

Publishing the DevOps MCP Server

This guide explains how to package and publish the DevOps MCP Server to PyPI so others can install it using `uv` or `uvx`.

Prerequisites

- Python 3.12 or higher
- `uv` package manager
- PyPI account (if publishing to PyPI)

Packaging and Publishing

We've provided a script that automates the build and publish process:

```
./build_and_publish.sh
```

This script will:

1. Check for required dependencies
2. Clean previous builds
3. Build the package (both wheel and source distribution)
4. Optionally publish to TestPyPI for testing
5. Optionally publish to PyPI for production use

Manual Publishing Steps

If you prefer to do the process manually:

1. Install build tools

```
uv pip install build twine
```

2. Clean previous builds

```
rm -rf dist/ build/ *.egg-info/
```

3. Build the package

```
python -m build
```

4. Check the package

```
twine check dist/*
```

5. Upload to TestPyPI (optional)

```
twine upload --repository testpypi dist/*
```

6. Upload to PyPI

```
twine upload dist/*
```

Versioning

When releasing a new version:

1. Update the version number in `pyproject.toml`
2. Update the changelog if applicable
3. Commit the changes
4. Tag the release: `git tag v0.1.0`
5. Push the tag: `git push origin v0.1.0`
6. Run the build and publish process

Installation Instructions for Users

After publishing, users can install the package using:

Using uv

```
uv pip install msrashed100-devops-mcp
```

Using uvx (for one-off execution)

```
uvx msrashed100-devops-mcp
```

Troubleshooting

- If you encounter permission errors when uploading to PyPI, ensure you're logged in with **twine login**
- If dependencies are missing, check the **dependencies** list in **pyproject.toml**
- For TestPyPI, you may need to create a separate account at <https://test.pypi.org/>