

Prerequisites

[Intro](#)

[VS Code](#)

[Git](#)

[Databricks CLI](#)

[Install Databricks CLI](#)

[Set up databricks cli config file](#)

[Set up Azure repo \(New Projects only\)](#)

[Next Steps](#)

Intro

This document contains guidance on the tools and software you need to install prior to deploying a new project or iterating on a currently deployed project.

VS Code

VS Code is the recommended editor for our Data Science projects as it directly integrates with Databricks and has support for practically every file type!

1. Open the Microsoft Store application
2. Search for Visual Studio Code
3. Click Get and follow the prompts

Git

Git is the version control software we use to keep track of project files and code versioning.

1. Download Git from here: [📄 Git - Downloading Package](#)

2. Follow the instructions in the installer.

3. Set your username and email:

- `git config --global user.name "Your Name"`
- `git config --global user.email "username@corpuk.net"`

4. Set a standard git commit template:

- Open windows command prompt and navigate to the root directory (It should default to this location with a new command prompt window. This will show as `C:\Users\corpuk_username>` next to the cursor)
- Type `notepad .gitmessage.txt` to create the template
- Paste the following into the newly created file, before saving and exiting the file:

- ```
1 Subject line (try to keep under 50 characters)
2
3 Multi-line description of commit, feel free to be detailed.
4
5 [Ticket: X]
```

- Enter the following into the command prompt to point git to the message template:

- `git config --global commit.template ~/.gitmessage.txt`

5. (Optional) Set VS Code as the git commit message editor. Enter the following into windows command prompt: `git config --global core.editor "code --wait"`

## Databricks CLI

Databricks CLI is used to interact with Databricks resources and set up the bundles that are used to deploy machine learning pipelines.

Databricks CLI version  $\geq 0.205.2$  is required to interact with the pipelines we have set up for project deployments.

### Install Databricks CLI

1. Open windows command prompt and type `databricks -v`. If you get an error that no such program exists then proceed to step 3. If you have a version  $< 0.205.2$  then proceed to step 2.
2. If you have CLI version  $\leq 0.18$  use pip to uninstall the legacy CLI (`pip uninstall databricks-cli`). If you have CLI version  $< 0.205.2$  and  $> 0.18$  you will need to find the `.exe` file on you system and delete it before proceeding to step 3. See this page for more details: [Databricks CLI migration](#)
3. Download the latest version of the CLI from the [Databricks cli release page](#)
4. Unzip the downloaded folder
5. You can leave the CLI `.exe` in the unzipped folder or move it to your home directory `C:\Users\corpuk_username`

### Set up databricks cli config file

1. Open windows command prompt and navigate to home directory `C:\Users\corpuk_username`
2. Enter `notepad .databrickscfg` to create the config file
3. Populate the resulting file with the following information for dev environment:
  - ```
1 [DS_PPD]
2 host = https://adb-2247447696331887.7.azuredatabricks.net
3 token = <your_token>
4 jobs-api-version = 2.1
```
4. You will need to generate your own token within the databricks UI:
 - In each workspace, click your username (top-right of screen) -> User Settings -> Developer -> Access Tokens
 - Paste the generated token into the relevant line of the `.databrickscfg` file

Set up Azure repo (New Projects only)

If this is the first time a project has been deployed with ML Ops then please ask [@Samuel.Thomas13](#) or [@Andrew Witherley](#) to set up a new repo.

New repos should only be created for business areas that have not had a deployment through ML Ops.

Check here for the list of repos: https://dev.azure.com/data3-dataops/Innovation/_git

Guidance on how to set up a new repo can be found here: [Creating a new repo](#)

Next Steps

You are now ready to start working on a new or existing project!