

HCLSoftware

DevOps Velocity MCP Server - Tech Preview



Contents

- Chapter 1. DevOps Velocity MCP Server - Tech**
- Preview..... 3**
 - MCP server setup in DevOps Velocity..... 4
 - Connecting to the DevOps Velocity MCP server using Claude Desktop..... 4
 - Connecting to the DevOps Velocity MCP server using GitHub Copilot in Visual Studio Code..... 6

Chapter 1. DevOps Velocity MCP Server - Tech Preview

A Model Context Protocol (MCP) server provides a structured way for GenAI clients to access tools, data, and actions through a consistent, secure interface. When you connect a GenAI client that supports MCP to the HCL DevOps Velocity (Velocity) MCP server, you can use natural-language requests to retrieve information from your Velocity.



Disclaimer:

This release contains access to the MCP server feature in Velocity as a Tech Preview and can be enabled by passing the `betaFeature=mcp` parameter while running the installer. The Tech Preview is intended for you to view the capabilities of MCP server offered by Velocity, and to provide your feedback to the product team. You are permitted to use the information only for evaluation purposes and not for use in a production environment. HCL® provides the information without obligation of support and "as is" without warranty of any kind.

The Velocity MCP server exposes predefined queries that let you view Value Stream Management data, metrics, build and deployment details, workflow gate statuses, and information about teams, applications, and workspaces. This capability provides a simple, conversational way to obtain operational insights without requiring navigation of the Velocity interface or creation of API requests.

The Velocity MCP server offers the following tools:

- applicationById
- builds
- deployments
- latestWorkflowGateChecks
- metrics
- teamsForCurrentUser
- teamspacesForTenant
- workflow
- workflowsForTenant
- workflowStatsByIdQuery

To use the Velocity MCP server, you must enable the MCP server feature flag when you run the installer. See [MCP server setup in DevOps Velocity on page 4](#).

You can currently connect to the Velocity MCP server using the following GenAI clients:

- Claude Desktop. See [Connecting to the DevOps Velocity MCP server using Claude Desktop on page 4](#)
- GitHub Copilot in Visual Studio Code. See [Connecting to the DevOps Velocity MCP server using GitHub Copilot in Visual Studio Code on page 6](#).

MCP server setup in DevOps Velocity

You can enable the Model Context Protocol (MCP) server in HCL DevOps Velocity (Velocity) to connect with GenAI clients, such as Claude Desktop or GitHub Copilot in VS Code. With this connection you can interact with Velocity and retrieve information, such as release details or value stream data.

To set up the MCP server, you must install Velocity on your Kubernetes or OpenShift cluster, generate a User Access Key (UAK), and configure the MCP server in your preferred GenAI client to use Velocity features through natural language interactions.

Perform the following steps to enable the MCP server in Velocity:

1. Open the terminal or command prompt and run the installation file with `betaFeature=mcp` parameter as shown in the following example.

```
./<path_to_installation_file>/<installation_file_name> --betaFeature=mcp
```

2. Install or upgrade Velocity on a Kubernetes or OpenShift platform.

For detailed instructions, see:

- [Installing on Kubernetes](#)
- [Installing on OpenShift](#)
- [Installing on Google Kubernetes Engine](#)
- [Upgrading Kubernetes-based installations](#)
- [Upgrading an OpenShift installation using Helm charts](#)

3. Log in to Velocity and create a user access key. See [Velocity](#).
4. Configure the Velocity MCP server in different Gen AI clients.

For detailed instructions, see:

- [Connecting to the DevOps Velocity MCP server using Claude Desktop on page 4](#)
- [Connecting to the DevOps Velocity MCP server using GitHub Copilot in Visual Studio Code on page 6](#)

Connecting to the DevOps Velocity MCP server using Claude Desktop

You can connect to the HCL DevOps Velocity (Velocity) Model Context Protocol (MCP) server by using the Claude Desktop application to interact with Velocity and retrieve the necessary data and insights.

Before you begin

You must have completed the following tasks:

- Installed Velocity and enabled the MCP server. See [MCP server setup in DevOps Velocity on page 4](#).
- Obtained the MCP Server URL. For example: `https://<velocity_hostname>/mcp`
- Ensured that Claude Desktop is installed.

You must have

1. Open Claude Desktop.
2. Go to **My Profile** → **Settings** → **Developer**.
3. Configure the MCP server by adding the JSON configuration.

```
{
  "mcpServers": {
    "velocity-mcp-server": {
      "command": "npx",
      "args": [
        "-y",
        "mcp-remote",
        "<velocity_mcp_server_url>",
        "-k",
        "--allow-http",
        "--header",
        "Authorization:${AUTH_HEADER}"
      ],
      "env": {
        "AUTH_HEADER": "UserAccessKey <velocity_user_access_key>",
        "NODE_TLS_REJECT_UNAUTHORIZED": "0"
      }
    }
  }
}
```



Notes:

- The `-k` argument allows self-signed certificates. Use it only if required.
- The `--allow-http` option enables non-https communication.
- The environment variable `NODE_TLS_REJECT_UNAUTHORIZED=0` should be used only in testing environments.

Property	Description
<velocity_mcp_server_url>	Specifies the full Velocity MCP server URL. For example: <code>https://<velocity_hostname>/mcp</code>
<velocity_user_access_key>	Provides the credentials necessary to access the Velocity MCP server. Create an User Access Key in Velocity and use it.

4. Save your changes and restart Claude Desktop.
5. Go to **My Profile** → **Settings** → **Developer**.
6. Verify the connection.
The Velocity MCP Server status should display **Running**.

Results

You have successfully connected to the Velocity MCP server by using Claude Desktop.

What to do next

You can run prompts related to Velocity to confirm a successful connection.

Connecting to the DevOps Velocity MCP server using GitHub Copilot in Visual Studio Code

You can connect to the HCL DevOps Velocity (Velocity) Model Context Protocol (MCP) server by using GitHub Copilot in Visual Studio (VS) Code to interact with Velocity and retrieve the necessary data and insights..

Before you begin

You must have completed the following tasks:

- Installed Velocity and enabled the MCP server. See [MCP server setup in DevOps Velocity on page 4](#).
- Obtained the MCP Server URL. For example: `https://<velocity_hostname>/mcp`
- Ensured that you have installed VS Code and the GitHub Copilot extension.

1. **Optional:** Open the terminal and enter the following command to allow the self-signed certificates if you are using a self-signed certificate during the Velocity installation.

```
export NODE_TLS_REJECT_UNAUTHORIZED=0
```

2. Enter the following command to open VS Code.

```
code .
```

3. Open the **Command Palette** (Ctrl+Shift+P or Cmd+Shift+P).
4. Search and select **MCP: Add Server**.
5. Choose **HTTP** as the server type, when prompted for the transport type.
The Velocity MCP Server uses HTTP-based transport.
6. Enter the **MCP Server URL**, in the **URL** field.
VS Code automatically generates the required `mcp.json` configuration.
7. Enter a name for the server. For example: Velocity MCP Server.
8. Select **Workspace**, when VS Code prompts you to choose where to save this configuration.

- **Global** – used for all VS Code workspaces
- **Workspace** – used only in the current folder/project



Note: Selecting **Workspace** automatically creates a `.vscode` folder and generates a `mcp.json` file inside it.



Important: You'll be prompted to authenticate via OAuth, but cancel the prompt by pressing **Esc** on your keyboard or clicking **Cancel**, as authentication is handled using the **User Access Key** defined in the `mcp.json` file.

9. Update the configuration details in the `mcp.json` file to pass the User Access key in the header as follows:

```
{
  "servers": {
    "velocity-mcp-server": {
      "url": "<velocity_mcp_server_url>",
      "type": "http",
      "headers": {
        "Authorization": "UserAccessKey <velocity_user_access_key>"
      }
    }
  },
  "inputs": []
}
```

Property	Description
<code><velocity_mcp_server_url></code>	Specifies the full Velocity MCP server URL. For example: <code>https://<velocity_hostname>/mcp</code>
<code><velocity_user_access_key></code>	Provides the credentials necessary to access the Velocity MCP server. Create an User Access Key in Velocity and use it.

10. Click **Restart**.

You must be able to see the connection successful logs and the discovered tools.

11. Verify the connection.

The Velocity MCP Server must display **Running** as the status.

Results

You have successfully connected to the Velocity MCP server by using GitHub Copilot in VS Code.

What to do next

You can run prompts related to Velocity to confirm a successful connection.