

Local Installation Guide for msrashed100-devops-mcp

Since we encountered issues publishing to PyPI, here's how to install and use the package locally on different machines.

Option 1: Direct Installation from Wheel File

The simplest way to install the package on another machine is to copy the wheel file and install it directly:

1. Copy the wheel file to the target machine:

```
# From the dist/ directory
msrashed100_devops_mcp-0.1.0-py3-none-any.whl
```

2. Install it using uv:

```
uv pip install /path/to/msrashed100_devops_mcp-0.1.0-py3-none-any.whl
```

3. Run the MCP server:

```
msrashed100-devops-mcp
```

Option 2: Create a Local Package Repository

—
PROF

You can create a simple local package repository to make installation easier:

1. Create a directory to serve as your repository:

```
mkdir -p ~/local-pypi
```

2. Copy the wheel file to this directory:

```
cp dist/msrashed100_devops_mcp-0.1.0-py3-none-any.whl ~/local-pypi/
```

3. Install from this local directory:

```
uv pip install --find-links ~/local-pypi msrashed100-devops-mcp
```

Option 3: Host on a Private PyPI Server

For team use, you can set up a private PyPI server:

1. Install a private PyPI server like [pypiserver](#):

```
pip install pypiserver
```

2. Start the server:

```
pypi-server -p 8080 ~/local-pypi/
```

3. Install from your private PyPI server:

```
uv pip install --index-url http://localhost:8080 msrashed100-devops-mcp
```

Option 4: Use Git Repository

You can also install directly from a Git repository:

1. Push your code to a Git repository (GitHub, GitLab, etc.)
2. Install using uv with the repository URL:

```
uv pip install git+https://github.com/yourusername/devops-mcp.git
```

Using with Claude or Other AI Assistants

To use this MCP server with Claude or other AI assistants:

1. Install the package using one of the methods above
2. Create an MCP configuration file (e.g., [mcp_settings.json](#)):

```
{  
  "mcpServers": {  
    "devops-server": {
```

```
"command": "msrashed100-devops-mcp",
"env": {
  "FASTMCP_LOG_LEVEL": "INFO",
  "AWS_PROFILE": "your-profile-name",
  "AWS_REGION": "us-east-1",
  "KUBECONFIG": "/path/to/your/kubeconfig"
},
"disabled": false,
"autoApprove": []
}
}
```

3. Configure your AI assistant to use this MCP server

Troubleshooting

- If you encounter dependency issues, make sure Python 3.12+ is installed on the target machine
- Check that all required environment variables are set correctly
- For Kubernetes tools, ensure the kubeconfig file is properly configured
- For AWS tools, make sure AWS credentials are properly set up