

# 安裝

安裝說明書

## 安裝

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

1. 安裝 OmniMessage 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 OmniMessage

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

# 配置
config :omnimessage_smpp,
  setting_name: value

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

配置

API\_BASE\_URL

OmniMessage Core 配置 URL

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

配置	配置	配置	配置
api_base_url	URL (URL)		-

OmniMessage Core 配置 URL 配置 REST API 配置 OmniMessage 配置

- 配置 SMPP 配置 OmniMessage 配置
- 配置 SMPP 配置
- 配置 OmniMessage
- 配置

配置“配置”配置 OmniMessage 配置

- ✓ 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

配置消息队列 SMPP ↔ OmniMessage 配置消息队列

配置

```
# 配置 IP 地址 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
```

項目	単位	型	初期値
smpp_poll_interval	ms	int	100

スmpp\_poll\_intervalは、スmppのポーリング間隔を指定します。

スmppのポーリング間隔は、スmppの稼働状況に応じて自動的に調整されます。

- スmppの稼働状況が**>100 TPS**の場合は、スmppのポーリング間隔は100-500msに調整されます。
- スmppの稼働状況が**10-100 TPS**の場合は、スmppのポーリング間隔は500-1000msに調整されます。
- スmppの稼働状況が**<10 TPS**の場合は、スmppのポーリング間隔は1000-2000msに調整されます。

スmppの稼働状況に応じて自動的に調整されるスmppのポーリング間隔は、スmppの稼働状況に応じて自動的に調整されます。

## VERIFY\_SSL\_PEER

SSLの稼働状況に応じて自動的に調整されるスmppのポーリング間隔は、スmppの稼働状況に応じて自動的に調整されます。

```
config :omnimessage_smpp,
  verify_ssl_peer: false
```

項目	単位	型	初期値
verify_ssl_peer	boolean	bool	false

スmppの稼働状況に応じて自動的に調整されるスmppのポーリング間隔は、スmppの稼働状況に応じて自動的に調整されます。

スmppの稼働状況に応じて自動的に調整されるスmppのポーリング間隔は、スmppの稼働状況に応じて自動的に調整されます。

- `true`の場合は、スmppの稼働状況に応じて自動的に調整されるスmppのポーリング間隔は、スmppの稼働状況に応じて自動的に調整されます。
- `false`の場合は、スmppの稼働状況に応じて自動的に調整されるスmppのポーリング間隔は、スmppの稼働状況に応じて自動的に調整されます。

スmppの稼働状況に応じて自動的に調整されるスmppのポーリング間隔は、スmppの稼働状況に応じて自動的に調整されます。

# SMSC\_NAME

Configuration

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

Field	Type	Value	Default
smpp_name	String	"smpp_gateway"	

Configuration

Configuration SMSC\_NAME

## SMPP Configuration

Configuration ESME SMSC Configuration

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	Carrier
Carrier	IP	2775	1-65535

Carrier SMPP Carrier TCP Carrier

Carrier 2775

Carrier

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

system\_id

Carrier



項目	単位	値
会社名	文字列	"company_user"

会社名を指定します。

会社名 - 会社名

## password

パスワード

項目	単位	値
パスワード	文字列	"secret_password"

パスワードを指定します。

パスワード

- パスワード
- パスワード
- パスワード

## tps\_limit

TPS制限

項目	単位	値	範囲
TPS制限	文字列	100	1-10000

TPS制限を指定します。

TPS制限

- TPS制限 70-80%
- TPS制限/TPS制限

- 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	值	说明
合作伙伴名称	字符串	"partner_acme"

合作伙伴名称

合作伙伴名称/系统ID

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"

监听地址列表

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 0 の場合は、接続数制限が機能しない
  - 設定値が 10000 を超える場合は、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.companay.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
  }
]
```

3

```
import Config

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

# 客户端
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
  }
]

# 服务器
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

安装依赖库

## 编译

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

编译成功后，生成可执行文件

## 运行

```
# 运行应用
sudo -u omnimessage-smpp /opt/omnimessage-smpp/bin/omnimessage-smpp console
```

按 `Ctrl+C` 退出

## 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

---

## 設定

- [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 □□□□

00	SMPP	00000000	ESME	0000	SMSC	00000000	SMPP	00000000000000000000
----	------	----------	------	------	------	----------	------	----------------------

SMPP □□□□□□□□

- 
- 
- 

[illegible]

## D

□□□□□□□□ SMPP □□□GSM-7□UCS-2 □□□

[illegible]

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

SMPP

□□□□ SMPP □□□□□□□□□□□□□□□□□□□□

# E

[illegible]

□□□□□□□□ SMPP □□□





**ISDN** 0000000000  
00000000000000

**J**

(0000)

**K**

0000  
00000000enquire\_link00000000000000

**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  
0000000000000000

## MT

MT 消息

## MSISDN

移动用户号码

## N

### NPI

NPI 号码 SMPP 消息 ISDN 消息

## O

消息

消息

消息

消息

## P

### PDU

消息 SMPP 消息 submit\_sm deliver\_sm

### Prometheus

消息

## Q

消息

消息

消息

消息

□□□□□□□□ SMPP □□□□□□

1111

TPS

## SMPP deliver\_sm

□□□□□□ SMPP □□□

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

□□□□□

**ESMEs**

□□□□□□□□ SMPP □□□

SMPP□□□□□□□□

□□□□□□□□ SMS □□□□□□□□□□

SMSC□□□□□□□□

□ SMPP □□□□□□□□ ESME□□□□□□□□□□ SMS □□□□□□□□□□□□□□□□□ □□□□□ □□□□□□□□  
□□□□□□ ESME □□□□□ SMSC□

## SSL/TLS

□ □ □ □ □ □ □ □ □ □

**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 Integration

### Prerequisites

SMPP integration requires Prometheus to be installed and running on your system.

OmniMessage Core provides an **OmniMessage** API that can be used to integrate with Prometheus.

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

### Installation

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

OmniMessage Prometheus integration

OmniMessage Prometheus integration on localhost

### Usage

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

### Configuration

OmniMessage `smpp_` configuration



```
- 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 - 请及时续费"
```

## 配置项

### smpp\_connection\_status

配置 Gauge

配置 SMPP 连接状态

配置

- 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

配置

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

配置

- 000000000000
- 000000000000
- 00000000

00000

## smpp\_messages\_sent\_total

Counter

SMPP00000000

00

connection\_status

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

- 00000000/00
- 0000/000000
- 0000000000

## smpp\_messages\_received\_total

Counter

SMPP00000000

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

Connection Attempts

48

smpp\_connection\_attempts\_total{bind\_name="vodafone\_uk",...} 48

- Connection Attempts
- Connection Attempts

smpp\_disconnection\_total

Counter

Disconnections

3

smpp\_disconnection\_total{bind\_name="vodafone\_uk",...} 3

- Disconnections
- Disconnections
- Disconnections

Uptime

smpp\_uptime\_seconds

Gauge

SMPP Uptime

📄

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

📄

- 📄📄📄📄
- 📄📄📄📄📄📄📄
- 📄📄📄📄📄📄

## OmniMessage API📄📄📄

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

📄📄**OmniMessage**📄📄📄📄📄📄📄

- omnimessage\_api\_requests\_total - 📄📄📄📄API📄📄📄
- omnimessage\_api\_request\_duration\_seconds - API📄📄📄
- omnimessage\_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

Dashboard

Query

Panel

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

Panel Stat

Panel

- `smpp_connection_status < 1`
- `smpp_connection_status == 1`

Panel

Panel

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

Panel Graph

Panel

Panel `{{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分钟内失败 {{ {{ $value }} }} 次"

    # 无流量
    - 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="omnimessage-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{{ $labels.bind_name }}API_BASE_URL{{ $labels.bind_name }}"

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

# 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分以内に送信受信0omniMessage APIエラー"

```

prometheus.yml

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




## Web

Web UI Prometheus

URL <https://your-server:8087>

SMPP →

- 送信受信
- 送信
- 送信受信
- 送信/受信
- 5送信

- 
- 
- 

□□□□□□

- 00000000000000000000
- 00000000000000000000000000/000000
- 00000000000000000000000000/000000
- 0000000000000000000000000000000000

□ □ □ □

□□□□□

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

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

# 1時間以内
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`

- 配置数据库驱动

## 2. 配置日志

- 配置SMPP → 配置
- 配置日志“配置”配置
- 配置日志配置

## 3. 配置数据库

- 配置数据库
- 配置数据库配置
- 配置数据库配置

## 4. 配置日志

- 配置数据库
- 配置数据库配置
- 配置数据库配置

## 5. 配置 Prometheus 配置

- `curl http://localhost:4000/metrics`
- 配置 Grafana 配置
- 配置数据库配置

□□□□

□□□□□□□□

- □□□□> 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 📖

📖

📖 **AI** **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. 設定ファイルの編集

```
config :omnimessage_smpp, :binds, [  
  # 設定...  
  
  # 設定  
  %{  
    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. サービスの再起動

```
# サービスを nano で Ctrl+X, Y, Enter  
  
# サービス  
sudo systemctl restart omnimessage-smpp
```

## 4. 確認

- サービスSMPP → サービス
- サービス
- サービス“サービス”サービス
- サービスサービス

## 5. サービス

- サービス

- 設置する `dest_smsc` 属性
- 属性値
- 属性名

## 属性値

属性値は **SMSC** 属性値と **ESMEs** 属性値

属性

### 1. 属性

- 属性 ID `partner_name`
- 属性名
- 属性値

### 2. 属性

- 属性 IP 属性
- 属性 TPS 属性
- 属性

属性

属性 **A** **Web UI** 属性

属性

属性

### 1. 属性

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

### 2. 属性

- 属性“属性”
- 属性
  - 属性 `partner_acme` 属性

- 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
- 各 IP 設定ファイル

各 **B**設定ファイル

設定ファイルの作成

例

### 1. 設定ファイル

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

### 2. 設定ファイルの内容

```
# 各 server_binds 設定
config :omnimessage_smpp, :server_binds, [
  # 各設定...

  # 各設定
  %{
    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
  }
]

# 各 listen 設定
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. 設定する

- IP 設定
- ポート `2775`
- ID `acme_corp`
- パスワード `secure_password_123`
- 設定完了

#### 5. 確認する

- 設定 SMPP → 確認
- 接続確認
- 送信確認
- IP 設定確認

### インストール

インストールするTPS のIP 設定

インストール

Web UI 設定

インストール

設定

#### 1. 設定

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

#### 2. 確認

- 接続確認

- 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. 重启服务

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

### 4. 验证

- 验证 SMPP → 验证
- 验证
- 验证
- 验证

## 部署

### 部署 SMPP 服务

#### 步骤

#### 1. 部署

- 部署/部署
- 部署

#### 2. 使用 Web UI 部署

- 验证 SMPP → 验证
- 验证
- 验证“验证”
- 验证

#### 3. 验证



- SMPP → /
- 
- “”
- 

4.

- -
- 

1.

- 
- 

2.

- 消息
- 消息
  - 消息
  - 消息
  - 消息 SMSC (dest\_smsc)
  - 消息
  - 消息

### 3. 消息

- 消息
- 消息 SMSC 消息

消息

消息

消息

### 1. 消息

- 消息 SMPP → 消息
- 消息
- 消息 消息

### 2. 消息

- 消息
- 消息
- 消息 dest\_smsc 消息
- 消息 deliver\_after 消息

### 3. 消息

- 消息 = 消息
- 消息
- 消息

#### 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

目录

### 1. 部署

- 配置 SMPP → 部署
- 配置 ID 验证
- 配置

### 2. IP 地址

- 配置 IP 地址
- 配置 IP 地址

### 3. 配置

- 配置
- 配置
- 配置

### 4. 部署

- 配置 Web UI 部署
- 配置“配置”部署

部署

部署 Prometheus 部署

部署

```
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. ログの確認

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

#### 4. 環境の確認

- サービスの有無
- SSL のインストール
- ディスク容量 `df -h`
- メモリ容量 `free -h`

#### 5. サービスの再起動

□ □ □ □ □ □ □ □ □

111




1. □□□□□□□

- 四角SMPP → 四角
- 四角四角四角
- 四角“四角四角”

2. □□□□□

- ☐ ☐☐☐☐☐☐☐
- ☐ ☐☐☐☐☐☐☐
- ☐ ☐☐☐☐☐☐

3. □□□□□

- ☐ 
- ☐ 
- ☐ 

4. □□□□□□□□

- ☐ 
- ☐ 
- ☐ 

555

□□□□□□□□□□

111

## 1. TPS

- 國際SMPP → 國際
- 國際
- 國際 TPS 國際

## 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
  - ☐ 常见问题
- 

# 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

Our SMPP implementation is based on OmniMessage Core. Our OmniMessage implementation is based on the following:

## OmniMessage XXXXXXXX

- XXXXXXXXXXXXXXXX
- XXXXXXXXXXXXXXXX
- Our API XXXXXXXX
- XXXXXX "Connection refused", "Timeout", "HTTP 503", "Connection reset"

XX

### 1. Our OmniMessage XXXXX

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

# XXXXXXXXXXXXX
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
3. 確認可能確認可能 `SMPP_POLL_INTERVAL`
4. 確認 OmniMessage 確認可能

### 確認 OmniMessage 確認可能確認可能

1. 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 `traceroute omnimessage-core.example.com`
4. 確認 SSL 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 HTTPS
3. 確認可能 `sudo systemctl restart omnimessage-smpp`

# 問題

## 問題文

問題

- 状態 "Disconnected" になる
- 接続が切断される
- 再接続する

問題の背景

### 1. 接続確認

確認

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

# ping
ping -c 3 smpp.carrier.com

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

確認

- DNS による IP 取得確認
- ping による接続確認
- telnet による接続確認

### 2. 接続

接続

- 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

- 配置项 30
- 配置项
- 配置项

### 3. 配置 TPS 配置

配置

- 配置项
- 配置项

配置项

- 配置 `tps_limit` 配置
- 配置 TPS 配置项 70-80%
- 配置项

### 4. 配置项

配置

- 配置项
- 配置项

配置项

- 配置项
- 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
- 性能测试

### 3. TPS 配置

配置

- 配置
- 配置

配置

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

### 4. 性能测试

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

API

- 认证和授权
- 数据格式

API

- 认证和授权
- 数据格式
- 错误处理和状态码

#### 4. 部署

API

- 设置 `expires` 头
- 配置 CORS

API

- 部署到生产环境
- 性能优化

## Web UI 开发

### 开发 Web 应用

API

- 使用 `https://your-server:8087` 作为 API 端点
- 配置 CORS

部署到生产环境

#### 1. 部署

API

```
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
- 安装证书

命令

- 生成证书

- 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 设置
- 设置



#### 4. 確認

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

---

確認

## Prometheus 確認

確認

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

- 使用 curl 命令测试接口
- 确保 Prometheus 端口 4000 已开放

### 3. 部署

配置

- 在 `/metrics` 目录下创建 `/prometheus` 和 `/stats` 目录
- 设置 Prometheus 监听 8087 端口

部署步骤

配置

- 创建配置文件
- 安装 Prometheus
- 启动服务

验证

#### 1. 本地测试

- 使用 curl 命令测试
- 验证 Prometheus 是否正常运行
- Prometheus 提供 `increase()` 和 `rate()` 函数

#### 2. 远程部署

- 部署到服务器

- 消息队列
- 消息中间件

3. 部署

- 部署环境
- 部署步骤
- 使用 Prometheus 监控

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
```

## 環境構築

- 仮想マシン構築
- ネットワーク構築

## 基本操作

- 仮想マシン起動
- ネットワーク接続
- 仮想マシン停止

## 応用操作

### 基本

- 仮想マシン起動
- ネットワーク
- 仮想マシン停止

### 応用

1. TPS 測定 - 仮想マシン
2. `queue_check_frequency` - 仮想マシン
3. API 呼び出し - 仮想マシン
4. 仮想マシン停止

## まとめ

- 仮想マシン構築と TPS
  - `queue_check_frequency` の確認
  - API の呼び出し
  - 仮想マシン停止
-

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

#### 確認

- map 関数
- " vs ' の区別
- 変数宣言
- `import Config`

#### 実行

- 実行コマンド
- 実行結果
- Elixir のインストール確認

### 開発環境

#### 前提

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

#### インストール

```
# 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](mailto:support@omnitouch.com)
- [] +61 XXXX XXXX 24/7
- [] []

### []

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



