Micro Focus Enterprise Server on Google Cloud Platform

A GCP blueprint

February 2022

Contents

Introduction

This document provides step-by-step instructions for you to quickly and confidently deploy the Micro Focus Enterprise Server 7.0 blueprint on Google Cloud Platform (GCP).

This document is intended for:

- A system integrator (SI) who is used to installing and configuring Enterprise Server deployments on premises and wants a template to use as a starting point for installing and configuring Enterprise Server deployments on GCP.
- An existing Micro Focus customer who has already deployed an IBM mainframe workload to Enterprise Server on premises and wants to see how easy it is to migrate these rehosted applications to GCP.
- Anyone interested in exploring the feasibility of moving an IBM mainframe workload to GCP and wants to experience a fully functioning production server for running mainframe applications.

The blueprint uses Terraform templates to create the required infrastructure within a GCP project; this includes the Micro Focus Enterprise Server product, containing a fully functioning demonstration application. The demonstration application, called BankDemo, is configured in a scalable and highly-available environment – a Performance and Availability Cluster (PAC) – and uses COBOL, CICS, Job Control Language (JCL), and Virtual Storage Access Method (VSAM) files.

Micro Focus Enterprise Server on GCP

Micro Focus Enterprise Server is an application deployment environment for IBM mainframe applications that have been running on the IBM z/OS operating system. Enterprise Server enables you to modernize and integrate mainframe applications with technologies such as .NET and Java. It also gives you the flexibility to deploy your application across virtual Linux and Microsoft Windows platforms or containers, on GCP.

Using Enterprise Server, your organization can:

- Move COBOL and PL/I mainframe applications to Linux or Windows on GCP with minimal change.
- Support both online CICS and IMS applications, as well as a batch environment to support the move of current jobs, job control, JCL, and batch utilities. Db2, IMS-DB, QSAM, and VSAM data can be transitioned into alternative database and file systems.
- Rapidly replicate business-critical functionality to new platforms, to support geographic, regulatory, line- of-business, or other key requirements.
- Meet your application reliability, availability, and serviceability requirements.
- Integrate with your security infrastructure, for appropriate application and system security.

The following table shows the main constituents of the mainframe, and their equivalent emulations within Enterprise Server:

IBM Mainframe Subsystem	Micro Focus Target
JES2, JES3	Compatible JES (JCL and Spool)
CICS and IMS TM	Enterprise Server Online Support
Batch, JCL	Enterprise Server Batch Support
COBOL	Micro Focus COBOL
PL/I	Micro Focus Open PL/I
REXX	Enterprise Server REXX Support
VSAM	Enterprise Server VSAM Support
IMS DB	Enterprise Server IMS DB Support
Db2	PostgreSQL, Microsoft SQL Server, IBM DB2 LUW, Oracle
z/OS, z/VSE	Linux, Windows, containers on Google Cloud

Costs and licenses

You are responsible for the cost of the GCP services used while running this blueprint. There is no additional cost for using it.

The Terraform templates supplied in this blueprint include configuration parameters that you can customize; however, apart from the parameters this tutorial instructs you to update, we recommend that you familiarize yourself with the out-of-the-box tutorial before you make any additional customizations. Some of these parameters, such as the specification of the virtual machines, will affect the cost of deployment. For cost estimates, refer to the GCP pricing calculator for each resource you will be using. Prices are subject to change.

This blueprint uses a Bring Your Own License (BYOL) model for Enterprise Server. How you license Enterprise Server for use with this blueprint depends on what sort of license you have:

- If you are an existing user of Enterprise Server you should contact your Micro Focus sales representative to discuss how you can evaluate the blueprint using one of your existing licenses.
- If you are not an existing user of Enterprise Server you can request a trial license from Micro Focus.

  This trial license for Enterprise Server on GCP is only intended to be used for a maximum of 90 days, and is not intended for production use. After the trial period, you are responsible for acquiring the necessary licenses directly from Micro Focus to continue using Enterprise Server on GCP.

Architecture of Enterprise Server on GCP

This section gives detailed information on the structure deployed by this blueprint on GCP.

When you apply the default Terraform infrastructure, the following resources are created and deployed into a new Virtual Private Cloud (VPC):

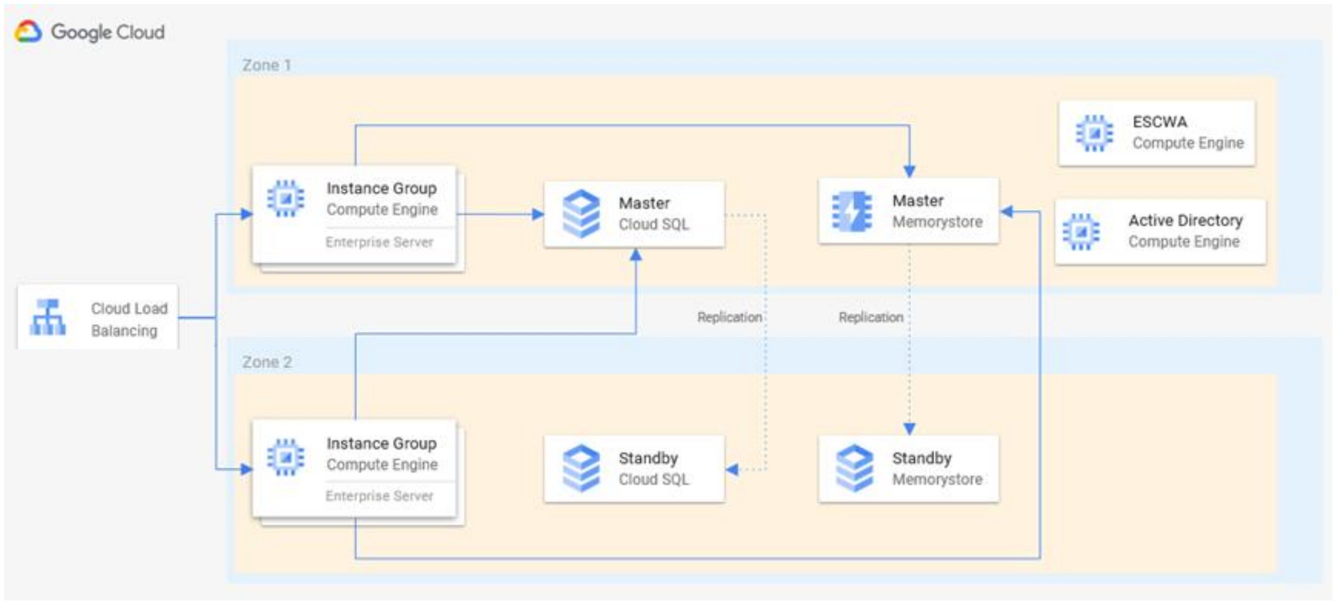


Figure 1: Architecture for Enterprise Server on GCP

Note: If you customize the supplied templates to deploy your own Enterprise Server applications to GCP, the resulting GCP architecture could be different to the one described in this section.

The blueprint sets up the following:

- A network load balancer to automatically distribute requests for the BankDemo demonstration application to the deployed Enterprise Server virtual machines and to handle 3270 traffic exposed via a public IP address.
- A single subnet containing administration and management utilities:
  - A virtual machine to host Enterprise Server Common Web Administration (ESCWA). ESCWA is a utility for viewing and managing enterprise servers running on different machines.
  - Two virtual machines, each to host Enterprise Server for Linux.
  - A master Memorystore for Redis instance, replicated to a standby Memorystore for Redis instance.
  - A master PostgreSQL 12 database instance, which stores the data files used by the BankDemo demonstration, replicated to a standby PostgreSQL 12 database instance.
  - An Active Directory domain controller.

Deployment

This section shows how to apply the necessary infrastructure to deploy Enterprise Server (including a demonstration application) into a GCP project. When you have deployed Enterprise Server, we recommend that you read the section *Running the BankDemo demonstration application* in order to validate that Enterprise Server was successfully deployed and also to familiarize yourself with it.

Prerequisites

Before deploying this blueprint, which will build the project infrastructure from the Terraform templates, this document assumes the following:

- You have a GCP-enabled Google account.
- You have created an empty cloud project for the purposes of this deployment.
- You have a valid license for Enterprise Server for Linux.

  This blueprint uses a Bring Your Own License (BYOL) model for Enterprise Server. If you are an existing user of Enterprise Server, you should contact your Micro Focus sales representative to discuss using one of your existing licenses. If you are not an existing user of Enterprise Server, you need to request a trial license from Micro Focus. For more information, see the section *Costs and licenses*.

It also assumes a familiarity with a Linux shell environment, and a level of competency with basic Linux shell commands.

Review the Micro Focus End User License Agreement

Before deploying this blueprint for Enterprise Server, review the terms of the Micro Focus End User License Agreement and Additional License Authorizations.

Deploy Enterprise Server from the Terraform templates

This section describes how to use the GCP project page to deploy the Enterprise Server blueprint using the Terraform templates.

The templates are available from a public GitHub repository. Download them to your GCP project, ready for deployment.

When deployed, the default configuration creates a Performance and Availability Cluster (PAC) consisting of two enterprise servers hosting a demonstration application.

As you become more familiar with the architecture, you can update the templates to change the configuration, but any such changes are out of scope of this document.

To run the application, you only need to make minimal changes to a variables file, which then propagates those changes throughout the architecture when it is built or rebuilt.

Note: You are responsible for the cost of the GCP services used while running this blueprint. For full details, see the pricing pages for each GCP service you will be using. Prices are subject to change.

Log on to the project

1. Sign in to https://console.cloud.google.com using your GCP-enabled Google credentials.
2. Select or create a project to which the blueprint will be deployed.

Set up the Cloud Shell environment

The Cloud Shell is a Linux-based development environment designed to allow you to interact with any project resources you have within GCP. For more information on the Google Cloud Shell, refer to https://cloud.google.com/shell/docs. We will use the shell to create new project resources in the current cloud project. Before we do that, we need to configure the environment:

1. At the top-right of the project home page, click (**Activate Cloud Shell**).

   The Cloud Shell opens in a pane at the bottom of the screen.
2. Obtain access credentials for your user to run commands for the current project:
   a. At the shell prompt, enter:

   ```
   gcloud auth login
   ```
   b. Follow the link displayed, select the required Google account when prompted, and then click **Allow**.
3. Click to copy the verification code, switch back to the shell and paste the code at the prompt.
4. Set the environment to default to the current project:
   a. To obtain the PROJECT_ID, enter:

   ```
   gcloud projects list
   ```
   b. Click **AUTHORIZE** if prompted.

   c. Set the environment to default to the current project:

   ```
   gcloud config set project <PROJECT_ID>.
   ```
   The project ID is appended to the shell prompt to indicate any actions apply to that project.

Obtain the blueprint source files

The blueprint files are a collection of application source files, configuration files and Terraform templates, supplied by Micro Focus. These files are hosted in a public Git repository.

1. At the shell prompt, navigate to a location at which to store a copy of the repository.
2. Clone the repository:

   ```
   git clone https://github.com/GoogleCloudPlatform/rapiddeploy-microfocus-es
   ```
   A copy of the repository is downloaded; we will refer to this location as the 'project folder' from now on.

Install the Enterprise Server license

You must supply your own Enterprise Server license for use with this demonstration. The license file (.mflic) must be placed within a specific location and then referenced correctly within the Terraform variables file (terraform.tfvars).

1. On the top-right of the Cloud Shell pane, click and select **Upload**.

2. Ensure that **File** is selected, then click **Choose Files**.
3. Select the appropriate license from your local machine.
4. Within the **Select a Destination Directory** section, navigate to the **eslicense** sub-folder within the project folder.
5. Click **UPLOAD**.

   The license file is uploaded.
6. At the shell prompt, navigate to the project folder and open **terraform.tfvars**.
7. Update the following variable, and then save the file:

Variable	Purpose
license_filename	The name of the .mflic file, including the extension. (The path defaults to the eslicense sub-folder within the project folder.)

   The Enterprise Server product will be properly licensed when the Terraform templates are applied to the project.

Apply the Terraform architecture

Terraform is an infrastructure as code (IaC) tool that allows you to build, change, and version the infrastructure required in GCP to host the demonstration application served from Enterprise Server.

Some of the contents of the Git repository you downloaded are Terraform template files. These are used to create the necessary resources within the GCP project. You should not need to edit most of the default settings within these templates, as they are designed to use a variables file (terraform.tfvars), containing the variables that you are most likely to change, but there are a few settings that you can update before applying the templates to the project

1. At the shell prompt, navigate to the project folder and initialize it to work with the Terraform files:

```
terraform init
```

2. Open terraform.tfvars, update the values for the following variables, and then save the file:

Variable	Purpose
name	The prefix used for resource names. Using this variable, you can create multiple configurations, as it ensures that the resource names are unique.
ssh_ip	Set this variable to the IP address of your local machine. This variable restricts ESCWA access to the specified IP address, which although optional, is recommended for security purposes.

3. Apply the Terraform architecture:

```
terraform apply -var-file=terraform.tfvars
```

4. Click **AUTHORIZE** if prompted.

   Terraform now calculates which resources it needs to create based on the contents of the template files and displays the ‘plan’.
5. Type **yes** to confirm the plan and proceed with the resource creation.

The resources are created; this can take a while. When the ‘Apply Complete!’ message is displayed, perform the post-installation tasks.

Post-installation tasks

Resources check

After you have successfully applied the Terraform architecture, your project should contain the following resources:

Cloud service	Resource
Network services > Load balancing	● A load balancer
Compute > Compute Engine > Instance templates	● Two enterprise server VMs ● An Active Directory VM ● An ESCWA VM
Databases > Memorystore > Redis	● A Redis instance
Databases > SQL	● Two PostgreSQL 12 instances

Each of these resources is prefixed with the *name* variable that you specified in terraform.tfvars. In these examples below 'name=mfdocs1' was used:

Resource	Example name
Load balancer	mfdoc1-loadbalancer
Enterprise server	mfdoc1-es-w32c
Active Directory	mfdoc1-activedirectory-001
ESCWA	mfdoc1-escwa-001
Redis instance	mfdoc1-redis
PostgreSQL instance	mfdoc1-db-8dc76es3

Note: The alphanumeric strings appended to some resources are auto generated to ensure resource names are unique within the configuration.

Start the BankDemo enterprise server regions

Each Enterprise Server contains a region from which the BankDemo application is served. You must start these regions before you can run the application.

You can administer the regions on both enterprise servers using ESCWA.

1. Log on to the ESCWA administration console – see Connecting to the Enterprise Server Common Web Interface (ESCWA).
2. Click **NATIVE**.
3. From the navigation pane, expand **Directory Servers**.

   Two directory servers are listed.

4. Expand the first server entry to display the enterprise server regions contained within.

5. Point to BNKDM, and once highlighted, click ▷ to start the region.
6. On the **Region Start Options** screen, ensure **Cold Start** is selected, and then click **START**.

   A popup message is displayed after a short while to indicate that the server is now running.

7. Repeat the same process to start BNKDM on the other directory server.

   When both BNKDM regions are running, you are ready to run the tutorial.

Tutorial

This section contains a tutorial, based around the BankDemo demonstration application. An overview of BankDemo is provided in the section *Introduction to the BankDemo demonstration application*.

You should only run through this section if you successfully completed all the steps in the Deployment section.

The section *Running the BankDemo demonstration application*, walks you through BankDemo's basic operation.

Two additional tutorials, described in the section *Investigating scale-out features in the BankDemo demonstration application*, take a closer look at Enterprise Server's scale-out features, made possible by Micro Focus’ Performance and Availability Cluster (PAC) architecture.

This section also contains information on housekeeping tasks that you might need to perform after running the tutorials.

Prerequisites

Before you run the BankDemo demonstration application that is used in these tutorials, you need to ensure that you have:

- Successfully deployed the supplied Terraform templates that build Enterprise Server.
- Successfully started the two BNKDM regions that contain the application – see *Start the BankDemo enterprise server regions*.
- Installed a TN3270 terminal emulator. You can use any TN3270 terminal emulator, but this guide shows the use of Micro Focus Rumba+, which is supplied with Micro Focus Enterprise Developer.

> **Note:** The instructions in these tutorials assume that when deploying the Terraform template files you used the default values supplied.

Introduction to the BankDemo demonstration application

BankDemo is a demonstration application for an imaginary banking company and provides simple banking functionality such as displaying accounts and transactions, transferring funds between accounts, and changing users' contact details. The source code for BankDemo is supplied with Enterprise Developer, which is a Micro Focus product that enables you to develop and maintain mainframe COBOL and PL/I applications.

This blueprint for Enterprise Server includes deployable files that have been built using Enterprise Developer. Running BankDemo after you have launched this blueprint enables you to verify that the blueprint has deployed as expected and that Enterprise Server is running successfully on GCP.

The BankDemo demonstration is configured as a Performance and Availability Cluster (PAC) using Memorystore for Redis as a Scale-Out Repository (SOR), and hosts the VSAM data files in a PostgreSQL 12 database instance:

We will demonstrate running the BankDemo application in both online mode and batch mode.

Running the BankDemo demonstration application

This section contains the steps you need to follow to run the application in online and batch modes.

Connecting to the BankDemo enterprise server

The BankDemo application runs on two enterprise servers that you must start manually to make the application available - see *Start the BankDemo enterprise server regions*.

Once the enterprise servers are running, use your TN3270 terminal emulator to connect to the load balancer within the GCP project configuration, using the following details:

```
load-balancer-IP-address:port
```

where:

- `load-balancer-IP-address` is the IP address of the load balancer. To obtain this value:

  a. On the Cloud Project page, click ☰ to open the navigation menu.

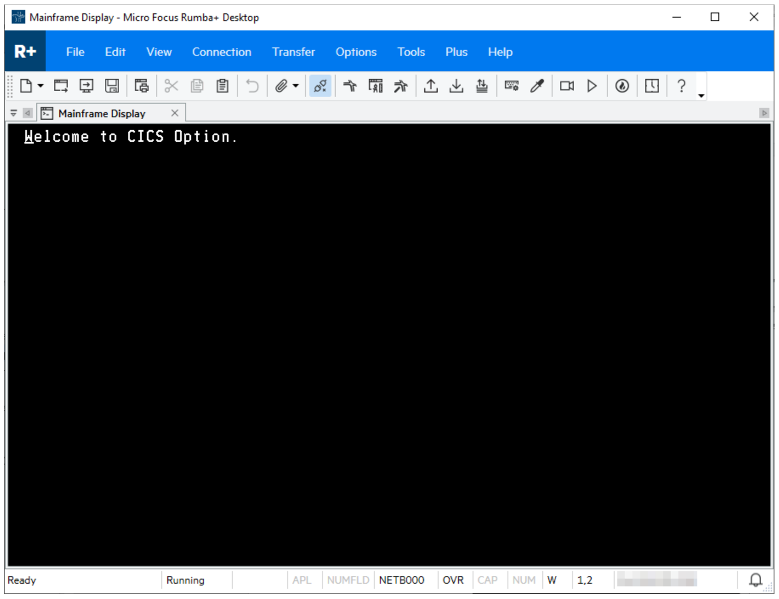
  b. Select **Network Services** > **Load balancing**.

  c. On the **Load balancing** page, click the name of the load balancer.

     The load balancer details are displayed. The **IP:Port** field shows the IP address on which to connect your TN3270 session.

- `port` is the port number displayed after the IP address. This should default to 5557.

  Once connected to the BankDemo enterprise server you will see a screen similar to the one in Figure 2.



```
Welcome to CICS Option.
```

Figure 2: The Welcome to CICS splash screen

Now that you are connected, you are ready to run the BankDemo application in online mode.

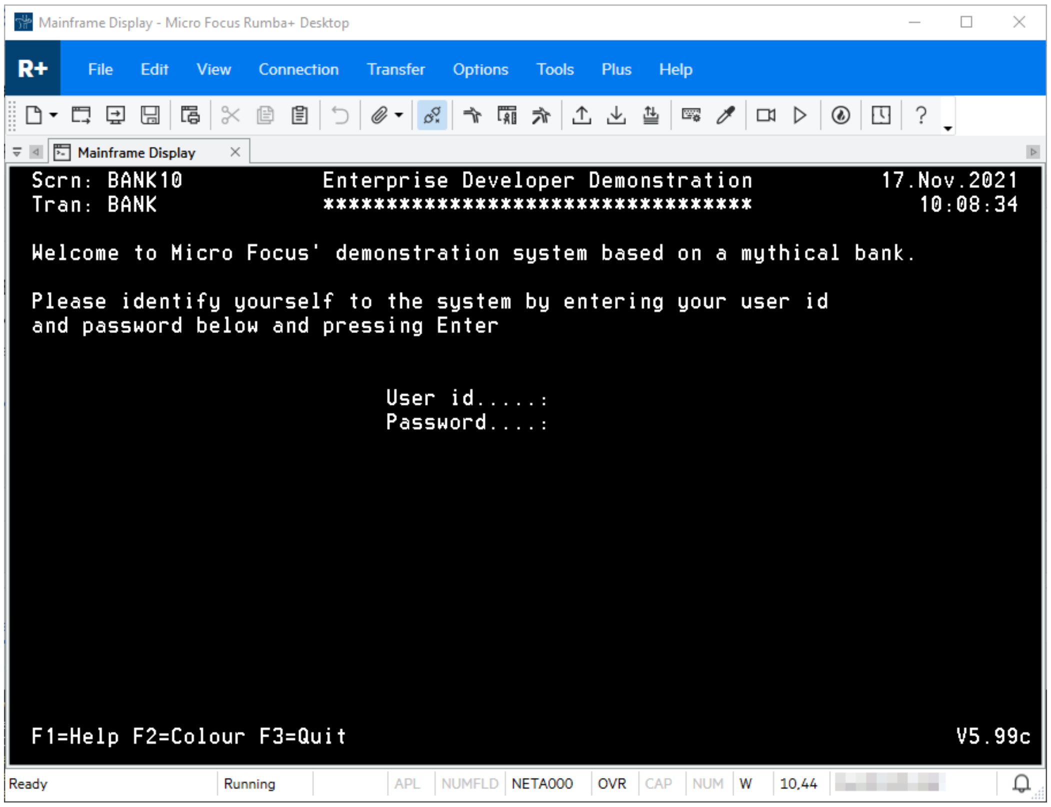
Running BankDemo in online mode

You can now perform the following steps to run the BankDemo application in online mode:

1. Press **Ctrl+Shift+Z** to clear the screen.
2. Enter the name of the transaction to run:

   `BANK`

The BankDemo sign-on screen is displayed, as shown in Figure 3.



```
Scrn: BANK10           Enterprise Developer Demonstration         17.Nov.2021
Tran: BANK             **********************************           10:08:34

Welcome to Micro Focus' demonstration system based on a mythical bank.

Please identify yourself to the system by entering your user id
and password below and pressing Enter


                            User id.....:
                            Password....:


F1=Help F2=Colour F3=Quit                                                V5.99c
```

Figure 3: The BankDemo sign-on screen

This screen being displayed is confirmation that the BankDemo application is running under Enterprise Server.

The rest of this section describes how you can try out some of BankDemo's features in online mode. If you don't want to try out these BankDemo features in online mode, your next step is to try running in batch mode: *Running BankDemo in batch mode*.

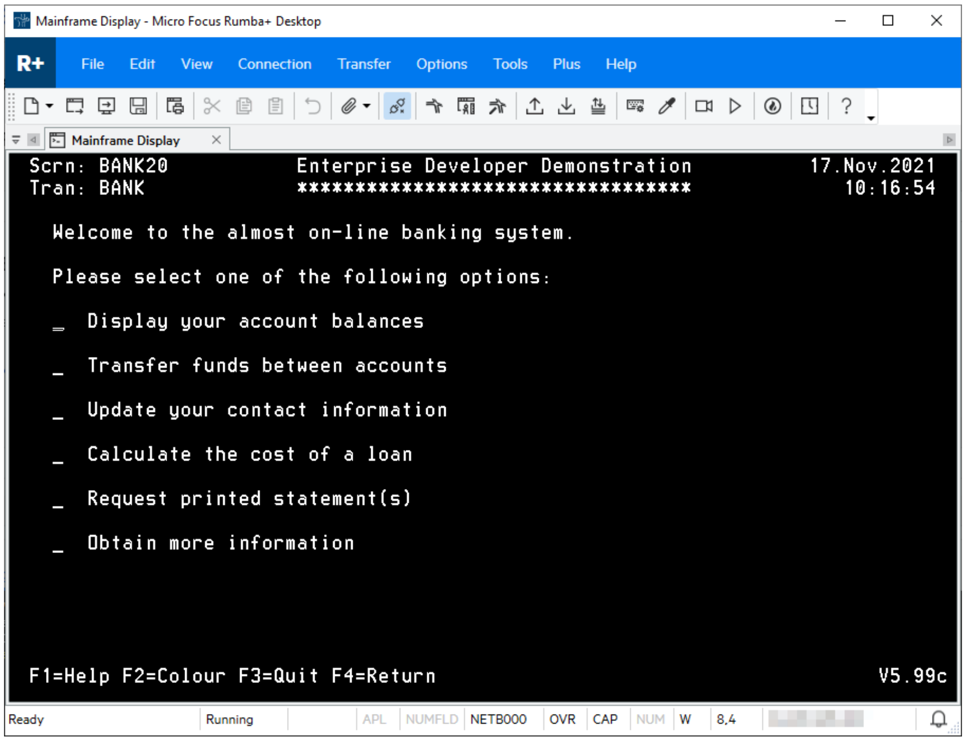
The remaining steps in this section enable you to verify that BankDemo can read and write data by logging in as a user, viewing the user's balance, transferring an amount between two of the user's accounts, and viewing the user's balance again to confirm that the transfer has happened.

Tip: Each screen of the BankDemo application has its own help page, which you can access by pressing **F1**.

3. From the BankDemo sign-on screen, specify a valid user ID, press **Tab**, and then enter a password. Valid user IDs are B0001 - B0036. This demonstration uses B0004.

   You can use any valid user ID, but if you choose a user other than B0004, the data presented on the screens that you see will be different from the data shown in the figures from this point on.

   The password can be any non-blank character string. Once successfully logged on, you see the BankDemo main options screen:



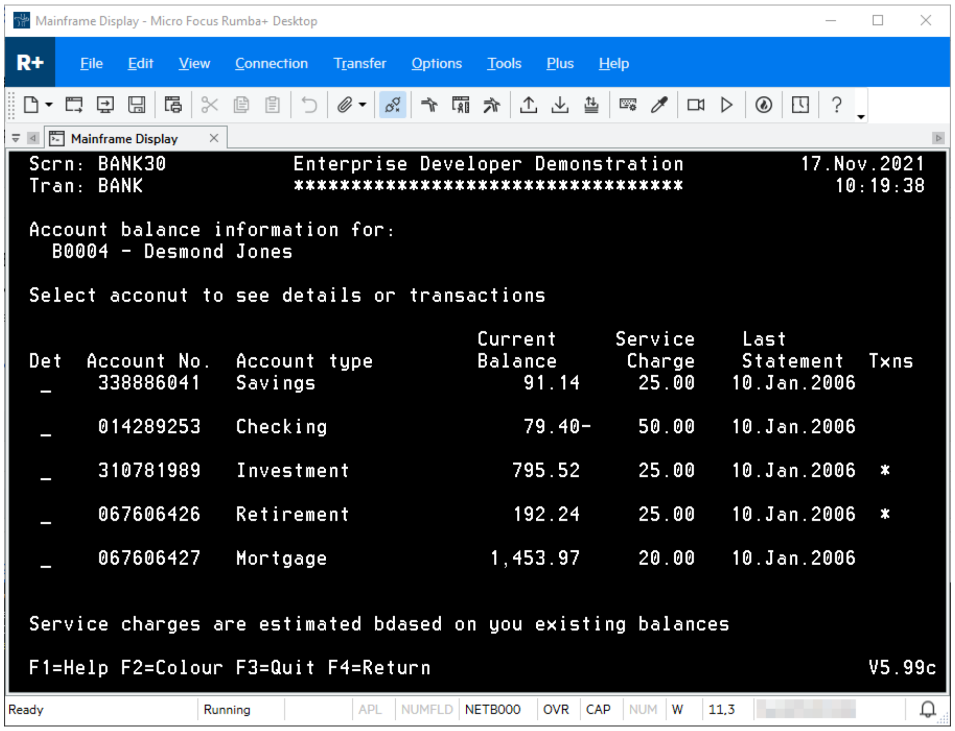
Mainframe Display - Micro Focus Rumba+ Desktop

```
 Scrn: BANK20             Enterprise Developer Demonstration           17.Nov.2021
 Tran: BANK               **********************************             10:16:54

   Welcome to the almost on-line banking system.

   Please select one of the following options:

   _  Display your account balances

   _  Transfer funds between accounts

   _  Update your contact information

   _  Calculate the cost of a loan

   _  Request printed statement(s)

   _  Obtain more information



 F1=Help F2=Colour F3=Quit F4=Return                                           V5.99c
```

Figure 4: The BankDemo main options screen

4. The cursor is positioned on the first option, **Display your account balances**. Press **X**, and then press **Enter**.

   The BankDemo account balances screen is displayed, showing the balances of the different accounts held by this user:



Mainframe Display - Micro Focus Rumba+ Desktop

```
 Scrn: BANK30             Enterprise Developer Demonstration          17.Nov.2021
 Tran: BANK               **********************************            10:19:38

 Account balance information for:
   B0004 - Desmond Jones

 Select acconut to see details or transactions

                                         Current      Service     Last
 Det  Account No.  Account type          Balance      Charge      Statement  Txns
  _     338886041  Savings                 91.14      25.00     10.Jan.2006

  _     014289253  Checking                 79.40-     50.00     10.Jan.2006

  _     310781989  Investment             795.52      25.00     10.Jan.2006  *

  _     067606426  Retirement             192.24      25.00     10.Jan.2006  *

  _     067606427  Mortgage             1,453.97      20.00     10.Jan.2006


 Service charges are estimated bdased on you existing balances

 F1=Help F2=Colour F3=Quit F4=Return                                          V5.99c
```

Figure 5: The BankDemo account balances screen

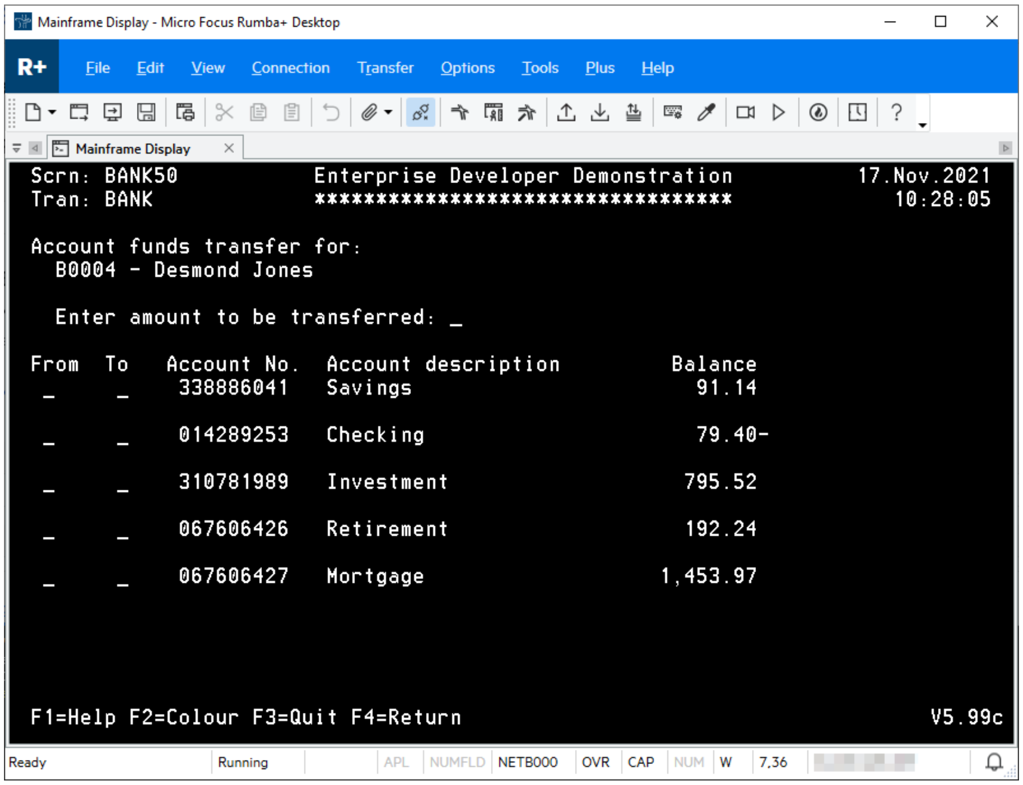
Note: There are two intentional spelling mistakes on the screen above; a later tutorial will demonstrate how we can fix these without incurring any down time.

5. Press **F4** to return to the BankDemo main options screen.

   Now we want to transfer some funds from one account to another. You can see from Figure 5 that this user's checking account is overdrawn, so we will transfer $80 from the savings account to the checking account.

6. Press **Tab** so that the cursor is positioned next to the **Transfer funds between accounts** option, press **X**, and then press **Enter**.

   The BankDemo balance transfer screen is displayed, showing the user's accounts, their balances, and options to move an amount from one account to another.



```
 Scrn: BANK50             Enterprise Developer Demonstration              17.Nov.2021
 Tran: BANK               **********************************                10:28:05

 Account funds transfer for:
   B0004 - Desmond Jones

   Enter amount to be transferred: _

 From  To     Account No.  Account description           Balance
  _     _      338886041   Savings                         91.14

  _     _      014289253   Checking                        79.40-

  _     _      310781989   Investment                     795.52

  _     _      067606426   Retirement                     192.24

  _     _      067606427   Mortgage                     1,453.97




 F1=Help F2=Colour F3=Quit F4=Return                                          V5.99c
```

Figure 6: The BankDemo balance transfer screen

7. The cursor is positioned next to the **Enter amount to be transferred** prompt. Specify `80.00` and then press **Tab**.

8. The cursor is now positioned in the **From** column of the savings account. This is the account that we want to transfer money from. Press **X**, and then press **Tab** twice.

9. The cursor is now positioned in the **To** column of the checking account. This is the account that we want to transfer money into. Press **X**, and then press **Enter**.

   The account balances are updated, and a message appears toward the bottom of the screen summarizing the transfer.

10. Press **F4** to return to the BankDemo main options screen.

    Although the balance transfer screen showed us the changed account balances after we had carried out the transfer, we will now display the account balances screen again. This is to confirm that the data has been updated and that we can read it.

11. The cursor is positioned on the first option, **Display your account balances**. Press **X**, and then press **Enter**.

    The account balances screen is displayed, showing the balances that reflect the changes that were made.

12. Press **F3** to exit BankDemo.

Running BankDemo in batch mode

You can now run the application in batch mode. The steps required to do this can be summarized as follows:

1. Close the BankDemo data files.
2. Submit the BankDemo JCL job.
3. View the output.
4. Reopen the BankDemo data files.

Closing the BankDemo data files

Running BankDemo in batch mode requires access to BankDemo's data files, but if you have previously run BankDemo interactively those data files may still be locked.

To close the data files so that you can run BankDemo in batch mode you need to do the following:

1. Still in your TN3270 terminal emulator, press **Ctrl+Shift+Z** to clear the screen.
2. Enter the following, which is the name of a supplied transaction to close BankDemo's data files:

   ```
   CFIL
   ```

   A message is displayed confirming that the data files have been closed.

Submitting the BankDemo JCL job

The JCL job that you can submit has been deployed to both Enterprise Server regions, and because they are set up in a PAC configuration, you can perform this submission on either region.

1. On the Cloud Project page, click ☰ to open the navigation menu.
2. Select **Compute Engines** > **VM instances**.
3. Locate one of the Enterprise Server instances and click its name.

   The details of the instance are displayed.

4. Click **SSH**.

   A new browser window is opened, and within it, a shell connecting you to the server. Your purpose for connecting is to run a COBOL command, and so you need to set the COBOL environment.

5. Switch to the user that has access to the deployed file:

   ```
   sudo su demouser
   ```

6. Enter the following command to set the environment:

```
. /opt/microfocus/EnterpriseDeveloper/bin/cobsetenv
```

A message is displayed to indicate that the environment is now set.

7. Run the cassub command to submit the JCL job to the spool queue:

   ```
   cassub -j/home/demouser/BankDemo_PAC/jcl/ZBNKSTMT.jcl -rBNKDM
   ```

   JCL messages confirm that the job has been submitted successfully.

Viewing the output

You can view output (which in this case is an account statement) for the job you have just submitted through ESCWA. Again, as these servers are configured as a PAC, you can use ESCWA to view the output on either Enterprise Server VM instance.

1. In your browser, connect to ESCWA – see Connecting to the Enterprise Server Common Web Interface (ESCWA).
2. Click **NATIVE** from the menu.
3. In the navigation pane, expand the structure to show one of the **BNKDM** enterprise server regions.
4. Click **BNKDM**, and then select **JES > Spool** from the menu at the top.
5. The job you submitted should be the only one displayed. If there are more than one job, you can ascertain which is yours by checking the date/time stamp for user: JESUSER.
6. Hover over the row that contains the job and click **View** (👁) to the right of the **COMPLETED** column.

   The JOB page displays informational messages that were generated while the job was running.
7. Scroll to the bottom of the screen where you can see a number of entries in the DD ENTRIES group, hover over the row that contains **PRINTOUT** in the **DD NAME** column and click **View** (👁) to the right of the **RECORDS** column.

   The output from the JCL job is displayed – a bank statement – which demonstrates that you have successfully run BankDemo in batch mode.

Reopening the BankDemo data files

After you have run BankDemo in batch mode, if you want to run it again in online mode, you must first reopen the data files.

1. Switch back to your TN3270 terminal emulator, and then press **Ctrl+Shift+Z** to clear the screen.
2. Enter the following, which is the name of a supplied transaction to open BankDemo's data files:

   ```
   OFIL
   ```

   A message is displayed confirming that the data files have been opened. You are now free to run BankDemo in online mode again.

Investigating scale-out features in the BankDemo demonstration application

This section contains two tutorials that enable you to take a closer look at Enterprise Server's scale-out features in action while running the BankDemo demonstration application.

The first tutorial looks at the basic setup that enables multiple Enterprise Server virtual machines to appear to function as a single entity, while the second shows how you can change a module of an application without having to stop and restart the application.

Introduction to Enterprise Server scale-out features

Enterprise server regions are typically deployed as scale-up servers. This means that performance is improved by adding or replacing the existing hardware components. Ultimately, a server's performance is limited by the machine's existing resources.

A Performance and Availability Cluster (PAC) enables you to configure enterprise server regions in a scale-out architecture. In a PAC, multiple enterprise server regions work together as a single logical entity.

A PAC can provide the following advantages over a single enterprise server region:

- Availability - A PAC that utilizes distributed enterprise server regions improves robustness to hardware or network issues.
- Performance - Multiple enterprise server regions work together to maximize the overall performance and throughput, potentially performing better than a single enterprise server region.

The enterprise server regions in a PAC need to be able to share synchronized user and system data. To facilitate the sharing and synchronicity requirement a PAC uses a data store, referred to as a Scale-Out Repository (SOR).

Multiple Enterprise Server virtual machines acting as a single logical entity

After you have successfully deployed Enterprise Server into GCP you can perform the steps in this tutorial to illustrate that multiple regions are working together.

1. Connect a TN3270 terminal emulator to the load balancer, as shown in Connecting to the BankDemo enterprise server.
2. Press **Ctrl+Shift+Z** to clear the screen.
3. Enter the following command to close the `BNKHELP` file across the PAC:

   ```
   CFMT CL BNKHELP
   ```

4. Log on to the ESCWA administration console – see Connecting to the Enterprise Server Common Web Interface (ESCWA).
5. Click **NATIVE** from the menu.
6. In the navigation pane, expand the structure to show one of the **BNKDM** enterprise server regions.
7. Click **BNKDM**, and then select **CICS > Resources** from the menu at the top.
8. Click **Active** in the drop-down list next to **RESOURCES**.

9. Expand the **FCT** item and select **BNKHELP**.

   The **State** field shows that the BNKHELP file is Closed. (This shows as Closed on the other BNKDM region too**.)**

10. Return to your TN3270 terminal emulator. If necessary, reconnect using the same credentials that you used at step 1.

11. Press **Ctrl+Shift+Z** to clear the screen.

12. Enter the following command to open the BNKHELP file across the PAC:

    ```
    CFMT OP BNKHELP
    ```

13. Return to ESCWA in your browser and refresh the page.

    Notice that the BNKHELP file is now marked as Open for reading and writing. (Again, this will have been updated on the other BNKDM region too.)

Updating modules in a live application

This tutorial illustrates a scenario in which some mistakes have been spotted in one of the BankDemo screens, the issues have been fixed, and you need to introduce the fixed module into the PAC without having to take down the application. Without the use of a PAC, you would not be able to introduce the fixed module without having to stop and restart the BankDemo application.

1. Connect a TN3270 terminal emulator to the load balancer, as shown in Connecting to the BankDemo enterprise server.
2. Press **Ctrl+Shift+Z** to clear the screen.
3. Enter the following command to run the BankDemo application:

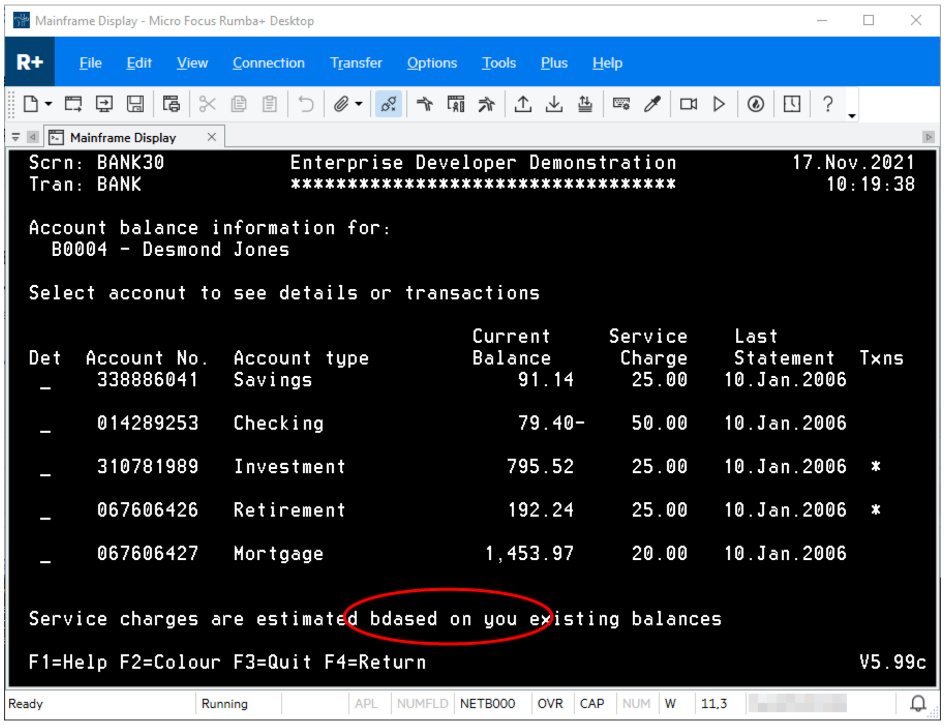
   ```
   BANK
   ```

   The BankDemo sign-on screen is displayed.

4. Specify `B0004` as the user ID, press **Tab**, then enter any non-blank character string as the password.

   The BankDemo main options screen is displayed. The cursor is positioned on the first option, **Display your account balances**.

5. Press **X** to select **Display your account balances**, and then press **Enter**.


Mainframe Display - Micro Focus Rumba+ Desktop
File Edit View Connection Transfer Options Tools Plus Help
Mainframe Display
Ready Running APL NUMFLD NETB000 OVR CAP NUM W 11,3

```
Scrn: BANK30          Enterprise Developer Demonstration         17.Nov.2021
Tran: BANK            **********************************            10:19:38

Account balance information for:
  B0004 - Desmond Jones

Select acconut to see details or transactions

                                        Current     Service    Last
Det  Account No.  Account type          Balance     Charge    Statement  Txns
 _    338886041   Savings                 91.14     25.00    10.Jan.2006

 _    014289253   Checking                79.40-    50.00    10.Jan.2006

 _    310781989   Investment             795.52     25.00    10.Jan.2006  *

 _    067606426   Retirement             192.24     25.00    10.Jan.2006  *

 _    067606427   Mortgage             1,453.97     20.00    10.Jan.2006


Service charges are estimated bdased on you existing balances

F1=Help F2=Colour F3=Quit F4=Return                                        V5.99c
```

Figure 7: Typos on the Account Balance display screen

The BankDemo account balances screen is displayed, showing the balances of the different accounts held by this user. Notice that there are two typos at the bottom of the screen: it says "bdased on you" instead of "based on your".

A corrected module was included with your enterprise server VM instances, so next you must connect to an instance and deploy the new module to your application.

6. Connect to one of the Enterprise Server regions – see Connecting to an Enterprise Server virtual machine.
7. Switch to a user that has access to the fixed module:

   ```
   sudo su demouser
   ```
8. Navigate to the module's location:

   ```
   cd /home/demouser/BankDemo_PAC
   ```
9. Rename the file, ready to deploy to the application:

   ```
   mv BBANK30P_Fixed.so BBANK30P.so
   ```
10. Deploy the file:

   ```
   caspac-aLoadCics=/home/demouser/BankDemo_PAC/BBANK30P.so     -sredis,<redis-
   instance:port> -nDEMOPAC
   ```
   - Where <redis-instance:port> is the primary endpoint displayed in the instance details page (**Memorystore** > **Redis** from the GCP project page).
   - DEMOPAC is the name of the PAC as displayed in ESCWA, within the SORs group.

A confirmation message is displayed when the new module is loaded.

11. Return to your TN3270 terminal emulator. (If necessary, reconnect using the same credentials that you used at step 1.)
12. Press **Ctrl+Shift+Z** to clear the screen.
13. Enter the following command to load the new copy of the **BBANK30P** module:

    ```
    CPMT NE BBANK30P
    ```

14. Press **Ctrl+Shift+Z** to clear the screen, type **BANK** and log on again.
15. Press **X** to select **Display your account balances**, and then press **Enter.**

    The BankDemo account balances screen is displayed, showing the balances of the different accounts held by this user. The two typos at the bottom of the screen are no longer there because the fixed module is now being used.

Housekeeping after running BankDemo

This section describes tasks you might want to perform after you have run the BankDemo demonstration application.

Stopping the BankDemo application

After you have finished with BankDemo, disconnect from the running enterprise server region, and then stop the region:

1. Disconnect your TN3270 emulator from the BankDemo application.
2. Switch back to ESCWA in your browser and perform the following for each BNKDM region:
   a. Click **GENERAL > Control**.

      The CONTROL page is displayed for the region that you selected.

   b. Click STOP.

      The **Region Stop Options** dialog box is displayed showing the region details and security configuration information. This dialog box also gives you the option of specifying operating system and security credentials to be used.

   c. Click STOP.

      After a few seconds the status of the region is shown as **Stopped**.
3. Repeat the same steps on the other BNKDM region.

Destroying the current configuration

After you have finished running the demonstration, a number of resources are still running within the cloud project, which are incurring a cost. A simple way to avoid unnecessary costs is to destroy the current configuration, that is, deleting the resources listed in Resources check; you will need to deploy the templates again in order to run the demonstration.

To destroy the configuration:

1. At the Cloud shell prompt, navigate to the project folder.
2. Destroy the current architecture:

```
terraform destroy
```

3. Click **AUTHORIZE** if prompted.

   Terraform now calculates which resources it will destroy.
4. Type **yes** to confirm.

The resources are destroyed; this can take a while. After this has completed, you can apply the Terraform architecture again.

Tip: If you receive an error whilst destroying the configuration, enter the following command and then retry the steps again:

```
terraform state rm module.sql-db.google_sql_user.default[0]
```

Commonly performed tasks for use in the tutorials

This section contains a number of tasks that you will need to perform at different stages as you follow the Enterprise Server tutorials. The steps described in this section are not part of the tutorials, but are listed here so that you can easily refer to them from the relevant points when you are running BankDemo.

Connecting to the Enterprise Server Common Web Interface (ESCWA)

All of the tutorials will require you to interact directly with the enterprise servers, either before you are able to run the BankDemo application, or during its execution (for example, when running in batch).

This section describes how to obtain the details required to log on, and then the logon process to access the administration console.

1. On the Cloud Project page, click ☰ to open the navigation menu.
2. Select **Compute Engines** > **VM instances**.
3. Locate the ESCWA VM instance and click its name.

   The details of the instance are displayed.
4. Scroll down the page to locate the Network Interfaces section and take note of the External IP address.
5. In your browser, navigate to **<external-ip>:10086**.

   The **Enterprise Server Common Web Interface** logon screen is displayed.
6. Enter the following details and click LOGON:

   - User Name: SYSAD
   - Password: SYSAD

   The ES Administration page is displayed.

Connecting to an Enterprise Server virtual machine

This section describes how to connect to one of the Linux-based Enterprise Server virtual machines within the project. Once you are connected to such a virtual machine you can use Enterprise Server as you would in a non-cloud environment.

> **Note:** The following steps assume that when deploying the Enterprise Server on GCP you used the default values within the Terraform templates. The default prefix for the two virtual machines is taken from the 'name' variable in terraform.tfvars. The rest of the name consists of '-es-' and a unique 4-char hash; for example, mf-env1-es-1n4t.

1. On the Cloud Project page, click ☰ to open the navigation menu.
2. Select **Compute Engine** > **VM Instances**.

   A list of VMs configured for this project are displayed.
3. Click the name of the required enterprise server region VM.

   The details for the server are displayed.
4. Click **SSH**.
5. Click **Connect**.

   A new browser window is opened, and within it, a shell connecting you to the server. Your purpose for connecting is likely to require the running of COBOL commands; if so, you need to set the COBOL environment.
6. Enter the following command to set the environment:

   ```
   . /opt/microfocus/EnterpriseDeveloper/bin/cobsetenv
   ```

   A message is displayed to indicate that the environment is now set.

Additional resources

This section provides links to information that is of particular use if you are not already familiar with Enterprise Server.

The Micro Focus website includes a wide variety of resources related to Enterprise Server. Many of these resources are freely available, while other resources require you to have a SupportLine login. To obtain a login, you need a valid license for Enterprise Server (or other relevant Micro Focus product). For licensing information, see Costs and licenses, earlier in this guide.

The following resources are freely available:

- Enterprise Server architecture

  https://www.microfocus.com/documentation/enterprise-developer/ed-latest/ES-WIN/GUID-B2ED168C-812D-4660-9A2C-F5A106E90FDD.html

- Enterprise Server instance architecture
  https://www.microfocus.com/documentation/enterprise-developer/ed-latest/ES-WIN/BKCACAINTRU005.html

- Scale-Out performance and availability clusters
  https://www.microfocus.com/documentation/enterprise-developer/ed-latest/ES-UNIX/GUID-F6E1BBB7-AEC2-45B1-9E36-1D86B84D2B85.html

- Enterprise Server on UNIX product documentation
  https://www.microfocus.com/documentation/enterprise-developer/ed-latest/ES-UNIX/index.html

- Enterprise Server Support Pack
  https://marketplace.microfocus.com/app-modernization/content/es-support-pack

- Micro Focus End User License Agreement & Additional License Authorizations

  https://www.microfocus.com/en-us/legal/software-licensing

- Requesting a trial license for Enterprise Server
  https://www.microfocus.com/products/enterprise-suite/enterprise-server/trial/

The following resource requires you to have a SupportLine login:

- Enterprise Server Troubleshooting Enablement Pack and Support Videos
  https://supportline.microfocus.com/examplesandutilities/TroubleShootingpack/index.aspx

Troubleshooting

Q. How should I configure my project to deploy the blueprint?

A. The provided scripts have been thoroughly tested and are designed to work with the default parameters supplied in the terraform.tfvars file. The service account being used to execute the scripts must also have sufficient permissions to create the necessary objects like virtual machines, network, database, memcache etc…

In cases where the scripts cannot be executed successfully, verify the terraform messages issued by the scripts, as well as the Google Cloud log files. In addition, you may have to verify the permissions given to the service account.

In more restricted GCP projects, you may also have to verify that no restrictions apply in the organizational policies that could cause the scripts to fail. Some important organizational policies that may affect the execution are:

- Restrict VPC peering usage
- Shielded VMs
- Define allowed external IPs for VM instances
- Define trusted image projects
- Restrict Authorized Networks on Cloud SQL instances

Q. Can I deploy this blueprint into an existing VPC instead of a new one being created upon deployment?

A. By default, a new VPC is created when you deploy the architecture; however, you can update the terraform.tfvars template file deploy to an existing VPC. Append the following settings to the template file:

```
Create_network=false
vpc_network = "<vpc-name>"
vpc_subnet = "<subnet-name>"
```

where *<vpc-name>* and *<subnet-name>* are the names of the existing resources.

Tip: The README.md, which is contained within the blueprint, contains descriptions of further settings with which to customize your deployment.

Q. My application encountered an error when running under Enterprise Server. What information do I need to include when informing Micro Focus of the issue, and what is the best way of obtaining that information?

A. For all troubleshooting exercises, gather as much information as possible about the state of the enterprise server region when the problem occurred and about the events leading up to the problem. This should include the date/time that the problem occurred, observations on what was happening in the system, how long it had been running, what the symptoms were, and how/where these were observed.

When a problem occurs, capture and provide specific logs, traces and dumps, and ideally the contents of various configuration files, directories, and the output from a number of operating system tools. Do this as soon as possible after the failure occurs.

You can use the MFESdiags diagnostic collection script— **mfesdiags.sh** —to collect the required Enterprise Server diagnostic information automatically. This script is available in the Enterprise Server Support Pack.

You should create a **.zip** file of the resultant data collection directory, and attach it to the Support Incident (SI) that you submit.

The script invokes the relevant "mfsupport" utility, which collects information about the operating system and machine hardware along with details about Micro Focus products installed.

As a minimum, if MFESdiags isn’t used to collect the data, you should collect and provide the following items as soon as possible after a failure (by zipping or ‘tar’ing the contents of the system/region directory):

- console.log
- log.html or log-*.html (for communications problems)
- Journal log (for MFDS/security problems)
- Any casdump or aux traces
- mfSupportInfo

However, the MFESdiags data collection script collects the preceding items, and other useful information details, as follows:

- All files from the Enterprise Server's System/Region Directory, including:
  - **console.log** and **console.bak**. The communications process log **log.html** or **log-*.html**.
  - Any trace diagnostic datasets, **casauxta.rec** and **casauxtb.rec**.
  - Any system abend dumps (**casdumpa.rec**, **casdumpb.rec** or **casdumpx.rec**).
  - Any HSF output files (***.csv**).
- The output from mfSupportInfo – this contains product and system information.
- The Resource Definition File (RDO/RDT), **dfhdrdat**.
- The Directory Server (MFDS) log file.
- The Directory Server configuration directory.

Notices