




AWS Mainframe Modernization is currently in preview. It is subject to change. We do not recommend using it for production workloads.



Setting up Enterprise Developer on Amazon AppStream

This document describes how to set up Micro Focus Enterprise Developer for one or more mainframe applications in order to maintain, compile, and test them using the Enterprise Developer features. The setup is based on the Amazon AppStream Windows images shared by AWS Mainframe Modernization with the customer and on the creation of Amazon AppStream fleets and stacks as described in [Setting up Amazon AppStream for use with AWS Mainframe Modernization Enterprise Analyzer and Enterprise Developer \(./set-up-appstream.html\)](#) .

Important

You must perform the steps of this setup when the Enterprise Analyzer fleet and stack are up and running.

 For a complete description of Enterprise Developer v7 features and deliverables, check out its [up-to-date online documentation \(v7.0\)](#) (<https://www.microfocus.com/documentation/enterprise-developer/ed70/ED-Eclipse/GUID-8D6B7358-AC35-4DAF-A445-607D8D97EBB2.html>) on the Micro Focus site.

In addition to Enterprise Developer  itself, Rumba (a tn3270 emulator) is also installed. Moreover, the image also offers [pgAdmin](https://www.pgadmin.org/) (<https://www.pgadmin.org/>), the feature-rich database administration and development platform for PostgreSQL, to allow efficient access to the application database during maintenance of source code in order to validate results . Finally, git-scm tools are also installed to allow for efficient interactions with CodeCommit repositories in addition to the Eclipse `git` interactive user interface included in Enterprise Developer.

If you need to access source code that is not yet loaded into CodeCommit repositories, but that is available in an Amazon S3 bucket, for example to perform the initial load of the source code into git, follow the procedure to create a virtual Windows disk as described in [Setting up Enterprise Analyzer on Amazon AppStream \(./set-up-ea.html\)](#) .

If you want to explore the service without using your own code, you can use the source code, sample data, and system definitions of the sample BankDemo application, which are available in the `C:\Users\Public\BankDemo` folder of the Amazon AppStream Windows instance. This directory encompasses a full Enterprise Developer template that will be used to load and execute the demo application on a local instance of Enterprise Developer running locally before it is pushed to the CI/CD code pipeline also provided by the AWS Mainframe Modernization. For more information, see the [Enterprise Build Tools](https://d1vi4vxke6c2hu.cloudfront.net/tutorial/tutorial-build.pdf) (<https://d1vi4vxke6c2hu.cloudfront.net/tutorial/tutorial-build.pdf>) tutorial.

Topics



- [Prerequisites \(#tutorial-ed-prerequisites\)](#)
- [Step 1: Setup by individual Enterprise Developer users \(#tutorial-ed-step1\)](#)
- [Step 2: Create the Amazon S3-based virtual disk on Windows \(optional\) \(#tutorial-ed-step2\)](#)
- [Step 3: Clone the repository \(#tutorial-ed-step3\)](#)
- [Subsequent sessions \(#tutorial-ed-step4\)](#)

Prerequisites

- One or more CodeCommit repositories loaded with the source code of the application to be maintained. The repository setup should match the requirements of the CI / CD pipeline above to create synergies by combination of both tools.
- IAM users with credentials to the CodeCommit repository or repositories defined by the account administrator for each application teammate according to the information in [Authentication and access control for AWS CodeCommit \(https://docs.aws.amazon.com/codecommit/latest/userguide/auth-and-access-control.html\)](https://docs.aws.amazon.com/codecommit/latest/userguide/auth-and-access-control.html) . The structure of those credentials is reviewed in [Authentication and access control for AWS CodeCommit \(https://docs.aws.amazon.com/codecommit/latest/userguide/auth-and-access-control.html\)](https://docs.aws.amazon.com/codecommit/latest/userguide/auth-and-access-control.html) and the complete reference for IAM authorizations for CodeCommit is in the [CodeCommit permissions reference \(https://docs.aws.amazon.com/codecommit/latest/userguide/auth-and-access-control-permissions-reference.html\)](https://docs.aws.amazon.com/codecommit/latest/userguide/auth-and-access-control-permissions-reference.html) : the administrator may define distinct IAM policies for distinct roles having credentials specific to the role for each repository and limiting its authorizations of the user to the specific set of tasks that he has to accomplish on a given repository. So, for each maintainer of the CodeCommit repository, the account administrator will generate a primary IAM user, grant this user permissions to access the required repository or repositories via selection proper IAM policy or policies for CodeCommit access.

Step 1: Setup by individual Enterprise Developer users

1. Obtain your IAM credentials:
 1. Connect to the AWS console at <https://console.aws.amazon.com/iam/> (<https://console.aws.amazon.com/iam/>) .
 2. Follow the procedure described in step 3 of [Setup for HTTPS users using Git credentials \(https://docs.aws.amazon.com/codecommit/latest/userguide/setting-up-gc.html\)](https://docs.aws.amazon.com/codecommit/latest/userguide/setting-up-gc.html) in the *AWS CodeCommit User Guide*.
 3. Copy the CodeCommit-specific user name and password that IAM generated for you, either by showing, copying, and then pasting this information into a secure file on your local computer, or by choosing **Download credentials** to download this information as a .CSV file. You need this information to connect to CodeCommit.
2. Start a session with Amazon AppStream based on the url received in the welcome email. Use your email as user name and create your password.
3. Select your Enterprise Developer stack.
4. On the menu page, choose **Desktop** to reach the Windows desktop streamed by the fleet.

Step 2: Create the Amazon S3-based virtual disk on Windows (optional)

If there is a need for Rclone (see above), create the Amazon S3-based virtual disk on Windows: (optional if all application artefacts exclusively come from CodeCommit access).

1. Copy the run-rclone.bat file provided in C:\Users\Public to your home folder C:\Users\PhotonUser\My Files\Home Folder via File Explorer. It might already be present in this directory if you previously installed Enterprise Analyzer in your Amazon AppStream account.

2. Update its config parameters with your Amazon S3 bucket name, your AWS access key and corresponding secret.
3. Use File Explorer to copy the file `run-rc\one.bat` provided in `C:\Users\Public` to the home folder.
4. Either run `run-rc\one.bat` from File Explorer or open a Windows command to run it. The source code of the application located in Amazon S3 bucket should now be visible in Windows File Explorer as disk `m2-s3` (as defined in heading line of `run-rc\one.bat`).

Step 3: Clone the repository

1. Navigate to the application selector menu in the top left corner of the browser window and select Enterprise Developer.
2. Complete the workspace creation required by Enterprise Developer in home folder by choosing `C:\Users\PhotonUser\My Files\Home Folder` (aka `D:\PhotonUser\My Files\Home Folder`) as location for the workspace.
3. In Enterprise Developer, clone your CodeCommit repository by going to the Project Explorer, right click and choose **Import, Import ..., Git, Projects** from **Git Clone URI**. Then, enter your CodeCommit-specific user name and password and complete the Eclipse dialog to import the code.

The CodeCommit git repository is now cloned in your local workspace.

Your Enterprise Developer workspace is now ready to start the maintenance work on your application. In particular, you can use the local instance of Microfocus Enterprise Server (ES) integrated with Enterprise Developer to interactively debug and run your application to validate your changes locally.

Note

The local Enterprise Developer environment, including the local Enterprise Server instance, runs under Windows while AWS Mainframe Modernization runs under Linux. We recommend that you run complementary tests in the Linux environment provided by AWS Mainframe Modernization after you commit the new application to CodeCommit and rebuild it for this target and before you roll out the new application to production.

Subsequent sessions

As you select a folder that is under Amazon AppStream management like the home folder for the cloning of your CodeCommit repository, it will be saved and restored transparently across sessions. Complete the following steps the next time you need to work with the application:

1. Start a session with Amazon AppStream based on the url received in the welcome email.
2. Login with your email and permanent password.
3. Select the Enterprise Developer stack.
4. Launch Rc\one to connect (see above) to the Amazon S3-backed disk when this option is used to share the workspace files.
5. Launch Enterprise Developer to do your work.