

Coverity Connect MCP Server - GitHub Repository Structure

📁 推奨ディレクトリ構成

coverity-connect-mcp/

- └── .github/
 - └── workflows/
 - └── test.yml # CI/CDテスト
 - └── publish.yml # パッケージ公開
 - └── security.yml # セキュリティスキャン
 - └── ISSUE_TEMPLATE/
 - └── bug_report.md
 - └── feature_request.md
 - └── security_issue.md
 - └── pull_request_template.md
- └── src/
 - └── coverity_mcp_server/
 - └── __init__.py
 - └── main.py # メインMCPサーバー
 - └── client.py # Coverity Connect クライアント
 - └── tools.py # MCPツール実装
 - └── resources.py # MCPリソース実装
 - └── prompts.py # MCPプロンプト実装
 - └── config.py # 設定管理
 - └── tests/
 - └── __init__.py
 - └── test_client.py
 - └── test_tools.py
 - └── test_integration.py
 - └── fixtures/
 - └── sample_projects.json
 - └── sample_defects.json
 - └── test_config.yaml
- └── docs/
 - └── installation.md
 - └── configuration.md
 - └── usage.md
 - └── api-reference.md
 - └── troubleshooting.md
 - └── examples/
 - └── basic-usage.md
 - └── ci-cd-integration.md
 - └── security-analysis.md
- └── examples/
 - └── claude_desktop_config.json # Claude Desktop設定例
 - └── docker-compose.yml # Docker実行例
 - └── config_templates/
 - └── projects.cfg.example
 - └── project_template.cfg
 - └── scripts/

```
| |—— setup.sh
| |—— setup.bat
| |—— health_check.py
|—— .gitignore
|—— .env.example          # 環境変数テンプレート
|—— pyproject.toml        # Python パッケージ設定
|—— requirements.txt      # 依存関係
|—— requirements-dev.txt  # 開発用依存関係
|—— Dockerfile           # Docker対応
|—— docker-compose.yml   # 開発環境
|—— LICENSE               # ライセンス (MIT推奨)
|—— README.md            # プロジェクト説明
|—— CHANGELOG.md         # 変更履歴
|—— CONTRIBUTING.md      # コントリビューションガイド
|—— SECURITY.md           # セキュリティポリシー
|—— manifest.json        # DXT用マニフェスト
```

マーケットプレイス公開について

対象マーケットプレイス

1. MCP Server Registry (公式)

- URL: <https://github.com/modelcontextprotocol/servers>
- 要件: オープンソース、MIT/Apache ライセンス
- 公開方法: Pull Request

2. PyPI (Python Package Index)

- パッケージ名: `coverity-connect-mcp`
- インストール: `pip install coverity-connect-mcp`

3. npm (Node.js Package Manager)

- DXTパッケージとして公開
- Claude Desktop Extension Store対応

4. Docker Hub

- コンテナイメージとして配布
- 企業環境での利用を想定

公開要件

✓ 必須要件

- オープンソースライセンス (MIT/Apache 2.0)
- セキュリティベストプラクティス準拠
- 包括的なドキュメント

- テストカバレッジ 80%以上
- CI/CD パイプライン

✅ 推奨要件

- Docker対応
- 複数OS対応 (Windows/macOS/Linux)
- 設定例とチュートリアル
- セキュリティ監査対応
- コミュニティサポート

📋 ライセンス考慮事項

推奨: MIT License

MIT License

Copyright (c) 2025 [Your Name]

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

知的財産権への配慮

- Coverity は Synopsys の商標
- Coverity Connect API の利用は適切な認証下で実行
- あなたの独自実装ロジックは保護される
- 企業秘密情報は除外

公開戦略

フェーズ1: GitHub公開 (即座に実行可能)

1. パブリックリポジトリ作成
2. 基本実装とドキュメント
3. CI/CDパイプライン設定
4. コミュニティフィードバック収集

フェーズ2: MCP Registry登録 (1-2週間後)

1. MCP公式レジストリにPR提出
2. コードレビュー対応
3. セキュリティ監査対応
4. 公式認定取得

フェーズ3: パッケージ配布 (1ヶ月後)

1. PyPI パッケージ公開
2. Docker Hub イメージ公開
3. DXT Extension作成
4. 企業向けサポート体制

フェーズ4: エコシステム拡張 (3ヶ月後)

1. プラグインアーキテクチャ
2. サードパーティ統合
3. エンタープライズ機能
4. 商用サポート検討

収益化可能性

オープンソース戦略

- **基本機能:** 完全無料・オープンソース
- **プレミアム機能:**
 - 企業向けサポート
 - 高度な分析機能
 - カスタム統合
 - トレーニング・コンサルティング

ビジネスモデル例

1. **フリーミアム**: 基本無料、高度機能有料
2. **サポート型**: 実装サポート・保守契約
3. **SaaS型**: クラウドホスト版提供
4. **コンサルティング**: Coverity導入支援

成功指標

技術指標

- GitHub Stars: 100+ (6ヶ月目標)
- PyPI Downloads: 1,000+/月
- Docker Pulls: 500+/月
- Issues Response Time: 48時間以内

コミュニティ指標

- Contributors: 10+
- Forks: 50+
- Community Discussions: アクティブ
- Documentation Quality: 高評価

この構成により、あなたのCoverity専門知識を活かした価値あるオープンソースプロジェクトとして、広く活用される可能性があります！