



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

Managed Runtime Tutorial

This tutorial shows how to deploy and test the BankDemo sample application in the AWS Mainframe Modernization managed runtime with the Micro Focus runtime engine.

Topics

- [Prerequisites \(#tutorial-runtime-prerequisites\)](#)
- [Step 1: Create a runtime environment \(#tutorial-runtime-step1\)](#)
- [Step 2: Create an application \(#tutorial-runtime-step2\)](#)
- [Step 3: Deploy an application \(#tutorial-runtime-step3\)](#)
- [Step 4: Import data sets \(#tutorial-runtime-step4\)](#)
- [Step 5: Start an application \(#tutorial-runtime-step5\)](#)
- [Step 6: Connect to the BankDemo sample application \(#tutorial-runtime-step6\)](#)

Prerequisites

Before you start the tutorial, make sure you complete the following prerequisites:

- Create an Amazon Relational Database Service PostgreSQL or Amazon Aurora PostgreSQL DB instance by following the instructions in [Creating a PostgreSQL DB instance and connecting to a database on a PostgreSQL DB instance \(https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_GettingStarted.CreatingConnecting.PostgreSQL.html\)](#) in the *Amazon RDS User Guide*.
- Upload the [BankDemo sample application](#) [\[https://d1vi4vxke6c2hu.cloudfront.net/demo/bankdemo_runtime.zip\]](https://d1vi4vxke6c2hu.cloudfront.net/demo/bankdemo_runtime.zip) to Amazon S3. For more information, see the [Enterprise Build Tools Tutorial](#) [\[https://d1vi4vxke6c2hu.cloudfront.net/tutorial/tutorial-build.pdf\]](https://d1vi4vxke6c2hu.cloudfront.net/tutorial/tutorial-build.pdf).
- (Optional) Create file storage of type Amazon Elastic File System or Amazon FSx for Lustre.

Create and configure an AWS Key Management Service key and AWS Secrets Manager secret

An AWS KMS key is required in order to create a secret in Secrets Manager. Follow these steps to securely store the credentials for the Amazon RDS database instance you created for this tutorial. For more information, see [Prerequisites \(#tutorial-runtime-prerequisites\)](#).

1. To create the key, follow the steps in [Creating keys \(https://docs.aws.amazon.com/kms/latest/developerguide/create-keys.html\)](#) in the *AWS Key Management Service Developer Guide*. You will need to edit the key policy for AWS Mainframe Modernization before you finish creating the key.
2. After you finish selecting the IAM users and roles that can use the key (step 15), edit the key policy to grant

AWS Mainframe Modernization principal decrypt permissions by adding (not replacing) the following policy statements:

```
{
  "Effect" : "Allow",
  "Principal" : {
    "Service" : "m2.amazonaws.com"
  },
  "Action" : "kms:Decrypt",
  "Resource" : "*"
}
```

3. To store the database credentials as a secret in Secrets Manager, follow the steps in [Create a secret](https://docs.aws.amazon.com/secretsmanager/latest/userguide/manage_create-basic-secret.html) (https://docs.aws.amazon.com/secretsmanager/latest/userguide/manage_create-basic-secret.html) in the *AWS Secrets Manager User Guide*. You will specify the key you created in the previous steps for the encryption key. You will also need to edit the resource permissions before you finish storing the secret.
4. In **Resource permissions - optional**, choose **Edit permissions**. In the editor, add a resource-based policy, such as the following, to share the secret with AWS Mainframe Modernization.

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Principal" : {
        "Service" : "m2.amazonaws.com"
      },
      "Action" : "secretsmanager:GetSecretValue",
      "Resource" : "*"
    }
  ]
}
```

Upload the data sets

Upload the data sets required for the BankDemo sample application to an Amazon S3 bucket, such as `s3://m2-tutorial`. The following JSON shows all the required data sets. Replace `$$S3_DATASET_PREFIX` with your Amazon S3 bucket that contains the catalog data. For example, `m2-tutorial/catalog`.

```
{
  "dataSets": [
    {
```

```

    "dataSet": {
      "location": "sql://ESPACDatabase/VSAM/CATALOG.DAT",
      "name": "CATALOG"
    },
    "externalLocation": {
      "s3Location": "$S3_DATASET_PREFIX/CATALOG.DAT"
    }
  },
  {
    "dataSet": {
      "location": "sql://ESPACDatabase/VSAM/SPLDSN.dat",
      "name": "SPLDSN"
    },
    "externalLocation": {
      "s3Location": "$S3_DATASET_PREFIX/SPLDSN.dat"
    }
  },
  {
    "dataSet": {
      "location": "sql://ESPACDatabase/VSAM/SPLJNO.dat?type=seq\\;
reclen=80,80",
      "name": "SPLJNO"
    },
    "externalLocation": {
      "s3Location": "$S3_DATASET_PREFIX/SPLJNO.dat"
    }
  },
  {
    "dataSet": {
      "location": "sql://ESPACDatabase/VSAM/SPLJOB.dat",
      "name": "SPLJOB"
    },
    "externalLocation": {
      "s3Location": "$S3_DATASET_PREFIX/SPLJOB.dat"
    }
  },
  {
    "dataSet": {
      "location": "sql://ESPACDatabase/VSAM/SPLMSG.dat",
      "name": "SPLMSG"
    },
    "externalLocation": {
      "s3Location": "$S3_DATASET_PREFIX/SPLMSG.dat"
    }
  },
  {
    "dataSet": {
      "location": "sql://ESPACDatabase/VSAM/SPLOUT.dat",

```

```

        "name": "SPLOUT"
    },
    "externalLocation": {
        "s3Location": "$S3_DATASET_PREFIX/SPLOUT.dat"
    }
},
{
    "dataSet": {
        "location": "sql://ESPACDatabase/VSAM/SPLSUB.dat",
        "name": "SPLSUB"
    },
    "externalLocation": {
        "s3Location": "$S3_DATASET_PREFIX/SPLSUB.dat"
    }
},
{
    "dataSet": {
        "location": "sql://ESPACDatabase
/VSAM/MFI01V.MFIDEMO.BNKACC.DAT?folder=/DATA",
        "name": "MFI01V.MFIDEMO.BNKACC"
    },
    "externalLocation": {
        "s3Location": "$S3_DATASET_PREFIX/data/MFI01V.MFIDEMO.BNKACC.DAT"
    }
},
{
    "dataSet": {
        "location": "sql://ESPACDatabase
/VSAM/MFI01V.MFIDEMO.BNKATYPE.DAT?folder=/DATA",
        "name": "MFI01V.MFIDEMO.BNKATYPE"
    },
    "externalLocation": {
        "s3Location": "$S3_DATASET_PREFIX/data/MFI01V.MFIDEMO.BNKATYPE.DAT"
    }
},
{
    "dataSet": {
        "location": "sql://ESPACDatabase
/VSAM/MFI01V.MFIDEMO.BNKCUST.DAT?folder=/DATA",
        "name": "MFI01V.MFIDEMO.BNKCUST"
    },
    "externalLocation": {
        "s3Location": "$S3_DATASET_PREFIX/data/MFI01V.MFIDEMO.BNKCUST.DAT"
    }
},
{
    "dataSet": {
        "location": "sql://ESPACDatabase

```

```

/VSAM/MFI01V.MFIDEMO.BNKHELP.DAT?folder=/DATA",
    "name": "MFI01V.MFIDEMO.BNKHELP"
  },
  "externalLocation": {
    "s3Location": "$S3_DATASET_PREFIX/data/MFI01V.MFIDEMO.BNKHELP.DAT"
  }
},
{
  "dataSet": {
    "location": "sql://ESPACDatabase
/VSAM/MFI01V.MFIDEMO.BNKTXN.DAT?folder=/DATA",
    "name": "MFI01V.MFIDEMO.BNKTXN"
  },
  "externalLocation": {
    "s3Location": "$S3_DATASET_PREFIX/data/MFI01V.MFIDEMO.BNKTXN.DAT"
  }
},
{
  "dataSet": {
    "location": "sql://ESPACDatabase/VSAM/YBATTSO.PRC?folder=/PRC\;
type=lseq\;reclen=80,80",
    "name": "YBATTSO.PRC"
  },
  "externalLocation": {
    "s3Location": "$S3_DATASET_PREFIX/prc/YBATTSO.prc"
  }
},
{
  "dataSet": {
    "location": "sql://ESPACDatabase/VSAM/YBNKEXTV.PRC?folder=/PRC\;
type=lseq\;reclen=80,80",
    "name": "YBNKEXTV.PRC"
  },
  "externalLocation": {
    "s3Location": "$S3_DATASET_PREFIX/prc/YBNKEXTV.prc"
  }
},
{
  "dataSet": {
    "location": "sql://ESPACDatabase/VSAM/YBNKPRT1.PRC?folder=/PRC\;
type=lseq\;reclen=80,80",
    "name": "YBNKPRT1.PRC"
  },
  "externalLocation": {
    "s3Location": "$S3_DATASET_PREFIX/prc/YBNKPRT1.prc"
  }
},
{

```

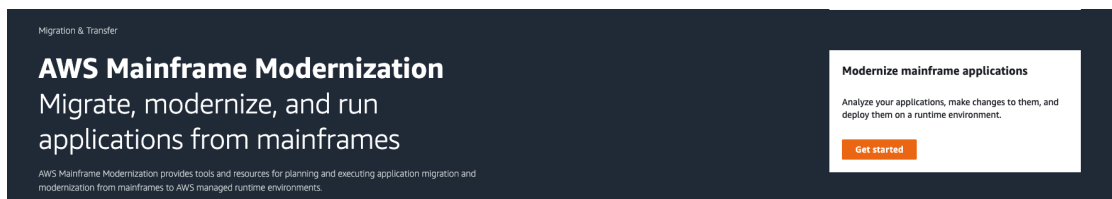
```

        "dataSet": {
            "location": "sql://ESPACDatabase/VSAM/YBNKSRT1.PRC?folder=/PRC\\;
type=lseq\\;reclen=80,80",
            "name": "YBNKSRT1.PRC"
        },
        "externalLocation": {
            "s3Location": "$S3_DATASET_PREFIX/prc/YBNKSRT1.prc"
        }
    },
    {
        "dataSet": {
            "location": "sql://ESPACDatabase/VSAM/KBNKSRT1.TXT?folder=
/CTLCARDS\\;type=lseq\\;reclen=80,80",
            "name": "KBNKSRT1"
        },
        "externalLocation": {
            "s3Location": "$S3_DATASET_PREFIX/ctlcards/KBNKSRT1.txt"
        }
    }
]
}

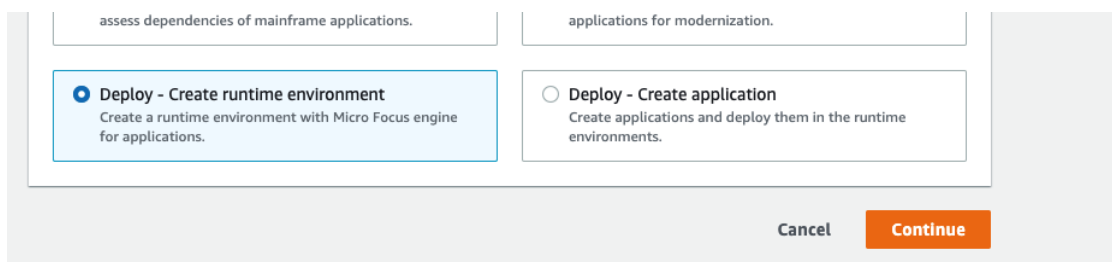
```

Step 1: Create a runtime environment

1. Open the [AWS Mainframe Modernization console](https://us-west-2.console.aws.amazon.com/m2/home?region=us-west-2#/) (https://us-west-2.console.aws.amazon.com/m2/home?region=us-west-2#/) and choose **Get started**.



2. On the **Get started** page, under **What would you like to do?**, choose **Deploy - Create runtime environment**, and then choose **Continue**.



3. On the **Create environment** page, under **Permissions (one time setup)**, choose **I grant AWS Mainframe Modernization the required permissions**. These permissions allow AWS Mainframe Modernization to create AWS resources on your behalf. This is a one time setup step, which will grant AWS Mainframe Modernization the necessary permissions. If permissions were previously granted, choose **Continue**.

Permissions (one time setup)

AWS Mainframe Modernization requires permissions to create environments and applications. [Learn more](#)

✔ You have granted AWS Mainframe Modernization the required permissions.

Cancel
Continue

4. Specify a name and optional description for the runtime environment, and choose **Next**.

AWS Mainframe Modernization > Environments > Create Environment

Step 1
Specify basic information

Step 2
Specify configurations

Step 3 - *Optional*
Attach storage

Step 4
Review and create

Specify basic information Info

Name and description

Environment name

Use only letters A-Z and a-z, numbers 0-9, or hyphens "-". The maximum length is 100 characters.

Environment description - *optional*

The description can be up to 500 characters.

Engine options

Micro Focus version

This engine provides a mainframe-compatible runtime for replatformed applications

Version 7.0.3 ▼

CloudWatch metrics and logging

CloudWatch metrics and logging are enabled so that you can monitor your environment and debug it later.

[Learn more](#)

Tags - *optional*

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

No tags associated with the environment yet.

Add new tag

You can add up to 50 more tags.

Cancel
Next

5. Under **Availability**, choose **High availability cluster**. Under **Resources**, choose an instance type and the number of instances you want. Then choose **Next**.

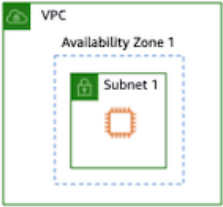
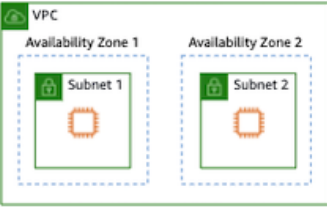
Specify configurations [Info](#)

Availability [Info](#)

Choose the availability pattern for your environment.

Standalone runtime environment
Sets up a single instance in a single availability zone. Does not guarantee high availability but costs less.

High availability cluster
Sets up redundant instances across two availability zones. Enables higher availability but costs more.

Resources

Instance type
Choose the instance type for your high availability cluster.

M2.m5.large ▼

Desired capacity
Specify the desired number of instances.

2

Security and network

Allow applications deployed to this environment to be publicly accessible.

Virtual Private Cloud (VPC)
Choose the VPC where you want to create the environment.

Default vpc-15 ▼

Subnets
Choose one or more subnets for a high availability setup.

Choose subnets ▼

subnet-56f1e | us-west-2a ✕

subnet-6685l | us-west-2b ✕

Security groups
Choose one or more security groups for the chosen VPC.

Choose security groups ▼

default ✕
default VPC security group

Cancel Previous **Next**

6. On the **Attach storage** page, optionally specify either Amazon EFS or Amazon FSx file systems. Then choose

Next.

Attach storage - *Optional* [Info](#)

EFS storage

Choose one or more existing EFS file systems. Specify a mount point for each system.

EFS file system Mount point - *optional*

You have reached the limit of 1 EFS.

FSx storage

Choose one or more existing FSx file systems. Specify a mount point for each system.

No EFS associated with this environment.

You can add up to 1 more FSx.

7. On the **Review and create** page, review all the configurations you selected for the runtime environment and choose **Create Environment**.

Step 2: Specify configurations Edit

Environment setup

| | |
|--------------------------------------------|-------------------------------------------------------------------|
| Availability High availability | Instance type M2.m5.large |
| VPC vpc-1! ↗ | Subnets subnet- ↗ subnet- ↗ |
| Security groups sg-9! ↗ | Public access to applications in this environment Enabled |

Step 3: Attach storage Edit

EFS storage (1)

| Storage ID | Storage name | Mount point |
|-------------------------|--------------|----------------|
| fs-0e ↗ | efs | /m2/mount/demo |

FSx storage

| Storage ID | Storage name | Mount point |
|--------------------------------------|--------------|-------------|
| No storage No storage to display. | | |

Cancel Previous Create environment

8. Wait until environment creation succeeds and the status changes to **Available**. It might take up to two minutes.

Environment "Tutorial" was created successfully. Deploy application ×

AWS Mainframe Modernization > Environments

Environments (7) Info ↻ Actions Create environment

| <input type="checkbox"/> | Environment name | Status | Engine | Version | Instance type |
|--------------------------|------------------|-----------|-------------|---------|---------------|
| <input type="checkbox"/> | Tutorial | Available | Micro Focus | 7.0.3 | M2.m5.large |

Step 2: Create an application

1. In the navigation pane, choose **Applications**. Then choose **Create application**.

AWS Mainframe Modernization > Applications

Applications (3) Info ↻ Actions Create application

| <input type="checkbox"/> | Name | Status | ARN | Description | Engine type | Creation time |
|--------------------------|------|--------|-----|-------------|-------------|---------------|
|--------------------------|------|--------|-----|-------------|-------------|---------------|

2. On the **Specify basic information** page, specify the application name and an optional description. You can

also provide optional tagging information. Then choose **Next**.

AWS Mainframe Modernization > Applications > Create application

Step 1
Specify basic information

Step 2
Specify resources and configurations

Step 3
Review and create

Specify basic information [Info](#)

Name and description


Application Name

Use only letters, numbers or hyphens. The maximum length is 100 characters.

Application Description - optional

The maximum length is 500 characters.

Engine type

 **MICRO FOCUS**® Micro Focus version
This engine provides a mainframe-compatible runtime for replatformed applications.

Tags - optional

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

No tags associated with the application yet.

You can add up to 50 more tags.

[Cancel](#)

3. On the **Specify resources and configurations** page, choose whether you want to specify the application definition using the inline editor or provide an Amazon S3 bucket where the application definition file is stored. Either type the application definition or provide the Amazon S3 location. Then choose **Next**.

AWS Mainframe Modernization > Applications > Create application

Step 1
Specify basic information

Step 2
Specify resources and configurations

Step 3
Review and create

Specify resources and configurations [Info](#)

Resources and configurations

Choose an approach to define the application

Specify the application definition with its resources and configurations using the inline editor

Use an application definition JSON file in an Amazon S3 bucket

```

1 {
2   "resources": [
3     {
4       "resource-type": "vsam-config",
5       "resource-id": "vsam-1",
6       "properties": {
7         "secret-manager-arn": "arn:aws:secretsmanager:us
8       }
9     },
10    {
11      "resource-type": "cics-resource-definition",
12      "resource-id": "resource-definition-1",
13      "properties": {
14        "file-location": "${s3-source}/RDEF",
15        "system-initialization-table": "BNKCICV"
16      }
17    },
18    {
19      "resource-type": "cics-transaction",
20      "resource-id": "transaction-1",
21      "properties": {
22        "file-location": "${s3-source}/transaction"
23      }
24    },
25    {
26      "resource-type": "mf-listener",
27      "resource-id": "listener-1",
28      "properties": {
29        "port": 6000,
30        "conversation-type": "tn3270"
31      }
32    },
33    {
34      "resource-type": "xa-resource",
35      "resource-id": "xa-resource-1",
36      "properties": {
37        "name": "XASQL",
38        "module": "${s3-source}/xa/ESPGSQLXA64.so",

```

JSON Ln 4, Col 21 Errors: 0 Warnings: 0

The maximum size of the JSON file is 500 kB.

Cancel Previous **Next**

You can use the following sample application definition file. Make sure to replace \$SECRET_ARN, \$S3_BUCKET, and \$S3_PREFIX with the correct values for you.

```

{
  "resources": [
    {
      "resource-type": "vsam-config",
      "resource-id": "vsam-1",
      "properties": {
        "secret-manager-arn": "$SECRET_ARN"
      }
    },
    {
      "resource-type": "cics-resource-definition",
      "resource-id": "resource-definition-1",
      "properties": {

```

```

        "file-location": "${s3-source}/RDEF",
        "system-initialization-table": "BNKCICV"
    }
},
{
    "resource-type": "cics-transaction",
    "resource-id": "transaction-1",
    "properties": {
        "file-location": "${s3-source}/transaction"
    }
},
{
    "resource-type": "mf-listener",
    "resource-id": "listener-1",
    "properties": {
        "port": 6000,
        "conversation-type": "tn3270"
    }
},
{
    "resource-type": "xa-resource",
    "resource-id": "xa-resource-1",
    "properties": {
        "name": "XASQL",
        "module": "${s3-source}/xa/ESPGSQLXA64.so",
        "secret-manager-arn": "$SECRET_ARN"
    }
},
{
    "resource-type": "jes-initiator",
    "resource-id": "jes-initiator-1",
    "properties": {
        "initiator-class": "A",
        "description": "initiator...."
    }
},
{
    "resource-type": "jcl-job",
    "resource-id": "jcl-job-1",
    "properties": {
        "file-location": "${s3-source}/jcl"
    }
}
],
"source-locations": [
    {
        "source-id": "s3-source",

```

```

    "source-type": "s3",
    "properties": {
      "s3-bucket": "$S3_BUCKET",
      "s3-key-prefix": "$S3_PREFIX"
    }
  ]
}

```

Note

This file is subject to change.

4. On the **Review and create** page, review the information you provided and choose **Create application**.

Review and create [Info](#)

Step 1: Specify basic information [Edit](#)

Basic information

| | |
|----------|-----------------------|
| Name | Description |
| BankDemo | Bank demo application |

Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

| Tag key | Tag value |
|--------------------------------|-----------|
| No tags No tags to display. | |

[Add tags](#)

Step 2: Specify resources and configurations [Edit](#)

Resources and configurations

Template

```

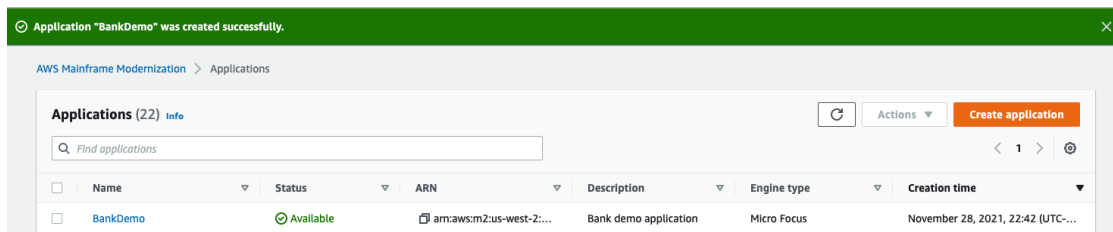
1  {
2    "resources": [
3      {
4        "resource-type": "vsam-config",
5        "resource-id": "vsam-1",
6        "properties": {
7          "secret-manager-arn": "arn:aws:secretsmanager:u
8        }
9      },
10     {
11      "resource-type": "cics-resource-definition",
12      "resource-id": "resource-definition-1",
13      "properties": {
14        "file-location": "${s3-source}/RDEF",
15        "system-initialization-table": "BNKCICV"
16      }
17     },
18     {
19      "resource-type": "cics-transaction",
20      "resource-id": "transaction-1",
21      "properties": {
22        "file-location": "${s3-source}/transaction"
23      }
24     }
25   ]
26 }

```

JSON Ln 1, Col 1 0 Errors: 0 0 Warnings: 0

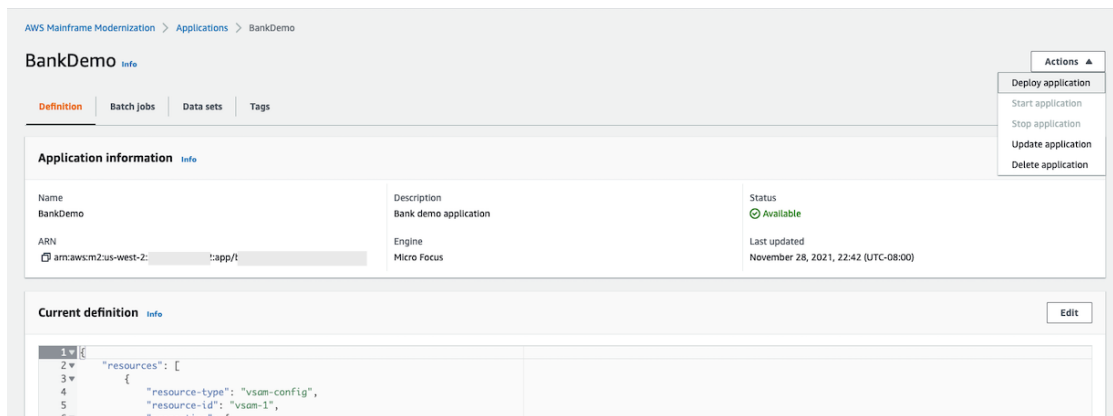
[Cancel](#) [Previous](#) [Create](#)

5. Wait until the application creation operation succeeds and the status changes to **Available**.

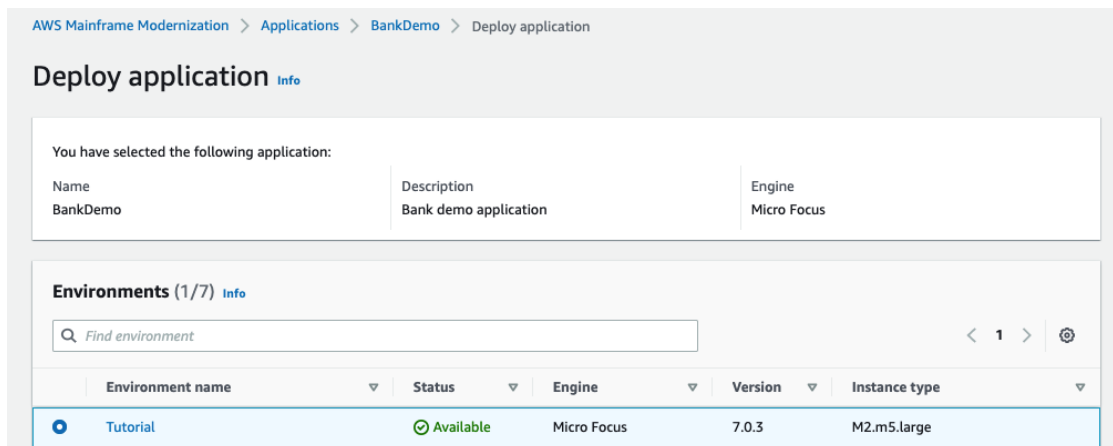


Step 3: Deploy an application

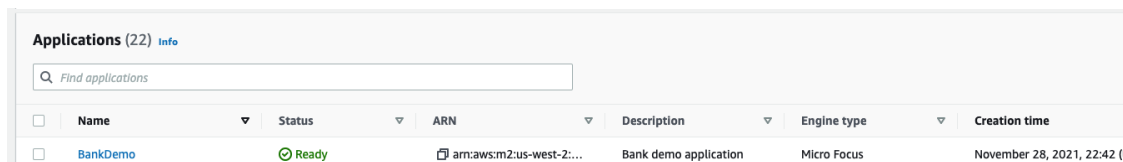
1. In the navigation pane, choose **Applications**, then choose **BankDemo**. Choose **Actions**, then choose **Deploy application**.



2. Choose the **Tutorial** environment and then choose **Deploy**.



3. Wait until the BankDemo application deployment succeeds and the status changes to **Ready**.



Step 4: Import data sets

1. In the navigation pane, choose **Applications**, then choose **BankDemo**. Choose the **Data sets** tab. Then choose **Import**.

AWS Mainframe Modernization > Applications > BankDemo > Import data set

Import data set [Info](#)

Import data set

Choose selection type

Use data set configuration JSON file in an Amazon S3 bucket
 Specify the data set configuration values separately

No items associated with the resource.

[Add new item](#)

[Cancel](#) [Submit](#)

2. Choose **Add new item** to add each data set and choose **Submit** after you provide the information for each data set.

AWS Mainframe Modernization > Applications > BankDemo > Import data set

Import data set [Info](#)

Import data set

Choose selection type

Use data set configuration JSON file in an Amazon S3 bucket
 Specify the data set configuration values separately

| Name | Location | External S3 location | |
|----------------------|------------------------|-------------------------|------------------------|
| CATALOG | sql://ESPACDatabase/VS | m2-tutorial/CATALOG.D/ | Remove |
| SPLDSN | sql://ESPACDatabase/VS | m2-tutorial/SPLDSN.dat | Remove |
| SPLJNO | sql://ESPACDatabase/VS | m2-tutorial/SPLJNO.dat | Remove |
| MFI01V.MFIDEMO.BNKAI | sql://ESPACDatabase/VS | m2-tutorial/data/MFI01\ | Remove |
| FI01V.MFIDEMO.BNKATY | sql://ESPACDatabase/VS | m2-tutorial/data/MFI01\ | Remove |
| YBATTSO.PRC | sql://ESPACDatabase/VS | m2-tutorial/prc/YBATTSS | Remove |

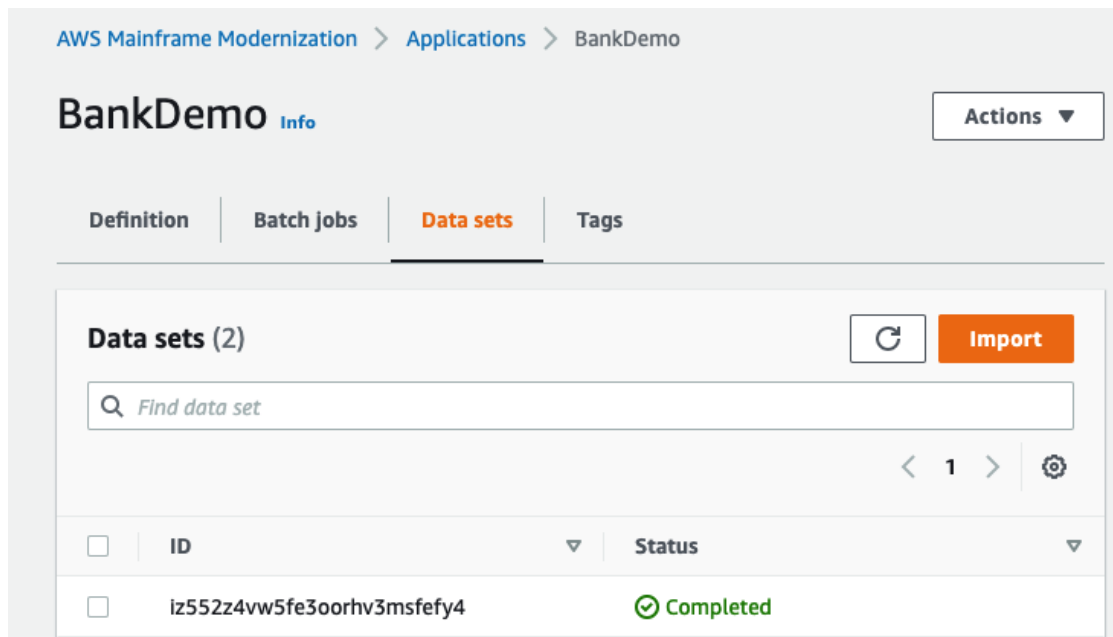
[Add new item](#)

[Cancel](#) [Submit](#)

The following table lists the data sets that you need to import. Replace `$$S3_DATASET_PREFIX` with your Amazon S3 bucket that contains the catalog data. For example, `S3_DATASET_PREFIX=s3://m2-tutorial/catalog`.

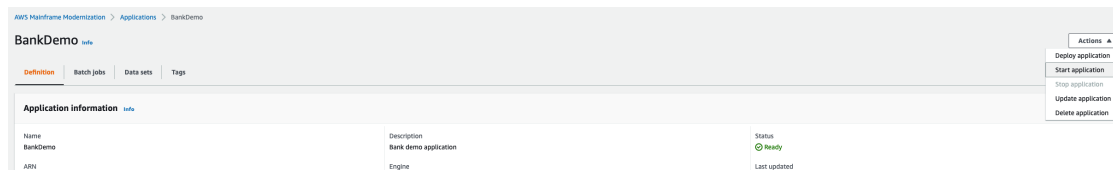
| Name | Location | External Amazon |
|-------------------------|---------------------------------------------------------------------------------|---------------------------------|
| CATALOG | sql://ESPACDatabase/VSAM/CATALOG.DAT | \$\$S3_DATASET_PR |
| SPLDSN | sql://ESPACDatabase/VSAM/SPLDSN.dat | \$\$S3_DATASET_PR |
| SPLJNO | sql://ESPACDatabase/VSAM/SPLJNO.dat?type=seq\;reclen=80,80 | \$\$S3_DATASET_PR |
| SPLJOB | sql://ESPACDatabase/VSAM/SPLJOB.dat | \$\$S3_DATASET_PR |
| SPLMSG | sql://ESPACDatabase/VSAM/SPLMSG.dat | \$\$S3_DATASET_PR |
| SPLOUT | sql://ESPACDatabase/VSAM/SPLOUT.dat | \$\$S3_DATASET_PR |
| SPLSUB | sql://ESPACDatabase/VSAM/SPLSUB.dat | \$\$S3_DATASET_PR |
| MFI01V.MFIDEMO.BNKACC | sql://ESPACDatabase/VSAM/MFI01V.MFIDEMO.BNKACC.DAT?folder=/DATA | \$\$S3_DATASET_PR/MFI01V.MFIDEM |
| MFI01V.MFIDEMO.BNKATYPE | sql://ESPACDatabase/VSAM/MFI01V.MFIDEMO.BNKATYPE.DAT?folder=/DATA | \$\$S3_DATASET_PR/MFI01V.MFIDEM |
| MFI01V.MFIDEMO.BNKCUST | sql://ESPACDatabase/VSAM/MFI01V.MFIDEMO.BNKCUST.DAT?folder=/DATA | \$\$S3_DATASET_PR/MFI01V.MFIDEM |
| MFI01V.MFIDEMO.BNKHELP | sql://ESPACDatabase/VSAM/MFI01V.MFIDEMO.BNKHELP.DAT?folder=/DATA | \$\$S3_DATASET_PR/MFI01V.MFIDEM |
| MFI01V.MFIDEMO.BNKTXN | sql://ESPACDatabase/VSAM/MFI01V.MFIDEMO.BNKTXN.DAT?folder=/DATA | \$\$S3_DATASET_PR/MFI01V.MFIDEM |
| YBATTSO.PRC | sql://ESPACDatabase/VSAM/YBATTSO.PRC?folder=/PRC\;type=lseq\;reclen=80,80 | \$\$S3_DATASET_PR/YBATTSO.prc |
| YBNKEXTV.PRC | sql://ESPACDatabase/VSAM/YBNKEXTV.PRC?folder=/PRC\;type=lseq\;reclen=80,80 | \$\$S3_DATASET_PR/YBNKEXTV.prc |
| YBNKPRT1.PRC | sql://ESPACDatabase/VSAM/YBNKPRT1.PRC?folder=/PRC\;type=lseq\;reclen=80,80 | \$\$S3_DATASET_PR/YBNKPRT1.prc |
| YBNKSRT1.PRC | sql://ESPACDatabase/VSAM/YBNKSRT1.PRC?folder=/PRC\;type=lseq\;reclen=80,80 | \$\$S3_DATASET_PR/YBNKSRT1.prc |
| KBNKSRT1 | sql://ESPACDatabase/VSAM/KBNKSRT1.TXT?folder=/CTLCARDS\;type=lseq\;reclen=80,80 | \$\$S3_DATASET_PR/KBNKSRT1.txt |

3. Wait until the data set import process completes and the status changes to **Completed**.

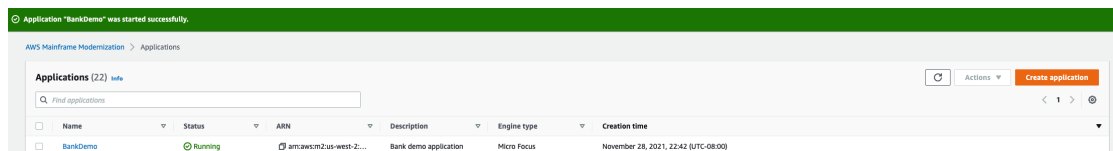


Step 5: Start an application

1. In the navigation pane, choose **Applications**, then choose **BankDemo**. Choose **Actions**, then choose **Start application**.

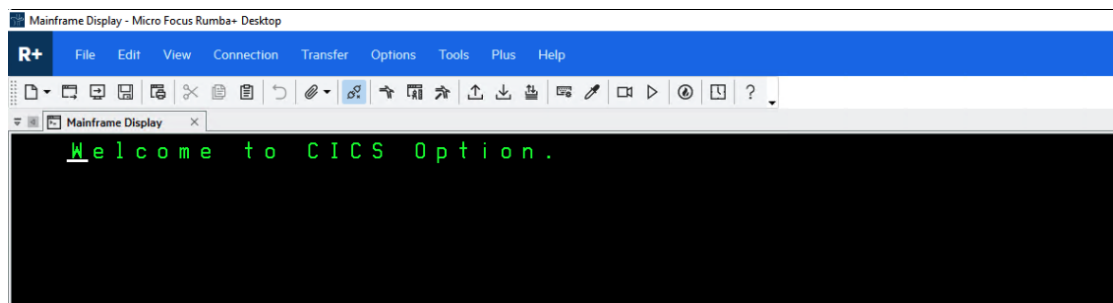


2. Wait until the BankDemo application starts successfully and the status changes to **Running**.

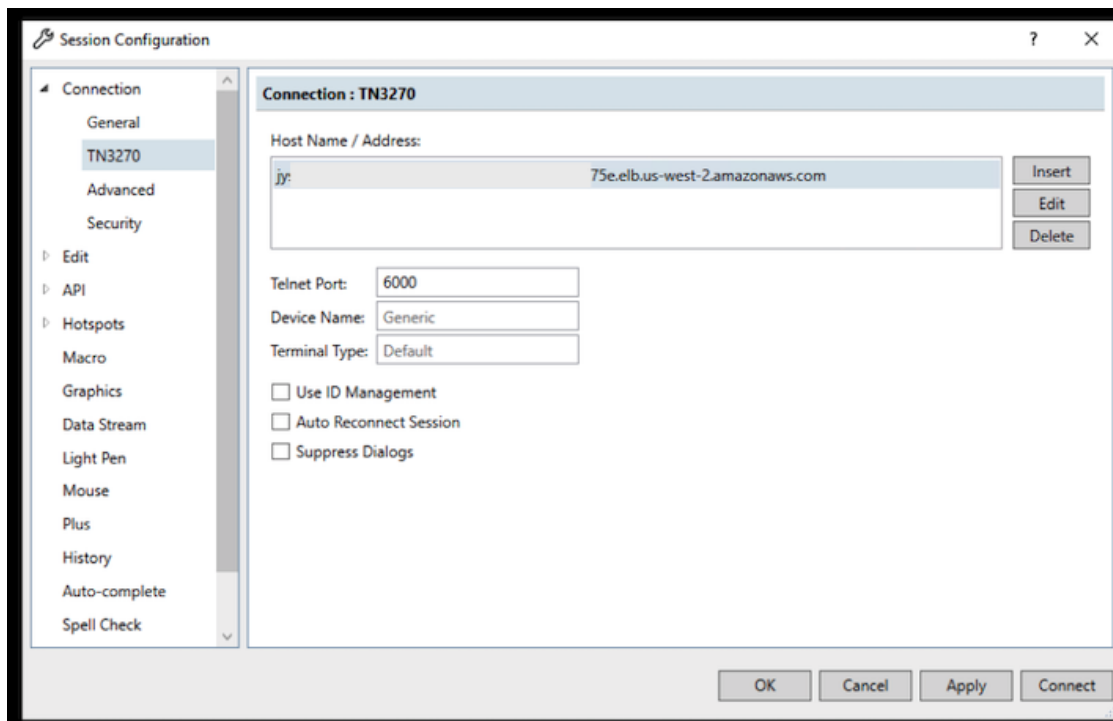
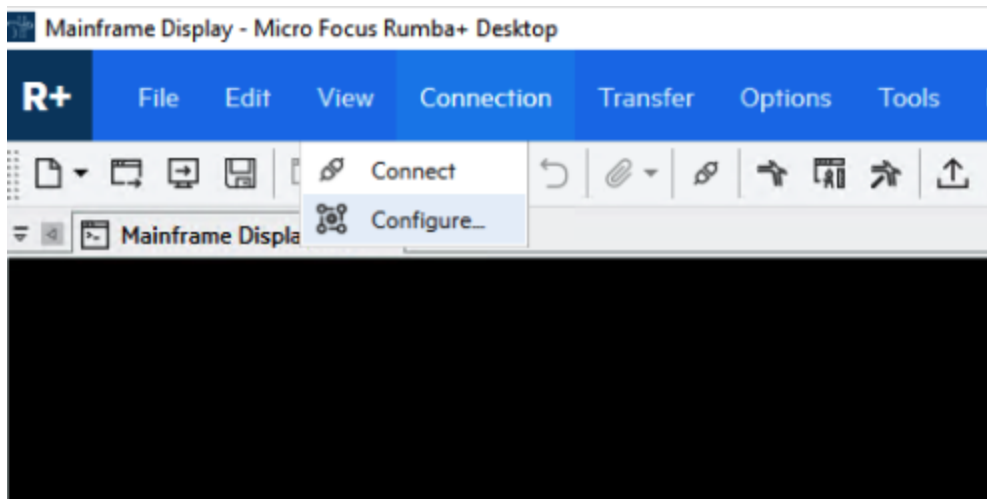


Step 6: Connect to the BankDemo sample application

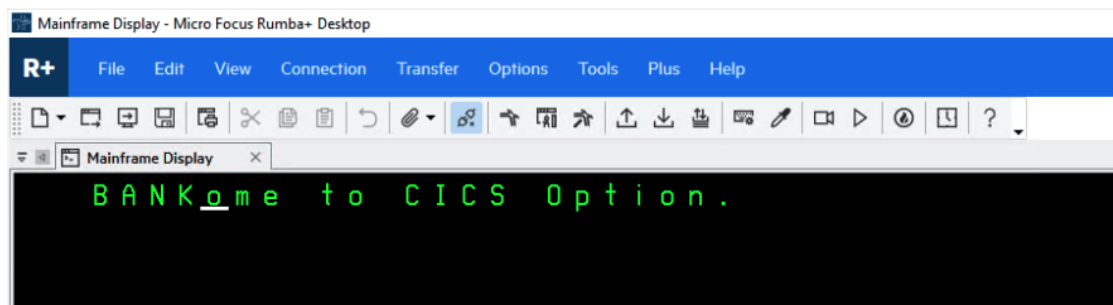
1. Start the terminal emulator you want. This tutorial uses Micro Focus Rumba+.



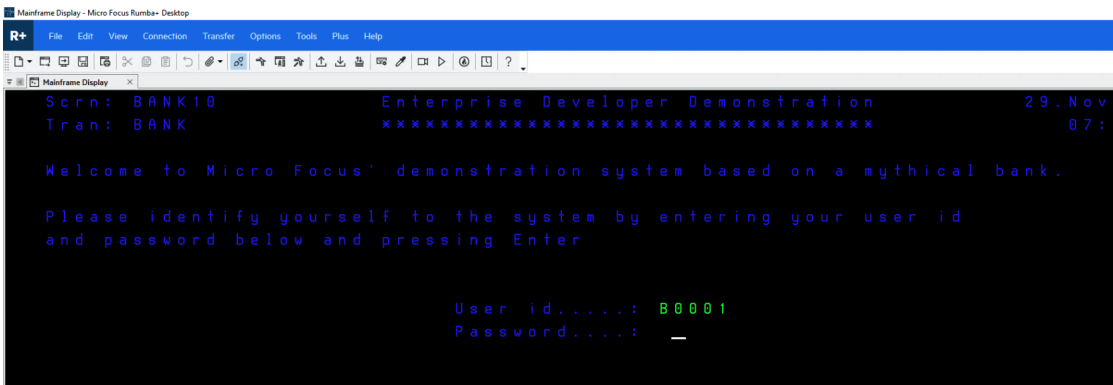
2. Choose **Connection**, then **Configure**, then **TN3270**.



3. Enter the BANK transaction name.



4. Type B0001 for the username and A for the password.



5. After you log in successfully, you can navigate through the BankDemo application.

