TIBCO ActiveMatrix® Service Grid - Container Edition

Quick Start

Version 1.0.0
December 2020

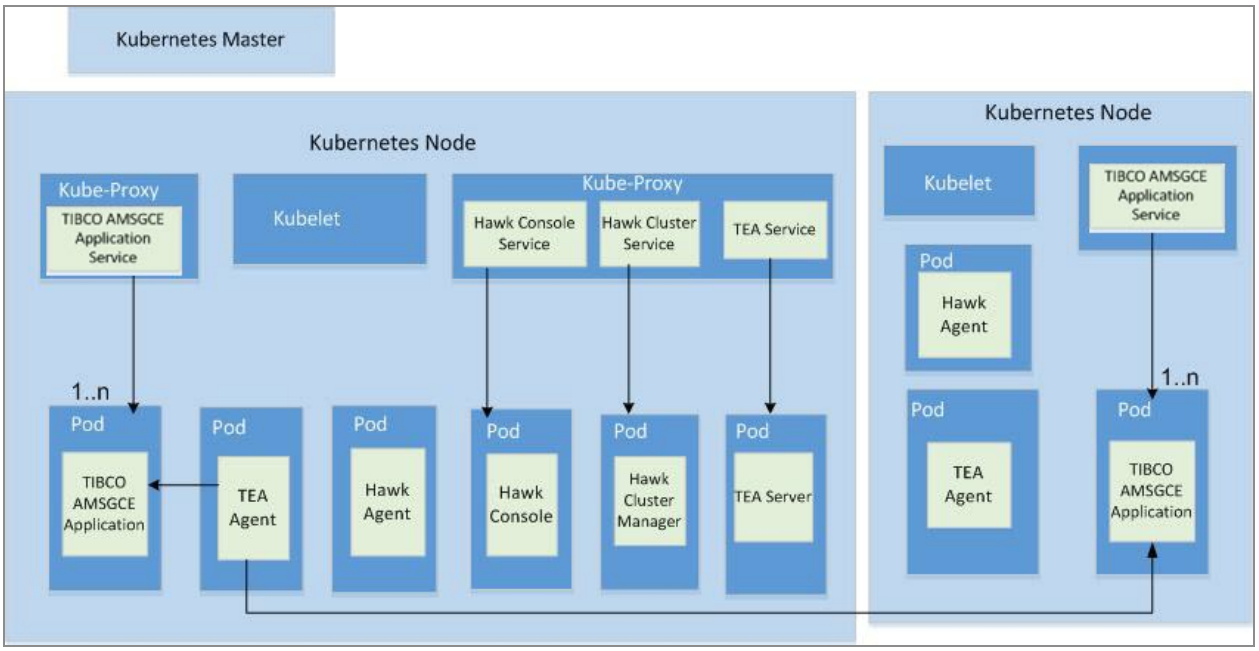
Document Updated: May 2021

Contents

Introduction

TIBCO ActiveMatrix® Service Grid - Container Edition enables you to run ActiveMatrix Service Grid applications on Docker and Kubernetes platforms. Managing a large number of distributed business applications: starting from deployment, integration, scaling, flexibility for future changes, and monitoring; pose a challenge to information technology departments. TIBCO ActiveMatrix products help solve many of these challenges.

The following figure illustrates the ActiveMatrix Service Grid - Container Edition architecture:



ActiveMatrix Service Grid - Container Edition is a container-based (Docker) system that is managed by an Kubernetes. Scaling, high availability, and fault tolerance are achieved by creating additional container instances. The only external infrastructure needed to run an ActiveMatrix Service Grid - Container Edition is a Kubernetes cluster.

In Kubernetes, each ActiveMatrix Service Grid - Container Edition application runs as a Kubernetes pod. Pods communicate with each other by using their IP addresses. Each pod contains a single ActiveMatrix Service Grid - Container Edition application container. Additional pods can be added to a node to scale the system. The system can also be scaled by creating a replica set across multiple Kubernetes nodes. You can expose the ActiveMatrix Service Grid - Container Edition application by using a Kubernetes service. In the container environment, an application is not logically split into multiple components

and run on multiple Kubernetes nodes; inter-component communication by using messaging bus or in-memory is not required. Inter-component communication is done by using Kubernetes services.

To monitor applications, TIBCO Enterprise Administrator (TEA) agent for ActiveMatrix Service Grid - Container Edition and TEA server are also deployed in the same cluster. In the Kubernetes cluster, the Application Monitor TEA agent automatically discovers all containers running ActiveMatrix Service Grid - Container Edition applications and displays their status on the Application Monitor dashboard. You can also monitor applications by using TIBCO Hawk® Container Edition.

Application Development

TIBCO ActiveMatrix Business Studio provides a common modeling, implementation, and deployment environment for different types of applications. TIBCO Business Studio provides an Eclipse-based design environment. Applications developed using the TIBCO ActiveMatrix design environment conform to the Service Component Architecture (SCA), which is a model for developing applications based on the service-oriented architecture (SOA).

The ActiveMatrix Service Grid - Container Edition installation package does not include TIBCO ActiveMatrix Business Studio with it. You can use the TIBCO ActiveMatrix Business Studio shipped with TIBCO ActiveMatrix Service Grid 3.4.0, 3.3.1, or 3.3.0 to develop applications. For more information about TIBCO Business Studio, see *TIBCO ActiveMatrix Service Grid Composite Development*.

Containerizing ActiveMatrix Service Grid Applications

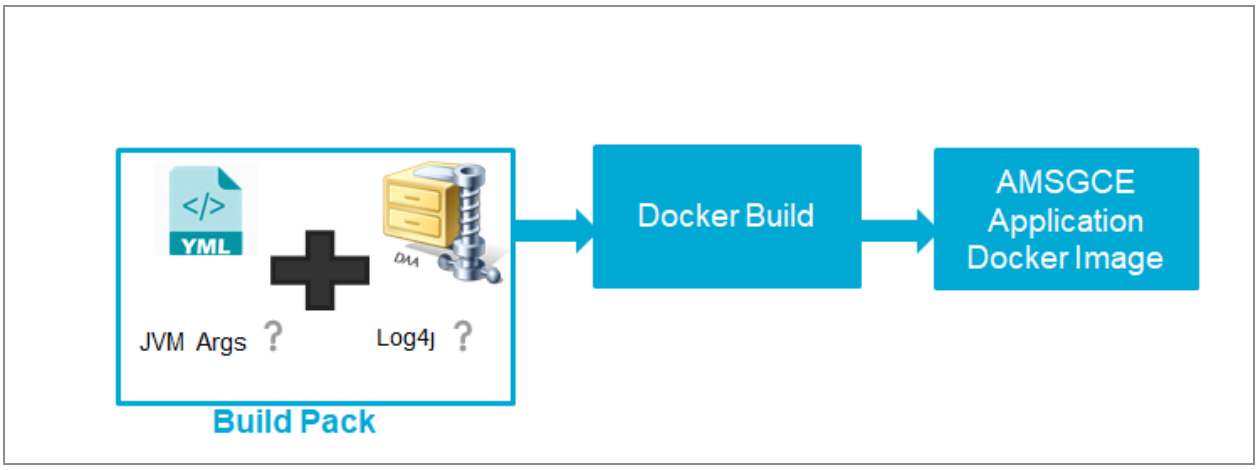
A TIBCO ActiveMatrix application comprises a common TIBCO ActiveMatrix runtime and an application-specific code running in the TIBCO ActiveMatrix runtime. Thus, to containerize a TIBCO ActiveMatrix application, TIBCO ActiveMatrix runtime and an application DAA are included in the Docker image. To containerize the TIBCO ActiveMatrix application, you need an application configuration or a build pack.

Application configuration contains the following artifacts:

Artifact	Description
Application DAA	A distributed application archive (DAA) is a package that contains TIBCO ActiveMatrix application and libraries. You can use TIBCO ActiveMatrix Business Studio shipped with TIBCO ActiveMatrix Service Grid 3.x to design and develop DAA and package application as DAA. The ActiveMatrix Service Grid - Container Edition installation package does not include TIBCO ActiveMatrix Business Studio.
Application Configuration YAML File	In TIBCO ActiveMatrix Service Grid 3.x, you can configure the application before deployment from the Administrator UI. All configurations are stored in the Administrator database. • ActiveMatrix Service Grid - Container Edition does not have an Administrator database. All configurations are stored in the YAML configuration file. • When building the Docker image of an application, the YAML file is copied to the Docker image. The YAML file has higher precedence as compared to DAA (that is, the YAML file overrides the configuration given in the DAA). • The Application Configuration YAML file with DAA and optionally, Log4j and JVM configuration are required to create a Docker image of an application.
Node JVM Arguments File (Optional)	This file contains JVM arguments that are used to create the ActiveMatrix Service Grid - Container Edition node in the container. • In ActiveMatrix Service Grid - Container Edition, each ActiveMatrix Service Grid - Container Edition node is a JVM environment. • The default file name is `node_jvm_parameters.config.yaml`. This file is used by image creation scripts. JVM arguments in this file are set as ActiveMatrix Service Grid - Container Edition node JVM arguments. If this file does not exist, then the default value is used.
Node Log4j Configuration File	This file contains Log4j configuration which is applied to the ActiveMatrix Service Grid - Container Edition node in the container. • In ActiveMatrix Service Grid - Container Edition, configuring logs only

Artifact	Description
(Optional)	on node level is supported. Logs cannot be configured at application level. • The default file name is `node-log4j.xml`. This file is used by image creation scripts and it is applied as TIBCO ActiveMatrix Service Grid - Container Edition node Log4j configuration. If this file is not provided when creating an application image, then the default file is used.

When the artifacts are generated, you can build the application Docker image:



Generating Artifacts for Image Building

You can generate the artifacts or configuration required to build an application Docker image in one of the following ways:

Scenario	Recommended Method
For Applications running in TIBCO ActiveMatrix Service Grid 3.x Setup	Using Application Extractor
Creating New Configuration	• Using DAA2Config Command-line tool • Using Application Configurator
Updating Existing Configuration	• Using Text Editor

Scenario	Recommended Method
	• Validating configuration by using the `checkConfig` command • Uploading existing configuration to Application Configurator

ActiveMatrix Service Grid - Container Edition Components

ActiveMatrix Service Grid - Container Edition consists of the Application Configurator (a web application for configurations), a DAA2Config command-line Tool, an Application Extractor to extract the application configuration, Application Monitor to monitor applications, and a Hawk microagent for monitoring.

Component	Description
Application Configurator	The Application Configurator is a web application to perform configuration tasks: • Upload and Configure TIBCO ActiveMatrix application • Generate the YAML configuration file for an application • Generate other optional configurations such as Log4j configuration and JVM configuration. • Configure a range of entities such as substitution variables, bindings, resource templates, and JVM arguments To perform any of these tasks, you must first upload the Distributed Application Archive (DAA) in the Application Configurator. For more information about the Application Configurator, see *TIBCO ActiveMatrix® Service Grid – Container Edition Administration*.
DAA2Config Command-Line Tool	The DAA2Config is a command-line tool to generate application configuration from DAA. This is a CLI alternative to the Application Configurator (which is GUI-based). All functionalities are similar in both.

Component	Description
	Script for this CLI tool: `amsgce-runtime-<version>/daautil/bin/DAA2Config.sh` For more information, see *TIBCO ActiveMatrix® Service Grid – Container Edition Administration*.
Application Monitor	You can monitor ActiveMatrix Service Grid - Container Edition applications deployed on Docker and Kubernetes platforms by using Application Monitor. ActiveMatrix Service Grid - Container Edition provides an Application Monitor agent, which allows you to monitor ActiveMatrix Service Grid - Container Edition applications. For more information, see *ActiveMatrix Service Grid - Container Edition Monitoring*.
Application Extractor	You can use the Application Extractor to extract application DAA and the application deployment configuration from ActiveMatrix Service Grid 3.x setup and use the extracted files to containerize the application. For more information, see *TIBCO ActiveMatrix® Service Grid - Container Edition Quick Start*.
Hawk Monitoring	ActiveMatrix Service Grid - Container Edition provides a Hawk microagent to monitor the applications by using Hawk® Container Edition. Each ActiveMatrix Service Grid - Container Edition node contains one Hawk microagent. Each ActiveMatrix Service Grid - Container Edition node contains one user application and system applications such as platform app, monitoring app, and mediation app. The Hawk agent runs on each Kubernetes node. For more information about Hawk concepts and procedures, see the Hawk® Container Edition documentation. For more information about Hawk microagent methods, see *ActiveMatrix Service Grid - Container Edition Monitoring*.

Note: ActiveMatrix Service Grid - Container Edition version is 1.0.0. However, the TIBCO ActiveMatrix Service Grid - Container Edition runtime node version is 5.0.0. The node version is displayed as 5.0.0 in logs and in the Application Monitor UI.

Deploying TIBCO ActiveMatrix Applications in Kubernetes

Once you have built an application Docker image, you can deploy TIBCO ActiveMatrix application by using Kubernetes deployment configuration file. A Kubernetes deployment file (a .yaml configuration file) describes how to deploy the Docker image of the TIBCO ActiveMatrix application to the Kubernetes cluster.

AMXCE Node

In ActiveMatrix Service Grid - Container Edition, one AMXCE node runs one TIBCO ActiveMatrix application. In Kubernetes, each ActiveMatrix Service Grid - Container Edition node runs as a Kubernetes pod. Kubernetes node is different from the AMXCE node. One Kubernetes node can run one or more AMXCE nodes or pods. Unlike TIBCO ActiveMatrix Service Grid, there is no concept of host in ActiveMatrix Service Grid - Container Edition. AMXCE nodes exist in an environment.

ActiveMatrix Service Grid - Container Edition and Kubernetes

ActiveMatrix Service Grid - Container Edition	Kubernetes
Application	Deployment: In Kubernetes, each TIBCO ActiveMatrix application can be deployed as Kubernetes deployment. Deployment allows you to describe the application life cycle such as image to use for application, the number of pods, and replicas to be used.
Application Instances	Pods: You can scale application

ActiveMatrix Service Grid - Container Edition	Kubernetes
	instances by scaling pods up and down.
Environment	Namespace: Namespaces are virtual clusters in Kubernetes.
Substitution Variable	ConfigMap: A ConfigMap stores configuration data for containers. Config map separates configurations from your pods and components. Secrets: Secrets are objects that stores sensitive information about your clusters such as user name and password in encrypted format.

Monitoring Applications

You can monitor applications running in Kubernetes Cluster from the Application Monitor. The Application Monitor displays data related to Kubernetes and the ActiveMatrix Service Grid - Container Edition node on a single dashboard. The dashboard displays data about Kubernetes resources such as running pods, deployment, namespace, and desired state. You can also monitor runtime entities such as environments, applications, nodes, and bindings in the enterprise.

The Application Monitor is a web UI that runs in TIBCO® Enterprise Administrator (TEA). ActiveMatrix Service Grid - Container Edition provides an agent for TIBCO Enterprise Administrator which allows you to monitor ActiveMatrix Service Grid - Container Edition applications.

For more information about monitoring applications, see *TIBCO ActiveMatrix® Service Grid - Container Edition Monitoring*.

Installation

ActiveMatrix Service Grid - Container Edition is easy to install. There is no Universal Installer. The `amsgce-runtime-<version>` folder contains all the Dockerfiles and scripts to build ActiveMatrix Service Grid - Container Edition components Docker images and to deploy the components on cloud platform.

System Requirements

- A machine with Docker installation (see the Readme for the Docker version to use) and initial setup based on your operating system, to generate Docker images. For complete information about Docker installation, see Docker Documentation.
- The machine on which you are building the Docker image must have Java 11 installed. `<JAVA_HOME>` must be added in the *Path* environment variable. Java 11 is the default and supported Java Runtime Environment (JRE) for ActiveMatrix Service Grid - Container Edition.

Procedure

1. From the TIBCO eDelivery website, download the `amsgce-runtime-<version>.zip`.
2. Copy `amsgce-runtime-<version>.zip` to your local machine and extract it.

What to do next

There are no other installation steps required to install ActiveMatrix Service Grid - Container Edition. You can use the scripts and Dockerfiles in `amsgce-runtime-<version>.zip` to build ActiveMatrix Service Grid - Container Edition components Docker images and deploy the components on cloud platform. For more information, see *TIBCO ActiveMatrix® Service Grid - Container Edition Cloud Deployment*.

Tutorial

The tutorial introduces you to the ActiveMatrix Service Grid - Container Edition complete cycle of application creation, development, deployment, and monitoring. It assumes that you are comfortable with the concepts and procedures of Docker and Kubernetes.

Setup Requirements

Make sure the following software are installed on your system:

- From the TIBCO eDelivery website, download the `amsgce-runtime-<version>.zip`. Copy `amsgce-runtime-<version>.zip` to your local machine and extract it. The `amsgce-runtime-<version>` folder contains all the Dockerfiles and scripts to build ActiveMatrix Service Grid - Container Edition components Docker images and deploy the components on cloud platform.
- Docker (see the Readme for the Docker version to use)
- Java Runtime Environment 11 or later
- Kubernetes command-line tool kubectl
- Minikube to run Kubernetes locally

This tutorial is recommended as an introduction to ActiveMatrix Service Grid - Container Edition:

1. Build and Run Application Configurator Docker Image
2. Generate Bookstore Application Configuration
3. Build and Run Bookstore Application Image
4. Build Application Monitor Image
5. Deploy Application Monitor to Kubernetes
6. Monitor Bookstore Application in Application Monitor
7. Testing the REST Service in Application Monitor

Build and Run Application Configurator Docker Image

To containerize the Bookstore application, you need to generate the application configuration YAML file. In Application Configurator, you can upload Bookstore application DAA and generate the application configuration YAML file. In this tutorial, you will build Application Configurator Docker image and run it in Docker.

Procedure

1. Navigate to `amsgce-runtime-<version>/applicationConfigurator/build`.
2. Build the Docker image of TIBCO ActiveMatrix Application Configurator by using the following script:

   ```
   ./build_amxce.sh
   ```

3. Run the Application Configurator Docker image:

   ```
   docker run --rm -p 8087:8087 -p 9998:9998 --name amxceac amxce_ac:1.0
   ```

   Here, port 8087 is the application port and port 9998 is used for monitoring.
4. You can access the Application Configurator by opening the following URL in a browser: `http://<hostname>:<hostport>/appconfig` or `http://<IP>:<hostport>/appconfig`.
   For example: `http://<hostname>:8087/appconfig/`

What to do next

By using the Application Configurator, you can configure an application and generate an application configuration YAML file. The next step is to generate the application configuration YAML file.

Generate Bookstore Application Configuration

The Bookstore sample application is provided at `amsgce-runtime-<version>\samples\bookstore\com.tibco.restbt.sample.bookstore`

Before creating the bookstore application Docker image, you must generate the application configuration from application DAA. Application configuration contains the YAML file and, Log4j configuration (optional), and JVM parameters files(optional).

You can generate application configuration in one of the following ways:

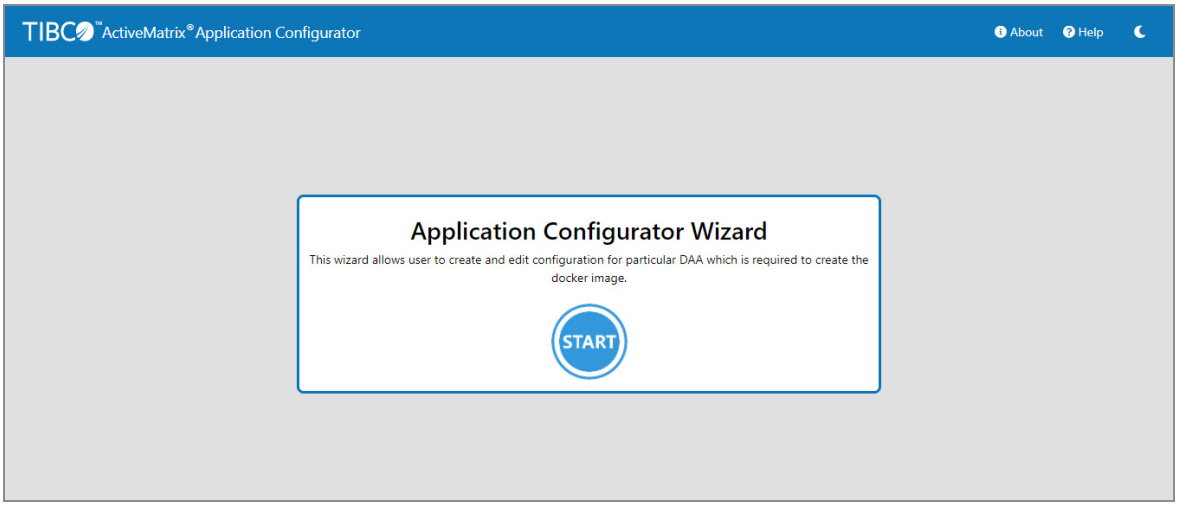
- Using Application Configurator UI
- Using DAA2Config command-line tool
- If the Bookstore application is already running in ActiveMatrix Service Grid 3.x setup, you can extract the application configuration by using the Application Extractor tool. See Application Extractor Tool.

Generating Application Configuration from the GUI (Application Configurator)

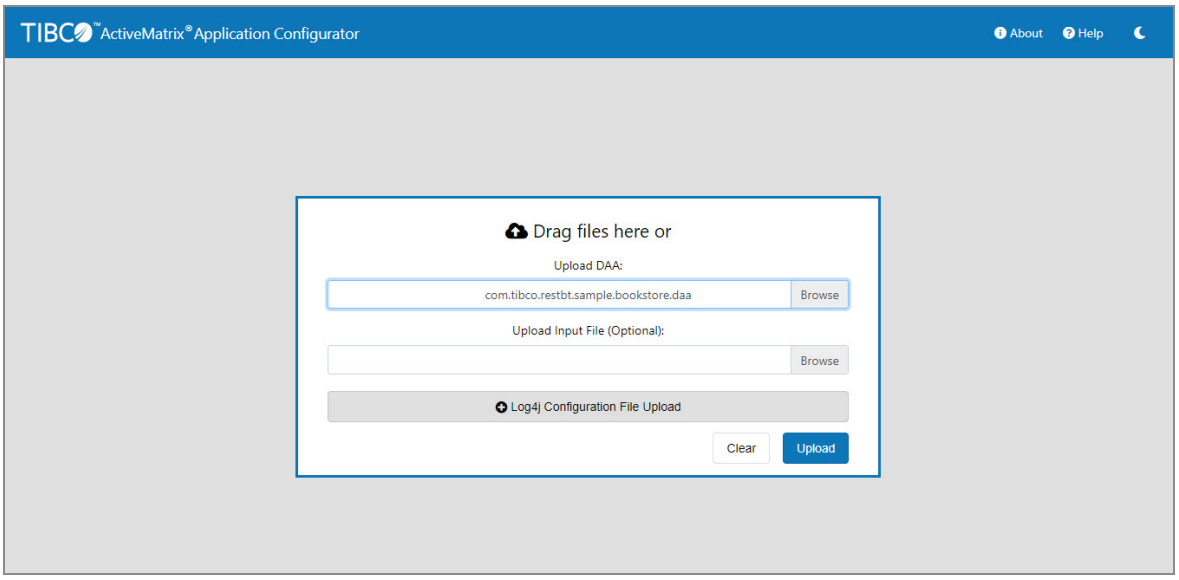
In this tutorial, you will upload the Bookstore application DAA to Application Configurator and complete the required configuration.

Procedure

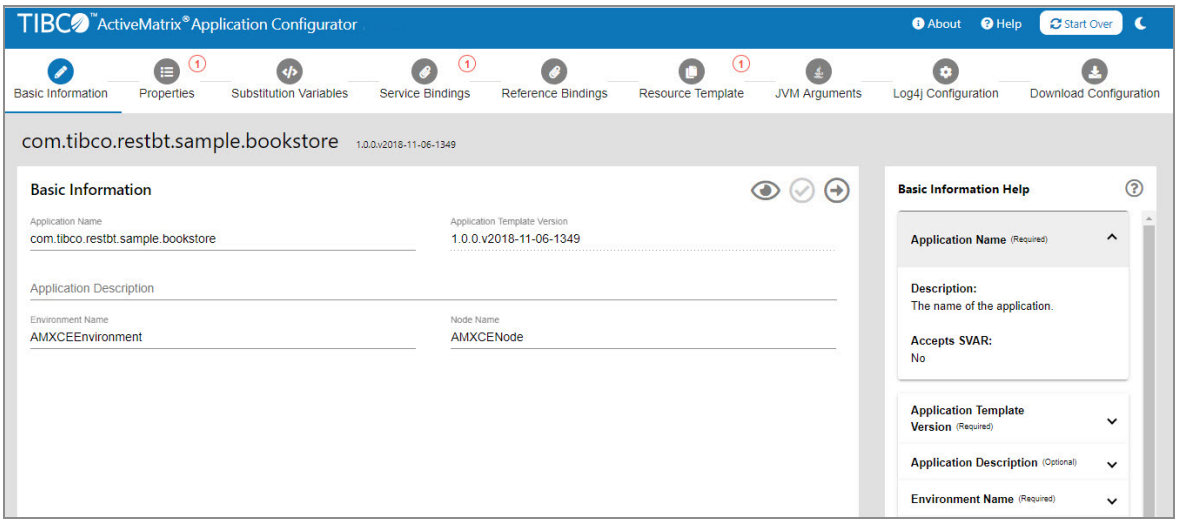
1. Access the Application Configurator UI by navigating to `http://<hostname>:8087/appconfig/`.



2. Upload the bookstore application DAA in Application Configurator.



3. Complete the required configuration. For information, see *TIBCO ActiveMatrix® Service Grid - Container Edition Administration.*



4. Download the application configuration `.zip` file to your machine.

What to do next

The downloaded `.zip` file contains the application DAA, application configuration YAML file, and optionally, Log4j configuration file and JVM arguments file. Specify the path of this `.zip` file in the next step to build the Bookstore application image.

Generating Application Configuration from the CLI (DAA2Config)

Instead of using the Application Configurator, you can use DAA2Config command-line tool to generate the application configuration YAML file with default values. In this tutorial, you will use `DAA2Config` command to generate the application configuration YAML file.

Procedure

1. Navigate to the `amsgce-runtime-<version>\daautil\bin` folder.
2. Run the following command to generate the YAML file from DAA:

   You can check the parameters to pass to the command by running the `./DAA2Config.sh help` command.

   For Linux Platform:

   ```
   ./DAA2Config.sh -daaFile /home/amsgce-runtime-
   1.0.0/samples/bookstore/com.tibco.restbt.sample.bookstore/Deployment
   Artifacts/com.tibco.restbt.sample.bookstore.daa
   ```

   For Windows Platform:

   ```
   DAA2Config.bat -daaFile C:\amsgce-runtime-
   1.0.0\samples\bookstore\com.tibco.restbt.sample.bookstore\Deployment
   Artifacts\com.tibco.restbt.sample.bookstore.daa
   ```

   The YAML file, node JVM parameters configuration file, and Log4j configuration file with default values are generated in the output location provided. If no output location is provided, all files are generated in the same folder where the DAA file is located.
3. To update the YAML file, you can use the text editor. After updating the YAML file, validate the configuration by using the `checkConfig` command.

   For Linux Platform:

   ```
   ./checkConfig.sh -daaFolder "/home/apps/bookstore" -debug true
   ```

   For Windows Platform:

   ```
   checkConfig.bat -daaFolder "C:\home\apps\bookstore" -debug true
   ```

What to do next

In the next step, specify the path of the folder containing the application configuration to build the Bookstore application image.

Build and Run Bookstore Application Image

After you have generated the Bookstore application configuration YAML file from application DAA, you can build the Bookstore application Docker image. In this tutorial, you will build the Bookstore application Docker image and run it in the Docker or Kubernetes.

Procedure

1. Navigate to `amsgce-runtime-<version>/runtime/build`.
2. To build the bookstore application Docker image, run the following command. Specify the location of the folder or the `.zip` file that contains the YAML file, DAA, and optionally, the Log4j and JVM configuration files by using `--app_location` argument. To check the options you can specify when running the command, run the following command:

   ```
   ./build_amxce.sh --help
   ```

   ```
   ./build_amxce.sh --image_tag bookstore:1.0 --app_location amsgce-
   runtime-
   <version>
   /samples/bookstore/deployment/com.tibco.restbt.sample.bookstore.zip
   ```

3. Execute the `docker run` command on the machine where you have created the application Docker image.

   ```
   docker run -d -p 7777:7777 --name bookstore bookstore:1.0
   ```

   The external port must be a port number that is used at the time of resource template creation. If you want to map a different port other than the port used for the YAML file, you can specify it when running the docker run command. If there are two HTTP connectors in the DAA, then while creating the container, you need to specify both the ports.

4. Check whether the container is created or not by running the following command:

```
docker ps
```

5. Access the application from `http://<hostname>:7777/bookstore/books`. Here, port must be the port you mapped when running the container.

Running Bookstore Application in Kubernetes

After you have built the Bookstore application Docker image, you can deploy the application in Kubernetes.

1. Create a Kubernetes deployment YAML file that is required to deploy an application. Sample YAML file is at `amsgce-runtime-<version>\samples\bookstore\deployment\bookstore-deployment.yaml`. Ensure that the Kubernetes deployment file contains correct application image name and port.
2. Run the `apply` command of `kubectl` utility to deploy an application to Kubernetes.

```
kubectl apply -f bookstore-deployment.yaml
```

3. Access this application at `http://<hostname>:31090/bookstore/books`.

What to do next

Bookstore application is now running in the Kubernetes. You can monitor the status of the running application and troubleshoot by using the Application Monitor. In the next step you will build and run the Application Monitor.

Build Application Monitor Image

The Application Monitor is a web UI that runs in TIBCO® Enterprise Administrator (TEA). ActiveMatrix Service Grid - Container Edition provides an agent for TIBCO Enterprise Administrator which allows you to monitor ActiveMatrix Service Grid - Container Edition applications. In this tutorial, you will build the Docker image of ActiveMatrix Service Grid - Container Edition TEA agent and TIBCO Enterprise Administrator Server.

Procedure

1. Navigate to `amsgce-runtime-<version>/teaagent/build.`
2. Build the ActiveMatrix Service Grid - Container Edition TEA agent Docker image by using the following script:

   ```
   ./build_amxceteaagent.sh
   ```

   You can list the most recently created images by using the following command:

   ```
   docker images
   ```

3. Create Docker image of TIBCO Enterprise Administrator Server. For more information, see `readme.md` at *TEA_HOME*`/docker` in TIBCO Enterprise Administrator Installation.

What to do next

After you have created the Docker images of TEA agent and TEA server, you can deploy the Application Monitor to Kubernetes in the next step.

Deploy Application Monitor to Kubernetes

After creating the Docker image for ActiveMatrix Service Grid - Container Edition TEA agent and TIBCO Enterprise Administrator server, you can deploy the Application Monitor to Kubernetes. Application Monitor needs the cluster view permission, so you must first create a service account with a cluster view role.

Procedure

1. Navigate to `amsgce-runtime-<version>/samples/kubernetes/Application_Monitor`.
2. To create a service account with a cluster view role, run the following command:

   ```
   kubectl apply -f amxce_authorization.yaml
   ```

3. In the `tea-server.yaml` file, update *<tea-server-docker-image>* with a TEA Server Docker image. Run the `apply` command of `kubectl` utility to deploy the TEA Server to Kubernetes.

   ```
   kubectl apply -f tea-server.yaml
   ```

4. In the `tea-agent.yaml` file, update <*amxceteaagent-image-name*> with TEA agent Docker image. Run the `apply` command of `kubectl` utility to deploy the TEA agent to Kubernetes.

```
kubectl apply -f tea-agent.yaml
```

5. Verify that the deployment has succeeded and that the pod is running. For example:

```
~$ kubectl get pod -n monitoring
NAME                                      READY   STATUS    RESTARTS
AGE
tea-agent-deployment-6c7f784fb8-pjr6g     2/2     Running   0
11s
tea-server-deployment-7699c48774-s46vf    1/1     Running   0
2m57s
```

6. To access the Application Monitor dashboard, navigate to: `http://<hostname>:31877/tea`.

What to do next

Now the Application Monitor is running in the Kubernetes platform. In the next step, you can monitor the status of Bookstore application and components of application in the Application Monitor UI.

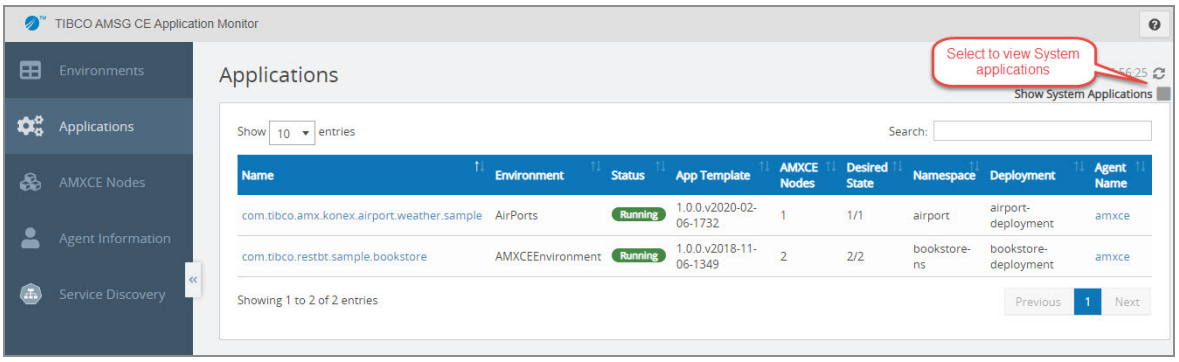
Monitor Bookstore Application in Application Monitor

The Application Monitor displays data related to Kubernetes and the ActiveMatrix Service Grid - Container Edition node on a single dashboard. You can view information about Bookstore application resources such as running pods, deployment, namespace, and desired state. You can also monitor runtime entities such as environments, applications, nodes, and bindings in the Bookstore application.

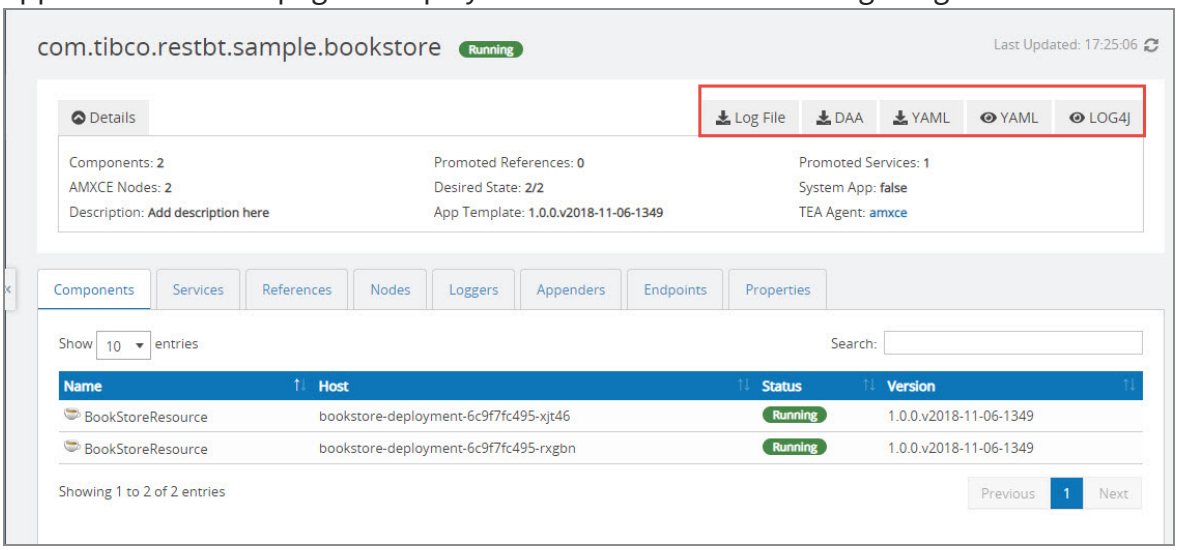
Procedure

1. Open a browser and navigate to `http://<hostname>:31877/tea`, where 31877 is the default port number.
2. Enter your login credentials and click **Sign In**. The default user name and the password are admin. The default timeout for a session is 30 minutes.

3. Click the ActiveMatrix Service Grid - Container Edition product card. The Application Monitor start page is displayed.
4. To open the Applications page and view the application status, click **Applications** in the sidebar.



5. To view application details, click `com.tibco.restbt.sample.bookstore`. The applications details page is displayed as shown in the following image:



What to do next

You can view information about Bookstore application and perform the different tasks from Application Monitor UI. For more information, see *TIBCO ActiveMatrix® Service Grid - Container Edition Monitoring.* In the next step, you will test the REST service of Bookstore application by using Swagger UI in the Application Monitor.

Testing the REST Service in Application Monitor

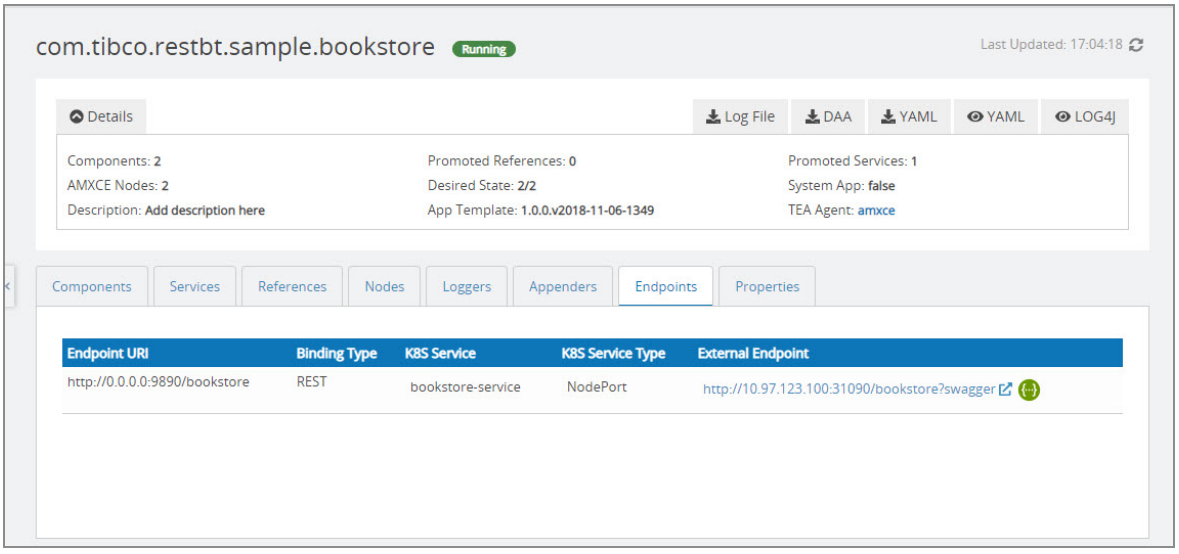
In TIBCO ActiveMatrix Service Grid, you can generate the swagger from (REST Binding Type) endpoint exposed in an application. Similarly, in ActiveMatrix Service Grid - Container Edition, you can view the Swagger response of the exposed REST endpoint. By using this feature, you can validate the response generation of the REST request by using Swagger UI.

Before you begin

- You must have Swagger UI hosted externally.
- You must provide the link to the Swagger UI hosted externally by using the environment variable `amxce_swaggerui` in the file `amsgce-runtime-<version>\samples\kubernetes\Application_Monitor\tea-agent.yaml`.
- To disable default Swagger UI validator, set `VALIDATOR_URL` to none in the `tea-agent.yaml` file.

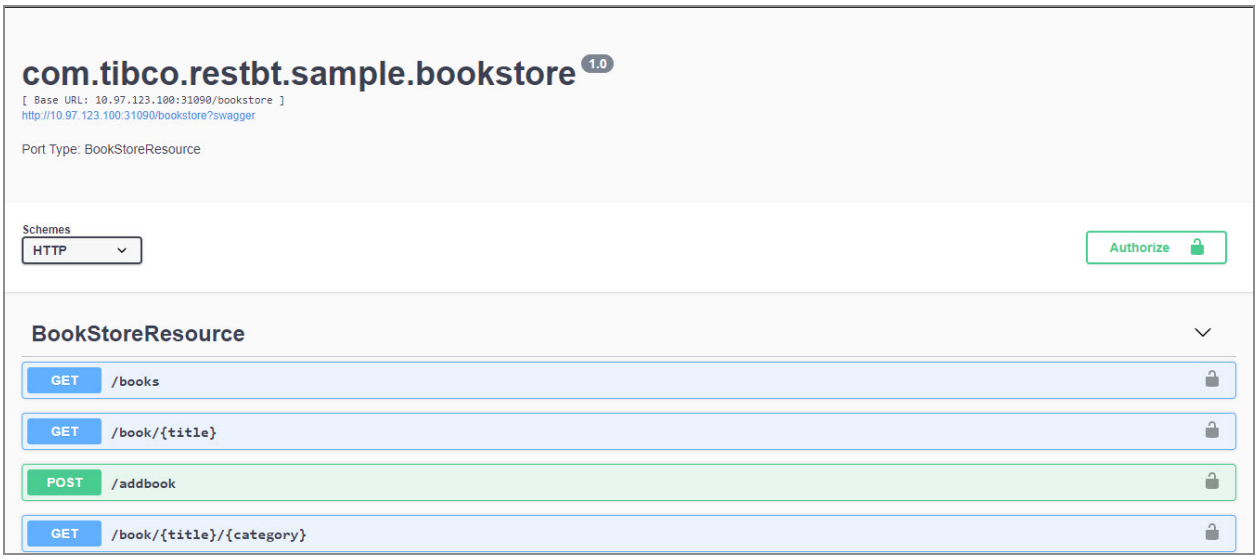
Procedure

1. In the Application Monitor UI, click **Applications**. A list of all the applications is displayed.
2. Select the com.tibco.restbt.sample.bookstore application.
3. Click the **Endpoints** tab.



4. Click the Swagger UI icon  next to REST Service Binding.

Swagger UI is generated for the selected REST Binding.



You can also access Swagger UI from the **Service Discovery** page.

Migrating ActiveMatrix Service Grid 3.x Applications to ActiveMatrix Service Grid - Container Edition

ActiveMatrix Service Grid - Container Edition provides a migration path for ActiveMatrix Service Grid 3.x applications. This document explains how to migrate from ActiveMatrix Service Grid 3.x to ActiveMatrix Service Grid - Container Edition and lists differences between the two. You can use the Application Extractor tool to extract configuration of applications running in the TIBCO ActiveMatrix Service Grid 3.x setup and use this configuration to containerize applications.

Use the following process flow to migrate applications:

1. Download the TIBCO ActiveMatrix® Service Grid - Container Edition software package from the TIBCO eDelivery. Extract the `amsgce-runtime-<version>.zip` file to a temporary directory on the machine.
2. Use the Application Extractor tool to extract the application configuration of single or multiple applications. For more information, see Application Extractor Tool.
3. Build the Docker images of applications and run them as containers. For more information, see "Containerizing ActiveMatrix Service Grid Applications" topic in *TIBCO ActiveMatrix® Service Grid - Container Edition Cloud Deployment*.

Application Extractor Tool

This tool is used to extract the TIBCO ActiveMatrix 3.x application configuration. You can use the extracted configuration to containerize an application.

To containerize any TIBCO ActiveMatrix 3.x application, you must first extract the configuration and deployment details of that application, build the image, and then run it as a container. Using the Application Extractor tool, you can extract Distributed Application Archive (DAA) and deployment configuration for any existing ActiveMatrix application and generate an output `.zip` file.

Note: Application Extractor tool supports TIBCO ActiveMatrix Service Grid 3.3.0 hotfix 23 and later versions.

Deploying the Application Extractor Tool

The extractor tool is a TIBCO ActiveMatrix 3.x based SOA application. To use this tool, you must first deploy it on an ActiveMatrix enterprise as any other ActiveMatrix application. For detailed steps, see the "Deploying Application Extractor" section in the *TIBCO ActiveMatrix® Service Grid - Container Edition Cloud Deployment*. You can also refer `Readme.txt` in the `amsgce-runtime-<version>\applicationExtractor` folder for the installation steps.

After this application is deployed to SystemNode under SystemEnvironment, it gives a Web UI that can be used to extract one or more applications for a given logical ActiveMatrix environment.

Accessing the Application Extractor Tool

Note: You must log in to ActiveMatrix Administrator as a super user.

To access the Application Extractor tool, navigate to: `http://<admin-machine>:<port-number-8120>/appextractor`.

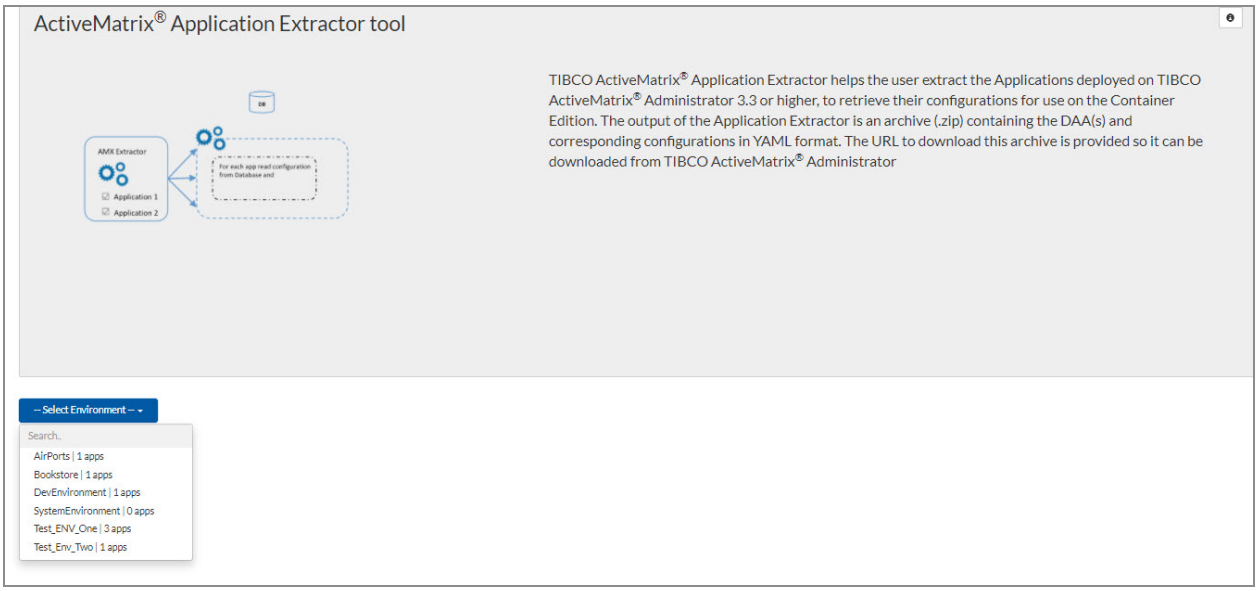
Here:

- <*admin-machine*>: URL of the machine on which the Administrator instance is running.
- <*port-number*>: Port number to access ActiveMatrix Administrator (typically, 8120).

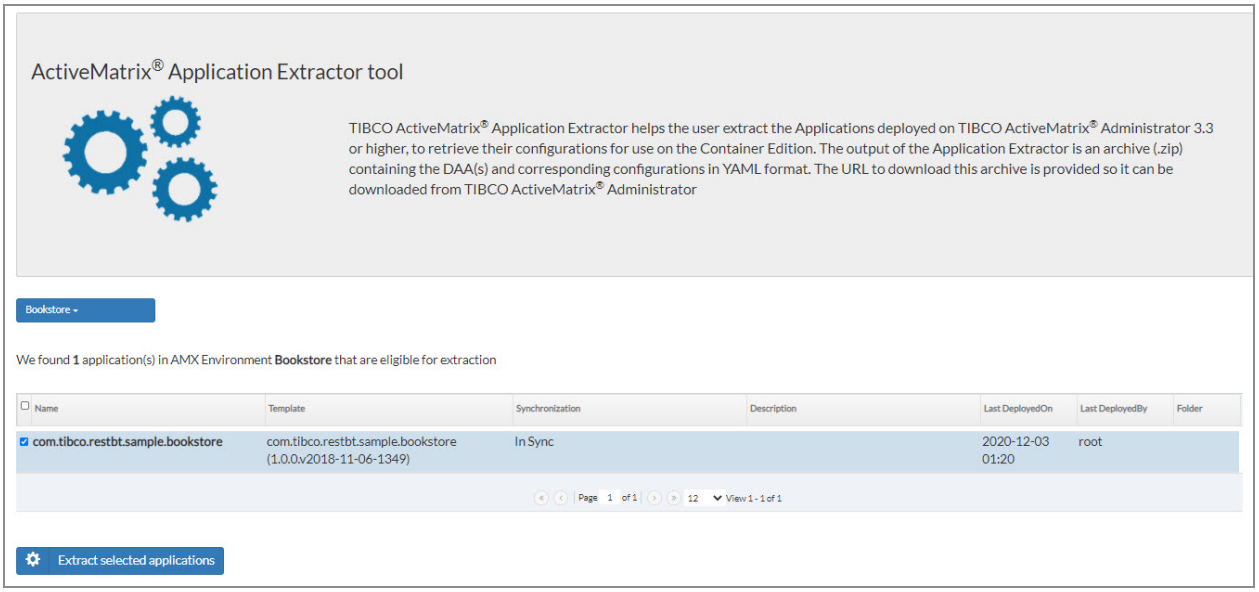
Extracting Application Configuration Using the Application Extractor Tool

1. Access the URL: `http://<admin-machine>:<port-number-8120>/appextractor` in a browser. Application Extractor UI is available without authentication but to download the `.zip` file, you must login to Administrator.
2. Click **Select Environment**.

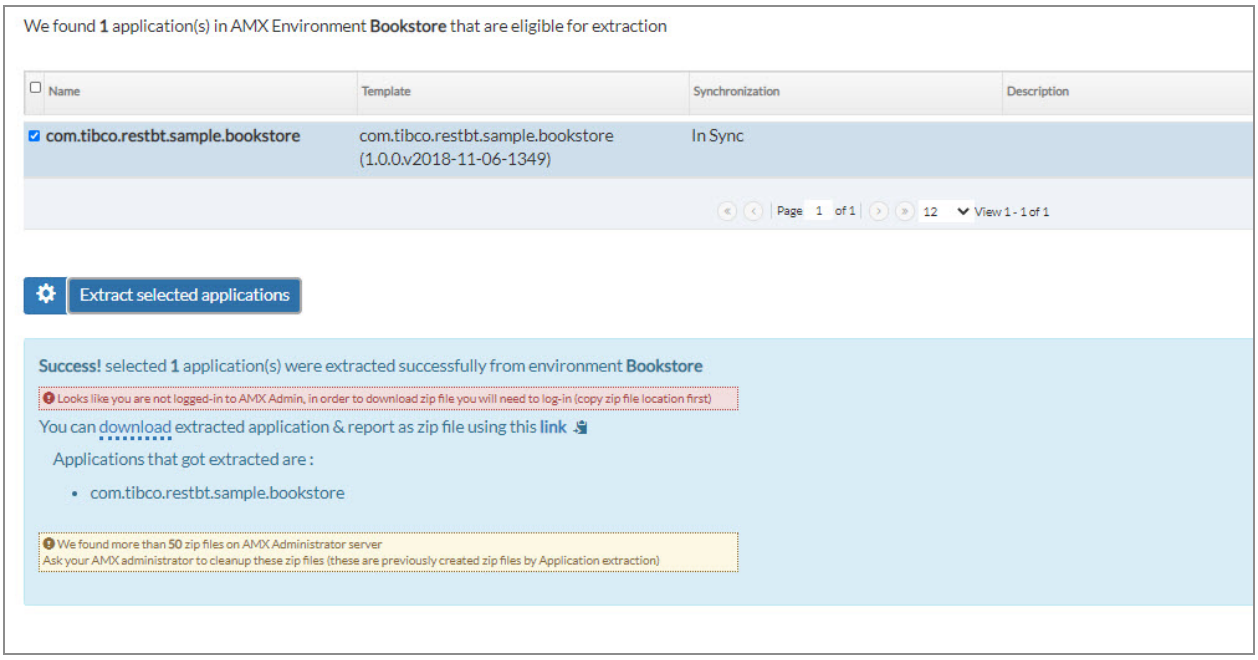
   A drop-down list with all the ActiveMatrix environments and number of applications within each environment is displayed. From the drop-down list, select the Environment from which you want to extract applications. A table shows all the applications within the selected environment.



3. Select a single application (or multiple applications) and click **Extract selected applications**. You can also select all applications by selecting the check box at the top.



On execution, a message appears on the screen as follows:



All the application information is extracted and stored as a `.zip` file on the server. If multiple applications are extracted, each application has a separate folder in the `.zip` file.

4. Click the **download** link to download the `.zip` file for the selected applications. You must log in to ActiveMatrix Administrator before downloading the `.zip` file.

Note: Only a super user can download the `.zip` file.

The tool generates a `.zip` file for selected application(s). The `.zip` file is named as *<EnterpriseName>_<EnvironmentName>_<DateTime>*`.zip`.

Each `.zip` file contains the following items. If multiple applications are selected for extraction, each application sub-folder within the exported .zip contains the following items:

- The `DAA` of the application (with the current `deployed appTemplate` version).
- The `*.config.yaml` file, which provides details of the current configuration.
- The `node_jvm_parameters.config.yaml` file, which provides the JVM configuration.
- A `node-log4j.xml` file, which provides the logging configuration details.
- The cert folder contains the keystore files that are used for the deployed application.
- An `Extract.report.text` file, which contains information about extraction and

errors encountered during export. In the `Extract.report.text` file, you can view all selected applications, successfully extracted applications, and failed applications.

When you complete the extraction, uninstall the Application Extractor from the SystemNode.

YAML Configuration File

This file represents the application and its configuration details. The structure of the file is as follows:

- General Information about application, environment, node, and template
- Properties
- Substitution Variables
- Services
- References
- Resources

This file represents a contract for building the image for containerization. In the case of multi-node deployment, it picks up random YAML file details. For more information about the YAML file, see "Understanding Application Configuration File" topic in the *TIBCO ActiveMatrix® Service Grid - Container Edition Administration*.

Limitations

The tool has the following limitations:

- Only the SOA application can be extracted.
- The application must only contain components like JAVA, Spring, Mediation, and WebApp.
- If the application or its components are mapped to multiple nodes, the application picks only one of the nodes from distribution and extracts information mainly from that node. If some required information is not available on the picked node, then other nodes are used to retrieve the information about:
    - Substitution variable values
    - Logging configuration
- If the application is using a shared library that was enabled separately from DAA

(such as JDBC Driver), those custom features need to be extracted and downloaded separately.

- If the application has a policy applied using TIBCO ActiveMatrix Policy Director, it is ignored.
- The application must be in the deployed (that is deployed, running, or stopped) state. The version of the application that is extracted depends on what version is deployed
- The application must be in `In Sync` state. An application in `Out of Sync` state cannot be extracted.
- If the application is in the middle of a rolling upgrade (that is, multiple versions are deployed to various nodes in the enterprise), the version for extraction is decided randomly.
- If the application uses multi-tenant features such as scoping, all that information is ignored.

Loggers

When extracting application configuration, loggers from node and application in TIBCO ActiveMatrix 3.x and default Log4j configuration in ActiveMatrix Service Grid - Container Edition are merged into the extracted `node-log4j.xml` file. When generating the log4j configuration file, loggers from an application have the highest priority, then loggers from node, and then loggers from default. The "root" loggers from node and application are skipped, the "root" logger from default is always used.

Logger appenders are assigned from the default configuration of ActiveMatrix Service Grid - Container Edition only.

Note: After extracting an application, the extracted `node-log4j.xml` file contains the `Console` appender, though the runtime node (`node-log4j.xml`) in TIBCO ActiveMatrix 3.x does not have the `Console` appender.

Removal of Specific JVM Arguments

When extracting the application configuration, the following JVM arguments are removed because they are not used in ActiveMatrix Service Grid - Container Edition.

- -Damx.securitymanager.enabled=false
- -Dcom.tibco.tibjms.connect.attempt.timeout=3000

- -XX:MaxPermSize=1024m
- -XX:PermSize=1024m

Differences between ActiveMatrix Service Grid 3.x and ActiveMatrix Service Grid - Container Edition

This section lists the differences between ActiveMatrix Container Edition and ActiveMatrix Service Grid 3.x in terms of functionality and migration support.

TIBCO ActiveMatrix Business Studio

TIBCO ActiveMatrix Business Studio is not bundled as part of ActiveMatrix Service Grid - Container Edition. You can use TIBCO ActiveMatrix Business Studio bundled with ActiveMatrix Service Grid 3.4.0, 3.3.1, or 3.3.0 to design applications.

Messaging Bus

Unlike ActiveMatrix Service Grid, Messaging Bus is not required for communication within the components of TIBCO ActiveMatrix applications or for message notification. Since the Messaging Bus is not required, there is no dependency on TIBCO EMS in ActiveMatrix Service Grid - Container Edition. TIBCO EMS is required for applications containing SOAP/JMS and JMS binding types, but it is limited to use at application level for business purpose.

TIBCO ActiveMatrix SPM Server and Dashboard

The TIBCO ActiveMatrix SPM Server and Dashboard are not available as part of ActiveMatrix Service Grid - Container Edition. You can use the Application Monitor in ActiveMatrix Service Grid - Container Edition to monitor running applications. Metrics similar to SPM can be exported by using various Prometheus exporters.

Policies

TIBCO ActiveMatrix Policy Director Governance to configure and enforce policies is not available in ActiveMatrix Service Grid - Container Edition. However, with few exceptions mentioned below, policies embedded or configured at the application design-time in TIBCO ActiveMatrix Business Studio or ActiveMatrix Administrator 3.x are supported. For dynamic Policy governance, an external API management tool can be used. Policies requiring Messaging Bus for enforcement are not supported because Messaging Bus is not

required in ActiveMatrix Service Grid - Container Edition as each application runs as a container.

Policy	Migration Support
Virtualization At Least Once	Not Supported
Virtualization At Most Once	Not Supported
Virtualization Transacted OneWay	Not Supported
JMS Transacted OneWay	Not Supported
Secure Virtualization	Not Supported
Logging	Not Supported
Prepare Before Undeploy	Not Supported

Resource Templates

Feature	Migration Support	Recommendation
Option to configure trust store from the Application Configurator by using **Configure SSL** wizard is not supported.	In Application Configurator, option of **Configure SSL** is not supported on the **SSL** tab of all resource templates.	You can use third-party tools such as OpenSSL.

Keystore and Password of Resource Template

If a resource template uses system keystore to store password, the password is extracted from the system keystore and stored in the YAML file. For keystore resource template, if it uses keystore file on the local file system, the keystore is bundled into the `.zip` file that you download from the Application Extractor.

Note: For HTTP connector resource template, the fields Idle Timeout, Request Header Size, Response Header Size, and Output Buffer Size are added in TIBCO ActiveMatrix Service Grid 3.4. If you are extracting the application configuration from TIBCO ActiveMatrix Service Grid 3.3.0 by using Application Extractor, the extracted YAML file does not contain these fields. However, if you upload the same application DAA to Application Configurator and generate the configuration, these fields are available in the YAML file.

Binding Types

TIBCO ActiveMatrix 3.x applications containing the following binding types are not supported or have limited support in ActiveMatrix Service Grid - Container Edition.

Binding Type	Migration Support	Comment
Virtualization	Not Supported	Applications containing the Virtualization binding type are not supported in the Application Configurator, Application Extractor, and Application Monitor. Since an application runs as one container, inter-component communication is not required through Messaging Bus or in-memory storage
JMS	Supported	Applications containing the JMS binding type can be uploaded in the Application Configurator and extracted in Application Extractor, but ability to configure these Applications in Application Configurator is limited.
Wired by Implementation with Virtualization Binding	Not Supported	Virtualization binding is not supported in ActiveMatrix Service Grid - Container Edition, therefore Wired by Implementation with Virtualization binding is not supported.

TIBCO Documentation and Support Services

How to Access TIBCO Documentation

Documentation for TIBCO products is available on the TIBCO Product Documentation website, mainly in HTML and PDF formats.

The TIBCO Product Documentation website is updated frequently and is more current than any other documentation included with the product. To access the latest documentation, visit https://docs.tibco.com.

Product-Specific Documentation

The following documentation for TIBCO ActiveMatrix® Service Grid - Container Edition is available on the TIBCO ActiveMatrix® Service Grid - Container Edition Product Documentation page:

- *TIBCO ActiveMatrix® Service Grid - Container Edition Release Notes*
- *TIBCO ActiveMatrix® Service Grid - Container Edition Cloud Deployment*
- *TIBCO ActiveMatrix® Service Grid - Container Edition Quick Start*
- *TIBCO ActiveMatrix® Service Grid - Container Edition Administration*
- *TIBCO ActiveMatrix® Service Grid - Container Edition Monitoring*

How to Contact TIBCO Support

You can contact TIBCO Support in the following ways:

- For an overview of TIBCO Support, visit http://www.tibco.com/services/support.
- For accessing the Support Knowledge Base and getting personalized content about products you are interested in, visit the TIBCO Support portal at https://support.tibco.com.
- For creating a Support case, you must have a valid maintenance or support contract with TIBCO. You also need a user name and password to log in to https://support.tibco.com. If you do not have a user name, you can request one by clicking Register on the website.

How to Join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can submit and vote on feature requests from within the TIBCO Ideas Portal. For a free registration, go to https://community.tibco.com.

Legal and Third-Party Notices