

# Coverity Connect MCP Server - 統合セットアップガイド

## 概要

このガイドでは、Coverity Connect MCP Serverの開発環境構築から本番環境デプロイまでの全手順を説明します。

## 前提条件

### システム要件

- **Python:** 3.8 以上
- **OS:** Windows 10/11, macOS, Linux
- **Claude Desktop:** 最新版
- **Git:** バージョン管理用

### 必要な権限

- Coverity Connect サーバーへのアクセス権限
- 管理者権限（パッケージインストール用）

---

## 開発環境セットアップ

### ステップ1: リポジトリのクローン

```
bash

# GitHubからクローン
git clone https://github.com/keides2/coverity-connect-mcp.git
cd coverity-connect-mcp
```

### ステップ2: Python環境の準備

```
bash

# 仮想環境作成（推奨）
python -m venv venv

# 仮想環境有効化
# Windows:
venv\Scripts\activate
# macOS/Linux:
source venv/bin/activate
```

## ステップ3: パッケージインストール

```
bash

# 開発用インストール
pip install -e .

# 開発用依存関係
pip install -e ".[dev]"
```

## ステップ4: 開発環境設定

### 4.1 環境変数設定 (Windows)

```
powershell

# テスト用ダミーサーバー設定
$env:COVERITY_HOST="localhost"
$env:COVERITY_PORT="5000"
$env:COVERITY_SSL="False"
$env:COVAUTHUSER="dummy_user"
$env:COVAUTHKEY="dummy_key"
$env:MCP_DEBUG="True"
$env:LOG_LEVEL="DEBUG"
```

### 4.2 環境変数設定 (macOS/Linux)

```
bash

# テスト用ダミーサーバー設定
export COVERITY_HOST="localhost"
export COVERITY_PORT="5000"
export COVERITY_SSL="False"
export COVAUTHUSER="dummy_user"
export COVAUTHKEY="dummy_key"
export MCP_DEBUG="True"
export LOG_LEVEL="DEBUG"
```

## ステップ5: ダミーサーバー起動

### 5.1 手動起動

```
bash

# ダミーサーバー起動
python examples/development/mock_server.py
```

## 5.2 便利スクリプト使用（Windows）

```
powershell  
  
# ダミーサーバー起動  
.\scripts\start_mock_server.bat
```

## ステップ6: MCPサーバー起動

### 6.1 手動起動

```
bash  
  
# 新しいターミナルで  
python -m coverity_mcp_server
```

### 6.2 便利スクリプト使用（Windows）

```
powershell  
  
# MCP開発サーバー起動  
.\scripts\start_dev.bat
```

## ステップ7: Claude Desktop設定

### 7.1 設定ファイル場所

#### Windows:

- `%APPDATA%\Claude\claude_desktop_config.json`
- または `%LOCALAPPDATA%\Claude\claude_desktop_config.json`

#### macOS:

- `~/Library/Application Support/Claude/claude_desktop_config.json`

#### Linux:

- `~/.config/claude/claude_desktop_config.json`

### 7.2 開発環境設定の追加

`examples/development/claude_desktop_config.json`の内容をClaude Desktopの設定ファイルに追加：

```
json
```

```
{
  "mcpServers": {
    "coverity-connect-dev": {
      "command": "python",
      "args": ["-m", "coverity_mcp_server"],
      "cwd": "/path/to/coverity-connect-mcp",
      "env": {
        "COVERITY_HOST": "localhost",
        "COVERITY_PORT": "5000",
        "COVERITY_SSL": "False",
        "COVAUTHUSER": "dummy_user",
        "COVAUTHKEY": "dummy_key",
        "MCP_DEBUG": "True",
        "LOG_LEVEL": "DEBUG"
      }
    }
  }
}
```

## ステップ8: 開発環境テスト

### 8.1 基本接続テスト

Claude Desktopで以下を実行：

Coverity MCPサーバーに接続できていますか？利用可能なツールとリソースを教えてください。

### 8.2 機能テスト

Coverityのプロジェクト一覧を取得してください

Coverityで重要度がHighの欠陥を検索してください

## ステップ9: 単体テスト実行

bash

```
# 全テスト実行
```

```
pytest
```

```
# カバレッジ付きテスト
```

```
pytest --cov=coverity_mcp_server
```

```
# 特定テストのみ
```

```
pytest tests/test_tools.py
```

## 本番環境デプロイ

### ステップ1: 本番環境準備

#### 1.1 Python環境セットアップ

```
bash
```

```
# 本番サーバーで
```

```
python -m venv /opt/coverity-mcp
```

```
source /opt/coverity-mcp/bin/activate
```

```
# または systemd 用
```

```
python -m venv ~/.local/share/coverity-mcp
```

#### 1.2 パッケージインストール

```
bash
```

```
# 本番用インストール
```

```
pip install -e .
```

```
# または PyPI からインストール (将来)
```

```
# pip install coverity-connect-mcp
```

### ステップ2: 本番環境設定

#### 2.1 環境変数設定

```
bash
```

```
# 実際のCoverity Connect設定
export COVERITY_HOST="your-coverity-server.company.com"
export COVERITY_PORT="8080"
export COVERITY_SSL="True"
export COVAUTHUSER="your-actual-username"
export COVAUTHKEY="your-actual-auth-key"
export MCP_DEBUG="False"
export LOG_LEVEL="INFO"
```

## 2.2 環境設定ファイル作成

```
bash

# 本番用環境ファイル作成
cp examples/production/.env.production /opt/coverity-mcp/.env
# 実際の値に編集
nano /opt/coverity-mcp/.env
```

## ステップ3: systemd サービス設定 (Linux)

### 3.1 サービスファイル作成

```
bash

sudo nano /etc/systemd/system/coverity-mcp.service
```

```
ini

[Unit]
Description=Coverity Connect MCP Server
After=network.target

[Service]
Type=simple
User=mcp-user
WorkingDirectory=/opt/coverity-mcp
Environment=PATH=/opt/coverity-mcp/bin
EnvironmentFile=/opt/coverity-mcp/.env
ExecStart=/opt/coverity-mcp/bin/python -m coverity_mcp_server
Restart=always
RestartSec=3

[Install]
WantedBy=multi-user.target
```

## 3.2 サービス有効化

```
bash

# サービス有効化
sudo systemctl daemon-reload
sudo systemctl enable coverity-mcp
sudo systemctl start coverity-mcp

# ステータス確認
sudo systemctl status coverity-mcp
```

## ステップ4: 本番Claude Desktop設定

### 4.1 本番設定の適用

`examples/production/claude_desktop_config.json`を参考に、実際の値で設定：

```
json

{
  "mcpServers": {
    "coverity-connect": {
      "command": "python",
      "args": ["-m", "coverity_mcp_server"],
      "cwd": "/opt/coverity-mcp",
      "env": {
        "COVERITY_HOST": "your-coverity-server.company.com",
        "COVERITY_PORT": "8080",
        "COVERITY_SSL": "True",
        "COVAUTHUSER": "your-username",
        "COVAUTHKEY": "your-auth-key",
        "MCP_DEBUG": "False",
        "LOG_LEVEL": "INFO"
      }
    }
  }
}
```

## ステップ5: 本番環境テスト

### 5.1 接続テスト

```
bash

# サーバー側でテスト
python -c "from coverity_mcp_server.coverity_client import CoverityClient; print('Import OK')"
```

## 5.2 Claude Desktop統合テスト

Claude Desktopで実際のCoverityデータにアクセス：

実際のCoverityプロジェクト一覧を表示してください

## ステップ6: 監視・ログ設定

### 6.1 ログ設定

```
bash

# ログディレクトリ作成
sudo mkdir -p /var/log/coverity-mcp
sudo chown mcp-user:mcp-user /var/log/coverity-mcp

# logrotateの設定
sudo nano /etc/logrotate.d/coverity-mcp
```

### 6.2 ヘルスチェック

```
bash

# 定期ヘルスチェックスクリプト
nano /opt/coverity-mcp/health_check.sh
```

## トラブルシューティング

### 開発環境

#### パッケージインストールエラー

```
bash

# 依存関係の更新
pip install --upgrade pip setuptools wheel
pip install -e . --force-reinstall
```

#### ダミーサーバー接続エラー

```
bash
```



```
# Flaskインストール確認
```

```
pip install flask
```

```
# ポート使用状況確認
```

```
netstat -an | grep :5000
```

## Claude Desktop接続失敗

1. 設定ファイルの構文確認
2. パス設定の確認
3. Claude Desktopの再起動

## 本番環境

### Coverity Connect接続エラー

```
bash
```

```
# 接続テスト
```

```
curl -k https://your-coverity-server:8080/ws/v9/configurationservice?wsdl
```

```
# 認証確認
```

```
echo $COVAUTHUSER
```

```
echo $COVAUTHKEY
```

## パフォーマンス問題

```
bash
```

```
# ログレベル調整
```

```
export LOG_LEVEL="WARNING"
```

```
# systemd設定最適化
```

```
sudo systemctl edit coverity-mcp
```

## 関連ドキュメント

- 開発詳細: `examples/development/README_dev.md`
- プロジェクト完了報告: `PROJECT_COMPLETION_REPORT.md`
- API仕様: `src/coverity_mcp_server/`
- テスト手順: `tests/README.md` (作成予定)

## サポート

- **GitHub Issues:** <https://github.com/keides2/coverity-connect-mcp/issues>
  - **Discussions:** <https://github.com/keides2/coverity-connect-mcp/discussions>
  - **ドキュメント:** プロジェクトREADME.md
- 

**更新日:** 2025年7月19日

**バージョン:** 1.0.0