How to deploy and run IBM® Maximo® Asset Management on Red Hat® OpenShift®

OpenShift is a platform-as-a-service system that is built around containers and uses container orchestration provided by Kubernetes.

Maximo Asset Management uses a WebSphere Liberty runtime to run the code in a containerized environment in Red Hat OpenShift. Running using traditional WAS ND is not supported within the containerized environment. The following link provides some information on the support for running Maximo code in WebSphere Liberty. Note that some limitations currently exist.

https://www.ibm.com/support/pages/maximo-asset-management-761-websphere-liberty-support

Note that this document does not describe how the Maximo database is set up and configured and assumes that the database is ready with the right version of Maximo code that's going to run on OpenShift. Always ensure that the right database schema of Maximo Asset Management is available when running Maximo code on any application server runtime environment. Information about how to set up the Maximo database is available from the IBM Knowledge Center documentation for Maximo Asset Management. To configure Db2®manually, please refer to the instructions available in the Manually Configuring Db2 topic.

Note that in order to run Maximo Asset Management on OpenShift, a minimum of Maximo Asset Management version 7.6.1.1 with the latest interim fix is required.

In order to run Maximo Asset Management on Red Hat OpenShift, you must complete the following tasks:

1. Understand Maximo workloads
2. Create container image for a Maximo workload
3. Deploy container image of a Maximo workload
4. Expose a Maximo workload

Understand Maximo workloads

The Maximo runtime is broken up into various workloads. A workload is a specific type of work that can be used to isolate the processing needs of that work so that it can be independently managed. For example, a UI workload is a type of work that allows a user to access the user interface of Maximo Asset Management using a web browser. A cron task workload is a type of work that allows background jobs to be run. When a UI workload and cron task workload are deployed on two separate machines, the CPU and memory consumed by these workloads do not affect each other, and these workloads can be independently scaled and managed based on the needs.

Maximo Asset Management provides six different workloads (UI, cron task, report, API, MEA, and JMS consumer) that can be used independently based on the processing and isolation needs. Maximo UI and cron task workloads are required to run Maximo Asset Management, and other workloads can be deployed based on the need. To run a workload, the Maximo code needs to be packaged to contain the needed code into an application bundle that can be used to create a container. Maximo Asset Management provides tools that can be used to build these packages for each workload. For each workload that needs to be run, a separate container image must be built and deployed into the containerized environment. The following table summarizes the various workloads of Maximo Asset Management and their purpose:

Workload	Bundle	Workload folder	Description
UI	Maximo User Interface	maximo-ui	The UI code and supporting code. Required because it is the primary interface for accessing Maximo Asset Management.
	Maximo-X	maximo-x	Bundle contains the static UI code for the Work Center UI. Required.
Cron	Maximo Cron	maximo-cron	Contains the code that is needed to run Maximo cron tasks. Required.
API	Maximo Application Programming Interface	maximo-api	Contains the older REST API and the new REST API code. Optional. Used only if the API needs to be available separately. By default, the REST API is bundled with the Maximo UI bundle.
JMS Consumer	Maximo JMS Consumer	maximo-jms-consumer	Used to continuously process the integration messages available on JMS queues by using message-driven beans (MDB). Optional. Used only if messages need to be processed.
MEA	Maximo Enterprise Adapter	maximo-mea	This bundle exposes the enterprise web services API. Optional. Used only if the web services API is used.
Report	Maximo Report	maximo-report	Contains the code that is needed to enable BIRT Report Only Server (BROS). Optional. Used to separate out the work load that is related to execution of reports that are submitted from the Maximo UI. The report engine is also bundled with the Maximo UI bundle, and the BROS server is used only if the BROS server URL is set up by using the Maximo property mxe.report.birt.viewrurl.

Create container image for a workload

Prerequisites:
To create container images, the following items are required:

1. A dedicated Maximo administration workstation.

   Maximo code is installed on this administration workstation using the standard Maximo installation procedures. This workstation is also where any new fix packs or feature packs, add-ons, industry solutions, or any customizations are applied to the installed Maximo code. This workstation can be used for two different purposes: creating the application bundles that are needed for building the container image and building a container image that is based on the application bundles. If this workstation is used for building the container image, additional disk space is needed to store the container image that is built. Note that this workstation is not used for running Maximo code, and the code will be run in the OpenShift environment.

2. A docker runtime

   Docker is a platform-as-a-service product that uses OS level virtualizations and allows software to be packaged as an image and run as a process by using containers. Docker provides a set of tools for building the images and for running the image code as a container. Ensure that Docker is installed on the workstation where the Maximo container image is built. Run `docker --version` from a command prompt to make sure Docker is installed successfully.

NOTE: The procedure outlined in this document assumes the Maximo administration workstation is used for creating the Maximo application bundles and for building the Maximo container image that is needed to run the Maximo code. It is possible to separate the creation of Maximo application bundles and the building of the container image using two different workstations, and this method requires copying the Maximo application bundles onto the workstation where the container image is built.

Creating a Maximo container image:

Before creating a Maximo container image, ensure that the prerequisites mentioned earlier are in place. As indicated in the Maximo workloads section, a separate container image needs to be built for each workload. The process of building a container image is the same for any workload. The only thing that differs is the configuration and code that's specific to the workload.

a) Configure files needed for building workload bundles.
b) Build the application bundle needed for the Maximo workload.
c) Build the container image needed for the Maximo workload.

Configure files needed for building workload bundles:

1. Every workload of Maximo Asset Management is run within the WebSphere Liberty runtime and requires application bundles to be created. These application bundles are created based on Java™ EE specification, and depending on the workload, it will be a Web Application Archive file (.war file) or an Enterprise Application Archive file (.ear file). WebSphere Liberty runtime

understands the bundles and loads the code to run based on a set of deployment descriptor files. These deployment descriptor files are typically in XML file format and are in different folders depending on the archive type. For ease of use, the deployment descriptors needed for building the workload bundles are put in the `deployment/was-liberty-default/config-deployment-descriptors` folder. Make sure to edit the files in this folder instead of the files in the applications\maximo folders.

2. Because the requirements of each workload vary, the configuration of WebSphere Liberty runtime that is used for running the workload also varies. The WebSphere Liberty runtime can be configured with a set of files (server.xml, jvm.options, certificates etc.) that are used to initialize the environment. These files are, by default, included for each workload in the `deployment\was-liberty-default\config-servers` folder. Edit these files before building the application bundles to ensure that the final container image has the appropriate configuration ready to run the workload.

Build the application bundle needed for the Maximo workload:

1. Open a Command Prompt or Terminal and change to the `deployment/was-liberty-default` folder.
2. Run the appropriate command to build the workload bundles (Java EE war/ear files). For example, to build the maximo ui workload, run the `buildmaximoui-war` and `buildmaximo-xwar` commands.
3. The workload bundles are created in the `deployment/was-liberty-default/deployment/maximo-ui/maximo-ui-server/apps` folder. The commands also copy the workload deployment descriptor files for the workload from the `deployment/was-liberty-default/config-deployment-descriptors` subfolder into the `deployment/was-liberty-default/deployment/`. Note that the `maximo-ui` subfolder created under `deployment/was-liberty-default/deployment/` folder is treated as a *workload deployment folder*, which will be used to create the container image. For each workload, a separate folder is created to distinguish the content associated with the workload.

Build the container image needed for the Maximo workload:

1. After the application bundle needed for the Maximo workload is built, a container image can be created that can be deployed into an OpenShift runtime.
2. In order to create the container image, a Docker file needs to be created containing instructions about the contents of the image. Since the workload is going to be based on the WebSphere Liberty runtime, a base image of WebSphere Liberty is needed. This image can be obtained from Docker hub https://hub.docker.com/r/ibmcom/websphere-liberty following the documentation. The document available at the link https://github.com/WASdev/ci.docker#building-an-application-image describes guidelines for creating the Docker file content. This base image can be extended to create a new image that includes the configuration and application bundle

needed for the workload. Note that for OpenShift, the UBI (Universal Base Image) images are the recommended images for using WebSphere Liberty.

3. To create the container image on a workstation, Docker must be installed and running on the machine where the container images need to be built.
4. To create the container images on the Maximo administration workstation, create a folder from where the docker images are going to be built, for example, if using windows, `c:\maximo-containers`. In this folder, create a subfolder for each workload. For example, for the Maximo UI, create a folder called `maximo-ui`. Note that the `maximo-ui` workload includes two bundle files (maximoui.war and maximo-x.war file). In each of the workload subfolders, create another subfolder called `MaximoServerConfig`.
5. Copy the Maximo application bundles and associated files for the workload from the *workload deployment folder* into the `MaximoServerConfig` folder.

   For example:

   Copy all files and sub-directories from
   `<maximo install folder>\deployment\was-liberty-default\deployment\maximo-ui`
   to
   `c:\maximo-containers\maximo-ui\MaximoServerConfig`

6. Create a file named `Dockerfile` in the workload folder (for example, c:\maximo-containers\maximo-ui) that contains the following content:

   ```
   FROM ibmcom/websphere-liberty:19.0.0.4-kernel-ubi-min
   COPY --chown=1001:0 MaximoServerConfig/apps/maximo-x.war /config/apps/
   COPY --chown=1001:0 MaximoServerConfig/apps/maximoui.war /config/apps/
   COPY --chown=1001:0 MaximoServerConfig/server.xml /config/
   COPY --chown=1001:0 MaximoServerConfig/jvm.options /config/
   RUN configure.sh
   ```

   The content in this file is indicating to the Docker build tool to use the base image of WebSphere Liberty available in Docker Hub and then add to that image the workload files and then configure the WebSphere Liberty.

   Note that the content of this file must be changed based on the workload deployment. In the example, Maximo UI bundles are used. Note that the only thing that changes in this file for each Maximo workload is the apps folder content.

7. Run the Docker build command to build the image. For example, the following command uses the `Dockerfile` from the current directory, builds an image, and tags the image with the maximo-project/maximo:1.0. The first part of the tag is the namespace name (`maximo-project`) followed by the tag name (`maximoui`) and ends with a version number (`1.0`). Always use this tag naming convention to ensure that it is easy to identify the image of the workload. For example, for the maximoui workload, run the following command:

   ```
   docker build -t maximo-project/maximoui:1.0 .
   ```

After the command is successfully run, it will create an image based on the `Dockerfile,` and the image is put into the local Docker image registry. Verify that the image exists in the local registry using the `docker image ls` command, which should list all the images available including the new one.

Deploy container image of a Maximo workload

Prerequisites:
In order to deploy container images, the following items are required:

1. Maximo application bundles needed for the workload are created.
2. Container images needed for the workloads are created.
3. OpenShift runtime environment is available with minimum system requirements to run Maximo Asset Management.
4. OpenShift project (also known as the namespace) created to isolate the Maximo workload deployments.

Deploying the container image in OpenShift:

Before the container image is deployed into OpenShift to run in the Kubernetes environment, the container image must be available from an image registry that is accessible to the OpenShift environment, and then the image must be deployed into an OpenShift project.

Create an OpenShift project:

A project must be created in OpenShift before the image can be deployed and run. A project establishes a namespace for organizing the artifacts that go into the project. If a project is not already created, create the project using the following command in the OpenShift environment after logging into OpenShift using the `oc login` command. Note that the project can also be created from the OpenShift Console.

```
oc new-project maximo-project
```

After the command is successful, a new project called maximo-project will be created in OpenShift along with a Kubernetes namespace of the same name. If a project already exists, use the following command to switch to the project

```
oc project maximo-project
```

Push the container image to an image registry accessible to OpenShift:

OpenShift comes, by default, with a built-in image registry, and it is also possible to configure OpenShift with an external image registry. Regardless of the image registry used by OpenShift, you must have access permission to view and deploy images into the image registry. The container image created in the local Docker image registry can be pushed into the remote image registry that is used by OpenShift using `docker push` command. Before the image can be pushed into the remote image registry, the image must be tagged with the `hostname:port/namespace/tagname:version` format. The `namespace` used here is the project name that is used in OpenShift. An OpenShift project can be created anytime with this name, if one is not already created. Note that OpenShift by default comes with an internal image registry. It is also possible to use external image registry. The instructions here only refer to pushing an image to the internal image registry. For example, to tag the container image of the maximoui created earlier, run the following command:

```
docker tag maximo-project/maximoui:1.0 my-docker-registry-host/maximo-
project/maximoui:1.0
```

where `my-docker-registry-host` is the hostname of the Docker image registry. After image tagging is successful, the image can be pushed to the remote Docker registry. You must be logged into the registry before the images can be pushed to the registry. Follow the procedures (mostly using `docker login`) recommended to log into the image registry first and then push the image. For example, to push the image of `maximoui` tagged earlier, run the following command:

```
docker push my-docker-registry-host/maximo-project/maximo:1.0
```

After the image is successfully pushed to the image registry, it is now available to the OpenShift environment to run the image.

Deploy an image into an OpenShift project:

To deploy an image, the image must be available in an image registry accessible to OpenShift and a project must be created, and a user must have access to deploy the image. The image can also be deployed from the OpenShift console. To deploy the workload image using OpenShift, run the following command:

```
      oc new-app  maximo-project/maximoui:1.0
```

This command will create the needed deployment and service to run a single instance of the workload. Maximo Asset Management also supports the ability to pass Maximo properties using environment variables to the workload container process. In OpenShift, the underlying Kubernetes supports the ability to store secrets, and the secrets can be configured to be passed to the workload container process, which allows better control over sensitive information. To pass a Maximo property as an environment variable, the name must be all upper case and the . (dot) in the property name must to be changed to _ (underscore). For example, the mxe.db.url property name becomes the MXE_DB_URL environment variable. During startup, the Maximo system loads the environment variables and looks for the prefix MXE and converts them to the internal property names. Note that currently this loading only works for property names that are all in lower case, Mixed case property names cannot be used. For example, to deploy maximo using environment variables, the following command can be executed:

```
oc new-app -e MXE_DB_URL=jdbc:db2://mydbhostname:50000/MAXIMODB  \
                      -e MXE_DB_USERNAME=maximo      \
                      -e MXE_DB_PASSWORD=maximo      \
                      maximo-project/maximoui:1.0
```

After the workload container image is deployed, the deployment can be checked using the `oc` command or from the OpenShift Console by switching to the specific project under which the workload is deployed. Run the following command to view the deployments that are running:

```
oc get pods -n maximo-project
```

Where the maximo-project is the project under which the workload is deployed.

Expose a Maximo workload

When the Maximo workloads are started in an OpenShift environment, they are not by default accessible outside of the environment, and the internal ports must be explicitly exposed to get access from outside the network. After the container image is deployed, it will be associated internally with the Kubernetes service, and the service can be exposed by running the following command:

```
oc expose svc/maximoui --hostname=myhostname
```

Where `maximoui` is the name of the image used for deploying the workload, and `myhostname` is the name of the host that resolves to the OpenShift Cluster. After this command is successful, the OpenShift environment will have a new route created that provides access to the workload. The route information can further be edited using the `oc route edit` command or from the OpenShift Console to specify any other DNS host name or to enable security and TLS termination. Please refer to the OpenShift documentation to configure routes.

After the deployment service is exposed by using the route, the Maximo application can be accessed using the context path that is used for the web applications by using the route that was configured. By default, the route will be configured with the / as the route path when exposed using the `oc expose` command. Note that the context path is specified in the WebSphere configuration file (server.xml) for the specific Web Module as part of creating the workload application bundle. In the case of the Maximo UI application, it is by default /maximo for accessing the application-based Maximo UI and /maximo-x for accessing the Work Center UI. Note that not all Maximo workloads need to be exposed to users, for example, the Maximo cron workload is just for running background jobs and does not require any user access.

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM® product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing IBM Corporation North Castle Drive, MD-NC119 Armonk, NY 10504-1785 US

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing Legal and Intellectual Property Law IBM Japan Ltd. 19-21, Nihonbashi-Hakozakicho, Chuo-ku Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing IBM Corporation North Castle Drive, MD-NC119 Armonk, NY 10504-1785 US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Red Hat®, JBoss®, OpenShift®, Fedora®, Hibernate®, Ansible®, CloudForms®, RHCA®, RHCE®, RHCSA®, Ceph®, and Gluster® are trademarks or registered trademarks of Red Hat, Inc. or its subsidiaries in the United States and other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM Online Privacy Statement

IBM Software products, including software as service solutions, (“Software Offerings”) may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session and persistent cookies that collect each user's name, user name, password, or other personally identifiable information for purposes of session management, authentication, single sign-on configuration or other usage tracking or functional purposes. These cookies can be disabled, but disabling them will also likely eliminate the functionality they enable.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at http://www.ibm.com/privacy and IBM's Online Privacy Statement at http://www.ibm.com/privacy/details in the section entitled "Cookies, Web Beacons and Other Technologies" and the "IBM Software Products and Software-as-a-Service Privacy Statement" at http://www.ibm.com/software/info/product-privacy.