host: "smpp-primary.carrier.net"
```

port

SMPP Carrier

Carrier	IP	Port	Host
Carrier	IP	2775	1-65535

Carrier SMPP Carrier TCP Carrier

Carrier 2775

Carrier

```
port: 2775 # Carrier  
port: 3000 # Carrier
```

system_id

Carrier

- 配置参数

配置

```
tps_limit: 10    # 配置
tps_limit: 50    # 配置
tps_limit: 100   # 配置
tps_limit: 1000  # 配置
```

配置

```
配置 = 100 TPS
配置 tps_limit = 70-80
配置 20-30 TPS 配置
```

queue_check_frequency

配置参数

配置	配置	配置	配置
配置	配置	1000	100-10000

配置参数

配置

- 配置 > **100 TPS** 配置 500-1000ms
- 配置 **10-100 TPS** 配置 1000-2000ms
- 配置 < **10 TPS** 配置 2000-5000ms

配置

- 配置 = 配置 API 配置
- 配置 = 配置 API 配置

Web UI 配置

SMPP 接口

SMPP 接口是 SMSC 和 ESME 之间的接口，用于发送和接收短信。

配置参数

```
config :omnimessage_smpp, :server_binds, [
  %{
    # 合作伙伴名称
    name: "partner_acme",

    # 系统ID
    system_id: "acme_corp",
    password: "acme_secret",

    # 允许的绑定类型
    allowed_bind_types: [:transmitter, :receiver, :transceiver],

    # IP 白名单
    ip_whitelist: ["192.168.1.0/24", "10.50.1.100"],

    # TPS 限制
    tps_limit: 50,

    # 队列检查频率
    queue_check_frequency: 1000
  }
]
```

合作伙伴名称

name

合作伙伴名称

名称	类型	默认值
合作伙伴名称	字符串	"partner_acme"

合作伙伴名称必须唯一

合作伙伴名称/密码必须唯一

system_id

Table 1: Example 1

Field	Type	Value
Company	String	"acme_corp"

Table 2: Example 2

Table 3: Example 3

password

Table 4: Example 4

Field	Type	Value
Password	String	"secure_password"

Table 5: Example 5

Table 6: Example 6

- Example 7
- Example 8
- Example 9

allowed_bind_types

Table 7: Example 10

Field	Type	Value
Allowed Bind Types	String	-

Table 8: Example 11

Table 9: Example 12

```
allowed_bind_types: [:transceiver] # [] transceiver
allowed_bind_types: [:transmitter, :receiver] # TX [] RX
allowed_bind_types: [:transmitter, :receiver, :transceiver] # []
```

ip_whitelist

ip_whitelist

ip_whitelist IP []

IP	Port	Protocol	Comment
192.168.1.100	80	HTTP	IP [] CIDR []

ip_whitelist - IP []

ip

- IP "192.168.1.100" /32
- CIDR "192.168.1.0/24" "10.0.0.0/8"
- ["192.168.1.0/24", "10.50.1.100"]

ip

```
# IP
ip_whitelist: []

# IP
ip_whitelist: ["203.0.113.50"]

# IP
ip_whitelist: ["203.0.113.50", "203.0.113.51"]

#
ip_whitelist: ["192.168.1.0/24"]

#
ip_whitelist: ["192.168.1.0/24", "10.50.1.100", "10.60.0.0/16"]
```

□□□□□

- /32 - 1 IP address
- /24 - 256 IP addresses
- /16 - 65,536 IP addresses
- /8 - 16,777,216 IP addresses

tps_limit

□ □ □ □ □ □

```
tps_limit 0 - 1 deliver_sm 0
```

queue_check_frequency

□□□□□□

queue_check_frequency - 检查队列的频率

Web UI ☐☐☐

--	--	--	--	--	--	--

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

配置

```
config :omnimessage_smpp, :listen, %{
  host: "0.0.0.0",
  port: 2775,
  max_connections: 100
}
```

监听

host

监听 IP 地址

名称	地址	监听地址	监听范围
监听	否	"0.0.0.0"	"0.0.0.0" 至 "127.0.0.1"

监听地址范围

地址

- "0.0.0.0" - 监听所有地址
- "127.0.0.1" - 监听本地地址
- "192.168.1.10" - 监听 IP 地址

port

监听 TCP 端口

名称	地址	端口	范围
监听	否	2775	1-65535

监听 SMPP 端口

监听 2775

max_connections

配置項目

項目	単位	初期値	範囲
max_connections	個	100	1-10000

設定方法: 設定ファイルの max_connections を変更する

注意事項

- 設定値が初期値より大きい場合は、再起動が必要
 - 設定値が初期値より小さい場合は、エラーが発生する
 - 設定値が 10-100 の範囲で設定される
-

□□□□□□

□□ **1**□□□□□□□□

```
import Config

config :omnimessage_smpp,
  api_base_url: "https://smcsc.company.com:8443",
  verify_ssl_peer: true,
  smsc_name: "smpp_prod"

config :omnimessage_smpp, :binds, [
  %{
    name: "att_primary",
    mode: :client,
    bind_type: :transceiver,
    host: "smpp.att.com",
    port: 2775,
    system_id: "company_user",
    password: "secure_pass_123",
    tps_limit: 100,
    queue_check_frequency: 1000
  }
]
```

2

```
import Config

config :omnimessage_smpp,
  api_base_url: "https://smcsc.company.com:8443"

config :omnimessage_smpp, :binds, [
  # []
  %{
    name: "att_us",
    mode: :client,
    bind_type: :transceiver,
    host: "smpp.att.com",
    port: 2775,
    system_id: "att_username",
    password: "att_password",
    tps_limit: 100,
    queue_check_frequency: 1000
  },

  # []
  %{
    name: "vodafone_uk",
    mode: :client,
    bind_type: :transceiver,
    host: "smpp.vodafone.co.uk",
    port: 2775,
    system_id: "voda_username",
    password: "voda_password",
    tps_limit: 50,
    queue_check_frequency: 1000
  }
]
```

03 3 0000000000000000

```
import Config

config :omnimessage_smpp,
  api_base_url: "https://smcsc.companay.com:8443"

# 0000
config :omnimessage_smpp, :binds, [
  %{
    name: "upstream_carrier",
    mode: :client,
    bind_type: :transceiver,
    host: "smpp.carrier.com",
    port: 2775,
    system_id: "my_username",
    password: "my_password",
    tps_limit: 100,
    queue_check_frequency: 1000
  }
]

# 00000000
config :omnimessage_smpp, :server_binds, [
  %{
    name: "partner_alpha",
    system_id: "alpha_corp",
    password: "alpha_secret",
    allowed_bind_types: [:transmitter, :receiver, :transceiver],
    ip_whitelist: ["203.0.113.0/24"],
    tps_limit: 50,
    queue_check_frequency: 1000
  },
  %{
    name: "partner_beta",
    system_id: "beta_inc",
    password: "beta_password",
    allowed_bind_types: [:transceiver],
    ip_whitelist: ["198.51.100.50"],
    tps_limit: 25,
    queue_check_frequency: 2000
  }
]
```

```
# 配置
config :omnimessage_smpp, :listen, %{
  host: "0.0.0.0",
  port: 2775,
  max_connections: 100
}
```

安装

安装依赖包

编译

```
# 编译 Elixir 项目
/opt/omnimessage-smpp/bin/omnimessage-smpp eval "File.read!
('config/runtime.exs')"
```

编译成功后，会在 bin 目录下生成可执行文件

运行

```
# 运行服务
sudo -u omnimessage-smpp /opt/omnimessage-smpp/bin/omnimessage-smpp console
```

按 `Ctrl+C` 停止服务

测试

1. 测试连接

```
sudo chmod 600 /opt/omnimessage-smpp/config/runtime.exs
sudo chown omnimessage-smpp:omnimessage-smpp /opt/omnimessage-smpp/config/runtime.exs
```

2. 設定項目

- 12 時間
- 設定項目
- 設定項目

3. IP 設定

- 設定項目 `ip_whitelist`
- 設定項目 `[]`

4. SSL 設定

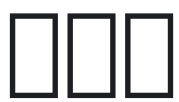
- 設定 `verify_ssl_peer: true` 設定項目

5. 設定項目

- 設定項目
- 設定項目/設定項目

目次

- [MONITORING.md](#) 設定項目
 - [OPERATIONS.md](#) 設定項目
 - [TROUBLESHOOTING.md](#) 設定項目
 - [README.md](#) 設定項目
-



□□□□

A

API□□□□□□□□

□□□□□□□□□□□□□□

□□□□

Web UI □□□□□□□□□□□□□□□□

B

□□

SMPP □□□□□□□□□□□□□□

□□

□□□□□□ SMPP □□□□□□□□□□□□□□

□□□□

SMPP □□□□□□

- □□□□□□□□
- □□□□□□□□
- □□□□□□□□

□□□□

□ SMPP □□□□□□□□□□□□□□□□ IP □□□

C

CIDR□□□□□□□□

□□ IP □□□□□□□□□□ 192.168.1.0/24 □□ 256 □ IP □□□□

□□□□

□□ SMPP □□□□□□□□ **ESME** □□□□□ **SMSC**□□□□□□□□ SMPP □□□□□□□□□□□□□□□□

□□□□

SMPP □□□□□□□□

- □□□□□□□□□□
- □□□□□□□□
- □□□□□□□□□□

□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

D

□□□□

□□□□□□□□□□ SMPP □□□GSM-7□UCS-2 □□□

□□□□

□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□**DLR**□

□□□□□□□□□□□□□□□□

dest_smsc

□□□□□□□□□□□□□□ SMPP □□□□□□□□□□

□□□□

□□□□ SMPP □□□□□□□□□□□□□□□□□□□□

E

□□□□

□□□□□ SMPP □□□□□□□□□□□□□□□□□□□□□□

ESM □

□□□□□□□□□□□□ SMPP □□□□

ISDN 0000000000
00000000000000

J

(0000)

K

0000
00000000enquire_link000000000000

KPI 00000000
000000000000000000000000

L

00
0 Prometheus 00000000000000000000 bind_name="vodafone_uk" 00

0000
0000 Web UI 000 Phoenix 00000

M

0000
00000000000000000000

00
00000000000 Prometheus 00000

MO 000000
000000000000000000

MT

Message Transfer

MSISDN

Mobile Station International
Subscriber Number

N

NPI

Number Portability
SMPP ISDN

O

Operator

Operator Number

Operator

Operator Number

P

PDU

SMPP submit_sm deliver_sm

Prometheus

Monitoring

Q

Queue

Queue Number

Queue

Queue Number

□□□□□□□□ SMPP □□□□□□

□□□□

TPS

SMPP deliver_sm

□□□□□□ SMPP □□□

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

□ □ □ □ □

[illegible]

□□□□□□□□ SMPP □□□

SMPP□□□□□□□□

□□□□□□□□ SMS □□□□□□□□□□

SMSC□□□□□□□□

[illegible]

SSL/TLS

[illegible]

Submit_SM

□□□□□□□□□□ SMPP PDU□

Submit_SM_Resp

submit_sm SMPP

ID

SMPP

T

TON

SMPP

TPS

SMPP

SMPP submit_sm

U

UCS-2

SMS 16 Unicode 70

V

□□□

□□□□□□□□□□□□□□□□

W

Web □□□

□□□□□□□□□□□□□□□□□□

□□□

□ IP □□□□

X

(□□□)

Y

(□□□)

Z

(□□□)

□□□□□□□□

□□□	□□
API	□□□□□□□□
CIDR	□□□□□□
DLR	□□□□
ESME	□□□□□□□□
GSM	□□□□□□□□
HTTP	□□□□□□□□
HTTPS	□□□□□□□□□□
IP	□□□□□□
ISDN	□□□□□□□□
KPI	□□□□□□□□
MO	□□□□□
MSISDN	□□□□□□□□□□□□
MT	□□□□□
NPI	□□□□□□□□
PDU	□□□□□□□□
SMPP	□□□□□□□□
SMSC	□□□□□□□□

項目	説明
SMS	短メッセージサービス
SSL	セキュアソケットレイヤー
TLS	トランスポートレイヤーセキュリティ
TON	トーン
TPS	トランザクション処理速度
UCS	ユニバーサルコーディングシステム
UI	ユーザーインターフェース
URL	ユニークリソースロケータ

目次

- **README.md** - 概要
 - **CONFIGURATION.md** - 設定
 - **OPERATIONS.md** - 運用
 - **MONITORING.md** - 監視
 - **TROUBLESHOOTING.md** - トラブルシューティング
-

Getting Started

SMPP Monitoring

Overview

SMPP monitoring is implemented using Prometheus and OmniMessage Core.

OmniMessage Core provides a **OmniMessage** API for monitoring SMPP.

1. **SMPP** - Monitoring
2. **OmniMessage API** - Configuration

Configuration

URL `http://your-server:4000/metrics`

Configure Prometheus to scrape

the `localhost` endpoint.

Usage

```
curl http://localhost:4000/metrics
```

Labels

Labels are used to identify the SMPP instance.

□ □ □ □ □

omnmessage_smpp_license_status

□□□ Gauge

--	--	--	--	--	--	--	--

11

- 1 = □□□□
- 0 = □□/□□□□

11

111

```
omnimessage_smpp_license_status 1
```

- 0000000000000000
- 0000000000000000SMPP000000
- Web UI000000000000

omnmessage_smpp

111

- `license_status == 0`
- SMPP
-
- UI

□□□□□

```
- alert: SMPP_License_Invalid
  expr: omnimessage_smpp_license_status == 0
  for: 1m
  labels:
    severity: critical
  annotations:
    summary: "SMPP许可证过期"
    description: "许可证过期 - 请检查许可证状态"
```

许可证

smpp_connection_status

仪表盘 Gauge

仪表盘 显示 SMPP 连接状态

值

- 1 = 成功
- 0 = 失败

配置

- bind_name - 绑定名称 "vodafone_uk"
- mode - 模式 "client" 或 "server"
- host - 主机地址
- port - 端口
- bind_type - SMPP 绑定类型
- system_id - 系统 ID

示例

```
smpp_connection_status{bind_name="vodafone_uk",mode="client",host="sn
1
```

仪表盘

- 0000000000000000
- 0000000000000000
- 00000000

000000

smpp_messages_sent_total

Counter

SMPP0000000000

00

connection_status

```
smpp_messages_sent_total{bind_name="vodafone_uk",mode="client",...}
150234
```

- 00000000/00
- 0000/000000
- 0000000000

smpp_messages_received_total

Counter

SMPP0000000000

00

connection_status

```
smpp_messages_received_total{bind_name="partner_acme",mode="server",...}
45123
```

📌

- 📶
- 📶MO📶
- 📶📶📶📶📶

📌📌📌

smpp_delivery_failures_total

📌 Counter

📌 📶📶📶📶📶

📌 📶

📌 📶connection_status📶

📌

```
smpp_delivery_failures_total{bind_name="vodafone_uk",mode="client",...}
234
```

📌

- 📶📶📶
- 📶📶📶📶📶
- 📶📶📶

📶📶📶

```
success_rate = (messages_sent - delivery_failures) / messages_sent
* 100
```

概要

smpp_bind_success_total

Counter

Bind 成功回数

Bind 成功回数

Bind

```
smpp_bind_success_total{bind_name="vodafone_uk",...} 45
```

Bind

- Bind 成功回数
- Bind 成功回数

smpp_bind_failures_total

Counter

Bind 失敗回数

Bind 失敗回数

Bind

```
smpp_bind_failures_total{bind_name="vodafone_uk",...} 3
```

Bind

- Bind 失敗回数
- Bind 失敗回数
- Bind 失敗回数

概要

smpp_connection_attempts_total

Counter

Counters

Count

smpp_connection_attempts_total{bind_name="vodafone_uk",...} 48

- Counters
- Counters

smpp_disconnection_total

Counter

Counters

Count

smpp_disconnection_total{bind_name="vodafone_uk",...} 3

- Counters
- Counters
- Counters

Counters

smpp_uptime_seconds

Gauge

Counters SMPP Counters

Count

📄

```
smpp_uptime_seconds{bind_name="vodafone_uk",...} 86400
```

📄

- 📄📄📄📄
- 📄📄📄📄📄📄📄
- 📄📄📄📄📄📄

OmniMessage API📄📄📄

📄📄📄📄📄📄SMPP📄📄📄📄📄**OmniMessage API**📄📄📄📄📄📄📄📄📄📄

📄📄**OmniMessage**📄📄📄📄📄📄📄

- `omnmessage_api_requests_total` - 📄📄📄📄API📄📄📄
- `omnmessage_api_request_duration_seconds` - API📄📄📄
- `omnmessage_queue_depth` - OmniMessage📄📄📄📄📄📄

📄📄📄📄📄📄📄📄📄📄

📄📄📄📄📄📄API📄📄📄

- "api.*connection refused" - 📄📄📄📄OmniMessage
- "api.*timeout" - OmniMessage📄📄
- "api.*http 503" - OmniMessage📄📄📄📄
- "api.*parse error" - 📄📄📄📄

Prometheus📄📄

📄📄📄📄📄📄

📄📄 `/etc/prometheus/prometheus.yml`🔍🔍🔍

```
scrape_configs:
  - job_name: 'omnimessage-smpp'
    scrape_interval: 15s
    static_configs:
      - targets: ['your-server:4000']
        labels:
          environment: 'production'
          service: 'omnimessage-smpp'
```

□□□□

```
scrape_configs:
  - job_name: 'omnimessage-smpp-instances'
    scrape_interval: 15s
    static_configs:
      - targets:
          - 'smpp-gw-1:4000'
          - 'smpp-gw-2:4000'
          - 'smpp-gw-3:4000'
        labels:
          environment: 'production'
```

□□□□

□□□□□□□□□□

```
scrape_configs:
  - job_name: 'omnimessage-smpp-instances'
    file_sd_configs:
      - files:
          - '/etc/prometheus/targets/smpp-*.json'
```

□□ /etc/prometheus/targets/smpp-production.json □

```
[
  {
    "targets": ["smpp-gw-1:4000", "smpp-gw-2:4000"],
    "labels": {
      "environment": "production",
      "datacenter": "us-east"
    }
  }
]
```

Grafana

```
smpp_connection_status{job="omnimessage-smpp"}
```

Stat

- < 1
- $== 1$

```
rate(smpp_messages_sent_total{job="omnimessage-smpp"}[5m])
```

Graph

{{bind_name}}

00000000

000

```
100 * (1 - (
    rate(smpp_delivery_failures_total{job="omnimessage-smpp"}[5m])
    /
    rate(smpp_messages_sent_total{job="omnimessage-smpp"}[5m])
))
```

0000 Gauge

000 00000-1000

000

- 000 < 95%
- 000 95-98%
- 000 > 98%

000000000000

000

```
smpp_uptime_seconds{job="omnimessage-smpp"} / 3600
```

0000 Stat

000 00

00000

Prometheus0000

000 /etc/prometheus/rules/smpp-alerts.yml0

```

groups:
- name: smpp_gateway
  interval: 30s
  rules:
    # 连接
    - alert: SMPPConnectionDown
      expr: smpp_connection_status == 0
      for: 2m
      labels:
        severity: critical
      annotations:
        summary: "SMPP{{ {{ $labels.bind_name }} }} 连接"
        description: "连接 {{ {{ $labels.bind_name }} }} 已断开2分钟"

    # 失败率
    - alert: SMPPHighFailureRate
      expr: |
        (
          rate(smpp_delivery_failures_total[5m])
          /
          rate(smpp_messages_sent_total[5m])
        ) > 0.05
      for: 5m
      labels:
        severity: warning
      annotations:
        summary: "{{ {{ $labels.bind_name }} }} 失败率过高"
        description: "失败率 {{ {{ $labels.bind_name }} }} 为 {{ {{ $value | humanizePercentage }} }}"

    # 绑定失败
    - alert: SMPPBindFailures
      expr: increase(smpp_bind_failures_total[10m]) > 3
      labels:
        severity: warning
      annotations:
        summary: "{{ {{ $labels.bind_name }} }} 绑定失败"
        description: "{{ {{ $labels.bind_name }} }} 在10分钟内发生了 {{ {{ $value }} }} 次"

    # 没有流量
    - alert: SMPPNoTraffic
      expr: rate(smpp_messages_sent_total[10m]) == 0

```

```

    for: 30m
    labels:
      severity: warning
    annotations:
      summary: "⚠️ {{ $labels.bind_name }} 连接异常"
      description: "{{ $labels.bind_name }} 连接异常30分钟内未恢复"

# 连接异常
- alert: SMPPFrequentDisconnections
  expr: increase(smpp_disconnection_total[1h]) > 5
  labels:
    severity: warning
  annotations:
    summary: "⚠️ {{ $labels.bind_name }} 连接异常"
    description: "{{ $labels.bind_name }} 连接异常 {{ $value }} 次"

# OmniMessage API异常
- alert: OmniMessageAPIUnreachable
  expr: |
    count(count_over_time({job="omnmessage-smpp"} |=
"api.*connection refused"[5m])) > 0
  for: 1m
  labels:
    severity: critical
  annotations:
    summary: "OmniMessage API异常"
    description: "SMPP连接异常OmniMessage API异常API_BASE_URL异常"

# OmniMessage API异常
- alert: OmniMessageAPITimeout
  expr: |
    count(count_over_time({job="omnmessage-smpp"} |=
"api.*timeout"[5m])) > 5
  for: 2m
  labels:
    severity: warning
  annotations:
    summary: "OmniMessage API异常"
    description: "连接异常API异常OmniMessage连接异常"

# 连接异常API异常
- alert: NoMessageFlow

```

```
    expr: rate(smpp_messages_sent_total[10m]) == 0 and
rate(smpp_messages_received_total[10m]) == 0
    for: 30m
    labels:
      severity: warning
    annotations:
      summary: "送信受信 - 0omniMessage"
      description: "送信30分間隔で0omniMessage APIが返らない"

```

prometheus.yml を編集

```
rule_files:
- '/etc/prometheus/rules/smpp-alerts.yml'
```

Web UI

Web UIはPrometheus

URL `https://your-server:8087`

SMPP →

- 送信受信
- 送信
- 送信受信
- 送信/受信
- 5分間隔

- 
- 
- 

--	--	--	--	--	--

- [illegible]

--	--	--	--

--	--	--	--

--	--	--	--	--

```
# 確認
sudo journalctl -u omnimessage-smpp -f

# 100行
sudo journalctl -u omnimessage-smpp -n 100

# 1時間前以降
sudo journalctl -u omnimessage-smpp --since "1 hour ago"

# エラー
sudo journalctl -u omnimessage-smpp -p err
```

Web UI

Web UI

-
-
-
-
-

- 時間あたりに送信できるメッセージ数
- 送信成功率
- 送信遅延時間/メッセージあたり
- 送信コスト/メッセージあたり
- 送信エラー率

送信遅延時間KPI

送信遅延時間

送信遅延時間

```
avg_over_time(smpp_connection_status[24h]) * 100
```

送信遅延時間 > 99.9%

送信遅延時間

送信遅延時間

```
rate(smpp_messages_sent_total[5m])
```

送信遅延時間

送信遅延時間

送信遅延時間

```
100 * (1 - rate(smpp_delivery_failures_total[5m]) /  
rate(smpp_messages_sent_total[5m]))
```

送信遅延時間 > 98%

📊 📈 📉 📊

📊 📈 📉 📊

```
rate(smpp_bind_success_total[1h]) * 3600
```

📊 < 10 📊 📈 📉 📊

📊 📈 📉 📊

1. 📊 📈

- 📊 📈 Prometheus 📊
- 📊 PagerDuty/OpsGenie 📊 24/7 📊
- 📊 📈

2. 📊 📈

- 📊 📈 Grafana 📊
- 📊 📈 📊 📈
- 📊 📈

3. 📊 📈

- 📊 📈
- 📊 📈
- 📊 📈

4. 📊 📈

- 📊 📈
- 📊 📈 TPS 📊
- 📊 📈 / 📊

5. 配置

- 配置API
 - 配置参数
 - 配置
-

配置参数

配置

配置 `smpp_connection_status`

- 0 = 配置
- 配置 = 配置

配置

配置 `smpp_delivery_failures_total`

- 配置 = 配置
- 配置 = 配置

配置

配置 `smpp_messages_sent_total` 配置

- 配置 = 配置TPS配置
- 配置API

配置

配置 `smpp_bind_failures_total`

- 配置 = 配置
- 配置system_id配置

□□□□

- **CONFIGURATION.md** - □□□□□□
 - **OPERATIONS.md** - □□□□
 - **TROUBLESHOOTING.md** - □□□□
 - **README.md** - □□□□□□□□
-



XXXXXX

XXXXXX OmniMessage Core

XXXX OmniMessage SMPP XXXXXXXX OmniMessage Core XXXXXXXXXXXXXXXXXXXX
OmniMessage - XXXXXXXXXXXXXXXX

XX OmniMessage XXXXX

- XX XXXXXXXX
- XX XXXXXXXXXXXX
- XX XXXXXXXX
- XX XXXXXXXXXXXX

XX **OmniMessage** XXXXX

```
# XX API XXXX
curl -k https://omnimessage-
core.example.com:8443/api/system/health

# XXXXXXXXXX API URL
grep api_base_url /opt/omnimessage-smpp/config/runtime.exs
```



XXXXXXXXXX

XXXXXXXXXXXXXXXXXX

1. XX **Web** XXXX

- URL: `https://your-server:8087`

- 配置数据库连接

2. 配置

- 配置SMPP → 配置
- 配置“”
- 配置

3. 配置

- 配置
- 配置
- 配置

4. 配置

- 配置
- 配置
- 配置

5. 配置 Prometheus

- `curl http://localhost:4000/metrics`
- 配置 Grafana
- 配置

□□□□

□□□□□□□□

- □□□□> 2 □□□□
- □□□□□□> 5%□
- □□□□□□
- □□□□□□

□□□ **MONITORING.md** □□□□□□□□

□□ **SMPP** □□

SMPP □□□□□□□□

SMPP □□□□□□□□□□ □□□□ □□□□□

□□ 1□**Web UI**□□□□

- □□□□□□□□□□□□□□
- □□□SMPP → □□□□□□ / □□□□□□□□
- □□□□□□□□□□□□□□
- □□□□□□□ Mnesia □□□□
- □□□□□□□□□□□□□□□□

□□ 2□□□□□

- □□□□□□□□□□□□□□
- □□□ `/opt/omnimessage-smpp/config/runtime.exs`
- □□□□ Elixir □□□□□□□□
- □□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□

□□□Web UI □□□□□□□□□□□□□□□□

📄 **CONFIGURATION.md** 📖

📁

📁 **ESME** 📁 **SMSC**

📁

- SMPP 📁/IP
- 📁 2775
- 📁 ID
- 📁
- 📁
- TPS 📁

📁

📁 **A** 📁 **Web UI**

📁

📁

1. 📁

- 📁 Web UI `https://your-server:8087`
- 📁 SMPP → 📁

2. 📁

- 📁 “📁”
- 📁
 - 📁 `vodafone_uk`
 - 📁 `smpp.vodafone.co.uk`
 - 📁 `2775`
 - 📁 **ID** `your_username`
 - 📁 `your_password`
 - 📁 `Transceiver`

- **TPS** 100
- 1000
- “”

3.

-
- SMPP →
- 10-30 “”
-

4.

-
- dest_smsc
-
-

B

1.

```
sudo nano /opt/omnimessage-smpp/config/runtime.exs
```

2. 配置连接参数

```
config :omnimessage_smpp, :binds, [  
  # 配置连接参数...  
  
  # 配置连接参数  
  %{  
    name: "vodafone_uk",  
    mode: :client,  
    bind_type: :transceiver,  
    host: "smpp.vodafone.co.uk",  
    port: 2775,  
    system_id: "your_username",  
    password: "your_password",  
    tps_limit: 100,  
    queue_check_frequency: 1000  
  }  
]
```

3. 保存并重启服务

```
# 保存并重启 nano 按 Ctrl+X, Y, Enter  
  
# 重启服务  
sudo systemctl restart omnimessage-smpp
```

4. 验证配置

- 验证 SMPP 连接是否成功
- 验证连接参数
- 验证“连接”是否成功
- 验证连接是否成功

5. 部署应用

- 部署应用

- 設定する `dest_smsc` 設定する
- 設定する
- 設定する

設定する

設定する **SMSC** 設定する **ESMEs** 設定する

設定

1. 設定

- 設定する ID `partner_name`
- 設定する
- 設定する

2. 設定

- 設定する IP 設定
- 設定する TPS 設定
- 設定する

設定

設定 **A** **Web UI** 設定

設定

設定

1. 設定

- 設定 Web UI `https://your-server:8087`
- 設定 SMPP → 設定

2. 設定

- 設定 “設定”
- 設定
 - 設定 `partner_acme` 設定

- 公司 ID `acme_corp`
 - 密码 `secure_password_123`
 - 电话号码
 - IP 地址 `203.0.113.0/24` 网段
 - TPS 每秒 `50`
 - 并发数 `1000`
- 配置“配置”

3. 配置数据库

- 数据库连接池配置
- 数据库

4. 配置网络

- 设置 IP 地址
- 端口 `2775`
- 公司 ID `acme_corp`
- 密码 `secure_password_123`
- 电话号码

5. 配置消息队列

- 配置 SMPP → 配置
- 配置

- 設定ファイル
- 設定 IP 設定ファイル

設定 **B** 設定ファイル

設定ファイルのインストール

設定

1. 設定ファイル

```
sudo nano /opt/omnimessage-smpp/config/runtime.exs
```

2. 設定ファイルのインストール

```
# 設定 server_binds 設定
config :omnimessage_smpp, :server_binds, [
  # 設定ファイル...

  # 設定ファイル
  %{
    name: "partner_acme",
    system_id: "acme_corp",
    password: "secure_password_123",
    allowed_bind_types: [:transmitter, :receiver,
:transceiver],
    ip_whitelist: ["203.0.113.0/24"],
    tps_limit: 50,
    queue_check_frequency: 1000
  }
]

# 設定ファイルのインストール
config :omnimessage_smpp, :listen, %{
  host: "0.0.0.0",
  port: 2775,
  max_connections: 100
}
```

3. 設定ファイル

```
sudo systemctl restart omnimessage-smpp
```

4. 設定ファイル

- 送信 IP 設定
- 送信ポート `2775`
- 送信 ID `acme_corp`
- 送信パスワード `secure_password_123`
- 送信者名

5. 送信設定

- 送信先 SMPP → 送信先
- 送信先
- 送信先
- 送信 IP 送信先

送信先

送信先 SMPP 送信先 IP 送信先

送信先

送信先 **Web UI** 送信先

送信先

送信先

1. 送信先

- 送信 Web UI `https://your-server:8087`
- 送信先 SMPP → 送信先
- 送信先 SMPP → 送信先

2. 送信先

- 送信先

- 設定“bind”
- 設定bind
 - 設定bindTPS 設定bindIP 設定bind/
- 設定“bind”

3. 設定bind

- 設定bind
- 設定bind
- 設定bindSMPP → 設定bind

4. 設定bind

- 設定bind
- 設定bind
- 設定bind

設定 B 設定bind

設定bind

設定

1. 設定bind

```
sudo nano /opt/omnimessage-smpp/config/runtime.exs
```

2. 設定bind

- 設定 :binds 設定 :server_binds 設定bind
- 設定bind
 - 設定bindTPS 設定bindIP 設定bind/
- 設定

```
%{
  name: "vodafone_uk",
  # ...
  tps_limit: 150, # 100
  password: "new_password" #
}
```

3. 再起動

```
sudo systemctl restart omnimessage-smpp
```

4. 確認

- SMPP →
-
-
-

インストール

インストール SMPP

インストール

1. インストール

- /
-

2. Web UI

- SMPP →
-
- “”
-

3. 確認

- SMPP → /
-
- “”
-

4.

- -
-

1.

-
-

2.

- 消息体
- 消息头
 - 消息ID
 - 消息类型
 - 目标 SMSC (`dest_smsc`)
 - 消息内容
 - 消息状态

3. 消息体

- 消息体结构
- 消息体中的 SMSC 地址

消息体结构

消息体中的 SMSC 地址

消息

1. 消息体

- 消息体 SMPP → 消息
- 消息体中的 SMSC 地址
- 消息体中的 消息内容

2. 消息体

- 消息体结构
- 消息体中的 SMSC 地址
- 消息体中的 `dest_smsc` 地址
- 消息体中的 `deliver_after` 地址

3. 消息体

- 消息 = 消息头 + 消息体
- 消息体中的 SMSC 地址
- 消息体中的 消息内容

4. 配置参数

- 配置参数
- 配置参数
- 配置参数

配置参数

配置参数

配置参数“配置”参数

配置

1. 配置参数

```
ping -c 3 carrier-smpp-server.com  
telnet carrier-smpp-server.com 2775
```

2. 配置参数

- 配置参数
- 配置参数
- 配置参数

3. 準備

- 準備SMPP → 準備/準備
- 準備 ID 準備
- 準備準備

4. 準備

- 準備SMPP → 準備
- 準備
- 準備“準備”
- 準備 10-30
- 準備準備“準備”

5. 準備

- 準備
- 準備
- 準備
- 準備 [TROUBLESHOOTING.md](#)

準備

準備

準備

- 配置/部署
- IP 地址分配
- 网络拓扑

步骤

1. 初始化

- 配置 SMPP → 初始化
- 配置 ID 信息
- 配置

2. 配置 IP 地址

- 配置 IP 地址
- 配置 IP 地址

3. 配置

- 配置
- 配置
- 配置

4. 配置

- 配置 Web UI 界面
- 配置“配置”界面

配置

配置 Prometheus 配置

配置

```
curl http://localhost:4000/metrics | grep smpp_connection_status
```

□□□□

```
smpp_connection_status{bind_name="vodafone_uk",...} 1
smpp_connection_status{bind_name="att_us",...} 1
```

□□□□ 1□□□□□

□□□□

□□□□□□

1. □□ Web UI → SMPP → □□□□
2. ❓❓□□□□□□
3. □□□□□□□□
4. □□□□□□□□□□□□□□
5. □□ [TROUBLESHOOTING.md](#)

□□□□□□□

1. □□□□□□□□□□
2. □□□□□□□□□□
3. □□□□□□□□□□□□
4. □□□□□□□□□□

□□□□□□

1. □□□□□□□□□□□□
 2. □□ `dest_smsc` □□□□□□
 3. □□ TPS □□□□□□□□
 4. □□ `queue_check_frequency` □□
-

目錄

目錄

目錄

1. 目錄

- 目錄
- 目錄
- 目錄

2. 目錄

- 目錄
- 目錄
- 目錄

3. 目錄

- 目錄 SMPP 目錄
- 目錄
- 目錄 IP 目錄

4. 目錄

- 目錄
- 目錄 TPS 目錄
- 目錄

目錄

目錄

- 目錄
- 目錄
- 目錄

000

```
# 000000
sudo systemctl status omnimessage-smpp

# 0000
sudo systemctl restart omnimessage-smpp

# 0000
sudo systemctl status omnimessage-smpp

# 0000
sudo journalctl -u omnimessage-smpp -n 50
```

00 **Web UI** 000

1. 0000000000 30-60 000000
2. 0000SMPP → 0000
3. 00000000001-2 000
4. 0000000000

00000

00000000000000

```
# 0000
sudo cp /opt/omnimessage-smpp/config/runtime.exs \
  /opt/omnimessage-smpp/config/runtime.exs.backup.$(date +%Y%m%d)

# 0000
sudo tar -czf /tmp/smpp-certs-$(date +%Y%m%d).tar.gz \
  /opt/omnimessage-smpp/priv/cert/
```

0000000000

```
# 確認
sudo cp /opt/omnimessage-smpp/config/runtime.exs.backup.YYYMMDD \
/opt/omnimessage-smpp/config/runtime.exs

# 再起動
sudo systemctl restart omnimessage-smpp
```

確認

確認インストール

確認

1. サービスの有無

```
sudo systemctl status omnimessage-smpp
```

2. サービスの起動

```
sudo systemctl start omnimessage-smpp
```

3. サービスのログ

```
sudo journalctl -u omnimessage-smpp -n 100
```

4. システムの状態

- サービスの有無
- SSL のインストール
- ディスクの使用状況 `df -h`
- メモリ使用状況 `free -h`

5. サービスの再起動

TPS 介紹

TPS

1. 什麼是 TPS

- 什麼是 SMPP → 通訊
- 什麼是 TPS
- 什麼是“TPS”

2. 什麼是 TPS

- 什麼是 TPS
- 什麼是 TPS
- 什麼是 TPS

3. 什麼是 TPS

- 什麼是 TPS
- 什麼是 TPS
- 什麼是 TPS

4. 什麼是 TPS

- 什麼是 TPS
- 什麼是 TPS
- 什麼是 TPS

TPS 介紹

TPS 介紹

TPS

1. 什麼是 TPS

- 什麼是 SMPP → 通訊
- 什麼是 TPS
- 什麼是 TPS TPS

2. 認證與授權

- 認證與授權
- 認證與授權

3. 系統架構

- 系統架構
- 系統架構

4. 系統部署與維護

- 系統部署與維護
- 系統部署與維護

系統架構

系統部署與維護

- ☐ 系統 SMPP 認證與授權
- ☐ 系統認證與授權
- ☐ 系統認證與授權
- ☐ 系統 Prometheus/Grafana 監控
- ☐ 系統認證與授權 > 98%

系統部署與維護

- ☐ 系統認證與授權
- ☐ 系統認證與授權
- ☐ 系統認證與授權
- ☐ 系統認證與授權
- ☐ 系統認證與授權

目录

- ☐ 简介
 - ☐ 快速入门
 - ☐ 部署指南
 - ☐ 配置 TPS 节点
 - ☐ 故障排除
-

附录

- **CONFIGURATION.md** - 配置文件
 - **MONITORING.md** - Prometheus 监控
 - **TROUBLESHOOTING.md** - 故障排除
 - **README.md** - 说明
-



XXXXXXXXXX

OmniMessage XXXX

XX SMPP XXXXXXXXXXXXXXXX OmniMessage CoreXXXX OmniMessage XXXXXXXXXXXXXXXX
XXXX

OmniMessage XXXXXXXX

- XXXXXXXXXXXXXXXXXXXX
- XXXXXXXXXXXXXXXXXXXX
- XXXAPI XXXXXXXX
- XXXXXX "Connection refused", "Timeout", "HTTP 503", "Connection reset"

XX

1. XX OmniMessage XXXXX

```
# XXXXX
curl -k -v https://omnimessage-
core.example.com:8443/api/system/health

# XXXXXXXXXXXX
ssh gateway-server 'curl -k https://omnimessage-
core.example.com:8443/api/system/health'
```

2. XXXXXX API URL

```
# 確認
grep -A1 'api_base_url' /opt/omnimessage-smpp/config/runtime.exs

# 確認可能
ping omnimessage-core.example.com
nc -zv omnimessage-core.example.com 8443
```

3. 確認可能 API 確認

```
# 確認 API 確認可能
sudo journalctl -u omnimessage-smpp -f | grep -i
'api\|omnimessage\|connect'

# 確認可能確認可能
sudo journalctl -u omnimessage-smpp -n 200 | grep -i error
```

確認可能

確認 OmniMessage 確認可能

1. 確認 OmniMessage 確認可能
2. 確認可能確認可能確認可能
3. 確認可能確認可能 `SMPP_POLL_INTERVAL`
4. 確認 OmniMessage 確認可能確認可能

確認 OmniMessage 確認可能確認可能

1. 確認可能確認可能確認可能 HTTPS
2. 確認 DNS 確認 `nslookup omnimessage-core.example.com`
3. 確認可能確認可能 `traceroute omnimessage-core.example.com`
4. 確認 SSL 確認可能確認可能 HTTPS

確認 API URL 確認可能

1. 確認 `/opt/omnimessage-smpp/config/runtime.exs`
2. 確認 `api_base_url` 確認可能確認可能確認可能 HTTPS
3. 確認可能 `sudo systemctl restart omnimessage-smpp`

問題

問題

問題

- 状態 "Disconnected" になる
- 接続が切れる
- 再接続しない

原因

1. 原因

原因

```
# DNS 確認
nslookup smpp.carrier.com

# ping
ping -c 3 smpp.carrier.com

# telnet
telnet smpp.carrier.com 2775

# nc
nc -zv smpp.carrier.com 2775
```

原因

- DNS 設定が正しくない
- ping 通じていない
- 接続が切れる

2. 対策

対策

- 看看 "bind failed" 和 "authentication error"
- Web UI → SMPP → 看看系统_id 看看

看看

- 看看系统_id
- 看看系统_id 看看
- 看看系统_id

3. IP 看看

看看

- 看看系统_id
- 看看系统_id 看看 IP

看看

- 看看系统_id IP

```
curl ifconfig.me
```

- 看看系统_id IP 看看
- 看看 IP 看看系统_id IP

4. 看看

看看

```
# 看看系统_id
sudo iptables -L -n | grep 2775

# 看看 UFW → Ubuntu/Debian
sudo ufw status | grep 2775

# 看看 firewalld → RHEL/CentOS
sudo firewall-cmd --list-ports | grep 2775
```

看看

```
# Ubuntu/Debian
sudo ufw allow out 2775/tcp

# RHEL/CentOS
sudo firewall-cmd --permanent --add-port=2775/tcp
sudo firewall-cmd --reload
```

□□□□□□

□□□

- □□□□□□□□□
- `smpp_disconnection_total` □□□□
- □□□□□□□□

□□□□□□□□□□

1. □□□□□

□□□

```
# □□□□
ping -c 100 smpp.carrier.com | grep loss

# □□□□□□
netstat -s | grep -i error
```

□□□□□

- □□□□□□□□□□
- □□□□□□□□ ISP
- □□□□□□/□□

2. □□□□□□

□□□

- 添加 "enquire_link timeout"
- 添加配置项

配置项

- 添加配置 30
- 添加配置项
- 添加配置项

3. 添加 TPS 配置

配置

- 添加配置项
- 添加配置项

配置项

- 添加 `tps_limit` 配置
- 将 TPS 配置项设置为 70-80%
- 添加配置项

4. 添加配置项

配置

- 添加配置项
- 添加配置项

配置项

- 添加配置项
- 添加配置项

□□□□□□

□□□□□

□□□

- □□□□□□
- `smpp_messages_sent_total` □□□
- □□□□□□

□□□□□□□□□□

1. □□□ `dest_smsc` □❖❖❖

□□□

- Web UI → □□ → □□□□ `dest_smsc` □□
- □ SMPP → □□□□□□□□□□□□□□

□□□□□

- □□□□ `dest_smsc` □□□□
- □□□□□□□□□□□□ `dest_smsc`
- □□ `dest_smsc` □ NULL□□□□□□□□

2. □□□□□□□□□□

□□□

- Web UI → □□ → □□ `deliver_after` □□
- □□□□□□□□□□□□□□

□□□

- □□□□□□□□□□□□ `deliver_after`
- □□□□□□□□□□□□□□

□□□□□

- 配置
- 配置

3. TPS 配置

配置

- 配置
- 配置

配置

- 配置 `tps_limit`
- 配置
- 配置 `CONFIGURATION.md`

4. 配置

配置

- 配置
- 配置

配置

```
# 配置
sudo systemctl restart omnimessage-smpp

# 配置
sudo journalctl -u omnimessage-smpp -f
```

配置

配置

- `smpp_delivery_failures_total` 配置
- 配置 "submit_sm_resp" 配置

- 00000000

0000000000

1. 00000000

000

- 0000000000
- 0000000000

00000000

- 0x0000000B - 0000
- 0x00000001 - 000000
- 0x00000003 - 0000

000000

- 0000000000 E.164
- 000000000000
- 00000000

2. 00000000

000

- 0000
- 0000
- 00

000000

- GSM-7000 160 000
- UCS-2000 70 000
- 0000000000
- 00000000

3. 000000

注意

- 所有文件都必须有权限
- 所有文件都必须有权限

权限设置

- 所有文件都必须有权限
- 所有文件都必须有权限
- 所有文件都必须有权限/所有文件

4. 权限

注意

- 所有 `expires` 文件
- 所有文件

权限设置

- 所有文件
- 所有文件都必须有权限

Web UI 安装

安装 Web 应用

注意

- 所有文件都必须有权限 <https://your-server:8087>
- 所有文件

所有文件都必须有权限

1. 安装

注意

```
sudo systemctl status omnimessage-smpp
```

????

```
# 启动服务
sudo systemctl start omnimessage-smpp

# 查看日志
sudo journalctl -u omnimessage-smpp -n 50
```

2. 端口 8087

检查

```
sudo ufw status | grep 8087
# 否
sudo firewall-cmd --list-ports | grep 8087
```

配置

```
# Ubuntu/Debian
sudo ufw allow 8087/tcp

# RHEL/CentOS
sudo firewall-cmd --permanent --add-port=8087/tcp
sudo firewall-cmd --reload
```

3. SSL 证书

步骤

- 生成证书请求
- 生成证书

安装

- 安装证书

- 安装 SSL 证书
- 安装 omnimessage

```
ls -l /opt/omnimessage-smpp/priv/cert/
```

4. 设置 URL

设置

- 设置 HTTPS 端口 HTTP
- 设置 IP/端口
- 设置 8087

Web UI 设置

设置

- 设置
- 设置
- 设置

设置

1. 设置

- Ctrl+F5 刷新
- 清除 cookies

2. 设置

- 按 F12
- 禁用 JavaScript
- 禁用

3. 设置

- 按 Chrome/Firefox/Edge 按钮
- 设置

4. 確認

```
sudo journalctl -u omnimessage-smpp -f
```

確認

Prometheus 確認

確認

- `curl http://localhost:4000/metrics` 確認
- Prometheus 確認
- 確認

確認

1. 確認

確認

```
sudo systemctl status omnimessage-smpp
```

確認

```
sudo systemctl start omnimessage-smpp
```

2. 確認

確認

```
# 本地测试
curl http://localhost:4000/metrics

# 远程测试
curl http://your-server-ip:4000/metrics
```

验证步骤

- 确保 Prometheus 正在运行
- 确保 Prometheus 监听了 4000

3. 验证结果

预期

- 访问 `/metrics` 返回 Prometheus 的 `/stats`
- 访问 4000 端口返回 8087

常见问题

原因

- 防火墙阻止
- 端口冲突
- 配置错误

解决方案

1. 检查防火墙

- 检查防火墙规则
- 禁用防火墙
- Prometheus 配置 `increase()` 或 `rate()`

2. 检查端口

- 检查端口占用

- 消息队列
- 数据库

3. 部署

- 部署消息队列
- 部署数据库
- 部署 Prometheus 监控系统

部署

CPU 部署

部署

```
top -p $(pgrep -f omnimessage-smpp)
```

部署

- 部署
- 部署
- 部署

部署

- 部署
- 部署 TPS
- 部署 CPU

部署

部署

```
ps aux | grep omnimessage-smpp
```

測試環境

- 測試環境配置
- 測試數據生成

測試結果

- 測試結果分析
- 測試結果總結
- 測試結果建議

測試總結

測試

- 測試環境
- 測試數據
- 測試結果

測試

1. TPS 測試 - 測試環境
2. `queue_check_frequency` - 測試數據
3. 測試 API 測試 - 測試結果
4. 測試結果總結

測試結果

- 測試結果分析 TPS
 - 測試 `queue_check_frequency` 測試數據
 - 測試 API
 - 測試結果總結
-

環境構築

インストール

前提

- 64bit OS
- 最低 "syntax error" 以上 "parse error"

インストール

```
# Elixir をインストール
/opt/omnimessage-smpp/bin/omnimessage-smpp eval "File.read!(
'config/runtime.exs')"
```

確認

- map を確認
- " vs ' の区別
- 変数の宣言
- `import Config`

実行

- 実行
- 実行
- Elixir の実行

確認

確認

- 実行
- 実行

確認

```
# 重启 omnimessage-smpp
sudo systemctl restart omnimessage-smpp

# 查看 omnimessage-smpp 状态
sudo systemctl status omnimessage-smpp

# 查看 omnimessage-smpp 日志
sudo journalctl -u omnimessage-smpp -n 50
```

检查

检查

检查

1. 检查系统资源

```
# 检查磁盘空间
df -h

# 检查内存
free -h

# CPU 使用率
uptime
```

2. 检查 omnimessage-smpp 服务

```
sudo systemctl status omnimessage-smpp
```

3. 检查日志

```
sudo journalctl -u omnimessage-smpp -n 200
```

4. 確認

```
sudo systemctl restart omnimessage-smpp
```

5. 確認

- 確認
- SSL 確認
- 確認
- 確認

6. 確認

```
# 確認
sudo cp /opt/omnimessage-smpp/config/runtime.exs.backup \
/opt/omnimessage-smpp/config/runtime.exs

# 確認
sudo systemctl restart omnimessage-smpp
```

7. 確認

確認

確認

確認

1. 確認 `cat /opt/omnimessage-smpp/VERSION`

2. 確認

```
sudo journalctl -u omnimessage-smpp -n 200 > /tmp/smpp-logs.txt
```

3. 確認

```
sudo cp /opt/omnimessage-smpp/config/runtime.exs  
/tmp/config.exs  
# [] /tmp/config.exs []
```

4. []

```
curl http://localhost:4000/metrics > /tmp/metrics.txt
```

5. []

```
uname -a > /tmp/system-info.txt  
free -h >> /tmp/system-info.txt  
df -h >> /tmp/system-info.txt
```

[]

- [] support@omnitouch.com
- [] +61 XXXX XXXX 24/7
- [] []

[]

- **OPERATIONS.md** - []
- **CONFIGURATION.md** - []
- **MONITORING.md** - []
- **README.md** - []

