TIBCO ActiveMatrix® Service Grid - Container Edition

Monitoring

Version 1.0.0
December 2020

Document Updated: May 2021

Contents

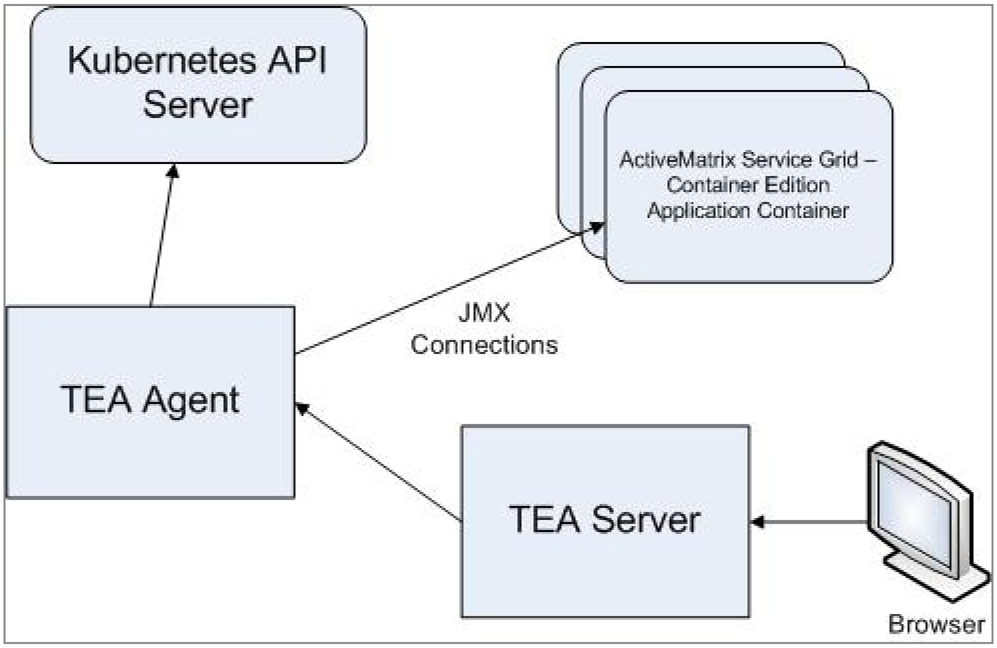
Monitoring Applications and Cloud Infrastructure from the Application Monitor

You can monitor the ActiveMatrix Service Grid - Container Edition applications deployed on the Docker and Kubernetes platforms from the Application Monitor. The Application Monitor is a web UI that functions in TIBCO® Enterprise Administrator (TEA).

ActiveMatrix Service Grid - Container Edition provides an agent for TIBCO Enterprise Administrator, which allows you to monitor and administer ActiveMatrix Service Grid - Container Edition applications. For more information about TIBCO Enterprise Administrator, see the TIBCO Enterprise Administrator documentation.

By using the Application Monitor, you can view data related to the Kubernetes and ActiveMatrix Service Grid - Container Edition nodes on a single dashboard. The dashboard displays data about Kubernetes resources such as running pods, deployments, namespaces, and states. You can also monitor runtime entities such as ActiveMatrix Service Grid - Container Edition environments, applications, nodes, and bindings in the enterprise.

Application Monitor architecture is as shown in the following diagram. From the browser, you can access the Application Monitor UI. Application Monitor runs in TIBCO Enterprise Administrator (TEA). The Application Monitor TEA agent communicates with the TIBCO Enterprise Administrator server for UI interactions and communicates with the ActiveMatrix Service Grid - Container Edition application containers by using JMX. Application Monitor creates two JMX connections to each ActiveMatrix Service Grid - Container Edition node or application, one to call JMX endpoints and get data and the second to register a JMX notification listener to detect any state change in the ActiveMatrix Service Grid - Container Edition node. Kubernetes related data is fetched from the Kubernetes API server.


Kubernetes API Server
ActiveMatrix Service Grid – Container Edition Application Container
JMX Connections
TEA Agent
TEA Server
Browser

Logging in to Application Monitor

You can log in to the Application Monitor from a Web browser.

Before you begin

- Ensure that you have deployed TIBCO Enterprise Administrator server container in the Kubernetes cluster. For more information, see *TIBCO ActiveMatrix® Service Grid - Container Edition Cloud Deployment*.
- Ensure that you have deployed TEA agent container in the Kubernetes cluster. For more information about deploying the Application Monitor in the Kubernetes cluster, see *TIBCO ActiveMatrix® Service Grid - Container Edition Cloud Deployment*.
- Ensure that ActiveMatrix Service Grid - Container Edition application is deployed in the Kubernetes cluster. For more information, see *TIBCO ActiveMatrix® Service Grid - Container Edition Cloud Deployment*.

Procedure

1. Open a browser and navigate to `http://<hostname>:31877/tea`. Here, 31877 is the default port number.

> **Note:** The default port number and other settings can be changed by modifying the `tea-server.yaml` Kubernetes deployment file, which is available at `amsgce-runtime-<version>\samples\kubernetes\Application_Monitor`.

2. Enter your login credentials and click **Sign In**. The default user name and password are `admin`. The default timeout for a session is 30 minutes.

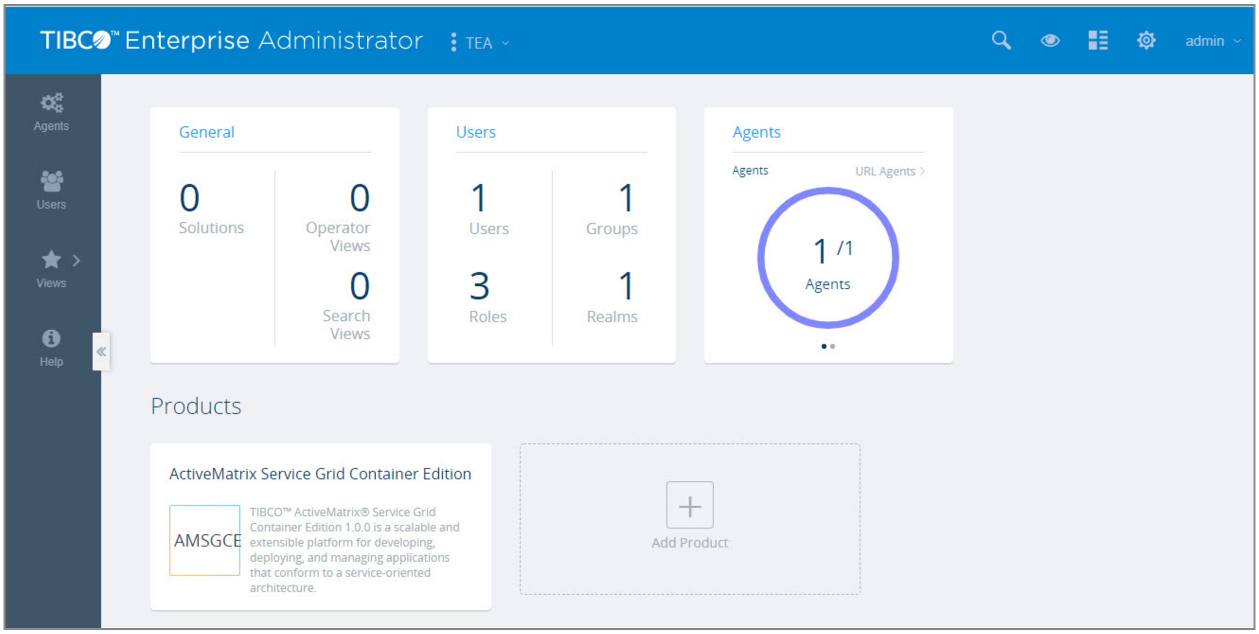
Result

On successful authentication, the TIBCO Enterprise Administrator landing page is displayed.

Navigating to the Application Monitor UI

Signing in to the Application Monitor displays the TIBCO Enterprise Administrator landing page. The user name with which you have logged in is shown on the title bar. The landing page displays cards with information about the general details, users, agents, and products exposed to the TIBCO Enterprise Administrator server. All the products (for example, ActiveMatrix Service Grid - Container Edition) exposed to the server are displayed as cards. You can click the links on the cards to drill down to further details.

The following image displays the TIBCO Enterprise Administrator landing page:

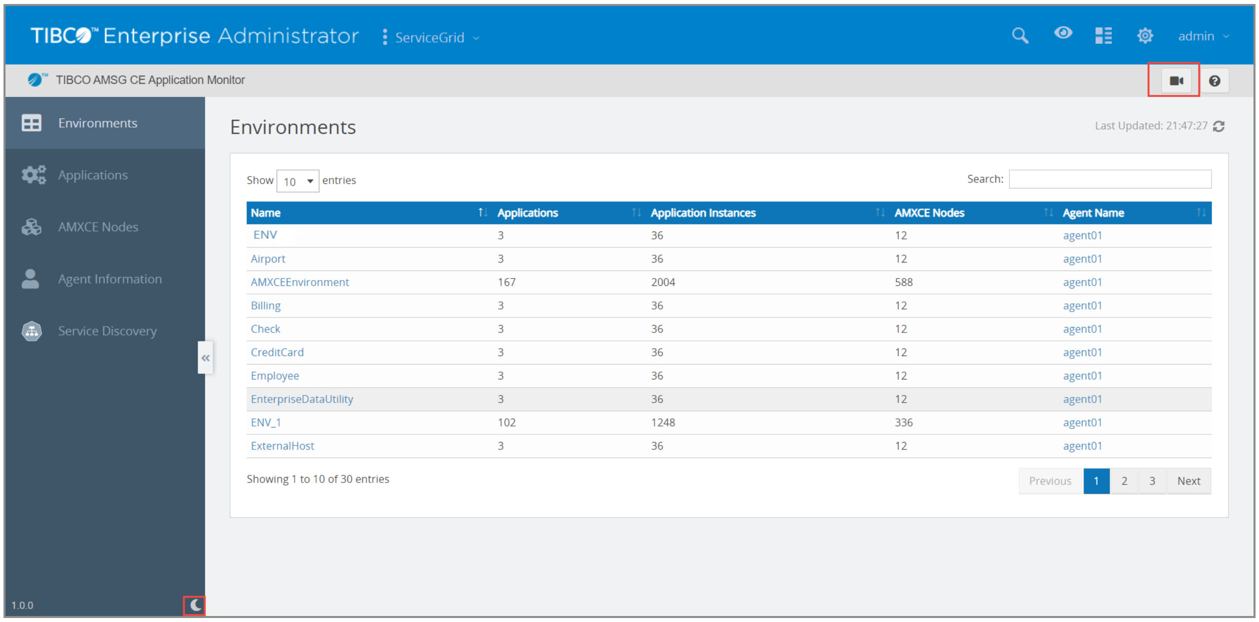


TIBCO™ ActiveMatrix® Service Grid Container Edition 1.0.0 is a scalable and extensible platform for developing, deploying, and managing applications that conform to a service-oriented architecture.

Application Monitor Start Page

To open the Application Monitor start page, click the ActiveMatrix Service Grid - Container Edition product card on the TIBCO Enterprise Administrator landing page. By default, the Application Monitor opens on the **Environments** page.

The following image displays the Application Monitor start page:



Name	Applications	Application Instances	AMXCE Nodes	Agent Name
ENV	3	36	12	agent01
Airport	3	36	12	agent01
AMXCEEnvironment	167	2004	588	agent01
Billing	3	36	12	agent01
Check	3	36	12	agent01
CreditCard	3	36	12	agent01
Employee	3	36	12	agent01
EnterpriseDataUtility	3	36	12	agent01
ENV_1	102	1248	336	agent01
ExternalHost	3	36	12	agent01

Showing 1 to 10 of 30 entries

Guided Tour

To get started, you can take a tour of the Application Monitor UI. To start the tour, click **Start Tour** 📹 in the upper-right corner of the Application Monitor.

Light and Dark Theme

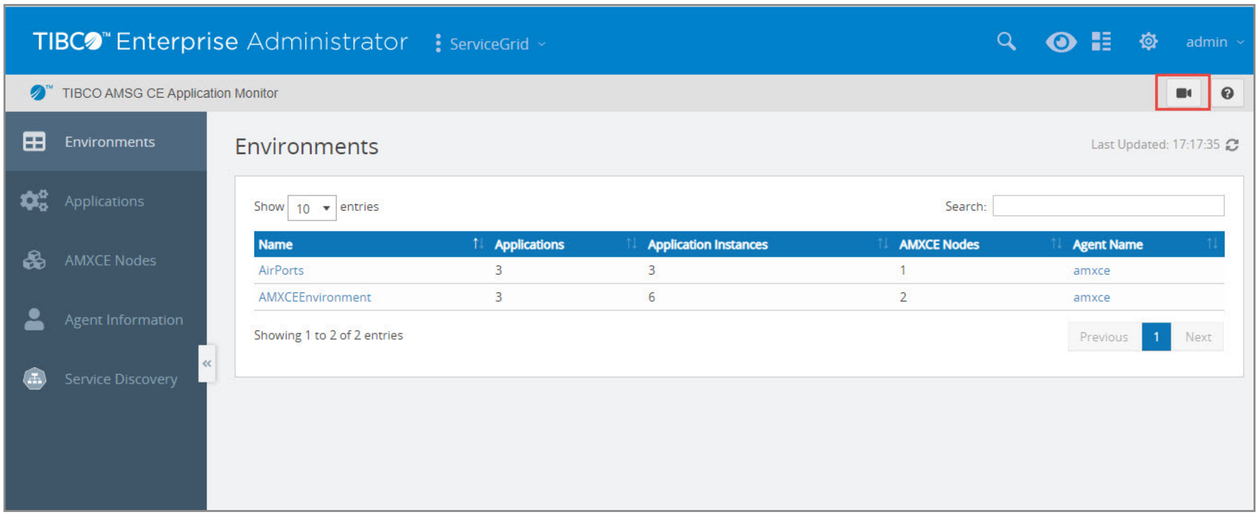
You can switch between light and dark theme by clicking the theme icon ☾ in the lower-left corner.

What to do next

- Viewing Environment Information
- Viewing Running Applications
- Viewing AMXCE Nodes Information

Viewing Environment Information

An environment is a logical grouping of applications and nodes. There can be multiple environments. In the Application Monitor, the **Environments** tab lists all environments.



Environments

Last Updated: 17:17:35

Show 10 entries

Search:

Name	Applications	Application Instances	AMXCE Nodes	Agent Name
AirPorts	3	3	1	amxce
AMXCEEnvironment	3	6	2	amxce

Showing 1 to 2 of 2 entries

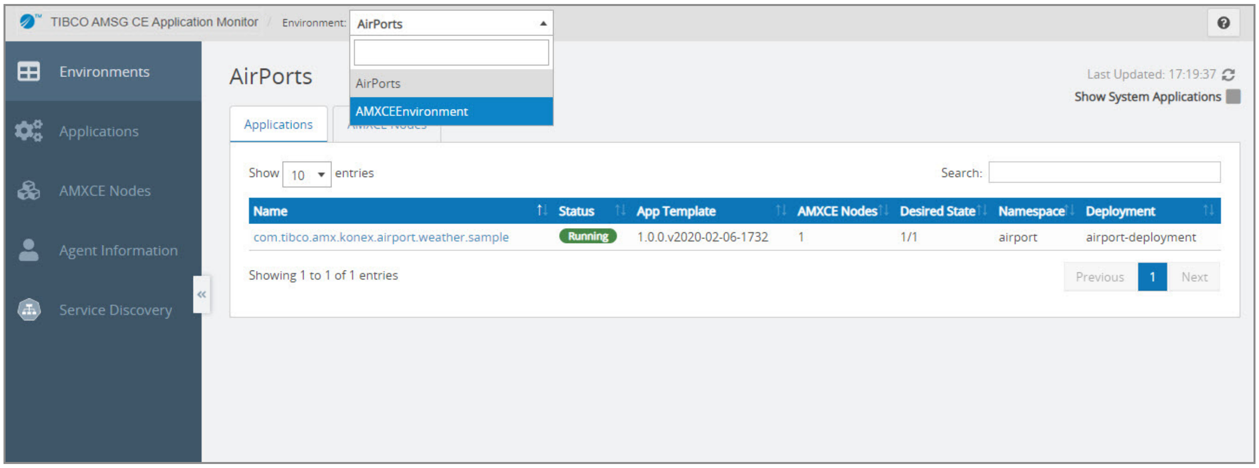
Previous 1 Next

For each environment, the following information is available:

Column	Description
Name	Name of the environment. Click the environment name to view its details.
Applications	Total number of applications running in the environment. Note that here the applications include platform (system) applications and user applications.
Application Instances	Total number of application instances running in the environment. This value changes once deployed on Kubernetes. On Kubernetes, you can change the number of replicas for each user application. An application instance is a complete copy of the application that is running on an AMXCE node. For example, when we deploy an AMXCE application to Kubernetes with one replica, we have one application instance. If we specify "replicas: 2" then we will have 2 application instances.
AMXCE Nodes	Total number of AMXCE nodes running in the environment.
Agent Name	Name of the Application Monitor agent.

Environment Details Page

On the environment details page, you can view the applications and AMXCE nodes in an environment. You can search and select an environment from the drop-down list and navigate to the environment details page.



Name	Status	App Template	AMXCE Nodes	Desired State	Namespace	Deployment
com.tibco.amx.konex.airport.weather.sample	Running	1.0.0.v2020-02-06-1732	1	1/1	airport	airport-deployment

Showing 1 to 1 of 1 entries

Applications in the Environment

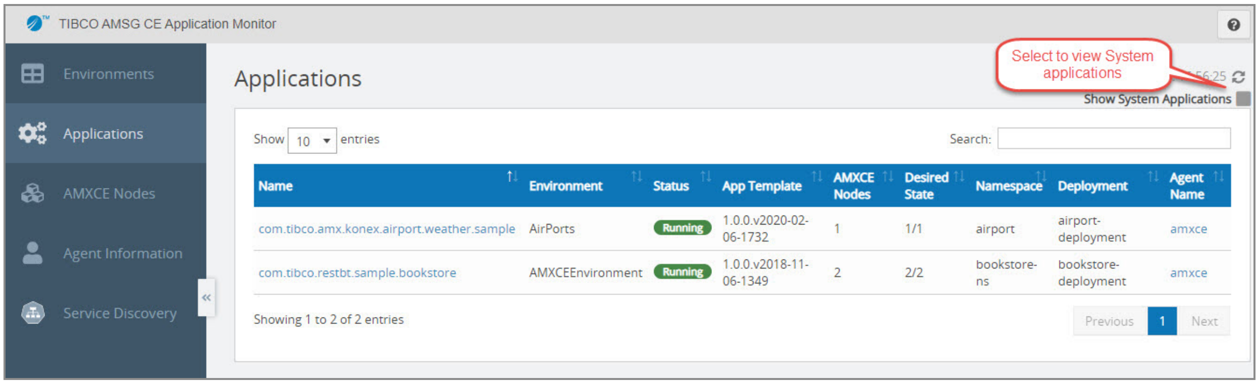
You can view the list of Applications running in an Environment. Click an Application name in the **Applications** tab of the selected environment details page. For more information about the fields, see Viewing Running Applications.

AMXCE Nodes in the Environment

You can view the list of AMXCE Nodes running in an Environment. Click a Node name in the **AMXCE Nodes** tab of the selected environment details page. For more information about the fields, see Viewing AMXCE Nodes Information.

Viewing Running Applications

You can view the user and system application's status on the **Applications** page. To open this page, click **Applications** from the sidebar.



Select to view System applications

Applications

Show 10 entries

Search:

Name	Environment	Status	App Template	AMXCE Nodes	Desired State	Namespace	Deployment	Agent Name
com.tibco.amx.konex.airport.weather.sample	AirPorts	Running	1.0.0.v2020-02-06-1732	1	1/1	airport	airport-deployment	amxce
com.tibco.restbt.sample.bookstore	AMXCEEnvironment	Running	1.0.0.v2018-11-06-1349	2	2/2	bookstore-ns	bookstore-deployment	amxce

Showing 1 to 2 of 2 entries

Previous 1 Next

The **Applications** page displays the following information:

Field	**Description**
Name	Name of the application.
Environment	ActiveMatrix Service Grid - Container Edition environment name in which the application is running.
Status	Specifies the status of the application. **Possible values:** • Running • Partially Running • Stopped The runtime state is a rollup value for all the application's components and bindings. Partial states mean that some of an application's components and bindings are in a different state than others.
App Template	The version of the application template from which the application was created.
AMXCE Nodes	Number of AMXCE nodes on which the application is running.

Field	Description
Desired State	Displays readiness state for the applications as ready/total replicas. This field is not applicable for system applications.
Namespace	Name of namespace in which application pods are deployed.
Deployment	Deployment name specified in the Kubernetes deployment file.
Agent Name	Name of the agent. You can click the agent name to view its details.

To View System Applications

By default, only user applications are displayed on the **Applications** page. To view the following system applications, select the **Show System Applications** check box in the upper-right corner of the **Applications** page:

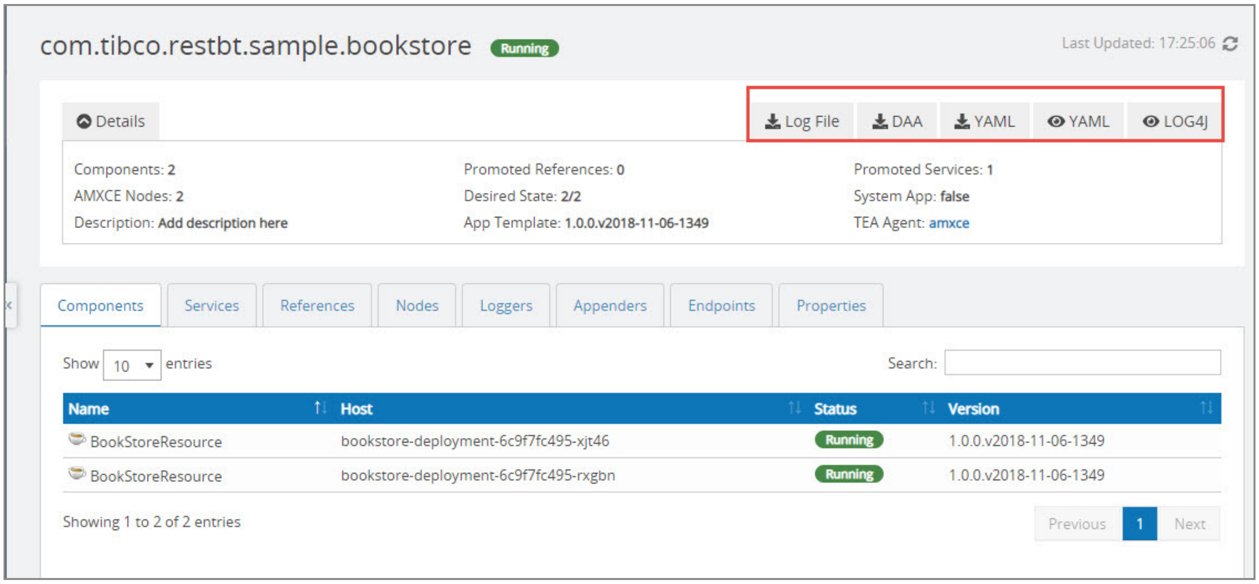
- `com.tibco.amx.platform`
- `com.tibco.amxce.platform.monitor`
- `com.tibco.amx.it.mediation.apt`
- `GovernanceControlDistribution` (Displayed only when Policy sets are used in applications)

See the following topics for information about Application details:

- Viewing Application General Information
- Viewing Application Components
- Viewing Application Services
- Viewing Application References
- Configuring Application Logging
- Application Endpoints
- Application Properties

Viewing Application General Information

To view application details, on the **Applications** page, select the application for which you want to view details. The applications detail page is displayed as shown in the following image.



com.tibco.restbt.sample.bookstore Running

Last Updated: 17:25:06

Components: 2
AMXCE Nodes: 2
Description: Add description here

Promoted References: 0
Desired State: 2/2
App Template: 1.0.0.v2018-11-06-1349

Promoted Services: 1
System App: false
TEA Agent: amxce

Show 10 entries

Search:

Name	Host	Status	Version
BookStoreResource	bookstore-deployment-6c9f7fc495-xjt46	Running	1.0.0.v2018-11-06-1349
BookStoreResource	bookstore-deployment-6c9f7fc495-rxgbn	Running	1.0.0.v2018-11-06-1349

Showing 1 to 2 of 2 entries

The **Details** section on the application detail page displays general information about the application. You can collapse and expand this section.

Name	Description
Components	Number of components in the application.
Promoted References	The number of promoted references.
Promoted Services	The number of promoted services.
AMXCE Nodes	The number of AMXCE nodes on which the application is deployed.
Desired State	Displays readiness state for the applications as ready/total replicas. This field is not applicable for System applications.

Name	Description
System App	Specifies whether the deployed application is user application or System application.
Description	The application description from the YAML file
App Template	The application template version.
TEA Agent	Click the agent name to view details.

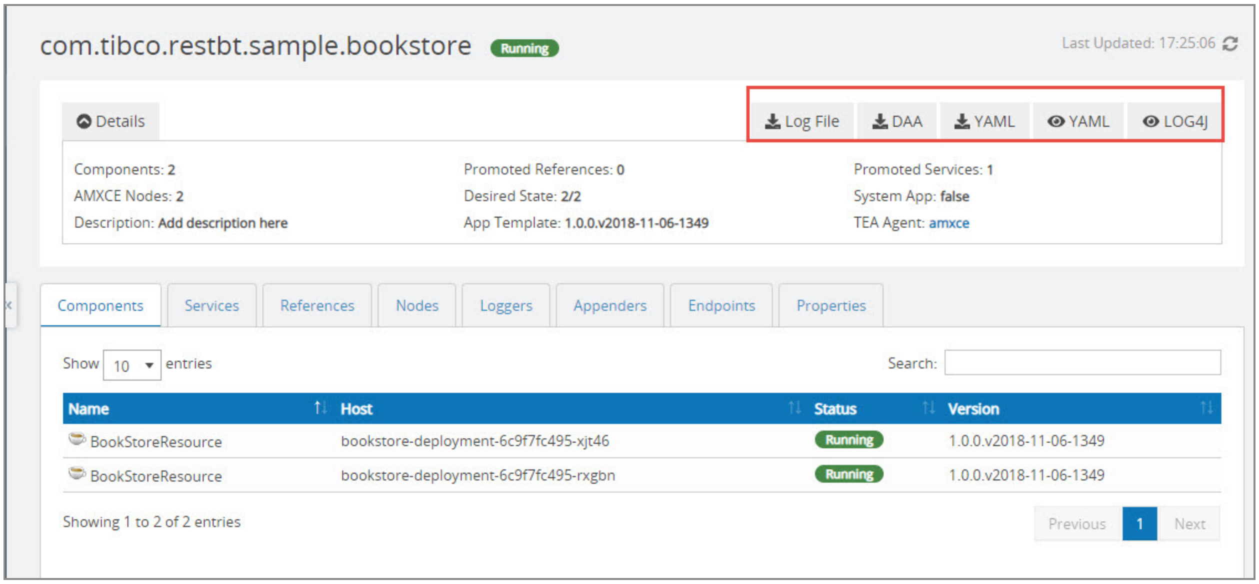
You can perform the following tasks from the application details page:

Task	Steps
Download Application Logs	To download all logs for an application, on the application details page, click **Log File** in the upper-right corner. This button will download all the logs available on the deployed nodes. The name of the downloaded file is `<ApplicationName>_TimeStamp.logs.zip`, the individual node log is included in a separate folder inside the zip file.
Download Application DAA	To download DAA for an application, on the application details page, click the **DAA** button in the upper-right corner. The downloaded `.zip` file contains the application DAA and any dependent DAA such as JDBC driver and custom features DAA. The name of the downloaded file is `<ApplicationName>.zip`.
Download Application Configuration YAML File	To download the YAML file for an application, on the application details page, click the **Download YAML** button in the upper-right corner. The name of the downloaded file is `<ApplicationName>.yaml.`
View Application Configuration YAML	To view the Application Configuration YAML file for an application, on the application details page, click **View**

Task	Steps
	YAML.
View Log4j XML Configuration	To view the Log4j configuration applied to an application, click **LOG4J** in the upper-right corner. You need not inspect the container every time to view the Log4j configuration.

Viewing Application Components

On the **Applications** page, select the application to view details. By default, the **Components** tab is displayed.



com.tibco.restbt.sample.bookstore

Components: 2
AMXCE Nodes: 2
Description: Add description here
Promoted References: 0
Desired State: 2/2
App Template: 1.0.0.v2018-11-06-1349
Promoted Services: 1
System App: false
TEA Agent: amxce

Show 10 entries

Search:

Name	Host	Status	Version
BookStoreResource	bookstore-deployment-6c9f7fc495-xjt46	Running	1.0.0.v2018-11-06-1349
BookStoreResource	bookstore-deployment-6c9f7fc495-rxgbn	Running	1.0.0.v2018-11-06-1349

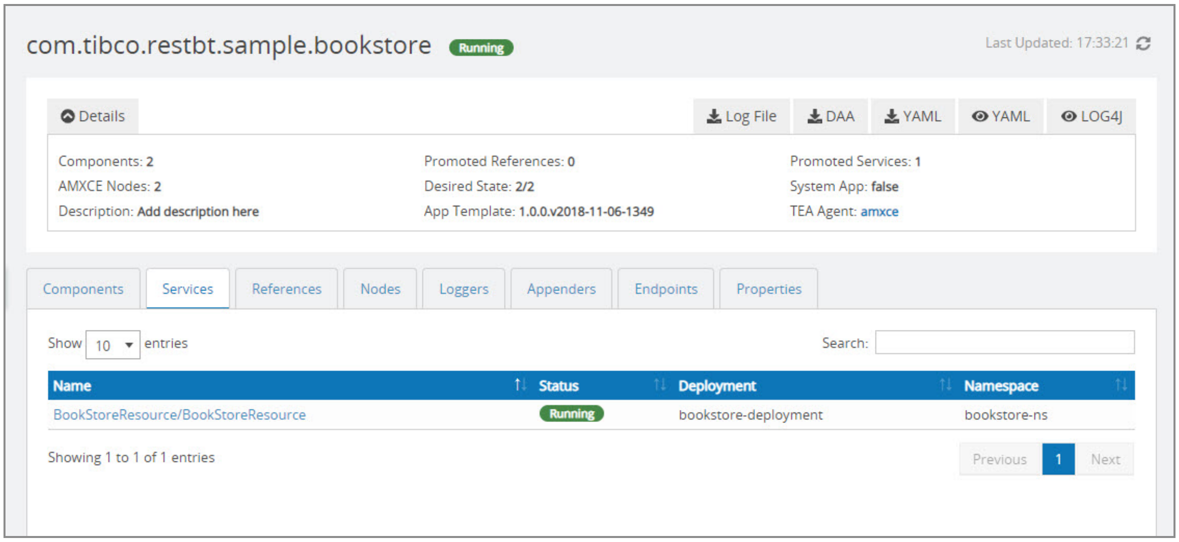
Showing 1 to 2 of 2 entries

This tab lists all the components in an application. You can get the following information:

- **Name:** Name of the component
- **Host:** Hostname is not the machine name it is combination of deployment name and container id.
- **Status:** Status of the component
- **Version:** Version of the component

Viewing Application Services

1. On the **Applications** page, select the application to view details.
2. To view the services in the application, click the **Services** tab. All services for an application are displayed. Each service is uniquely identified by deployment and namespace.



com.tibco.restbt.sample.bookstore Running

Last Updated: 17:33:21

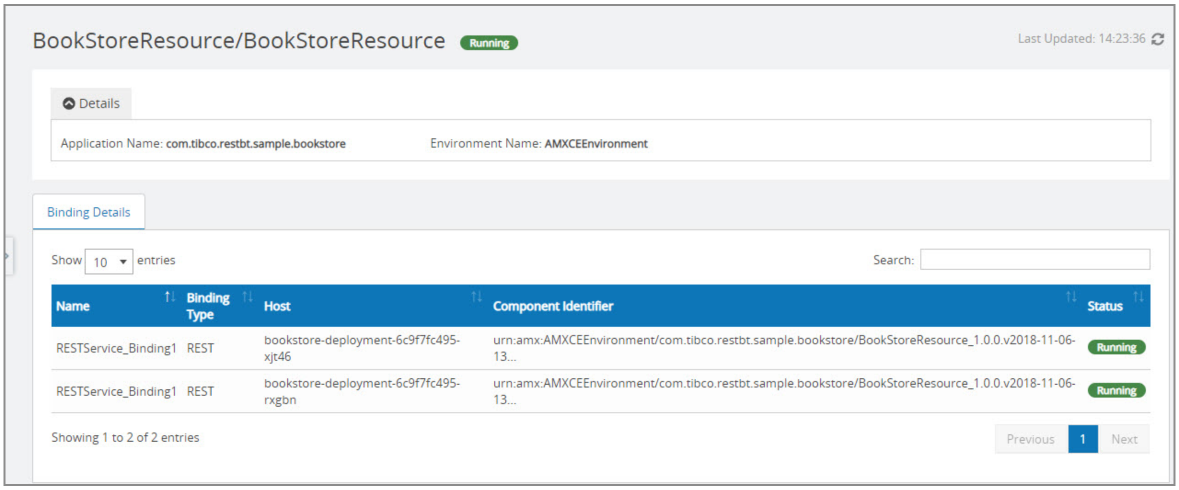
Components: 2
AMXCE Nodes: 2
Description: Add description here
Promoted References: 0
Desired State: 2/2
App Template: 1.0.0.v2018-11-06-1349
Promoted Services: 1
System App: false
TEA Agent: amxce

Show 10 entries Search:

Name	Status	Deployment	Namespace
BookStoreResource/BookStoreResource	Running	bookstore-deployment	bookstore-ns

Showing 1 to 1 of 1 entries

3. To view details about a service, click the service name. All bindings in the service are listed.



BookStoreResource/BookStoreResource Running

Last Updated: 14:23:36

Application Name: com.tibco.restbt.sample.bookstore
Environment Name: AMXCEEnvironment

Show 10 entries Search:

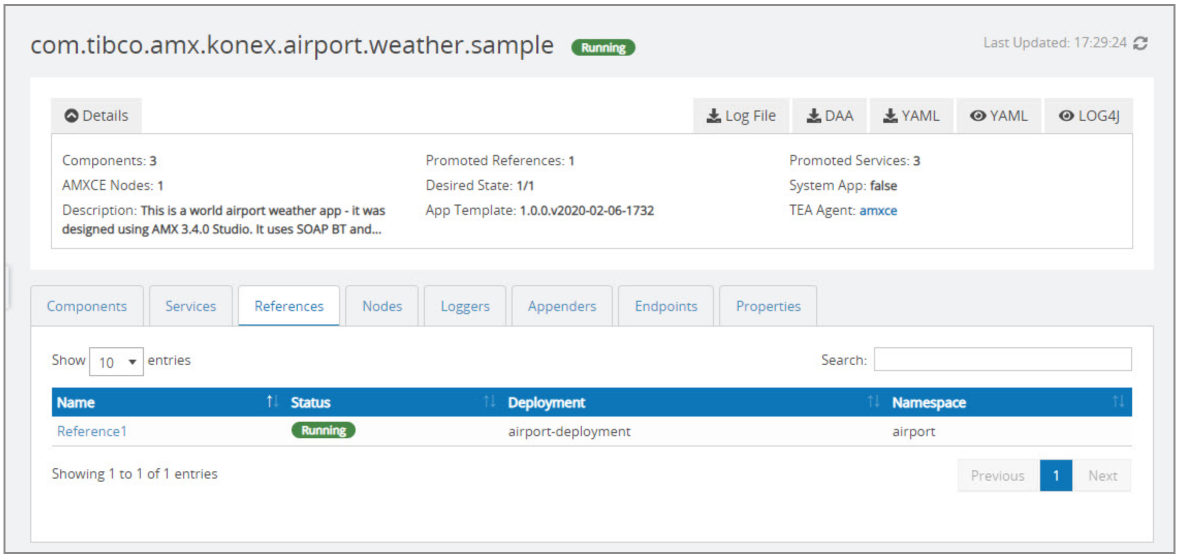
Name	Binding Type	Host	Component Identifier	Status
RESTService_Binding1	REST	bookstore-deployment-6c9f7fc495-xjt46	urn:amx:AMXCEEnvironment/com.tibco.restbt.sample.bookstore/BookStoreResource_1.0.0.v2018-11-06-13...	Running
RESTService_Binding1	REST	bookstore-deployment-6c9f7fc495-rxgbn	urn:amx:AMXCEEnvironment/com.tibco.restbt.sample.bookstore/BookStoreResource_1.0.0.v2018-11-06-13...	Running

Showing 1 to 2 of 2 entries

Viewing Application References

1. On the **Applications** page, select an application to view details.

2. To view references in the application, click the **References** tab. All references in the application are displayed.



com.tibco.amx.konex.airport.weather.sample Running

Last Updated: 17:29:24

Components: **3**
AMXCE Nodes: **1**
Description: **This is a world airport weather app - it was designed using AMX 3.4.0 Studio. It uses SOAP BT and...**

Promoted References: **1**
Desired State: **1/1**
App Template: **1.0.0.v2020-02-06-1732**

Promoted Services: **3**
System App: **false**
TEA Agent: amxce

Show 10 entries Search:

Name	Status	Deployment	Namespace
Reference1	Running	airport-deployment	airport

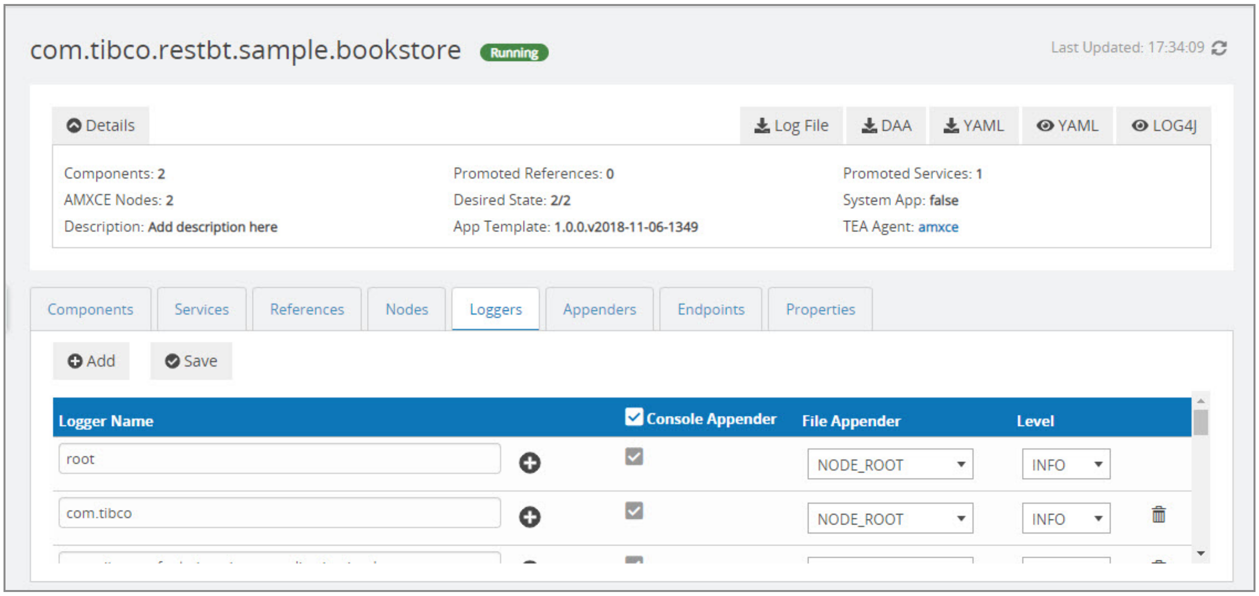
Showing 1 to 1 of 1 entries

3. To view details about a reference, click the reference name. All reference bindings are listed.

Configuring Application Logging

You can view the Log4j configuration for an application on the **Loggers** and **Appenders** tabs. The same configuration is generated in the `node-log4j.xml` file.

You can add, update, or delete loggers and appenders. Multiple appenders can be assigned to a logger. Any changes made to this configuration are applied to all the nodes at runtime.



com.tibco.restbt.sample.bookstore

Components: 2
AMXCE Nodes: 2
Description: Add description here
Promoted References: 0
Desired State: 2/2
App Template: 1.0.0.v2018-11-06-1349
Promoted Services: 1
System App: false
TEA Agent: amxce

Logger Name	☑ Console Appender	File Appender	Level
root	☑	NODE_ROOT	INFO
com.tibco	☑	NODE_ROOT	INFO

Note: You cannot create a logger with the name 'root' (or any variant of this by case sensitivity). If you try to add such a logger, a warning message is displayed. You cannot rename the existing root logger.

Fields on the Logger Tab

Property	Required?	Editable?	Description
Logger Name	Y	Y	The name of the logger.
Appender	Y	Y	The destination to which log events are appended.
Console Appender	N	Y	Select the check box in the **Console Appender** column to display logs on the console.
Level	Y	Y	All events of a level equal to or lower (less verbose) than the specified level are logged. For the Info level, Info, Warn, Error and Fatal events are logged. • TRACE: All events. • DEBUG: Fine-grained informational events used for debugging an application.

Property	Required?	Editable?	Description
			• INFO: Coarse-grained informational messages that highlight the progress of the application. • WARN: Potentially harmful events. • ERROR: Errors that allow the application to continue running. • FATAL: Errors that cause the application to fail. • OFF: Blocks passing messages to a parent

Appender Tab

Property	Description
Name	Name of the logging appender.
Location	The fully-qualified path to the log file.
Max Size(KB)	The maximum size of each log file in kilobytes. **Default:** 10240 KB
Max Backup Num	The number of log files to keep. When a log file reaches the maximum size, a new log file is created. After the number of files matches the number specified, the oldest is deleted when a new file is created. Each file is appended with a number. **Default:** 25 That is, to specify the Max Backup value, you can define a substitution variable: `%%node_log_max_backup_index%%`
Pattern Layout	Controls the format of the log entries for a clear text file appender. Conforms to the log4j pattern layout. **Default:**

Appender Tab(Continued)

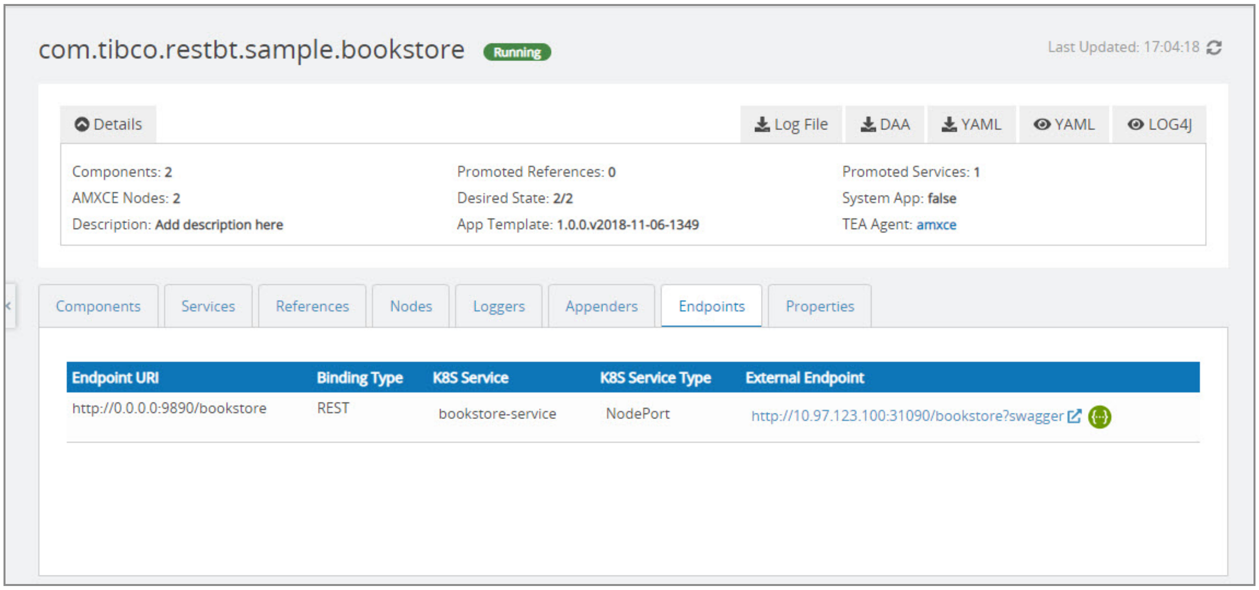
Property	Description

	`"%d{dd MMM yyyy HH:mm:ss,SSS} [%t] [%-5p] %c %X{_cl.correlationId} - %m%n"`
	This string prints the date, the name of the thread that generated the event, the level of the logged event, the category of the logged event, a correlation ID (an enrichment field), a message, and a line separator. For example:
	`17 Dec 2009 16:43:41,250 [Job_Executor2] [INFO ] com.tibco.amf.hpa.tibcohost.node.TibcoHostNode. - Successfully finished processing of RDA rda6705267566599374829.zip`
	In addition to the default format, ActiveMatrix Service Grid - Container Edition also supports the pattern layouts extended with enrichment fields.
	`%R{_cl.physicalCompId.matrix.host} %d'{dd MMM yyyy HH:mm:ss,SSS}' [%t] [%-5p] %c - %m%n`
	Note: Pattern layout is not validated in the Application Configurator. Ensure that pattern layout conforms to the Log4j pattern layout.

Note: If the same appender is assigned to the logger with the different level, the one with the highest level is only assigned to the logger.

Application Endpoints

The **Endpoints** tab of the application details page lists all the endpoints exposed by an application.



com.tibco.restbt.sample.bookstore

Last Updated: 17:04:18

Components: 2
AMXCE Nodes: 2
Description: Add description here

Promoted References: 0
Desired State: 2/2
App Template: 1.0.0.v2018-11-06-1349

Promoted Services: 1
System App: false
TEA Agent: amxce

Endpoint URI	Binding Type	K8S Service	K8S Service Type	External Endpoint
http://0.0.0.0:9890/bookstore	REST	bookstore-service	NodePort	http://10.97.123.100:31090/bookstore?swagger

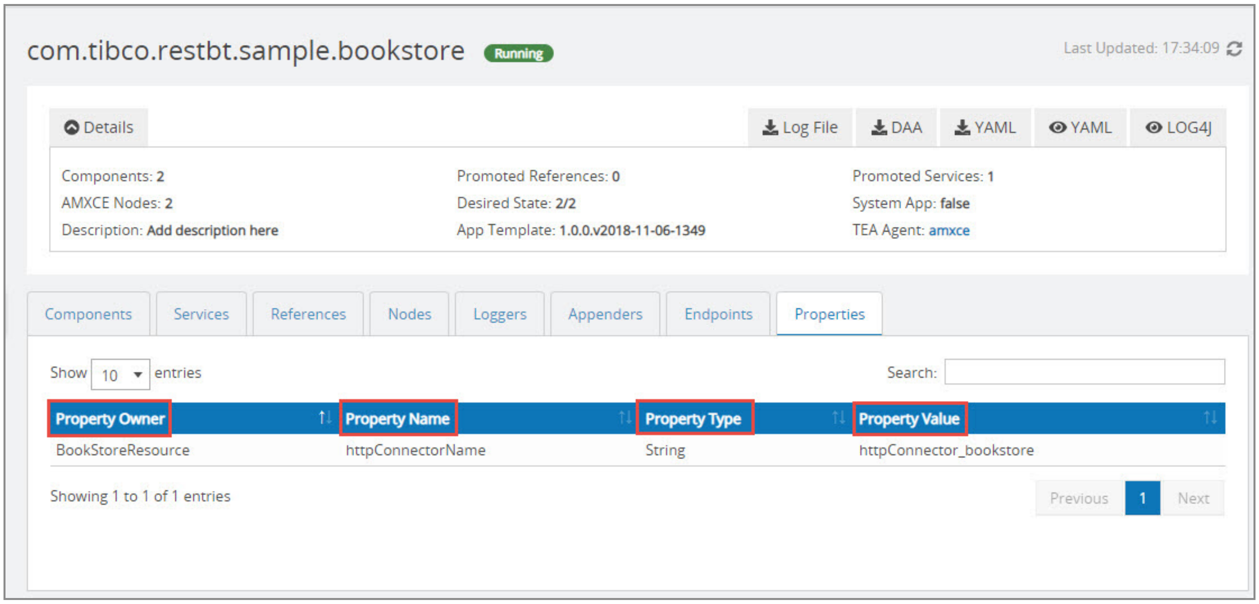
You can get the following information about endpoints:

Name	Description
Endpoint URI	The endpoint URI.
Binding Type	The binding type can be SOAP or REST.
K8S Service	Kubernetes service name.
K8S Service Type	Kubernetes service type. It can be NodePort, LoadBalancer, or ClusterIP. **Note:** If you have not mapped any Kubernetes service, then the **K8S** and **K8S Service Type** fields have values as NA.
External Endpoint	Click the **External URL** to view the respective WSDL or JSON structure based on the binding type. For more information about accessing Swagger UI, see Accessing Swagger UI from the Application Monitor.

Note: JMS binding is not supported.

Application Properties

The **Properties** tab lists the properties in an application as shown in the following image. For more information about Application Properties, see *TIBCO ActiveMatrix® Service Grid - Container Edition Administration*.



com.tibco.restbt.sample.bookstore

Last Updated: 17:34:09

Components: 2
AMXCE Nodes: 2
Description: Add description here

Promoted References: 0
Desired State: 2/2
App Template: 1.0.0.v2018-11-06-1349

Promoted Services: 1
System App: false
TEA Agent: amxce

Show 10 entries

Search:

Property Owner	Property Name	Property Type	Property Value
BookStoreResource	httpConnectorName	String	httpConnector_bookstore

Showing 1 to 1 of 1 entries

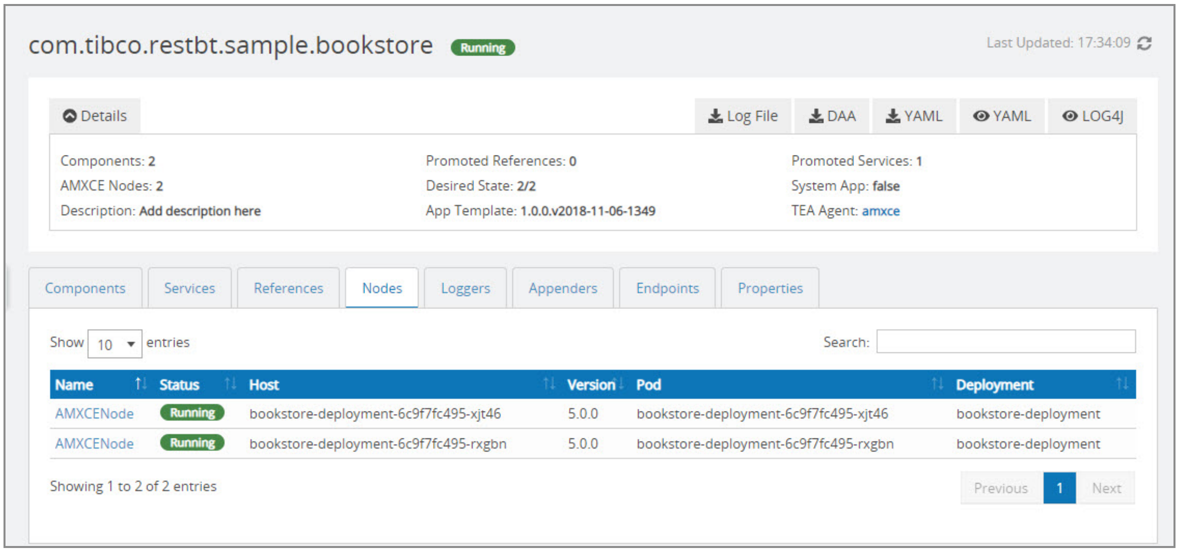
Viewing AMXCE Nodes Information

AMXCE node is the runtime environment for applications. Nodes exist in an environment and are managed by containers. In ActiveMatrix Service Grid - Container Edition, a node runs as an integration node. A node acts as a sandbox for applications. One application instance runs on one AMXCE node.

You can navigate to nodes page in one of the following ways:

From the Applications Page

1. On the **Applications** page, select an application to view details.
2. To view the AMXCE node information on which the application is running, click the **Nodes** tab. The AMXCE nodes on which the application is running are displayed. Click a node name to view its details.



com.tibco.restbt.sample.bookstore Running

Last Updated: 17:34:09

Components: **2**
AMXCE Nodes: **2**
Description: **Add description here**
Promoted References: **0**
Desired State: **2/2**
App Template: **1.0.0.v2018-11-06-1349**
Promoted Services: **1**
System App: **false**
TEA Agent: amxce

Show 10 entries

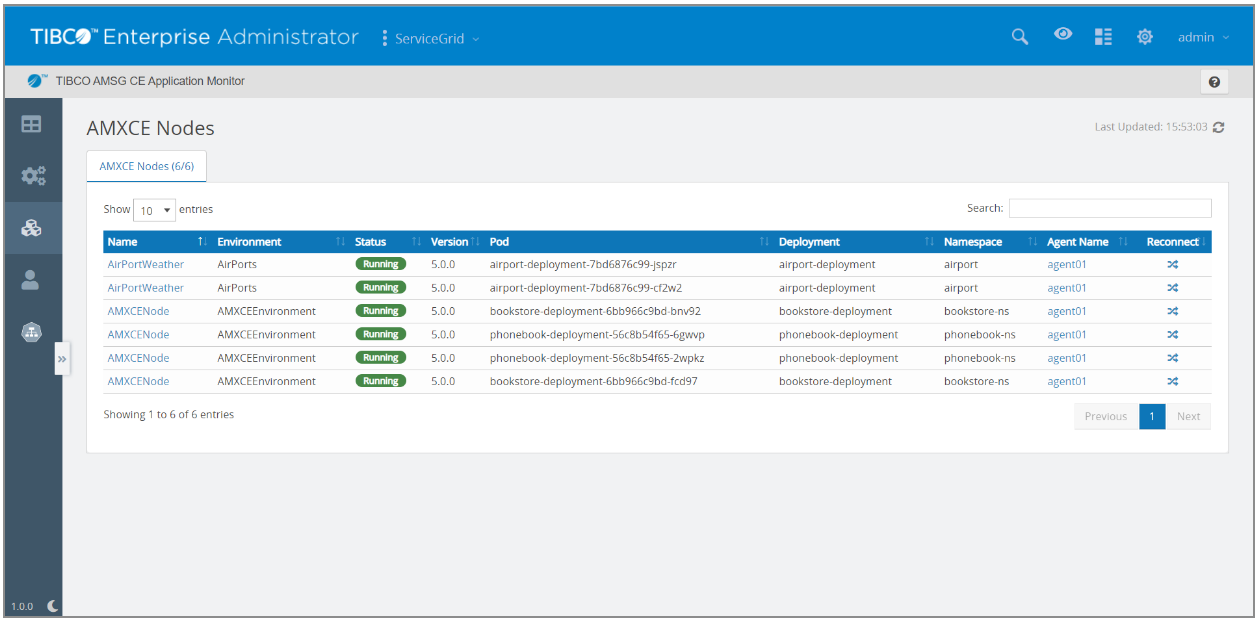
Search:

Name	Status	Host	Version	Pod	Deployment
AMXCENode	Running	bookstore-deployment-6c9f7fc495-xjt46	5.0.0	bookstore-deployment-6c9f7fc495-xjt46	bookstore-deployment
AMXCENode	Running	bookstore-deployment-6c9f7fc495-rxgbn	5.0.0	bookstore-deployment-6c9f7fc495-rxgbn	bookstore-deployment

Showing 1 to 2 of 2 entries

From the AMXCE Nodes tab

To view the AMXCE nodes information, in the Application Monitor, click the **AMXCE Nodes** tab.



TIBCO™ Enterprise Administrator
ServiceGrid
admin
TIBCO AMSG CE Application Monitor
1.0.0

AMXCE Nodes

Last Updated: 15:53:03

AMXCE Nodes (6/6)

Show 10 entries

Search:

Name	Environment	Status	Version	Pod	Deployment	Namespace	Agent Name	Reconnect
AirPortWeather	AirPorts	Running	5.0.0	airport-deployment-7bd6876c99-jspzr	airport-deployment	airport	agent01	🔀
AirPortWeather	AirPorts	Running	5.0.0	airport-deployment-7bd6876c99-cf2w2	airport-deployment	airport	agent01	🔀
AMXCENode	AMXCEEnvironment	Running	5.0.0	bookstore-deployment-6bb966c9bd-bnv92	bookstore-deployment	bookstore-ns	agent01	🔀
AMXCENode	AMXCEEnvironment	Running	5.0.0	phonebook-deployment-56c8b54f65-6gwvp	phonebook-deployment	phonebook-ns	agent01	🔀
AMXCENode	AMXCEEnvironment	Running	5.0.0	phonebook-deployment-56c8b54f65-2wpkz	phonebook-deployment	phonebook-ns	agent01	🔀
AMXCENode	AMXCEEnvironment	Running	5.0.0	bookstore-deployment-6bb966c9bd-fcd97	bookstore-deployment	bookstore-ns	agent01	🔀

Showing 1 to 6 of 6 entries

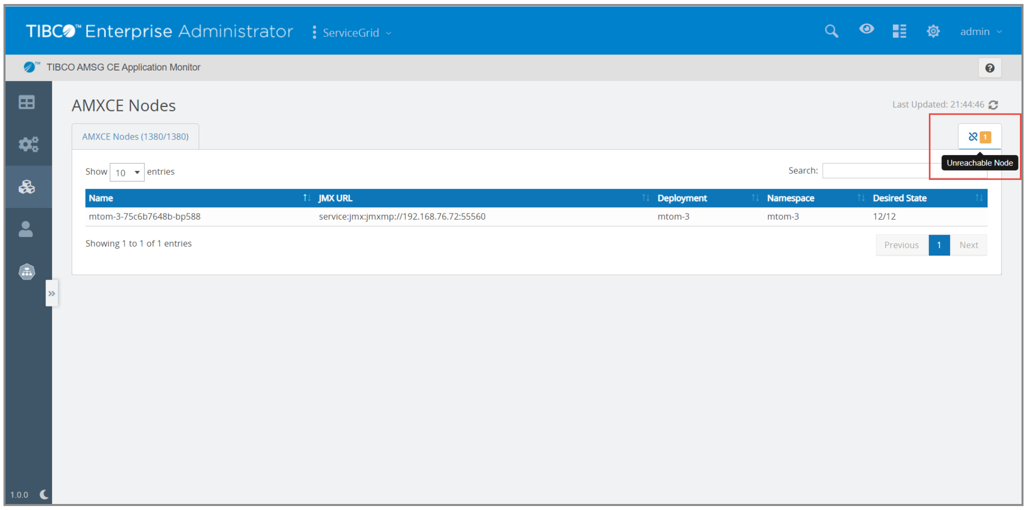
AMXCE Nodes (*x*/*y*) at the top indicates that *x* is the count of nodes that are registered with the Application Monitor agent and *y* is the number of total nodes present in the cluster.

Column	Description
Name	AMXCE node name. Click the node name to view its details.
Environment	Name of the ActiveMatrix Service Grid - Container Edition environment in which the node is running.
Status	Specifies the status of the entity. **Possible values:** • Running • Partially Running: Node status is rollup of all components, bindings, services status running on the node. If any component, binding, or service is in partially running state then the node will be in the Partially Running state. • Stopped
Version	AMXCE node version.
Agent Name	Name of the AMXCE agent. Click the agent name to view its details.
Reconnect	Reconnecting involves recreating the JMX connections from the Application Monitor to the ActiveMatrix Service Grid - Container Edition applications and refreshes the status of the AMXCE node and entities running on the node. This feature can be used to fix the inconsistent status of the Application Monitor and runtime. This reconnect operation applies to the selected node and the entities running on the selected node only.
The following fields are displayed for a Kubernetes setup:	
Pod	Name of the pod. Pods are the smallest deployable units of computing that you can create and manage in Kubernetes. It is a group of one or more containers, with shared storage or network resources, and a specification for how to run the containers.
Namespace	Name of the namespace used in the configuration (`.yaml`) file.

Column	Description
	Kubernetes supports multiple virtual clusters backed by the same physical cluster. These virtual clusters are called namespaces. Namespaces provide a scope for names. • Names of resources need to be unique within a namespace but not across namespaces. • Namespaces cannot be nested inside one another and each Kubernetes resource can only be in one namespace. • Namespaces are a way to divide cluster resources between multiple users.
Deployment	Name of the deployment in which application is deployed. A deployment provides declarative updates for the pods ReplicaSets. You can describe a desired state in a deployment and the Deployment Controller changes the actual state to the desired state at a controlled rate. You can define deployments to create new ReplicaSets or to remove existing deployments and adopt all their resources with new deployments.

You can perform the following tasks:

Task	Description
View Unreachable Nodes	When you start an Application Monitor TEA agent, all the AMXCE nodes are started. Sometimes, a JMX connection cannot be created for a specific AMXCE node. To view the nodes that have not started, click the **Unreachable Node** icon in the upper-right corner. The count of unreachable nodes is also displayed in the upper-right corner. Unreachable Node icon is displayed only when there is one or more unreachable nodes in the cluster, otherwise this icon is not displayed. The AMXCE node is unreachable only if the pod is running but the agent is unable to create a JMX connection.

<table>
<tr><th>Task</th><th>Description</th></tr>
<tr><td></td><td>

TIBCO Enterprise Administrator
ServiceGrid
admin
TIBCO AMSG CE Application Monitor
1.0.0
Unreachable Node
Previous
1
Next

AMXCE Nodes

Last Updated: 21:44:46

AMXCE Nodes (1380/1380)

Show 10 entries

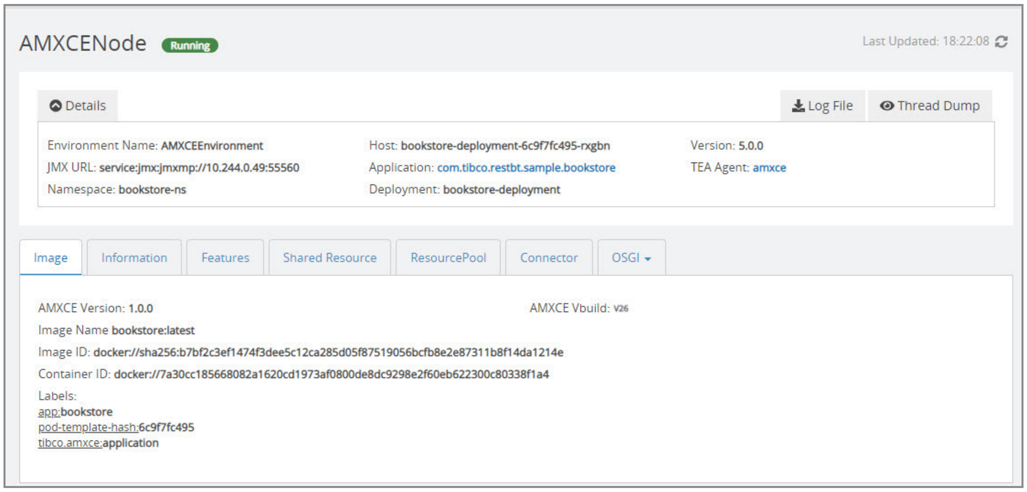
Search:

<table>
<tr><th>Name</th><th>JMX URL</th><th>Deployment</th><th>Namespace</th><th>Desired State</th></tr>
<tr><td>mtom-3-75c6b7648b-bp588</td><td>service:jmx:jmxmp://192.168.76.72:55560</td><td>mtom-3</td><td>mtom-3</td><td>12/12</td></tr>
</table>

Showing 1 to 1 of 1 entries
</td></tr>
<tr><td>View Node Details</td><td>
On the <b>AMXCE Nodes</b> page, select a node to view its details. The <b>Details</b> pane on the node details page displays general information about the node. You can collapse and expand this pane.

Click the application name to get more information about the user application running on that node. Click the TEA agent name to view its details. Host is the name of the container in which the node is running.

The node details page is displayed as shown in the following image:



AMXCENode

Last Updated: 18:22:08

Environment Name: AMXCEEnvironment
JMX URL: service:jmx:jmxmp://10.244.0.49:55560
Namespace: bookstore-ns
Host: bookstore-deployment-6c9f7fc495-rxgbn
Application: com.tibco.restbt.sample.bookstore
Deployment: bookstore-deployment
Version: 5.0.0
TEA Agent: amxce

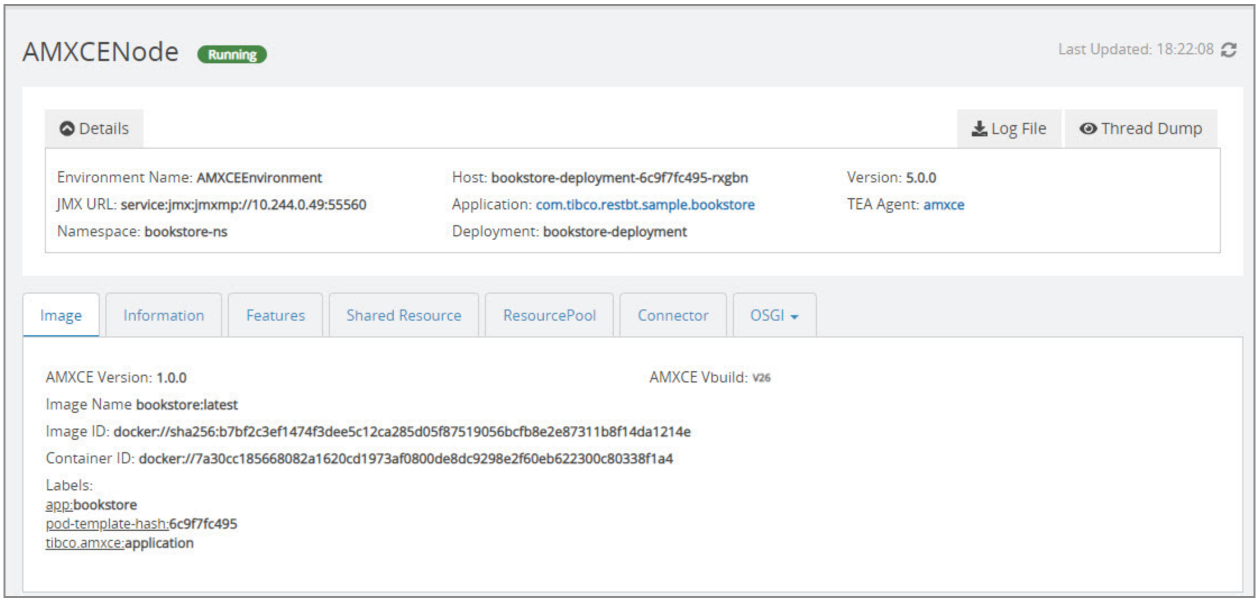
AMXCE Version: 1.0.0
AMXCE Vbuild: v26
Image Name bookstore:latest
Image ID: docker://sha256:b7bf2c3ef1474f3dee5c12ca285d05f87519056bcfb8e2e87311b8f14da1214e
Container ID: docker://7a30cc185668082a1620cd1973af0800de8dc9298e2f60eb622300c80338f1a4
Labels:
app:bookstore
pod-template-hash:6c9f7fc495
tibco.amxce:application
</td></tr>
<tr><td>Download</td><td>To download logs for a node, on the node details page, click <b>Log File</b>. The</td></tr>
</table>

Task	Description
Node Logs	name of the downloaded `.zip` file is `HostName_NodeName_TimeStamp.logs.zip`.
View Thread Dump	Thread Dump captures information about threads for the node. To view Thread Dump, on the node details page, click the **Thread Dump** button. You can view information such as daemon thread count, thread count, peak thread count, and total started thread count.

Image Tab

The **Image** tab lists all the image related information from the image manifest created as shown in the image.

Note: The information for **Labels** is set by Kubernetes.



AMXCENode

Last Updated: 18:22:08

Environment Name: AMXCEEnvironment

JMX URL: service:jmx:jmxmp://10.244.0.49:55560

Namespace: bookstore-ns

Host: bookstore-deployment-6c9f7fc495-rxgbn

Application: com.tibco.restbt.sample.bookstore

Deployment: bookstore-deployment

Version: 5.0.0

TEA Agent: amxce

AMXCE Version: 1.0.0

AMXCE Vbuild: V26

Image Name bookstore:latest

Image ID: docker://sha256:b7bf2c3ef1474f3dee5c12ca285d05f87519056bcfb8e2e87311b8f14da1214e

Container ID: docker://7a30cc185668082a1620cd1973af0800de8dc9298e2f60eb622300c80338f1a4

Labels:
app:bookstore
pod-template-hash:6c9f7fc495
tibco.amxce:application

Viewing and Downloading Node Health Check Information

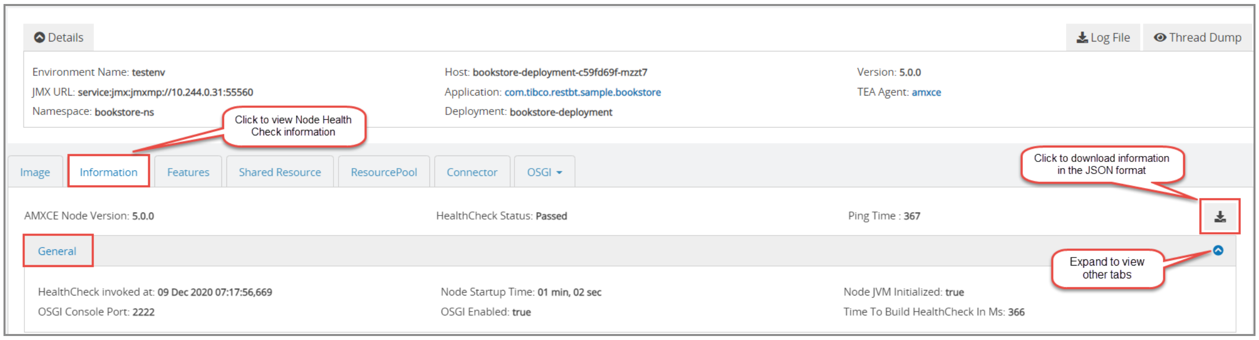
The following Health Check information about a runtime node is provided:

- General information such as the Node Startup Time and the Time to build the

HealthCheck and so on.

- JVM Information such as JVM arguments for AMXCE Node, thread information and so on.
- JVM memory usage with details of allocated, free, maximum, and used values.
- Machine-specific information. It provides details such as machine name, Username, CPU count, operating system, memory, and swap space.
- TIBCO `CONFIG_HOME` information.

To view Health Check information, on the node details page, go to the Information tab. Expand the sections on the page to view information.



Environment Name: testenv
JMX URL: service:jmx:jmxmp://10.244.0.31:55560
Namespace: bookstore-ns
Host: bookstore-deployment-c59fd69f-mzzt7
Application: com.tibco.restbt.sample.bookstore
Deployment: bookstore-deployment
Version: 5.0.0
TEA Agent: amxce

Click to view Node Health Check information

Click to download information in the JSON format

AMXCE Node Version: 5.0.0
HealthCheck Status: Passed
Ping Time : 367

Expand to view other tabs

HealthCheck invoked at: 09 Dec 2020 07:17:56,669
OSGI Console Port: 2222
Node Startup Time: 01 min, 02 sec
OSGI Enabled: true
Node JVM Initialized: true
Time To Build HealthCheck In Ms: 366

Note: Each AMXCE node has the following JVM arguments set by default:

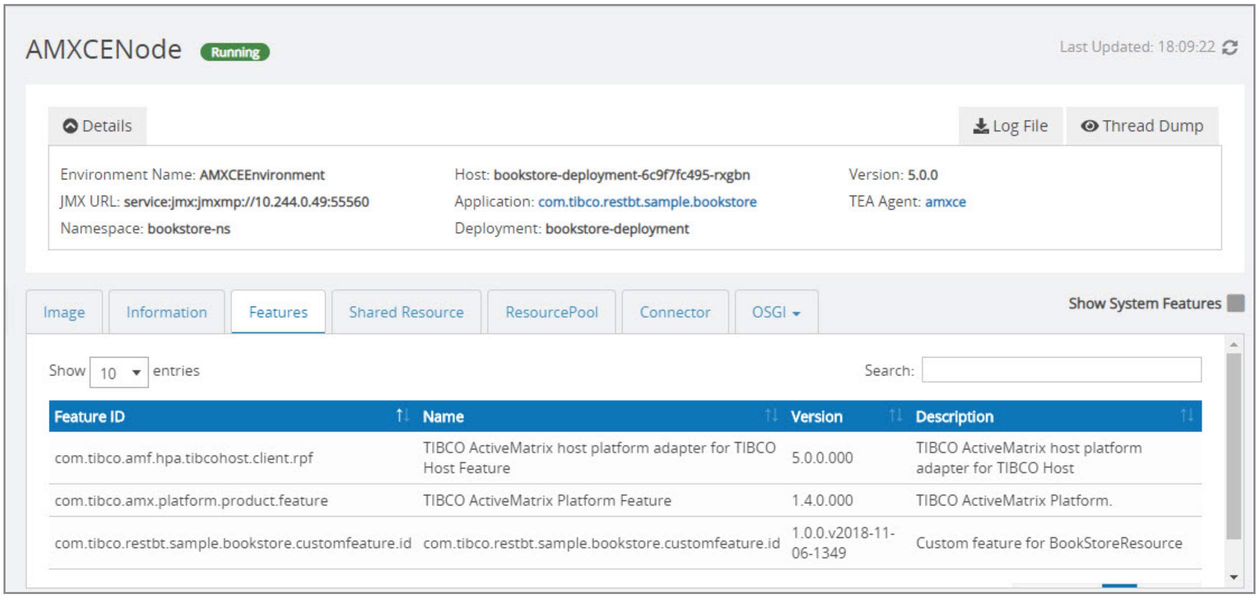
- `com.tibco.noMessagingBus=true`
- `com.tibco.amx.rad=true`

To Download Health Check Information (JSON)

To download the node health check information data as a JSON file, click **Download JSON** in the upper-right corner as highlighted in the above image. This JSON file contains all the data displayed on the UI. The default name of the file that gets downloaded is `Node_Name_TIME_STAMP.json`.

Node Features Reference

The **Features** tab lists all the features enabled on the AMXCE node. To view the System features, select the **Show System Features** check box in the upper-right corner. For more information about the features, see ActiveMatrix Service Grid documentation.

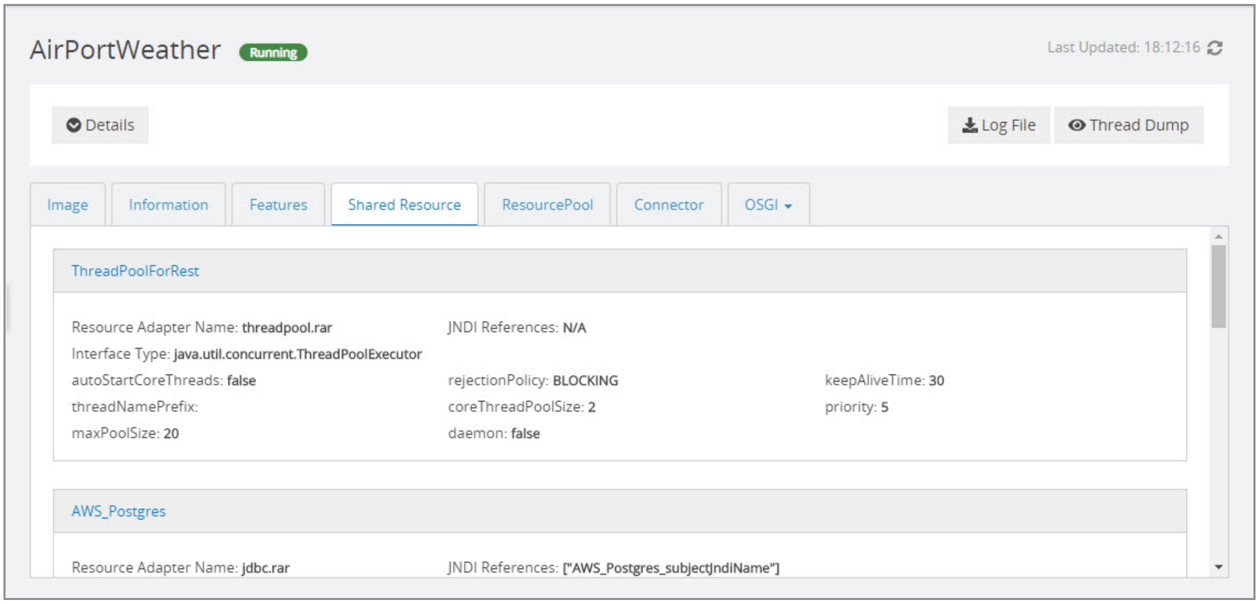


Environment Name: AMXCEEnvironment
JMX URL: service:jmx:jmxmp://10.244.0.49:55560
Namespace: bookstore-ns
Host: bookstore-deployment-6c9f7fc495-rxgbn
Application: com.tibco.restbt.sample.bookstore
Deployment: bookstore-deployment
Version: 5.0.0
TEA Agent: amxce

Feature ID	Name	Version	Description
com.tibco.amf.hpa.tibcohost.client.rpf	TIBCO ActiveMatrix host platform adapter for TIBCO Host Feature	5.0.0.000	TIBCO ActiveMatrix host platform adapter for TIBCO Host
com.tibco.amx.platform.product.feature	TIBCO ActiveMatrix Platform Feature	1.4.0.000	TIBCO ActiveMatrix Platform.
com.tibco.restbt.sample.bookstore.customfeature.id	com.tibco.restbt.sample.bookstore.customfeature.id	1.0.0.v2018-11-06-1349	Custom feature for BookStoreResource

Node Shared Resources Reference

The **Shared Resource** tab lists all the managed resources used by the node. The keystore password is not displayed for security.



ThreadPoolForRest

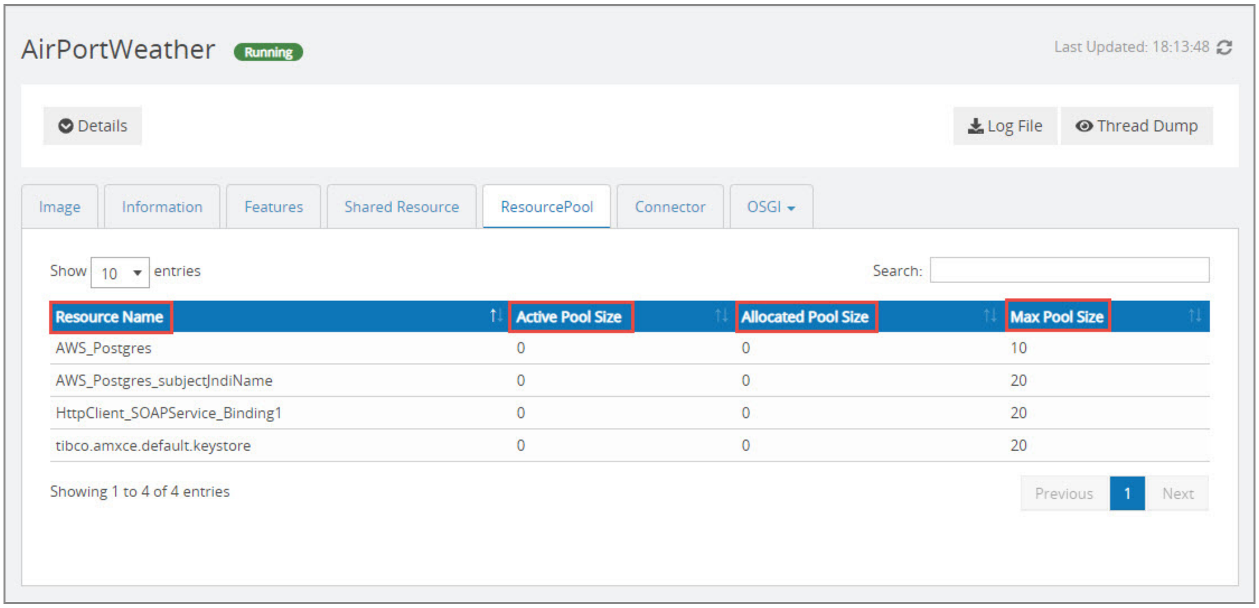
Resource Adapter Name: threadpool.rar
JNDI References: N/A
Interface Type: java.util.concurrent.ThreadPoolExecutor
autoStartCoreThreads: false
rejectionPolicy: BLOCKING
keepAliveTime: 30
threadNamePrefix:
coreThreadPoolSize: 2
priority: 5
maxPoolSize: 20
daemon: false

AWS_Postgres

Resource Adapter Name: jdbc.rar
JNDI References: ["AWS_Postgres_subjectJndiName"]

Viewing Resource Pool Information

The **ResourcePool** tab displays all the shared resource pools and the related information as shown in the following image:

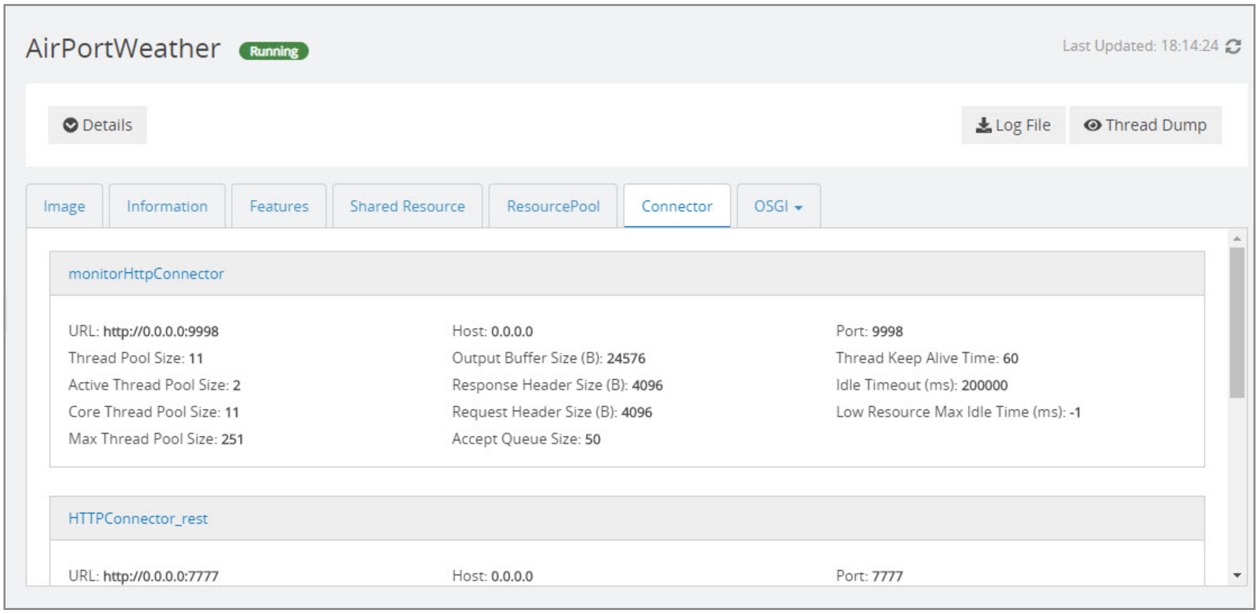


Resource Name	Active Pool Size	Allocated Pool Size	Max Pool Size
AWS_Postgres	0	0	10
AWS_Postgres_subjectJndiName	0	0	20
HttpClient_SOAPService_Binding1	0	0	20
tibco.amxce.default.keystore	0	0	20

Showing 1 to 4 of 4 entries

Viewing HTTP Connectors Information

The **Connector** tab lists the deployed or used connectors from the node.



monitorHttpConnector

URL: http://0.0.0.0:9998
Thread Pool Size: 11
Active Thread Pool Size: 2
Core Thread Pool Size: 11
Max Thread Pool Size: 251

Host: 0.0.0.0
Output Buffer Size (B): 24576
Response Header Size (B): 4096
Request Header Size (B): 4096
Accept Queue Size: 50

Port: 9998
Thread Keep Alive Time: 60
Idle Timeout (ms): 200000
Low Resource Max Idle Time (ms): -1

HTTPConnector_rest

URL: http://0.0.0.0:7777
Host: 0.0.0.0
Port: 7777

Viewing Node Specific OSGi Information

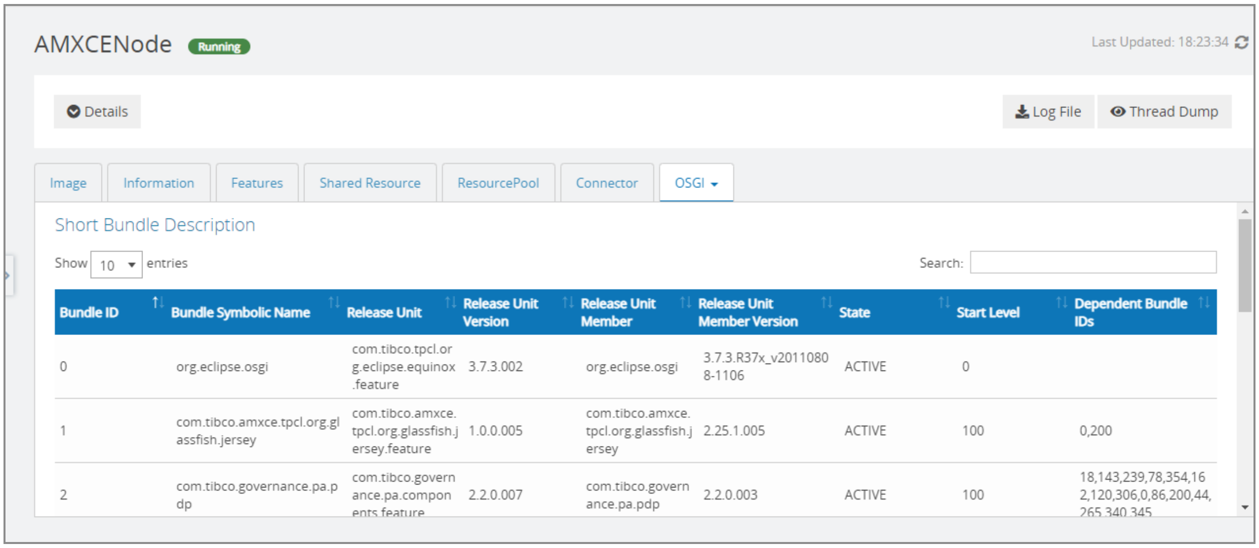
The **OSGi** tab provides node-specific OSGi information that can be used for troubleshooting and diagnostic purposes without requiring special OSGi configuration and node restart. The tool provides a visual representation of the information normally provided by the OSGi console commands, which require configuring the OSGi console port and node restart. The OSGi tab has a dropdown list, which corresponds to the "Short Bundle Description", "Endpoint Details", "Implementation Component Details", and "Export Bundle Search Utility" pages.

Using this tool, you can:

- Gather OSGi details of deployed nodes and generate a tabular report. The tool provides details of nodes, bundles, endpoints, and implementation components.
- Search for specific values in the current report.
- Refresh or reload data in the current report.
- Export data from a report to a JSON file (`.json`) file.
- Search for specific packages and export the details to a `.json` file

Short Bundle Description

The Bundle page shows details about the currently installed bundles. You can use the search function to get a list of individual packages and their importers and exporters. To search for values in the current diagnostic report or table, enter the text in the search field at the right side. The filtered results are displayed in the current table.



Short Bundle Description

Show 10 entries

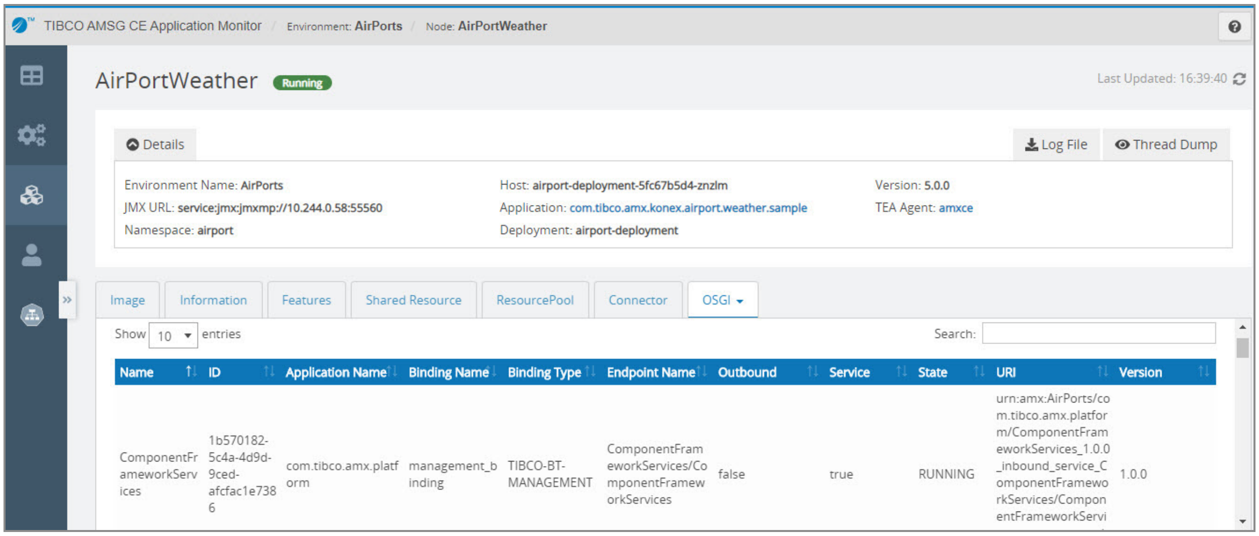
Search:

Bundle ID	Bundle Symbolic Name	Release Unit	Release Unit Version	Release Unit Member	Release Unit Member Version	State	Start Level	Dependent Bundle IDs
0	org.eclipse.osgi	com.tibco.tpcl.org.eclipse.equinox.feature	3.7.3.002	org.eclipse.osgi	3.7.3.R37x_v20110808-1106	ACTIVE	0	
1	com.tibco.amxce.tpcl.org.glassfish.jersey	com.tibco.amxce.tpcl.org.glassfish.jersey.feature	1.0.0.005	com.tibco.amxce.tpcl.org.glassfish.jersey	2.25.1.005	ACTIVE	100	0,200
2	com.tibco.governance.pa.pdp	com.tibco.governance.pa.components.feature	2.2.0.007	com.tibco.governance.pa.pdp	2.2.0.003	ACTIVE	100	18,143,239,78,354,162,120,306,0,86,200,44,265,340,345

The Bundle page provides the following information:

Item	Description
Bundle ID	Unique identification number given by the OSGi framework to each bundle.
Bundle Symbolic Name	Unique identifier of a bundle (MANIFEST).
Release Unit	Identifier of the bundle application template.
Release Unit Version	Version of the bundle application template.
Release Unit Member	Identifier of the bundle application custom feature.
Release Unit Member Version	Version of the bundle application custom feature.
Start Level	The order in which the bundles start.
State	State of the bundle, ACTIVE or RESOLVED.
Dependent Bundle IDs	Unique identification numbers given by the OSGi framework to the dependent bundles.

Endpoint Details



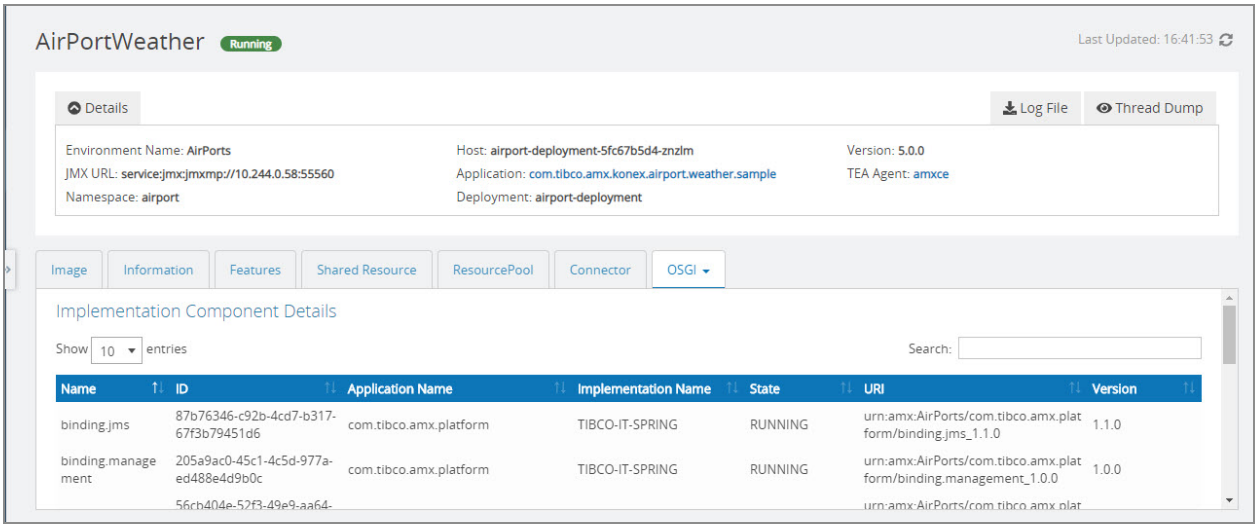
Environment Name: AirPorts
JMX URL: service:jmx:jmxmp://10.244.0.58:55560
Namespace: airport
Host: airport-deployment-5fc67b5d4-znzlm
Application: com.tibco.amx.konex.airport.weather.sample
Deployment: airport-deployment
Version: 5.0.0
TEA Agent: amxce

Name	ID	Application Name	Binding Name	Binding Type	Endpoint Name	Outbound	Service	State	URI	Version
ComponentFrameworkServices	1b570182-5c4a-4d9d-9ced-afcfac1e7386	com.tibco.amx.platform	management_binding	TIBCO-BT-MANAGEMENT	ComponentFrameworkServices/ComponentFrameworkServices	false	true	RUNNING	urn:amx:AirPorts/com.tibco.amx.platform/ComponentFrameworkServices_1.0.0_inbound_service_ComponentFrameworkServices/ComponentFrameworkServi	1.0.0

The **Endpoint** tab provides the following information:

Item	Description
Name	Endpoint application component name.
ID	Endpoint application component unique ID.
Application Name	Endpoint application name.
Binding Name	Endpoint application binding name.
Binding Type	Endpoint application type.
Endpoint Name	Endpoint name.
Outbound	Is promoted reference.
Service	Is promoted service.
State	Endpoint application component status.
URI	Universal Resource Identification (URI) for that particular endpoint.
Version	Endpoint application template version.

Implementation Component Details



Environment Name: AirPorts
JMX URL: service:jmx:jmxmp://10.244.0.58:55560
Namespace: airport

Host: airport-deployment-5fc67b5d4-znzlm
Application: com.tibco.amx.konex.airport.weather.sample
Deployment: airport-deployment

Version: 5.0.0
TEA Agent: amxce

Implementation Component Details

Name	ID	Application Name	Implementation Name	State	URI	Version
binding.jms	87b76346-c92b-4cd7-b317-67f3b79451d6	com.tibco.amx.platform	TIBCO-IT-SPRING	RUNNING	urn:amx:AirPorts/com.tibco.amx.platform/binding.jms_1.1.0	1.1.0
binding.management	205a9ac0-45c1-4c5d-977a-ed488e4d9b0c	com.tibco.amx.platform	TIBCO-IT-SPRING	RUNNING	urn:amx:AirPorts/com.tibco.amx.platform/binding.management_1.0.0	1.0.0
	56cb404e-52f3-49e9-aa64-				urn:amx:AirPorts/com.tibco.amx.plat	

The **Implementation Component Details** page provides the following information:

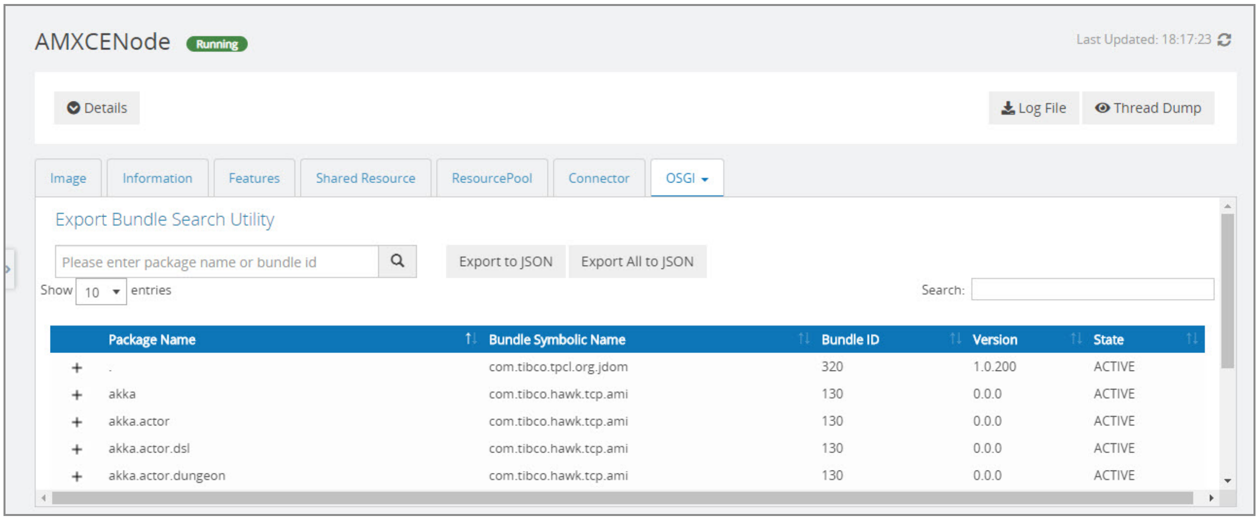
Item	Description
Name	Name of the implementation component.
ID	Unique ID of the implementation component unique.
Application Name	Name of the application on the implementation component.
Implementation Type	Type of implementation component (such as Java and Mediation).
State	Status of the implementation component.
URI	URI for the specific implementation component.
Version	Application template version of the implementation component.

Searching for Packages

From the **Export Bundle Search Utility** page, you can search for packages based on the package name or bundle ID. This page provides information about the hierarchy of the package. For example:

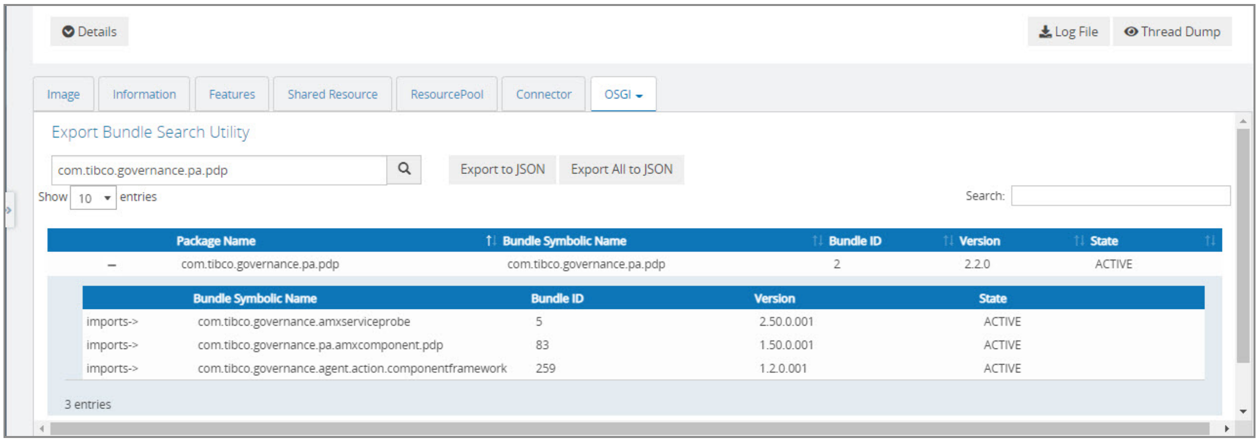
- The bundle exporting the searched package.
- Bundles that are importing the searched package.

To export the search results (details of the package and imported data) to a `.json` file, click **Export to JSON**. To export the hierarchy of all the OSGi packages to a `.json` file, click **Export All to JSON**.



	Package Name	Bundle Symbolic Name	Bundle ID	Version	State
+	.	com.tibco.tpcl.org.jdom	320	1.0.200	ACTIVE
+	akka	com.tibco.hawk.tcp.ami	130	0.0.0	ACTIVE
+	akka.actor	com.tibco.hawk.tcp.ami	130	0.0.0	ACTIVE
+	akka.actor.dsl	com.tibco.hawk.tcp.ami	130	0.0.0	ACTIVE
+	akka.actor.dungeon	com.tibco.hawk.tcp.ami	130	0.0.0	ACTIVE

In the search field, enter the package name or bundle ID and click 🔍. The details of the package are displayed as shown below:



com.tibco.governance.pa.pdp

	Package Name	Bundle Symbolic Name	Bundle ID	Version	State
–	com.tibco.governance.pa.pdp	com.tibco.governance.pa.pdp	2	2.2.0	ACTIVE

	Bundle Symbolic Name	Bundle ID	Version	State
imports->	com.tibco.governance.amxserviceprobe	5	2.50.0.001	ACTIVE
imports->	com.tibco.governance.pa.amxcomponent.pdp	83	1.50.0.001	ACTIVE
imports->	com.tibco.governance.agent.action.componentframework	259	1.2.0.001	ACTIVE

3 entries

Application Monitor Agent

Application Monitor uses the Application Monitor agent to monitor ActiveMatrix Service Grid - Container Edition nodes and applications deployed in an environment. Application Monitor agent is the back-end server process that provides a monitoring functionality for an ActiveMatrix Service Grid - Container Edition application and cluster. The agent communicates with the TIBCO Enterprise Administrator server for UI interactions and communicates with the ActiveMatrix Service Grid - Container Edition application containers by using JMX.

Agent Metadata

Every agent has its own metadata. This metadata includes agent type and agent version. In ActiveMatrix Service Grid - Container Edition, the agent type is AMXCE and version is 1.0.0.

> **Note:** If there is any change regarding the metadata of an agent, then you must unregister the agent and register again.

Agent Properties

Agent properties are used when auto-registering an agent. You can configure these properties in the Kubernetes deployment configuration file. Sample file is provided at `amsgce-runtime-<version>\samples\kubernetes\Application_Monitor`. For more information, see *TIBCO ActiveMatrix® Service Grid - Container Edition Cloud Deployment*.

Agent Configuration Details

The configuration details such as (Name, Type, Version and so on) are set for every agent before registering on the TEA server. The configuration details are displayed on the console when registering an agent. You can view this information in the agent logs as well.

User Permissions

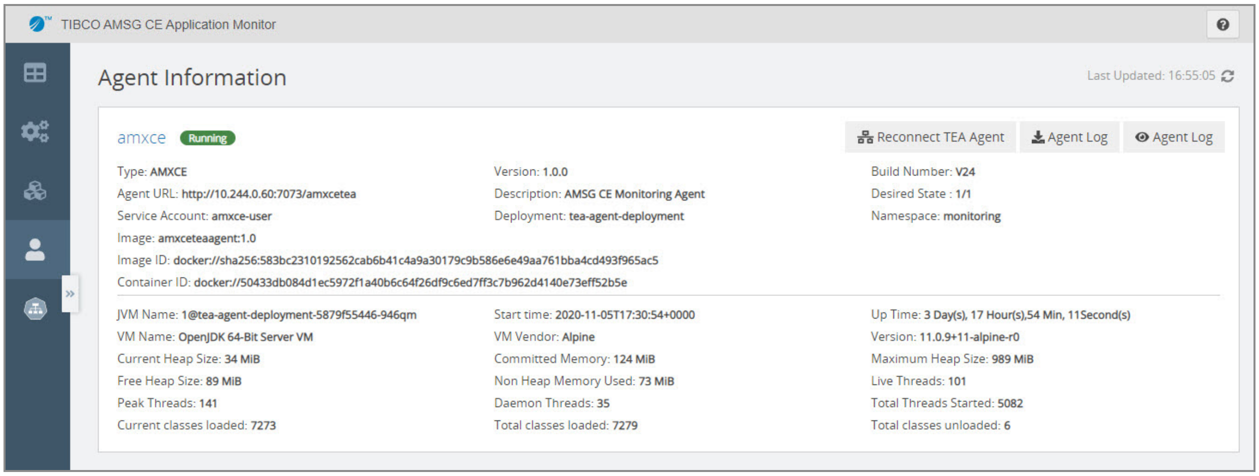
Every TEA agent has some permissions assigned to it. By default, the Application Monitor agent has read-write permissions. On the TEA, Administrator can assign users to different groups and roles. Each user role has some permissions enabled for it. The user with the assigned role can perform all the operations that are enabled for that role. For more information, see TIBCO Enterprise Administrator documentation.

Viewing Agent Information

The ActiveMatrix Service Grid - Container Edition installation package includes the Application Monitor agent that can be used to monitor applications.

The Application Monitor agent is the back-end server process that provides a monitoring functionality for an ActiveMatrix Service Grid - Container Edition application and cluster. The agent communicates with the TIBCO Enterprise Administrator server for UI interactions and communicates with the ActiveMatrix Service Grid - Container Edition application containers by using JMX.

To view the agent information, click **Agent Information** on the sidebar.



Type: AMXCE
Version: 1.0.0
Build Number: V24
Agent URL: http://10.244.0.60:7073/amxcetea
Description: AMSG CE Monitoring Agent
Desired State : 1/1
Service Account: amxce-user
Deployment: tea-agent-deployment
Namespace: monitoring
Image: amxceteaagent:1.0
Image ID: docker://sha256:583bc2310192562cab6b41c4a9a30179c9b586e6e49aa761bba4cd493f965ac5
Container ID: docker://50433db084d1ec5972f1a40b6c64f26df9c6ed7ff3c7b962d4140e73eff52b5e

JVM Name: 1@tea-agent-deployment-5879f55446-946qm
Start time: 2020-11-05T17:30:54+0000
Up Time: 3 Day(s), 17 Hour(s),54 Min, 11Second(s)
VM Name: OpenJDK 64-Bit Server VM
VM Vendor: Alpine
Version: 11.0.9+11-alpine-r0
Current Heap Size: 34 MiB
Committed Memory: 124 MiB
Maximum Heap Size: 989 MiB
Free Heap Size: 89 MiB
Non Heap Memory Used: 73 MiB
Live Threads: 101
Peak Threads: 141
Daemon Threads: 35
Total Threads Started: 5082
Current classes loaded: 7273
Total classes loaded: 7279
Total classes unloaded: 6

The following information is displayed on the **Agent Information** page.

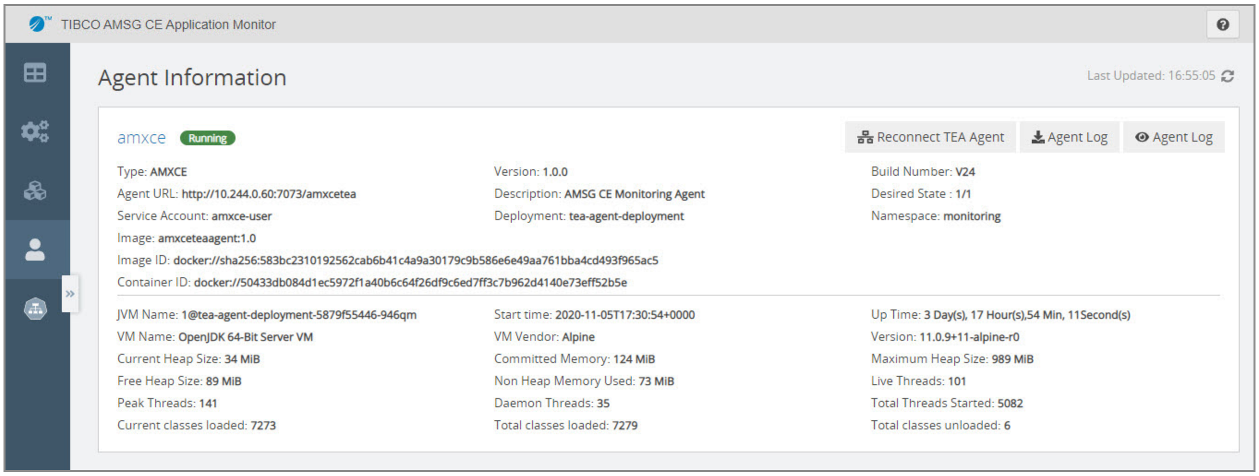
Field	Description
Name	Name of the Application Monitor agent configured.
Status	Agent runtime status.
Type	Agent type (that is, AMXCE)
Version	Agent version (that is, 1.0.0)
Build No	ActiveMatrix Service Grid - Container Edition build number.
Agent URL	URL must be in the following format: `http://<ip:port>/amxcetea` Application Monitor TEA agent self URL. Agent URL is required to manually register agent with TIBCO Enterprise Administrator server. **Default port:** 7073
Description	Description of the Application Monitor agent.

This page also displays Kubernetes deployment information such as Namespace, Deployment name, Desired State, Image name, Image ID, and Container ID.

This page also includes JVM information such as JVM Name, Start time, Up time and so on.

Viewing or Downloading Agent Logs

You can view or download the logs for an agent on the **Agent Information** page. Click the **Agent Log** button to download logs or you can also view logs in a browser by clicking the **Agent Log** button.



Type: AMXCE
Agent URL: http://10.244.0.60:7073/amxcetea
Service Account: amxce-user
Image: amxceteaagent:1.0
Image ID: docker://sha256:583bc2310192562cab6b41c4a9a30179c9b586e6e49aa761bba4cd493f965ac5
Container ID: docker://50433db084d1ec5972f1a40b6c64f26df9c6ed7ff3c7b962d4140e73eff52b5e
Version: 1.0.0
Description: AMSG CE Monitoring Agent
Deployment: tea-agent-deployment
Build Number: V24
Desired State : 1/1
Namespace: monitoring

JVM Name: 1@tea-agent-deployment-5879f55446-946qm
VM Name: OpenJDK 64-Bit Server VM
Current Heap Size: 34 MiB
Free Heap Size: 89 MiB
Peak Threads: 141
Current classes loaded: 7273
Start time: 2020-11-05T17:30:54+0000
VM Vendor: Alpine
Committed Memory: 124 MiB
Non Heap Memory Used: 73 MiB
Daemon Threads: 35
Total classes loaded: 7279
Up Time: 3 Day(s), 17 Hour(s),54 Min, 11Second(s)
Version: 11.0.9+11-alpine-r0
Maximum Heap Size: 989 MiB
Live Threads: 101
Total Threads Started: 5082
Total classes unloaded: 6

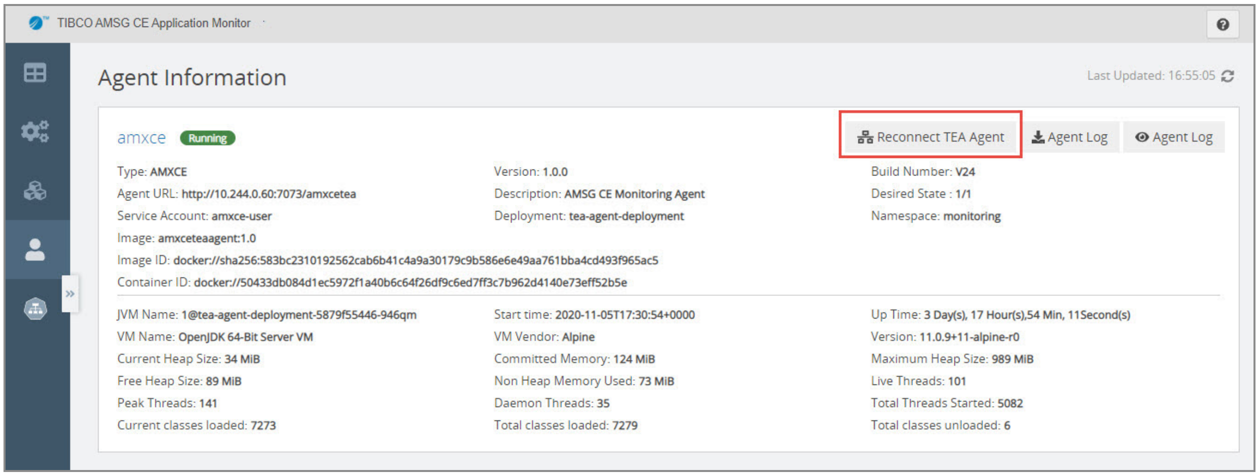
The default name of the log file is `<Agent Name>-teaagent_TimeStamp.log`.

Reconnecting TEA Agent

This feature is used to resolve state inconsistencies between the Application Monitor and runtime, for example: whenever the data displayed in Application Monitor is not in sync with the configurations provided or incorrect data is displayed for the field. Reconnecting to the TEA agent involves re-creating all the JMX connections from the Application Monitor to the ActiveMatrix Service Grid - Container Edition applications and refreshing the status of all entities.

Procedure

1. In the Application Monitor, click **Agent Information**.
2. Click **Reconnect TEA Agent**. The status of nodes and entities running on the nodes is re-emitted.



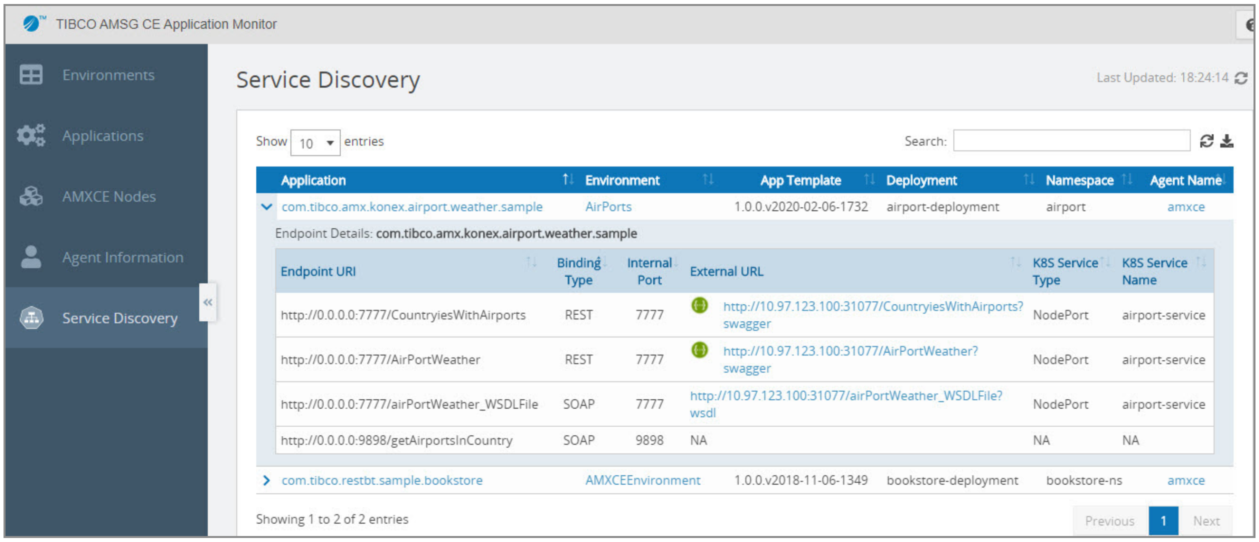
Type: AMXCE
Agent URL: http://10.244.0.60:7073/amxcetea
Service Account: amxce-user
Image: amxceteaagent:1.0
Image ID: docker://sha256:583bc2310192562cab6b41c4a9a30179c9b586e6e49aa761bba4cd493f965ac5
Container ID: docker://50433db084d1ec5972f1a40b6c64f26df9c6ed7ff3c7b962d4140e73eff52b5e
Version: 1.0.0
Description: AMSG CE Monitoring Agent
Deployment: tea-agent-deployment
Build Number: V24
Desired State : 1/1
Namespace: monitoring

JVM Name: 1@tea-agent-deployment-5879f55446-946qm
VM Name: OpenJDK 64-Bit Server VM
Current Heap Size: 34 MiB
Free Heap Size: 89 MiB
Peak Threads: 141
Current classes loaded: 7273
Start time: 2020-11-05T17:30:54+0000
VM Vendor: Alpine
Committed Memory: 124 MiB
Non Heap Memory Used: 73 MiB
Daemon Threads: 35
Total classes loaded: 7279
Up Time: 3 Day(s), 17 Hour(s),54 Min, 11Second(s)
Version: 11.0.9+11-alpine-r0
Maximum Heap Size: 989 MiB
Live Threads: 101
Total Threads Started: 5082
Total classes unloaded: 6

Service Discovery

The **Service Discovery** page lists the information related to the applications and their exposed endpoints. The **Service Discovery** tab displays all the endpoints details on a single page.

Expand an application to get more information about application endpoints.



Application	Environment	App Template	Deployment	Namespace	Agent Name
com.tibco.amx.konex.airport.weather.sample	AirPorts	1.0.0.v2020-02-06-1732	airport-deployment	airport	amxce

Endpoint Details: com.tibco.amx.konex.airport.weather.sample

Endpoint URI	Binding Type	Internal Port	External URL	K8S Service Type	K8S Service Name
http://0.0.0.0:7777/CountryiesWithAirports	REST	7777	http://10.97.123.100:31077/CountryiesWithAirports?swagger	NodePort	airport-service
http://0.0.0.0:7777/AirPortWeather	REST	7777	http://10.97.123.100:31077/AirPortWeather?swagger	NodePort	airport-service
http://0.0.0.0:7777/airPortWeather_WSDLFile	SOAP	7777	http://10.97.123.100:31077/airPortWeather_WSDLFile?wsdl	NodePort	airport-service
http://0.0.0.0:9898/getAirportsInCountry	SOAP	9898	NA	NA	NA

Application	Environment	App Template	Deployment	Namespace	Agent Name
com.tibco.restbt.sample.bookstore	AMXCEEnvironment	1.0.0.v2018-11-06-1349	bookstore-deployment	bookstore-ns	amxce

Showing 1 to 2 of 2 entries

About Endpoints in an application

Name	Description
Endpoint URI	The endpoint URI.
Binding Type	The binding type can be SOAP or REST.
Internal Port	The container port used.
External URL	Click the URL to view the respective WSDL or JSON structure based on the binding type. You can also access Swagger UI by clicking the Swagger UI icon {…}. For more information, see Accessing Swagger UI from the Application Monitor.
K8S Service Type	The Kubernetes service type can be NodePort, LoadBalancer, or ClusterIP. **Note:** If you have not mapped any Kubernetes service, then the Kubernetes Service and Kubernetes Service Type fields have values as 'NA'.
K8S Service Name	Kubernetes service name.

Note: SOAP/JMS Binding type is not supported.

Download JSON File

To download the application endpoints data as a JSON file, click **Download JSON** ⤓ in the upper-right corner. This JSON file contains all the data displayed on the UI. The default name of the file that gets downloaded is `Service_TIME_STAMP.json`.

Refresh from Cache

You can synchronize the data on the **Service Discovery** tab with the runtime by clicking **Refresh from Cache** ⟳ in the upper-right corner. If you make any changes in the runtime configurations, for example, changing the internal port numbers or adding or removing Kubernetes service, then you can synchronize the data on the UI by using this feature.

Visualizing and Testing REST Service Bindings Using Swagger

Swagger scans the application code and exposes the documentation on the URL. You can consume this URL (a JSON document) to understand the capabilities of the REST service without accessing the actual source code and documentation.

For more information on Swagger, see https://swagger.io/.

Swagger is built around OpenAPI Specification. Currently, it uses Swagger Specification version 2.0 and Application API version 1.0. The specification defines a set of files required to describe the API. The files can be written in YAML or JSON format. These files can then be used by the Swagger UI to display the API. For more information on the OpenAPI Specification, see https://swagger.io/specification/.

Using Swagger UI integrated with the Application Monitor, you can:

- View all REST service endpoints and operations on each endpoint implemented by the REST service.
- Examine the inputs and outputs for each operation in JSON format with detailed schema.
- Allow end developers to effortlessly interact and try out every single operation that the API exposes for easy consumption. Specify JSON, BJSON, or XML as the content type for a Request and Response for each operation.
- Invoke an operation and receive a live response for the input specified.
- Generate and download a Swagger.json file.

Accessing Swagger UI from the Application Monitor

In TIBCO ActiveMatrix Service Grid, you can generate the Swagger from (REST binding type) endpoints 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 the Swagger UI.

Before you begin

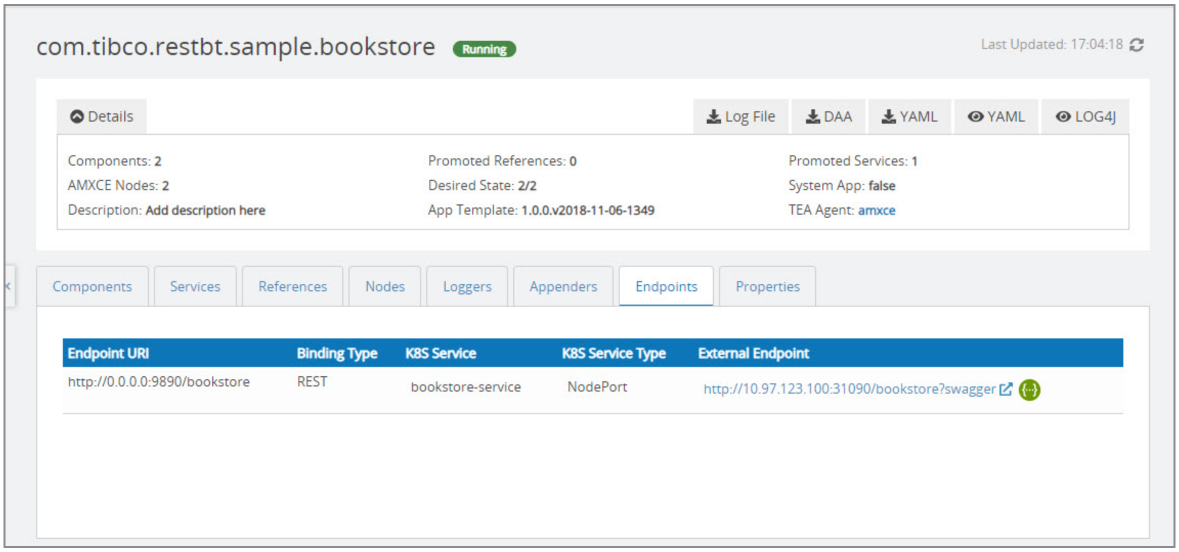
- You must have Swagger UI hosted externally.
- You must provide the link to the Swagger UI hosted externally by using the

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 sidebar, click **Applications**. A list of all the applications is displayed.
2. Select the required application.
3. Click the **Endpoints** tab.

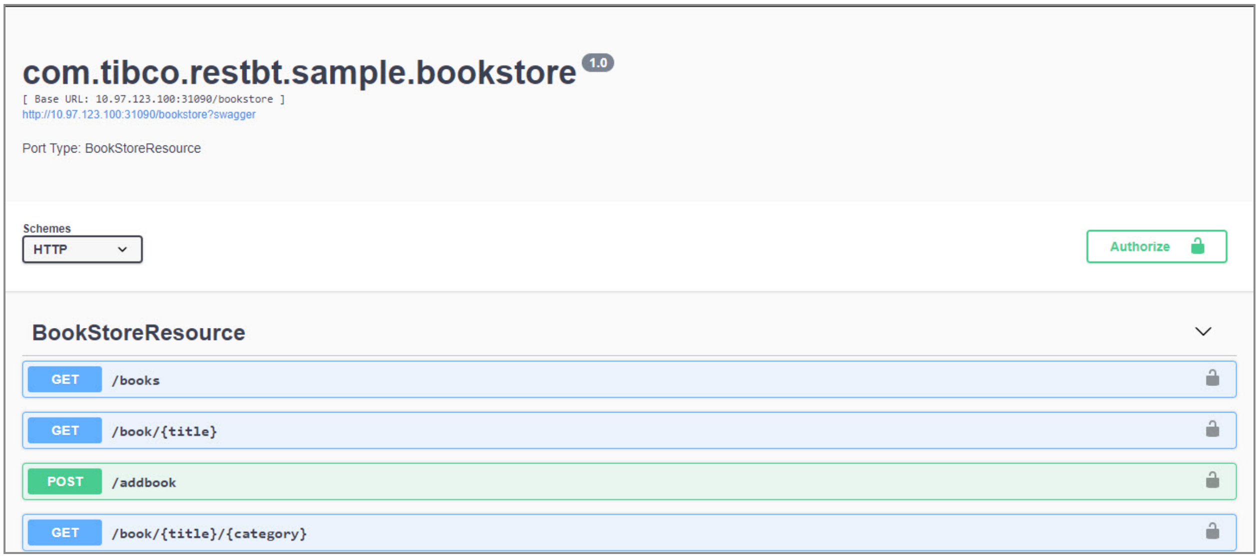


Components: 2
AMXCE Nodes: 2
Description: Add description here
Promoted References: 0
Desired State: 2/2
App Template: 1.0.0.v2018-11-06-1349
Promoted Services: 1
System App: false
TEA Agent: amxce

Endpoint URI	Binding Type	K8S Service	K8S Service Type	External Endpoint
http://0.0.0.0:9890/bookstore	REST	bookstore-service	NodePort	http://10.97.123.100:31090/bookstore?swagger

4. Click the Swagger UI icon next to REST service binding.

The Swagger UI is generated for the selected REST binding.



com.tibco.restbt.sample.bookstore 1.0

[Base URL: 10.97.123.100:31090/bookstore]
http://10.97.123.100:31090/bookstore?swagger

Port Type: BookStoreResource

Schemes
HTTP

BookStoreResource

GET /books
GET /book/{title}
POST /addbook
GET /book/{title}/{category}

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

Troubleshooting

Reconnect TEA agent

Whenever the data displayed in Application Monitor is not in sync with the configurations provided or incorrect data is displayed for the field, you can use the **Reconnect TEA Agent** option. Reconnecting TEA agent creates new JMX connections to the entities and re-emits the status of all entities. For more information, see Reconnecting TEA Agent.

View Unreachable Nodes

While starting an Application Monitor TEA agent, all the AMXCE nodes are started. Sometimes, a JMX connection cannot be created for a specific AMXCE node. To view the nodes that have not started, click the **Unreachable Node** icon in the upper-right corner on the **AMXCE Nodes** page. The count of unreachable nodes is also displayed in the upper-right corner. The AMXCE node is unreachable only if the pod is running but the agent is unable to create a JMX connection. You can find pod id and namespace from the **Unreachable Node** tab. You can navigate to kubectl and check that pod status, and logs to troubleshoot the issue.

Entity Level Logs

To troubleshoot the entity-specific issues, you can use loggers. By downloading the entity-specific logs, you can debug the errors. By changing or adding different logs and levels, you can generate detailed log information for the entity. You can view or download logs for agents, AMXCE nodes, and applications. You can use Docker logs or kubectl logs along with Application Monitor logs. The Application Monitor uses logback for logger configuration. It has a default `logback.xml` file, located at `amsgce-runtime-<version>/teaagent/config/logback.xml`. To update the logging configuration, change this file before creating the TEA agent Image. After the image is built, you can mount the logging file but you will need to restart the TEA agent container.

Application Status

The Application Monitor displays the state of an application by aggregating and summarizing the state based on all its components, including the bindings. For example, If some application components are running and some are explicitly stopped, the runtime state shows `Partially Running`.

If an application is showing a different status than expected, check the status of different application entities such as components, services, references, and service or reference bindings. As the status of all child entities is included in the application status, you can identify the exact entity that caused the issue.

AMXCE Node Health Check

For troubleshooting issues related to the AMXCE node, you can use the AMXCE node Health Check feature. AMXCE Node Health Check is enabled by default. Health check enables you to view information about the threads that are used and JVM memory being used. For more information, see Viewing and Downloading Node Health Check Information.

OSGi Diagnostic Console

The OSGi diagnostic tool provides node-specific OSGi information that can be used for troubleshooting and diagnostic purposes without requiring special OSGi configuration and node restart. The tool provides a visual representation of the information normally provided by the OSGi console commands that require configuring OSGi console port and node restart. You can connect to OSGi command-line by using the default port 2222 which is configurable. For more information, see Viewing Node Specific OSGi Information.

Monitoring Applications by Using TIBCO Hawk Microagent

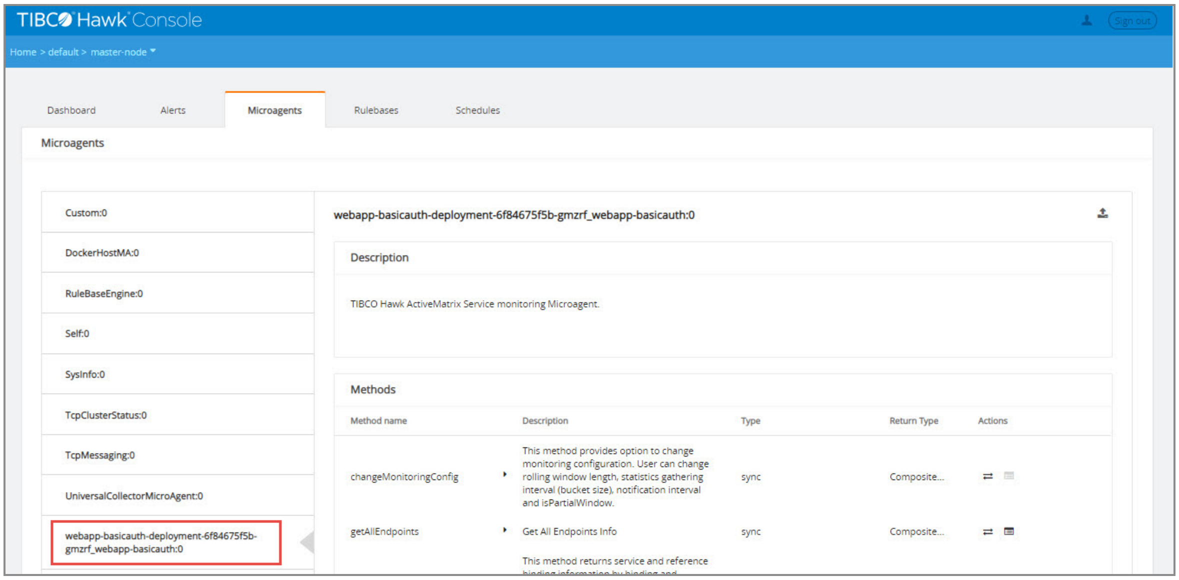
ActiveMatrix Service Grid - Container Edition embeds a TIBCO Hawk microagent whose methods enable you to monitor and manage deployed ActiveMatrix Service Grid - Container Edition applications. You can use the Hawk Console to perform many administrative functions. There is one Hawk microagent for each AMXCE application or node. The Hawk agent runs on each Kubernetes node. The Hawk agent runs as a daemon set.

Before you begin

You must deploy TIBCO Hawk - Container Edition components with ActiveMatrix Service Grid - Container Edition on Kubernetes. For complete instructions, see "Deploying TIBCO Hawk - Container Edition in Kubernetes" topic in *TIBCO ActiveMatrix® Service Grid - Container Edition Cloud Deployment.*

Procedure

1. Navigate to `http://<Console_host_IP>:<Host_port>/HawkConsole`. For example: http://<*host*>:31083/HawkConsole. Where 31083 is default NodePort specified in the sample `hawk.yaml` file.
2. On the Hawk Console login page, enter your login credentials for the Hawk Console and click **Login**. The default user name and password are admin.
3. Navigate to Domain card > Hawk Agent > Agent Dashboard. Click on the **Microagents** tab. The microagent default display name is `<HostName>_<AMXCE Node Name>`. Here `<HostName>` is combination of Kubernetes deployment name and container id. This microagent provides methods for monitoring ActiveMatrix Service Grid - Container Edition applications.



webapp-basicauth-deployment-6f84675f5b-gmzrf_webapp-basicauth:0

Description

TIBCO Hawk ActiveMatrix Service monitoring Microagent.

Methods

Method name	Description	Type	Return Type	Actions
changeMonitoringConfig	This method provides option to change monitoring configuration. User can change rolling window length, statistics gathering interval (bucket size), notification interval and isPartialWindow.	sync	Composite...	⇄
getAllEndpoints	Get All Endpoints Info	sync	Composite...	⇄
	This method returns service and reference			

Microagent Display Name

The microagent default display name is `<HostName>_<AMXCE Node Name>` as highlighted in the previous image. Here `<HostName>` is combination of Kubernetes deployment name and container id.

You can override this by setting the following environment variable name in the Kubernetes deployment YAML file:

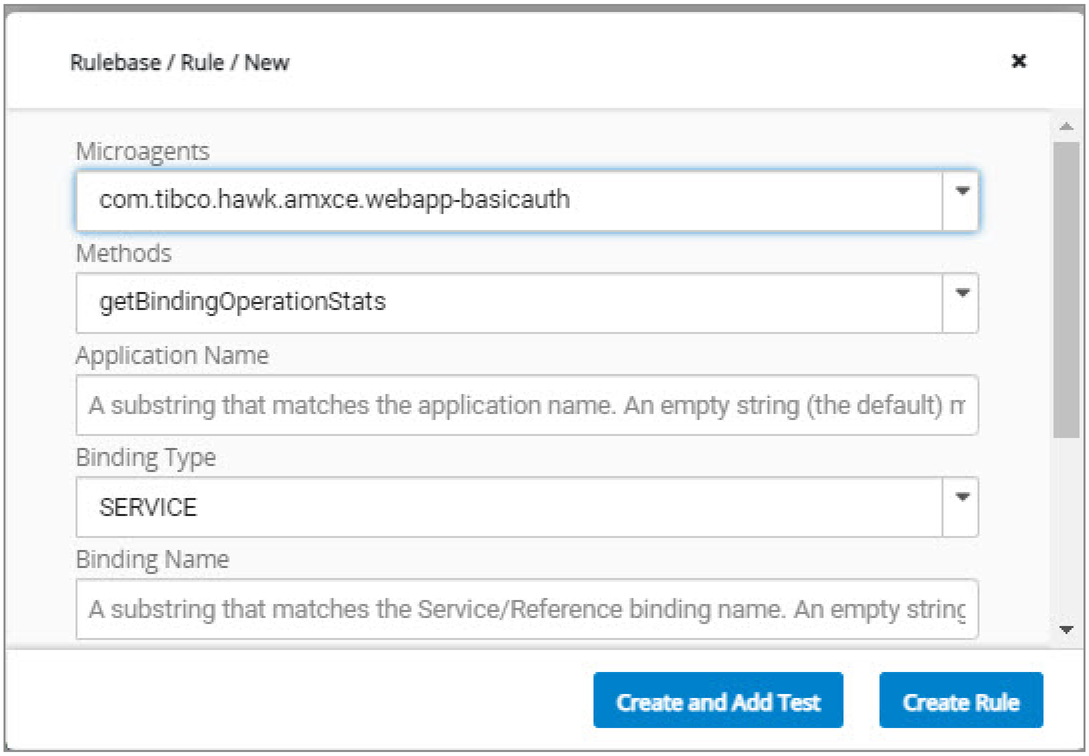
`microagent_name_suffix`: After setting this environment variable, microagent display name changes to `<HostName>_<microagent_name_suffix>`.

Microagent Name

The default name of a microagent is `com.tibco.hawk.amxce.<AMXCE Node Name>` as highlighted in the following image.

You can override this by setting the following environment name in the Kubernetes deployment YAML file:

`microagent_name_suffix`: After setting this environment variable, the microagent name changes to `com.tibco.hawk.amxce.<microagent_name_suffix>`.



Rulebase / Rule / New

Microagents
com.tibco.hawk.amxce.webapp-basicauth

Methods
getBindingOperationStats

Application Name
A substring that matches the application name. An empty string (the default) m

Binding Type
SERVICE

Binding Name
A substring that matches the Service/Reference binding name. An empty strinc

Methods

The methods in the microagent are used to change monitoring configurations and return components and bindings information.

changeMonitoringConfig

The changeMonitoringConfig method provides the option to change monitoring configuration.

Users can change rolling window length, statistics gathering interval (bucket size), notification interval, and isPartialWindow. This modification overrides configuration details provided by monitoring application and resets after the monitoring application restarts.

Type

Synchronous, `IMPACT_ACTION`.

Arguments

Name	Type	Description
Rolling Window Length (in seconds)	Long	Select rolling window length in seconds. **Possible values:** 60, 120, 180, 240, 300
Statistics Gathering Interval (in seconds)	Long	Select statistics gathering interval in seconds. **List box options:** 5, 10, 15, 20, 30, 40, 50, 60
Notification Interval (in seconds)	Long	Select notification interval in seconds. **List box options:** 10, 15, 20, 30, 40, 50, 60
Is Partial Window	Boolean	Select true or false for the sliding window. Statistics include the current working bucket. **Valid values:** true, false

Returns

None.

getBindingInfo

The getBindingInfo method returns the service and reference binding information by binding and application name. It serves as the regular expression used to filter the bindings returned.

Type

Synchronous, `IMPACT_INFO`.

Arguments

Name	Type	Description
Application Name	String	A substring that matches the application name. The substring can also be a regular expression. **Default:** Empty string (An empty string matches all names.)
Binding Name	String	A substring that matches the name of the service or reference binding. The substring can also be a regular expression. **Default:** Empty string (An empty string matches all names.)
Runtime State	String	Select a runtime state of the binding.

Returns

Type: `COM.TIBCO.hawk.talon.TabularData`

Name	Type	Description
AMXCE Host	String	Name of the AMXCE host.
AMXCE Node	String	Name of the AMXCE node.
Environment	String	Name of the environment.
Application	String	Name of the application.
Binding	String	Name of the binding.
Binding Type	String	Binding type (service or reference).
Contract	String	Contract name (service or reference name).
Runtime State	String	Runtime state of the binding. **Possible values:**

Name	Type	Description
		- RUNNING - NOT_RUNNING - STOPPED - LOST_CONTACT - INSTALLED - UNINSTALLED - START_FAILED - WAITING_FOR_DEPENDENCIES
Binding URI	String	Binding URI.
Time Stamp	String	Timestamp of the runtime state change of binding.

onComponentOperationStats

The onComponentOperationStats method provides notification in each interval period defined in monitoring configuration with statistics.

The statistics include component service and reference operation for the last `<n>` by operation name, component service or reference name, component name, application name and type and it serves as the regular expression used to filter the operations statistics returned.

Type

Asynchronous, `IMPACT_INFO`.

Arguments

Name	Type	Description
Operation Name	String	A substring that matches the operation name. The substring can also be a regular expression.

Name	Type	Description
		Default: Empty string (An empty string matches all names.)
Contract Name	String	A substring that matches the component service or reference name. The substring can also be a regular expression. **Default:** Empty string (An empty string matches all names.)
Component Name	String	A substring that matches the component name. The substring can also be a regular expression. **Default:** Empty string (An empty string matches all names.)
Application Name	String	A substring that matches the application name. The substring can also be a regular expression. **Default:** Empty string (An empty string matches all names.)
Contract Type	String	Select type. **Possible values:** • **<none>:** All component services and references. • **SERVICE:** All component services. • **REFERENCE:** All component references.

Return

Type: `COM.TIBCO.hawk.talon.TabularData`

Name	Type	Description
AMXCE Host	String	Name of the AMXCE host.
AMXCE Node	String	Name of the node.
Environment	String	Name of the environment.
Application	String	Name of the application.

Name	Type	Description
Component	String	Name of the component.
Contract	String	Name of the contract.
Contract Type	String	Contract type (service or reference).
Operation Name	String	Name of the service operation.
URI	String	Component URI.

onBindingOperationStats

The onBindingOperationStats method provides notification in each interval period defined in monitoring configuration with statistics of service and reference binding operation for last `<n>` by operation name, binding name, application name and binding type. It serves as the regular expression used to filter the operations statistics returned.

Type

Asynchronous, `IMPACT_INFO`.

Arguments

Name	Type	Description
Operation Name	String	A substring that matches the operation name. The substring can also be a regular expression. **Default:** Empty string (An empty string matches all names.)
Binding Name	String	A substring that matches the name of the service or reference binding. The substring can also be a regular expression. **Default:** Empty string (An empty string matches all names.)
Application	String	A substring that matches the application name. The substring can also

Name	Type	Description
Name		be a regular expression. **Default:** Empty string (An empty string matches all names.)
Binding Type	String	Select type. **Possible values:** • **<none>:** All service and reference bindings. • **SERVICE:** All service binding. • **REFERENCE:** All reference bindings.

Return

Type: `COM.TIBCO.hawk.talon.TabularData`

Name	Type	Description
AMXCE Host	String	Name of the ActiveMatrix Service Grid - Container Edition host.
Node	String	Name of the node.
Environment	String	Name of the environment.
Application	String	Name of the application.
Binding	String	Service or reference binding name.
Binding Type	String	Binding type (component service or reference name).
Contract	String	Name of the contract.
Operation	String	Name of the service operation.
Binding URI	String	Service or reference binding URI.
Endpoint Direction	String	Endpoint direction (inbound or outbound).

ListAllStatus

The method ListAllStatus returns runtime status of all AMXCE entities such as host, node, components, and bindings.

Type

Open, Synchronous, `IMPACT_INFO`.

Arguments

None.

Returns

Type: `COM.TIBCO.hawk.talon.TabularData`

Name	Type
Index	Integer
Impl	String
Identifier	String
App	String
Host	String
State	String

getAllEndpoints

The getAllEndpoints method returns information about the endpoints.

Type

Open, Synchronous, `IMPACT_INFO`

Arguments

None

Returns

Type: `COM.TIBCO.hawk.talon.CompositeData`

Name	Type
HTTP Connector	String
Endpoint URI	String
Binding Type	String
Protocol	String
Host	String
Port	String

getBindingOperationStats

The getBindingOperationStats method provides statistics of service and reference binding operation execution by operation name, service, or reference binding name, application name, and binding type. It serves as the regular expression used to filter the operations statistics returned.

Type

Synchronous, `IMPACT_INFO`.

Arguments

Name	Type	Description
Application	String	A substring that matches the application name. The substring can also

Name	Type	Description
Name		be a regular expression. **Default:** Empty string (An empty string matches all names.)
Binding Name	String	A substring that matches the name of the binding. The substring can also be a regular expression. **Default:** Empty string (An empty string matches all names.)
Operation Name	String	A substring that matches the operation name. The substring can also be a regular expression. **Default:** Empty string (An empty string matches all names.)
Binding Type	String	Select type. **Possible values:** • **<none>:** All service and reference bindings. • **SERVICE:** Only service binding. • **REFERENCE:** Only reference bindings.

Return

Type: `COM.TIBCO.hawk.talon.TabularData`

Name	Type	Description
AMXCE Host	String	Name of the AMXCE host.
AMXCE Node	String	Name of the node.
Environment	String	Name of the environment.
Application	String	Name of the application.
Binding	String	Service or reference binding name.

Name	Type	Description
Binding Type	String	Binding type (component service or reference name).
Contract	String	Name of the contract.
Operation	String	Service operation name.
Binding URI	String	Service or reference binding URI.
Endpoint Direction	String	Endpoint direction (inbound or outbound).

getBindingStats

The getBindingStats method provides statistics of service and reference binding execution by binding name, application name, and type. It serves as the regular expression used to filter the binding statistics that are returned.

Type

Open, Synchronous, `IMPACT_INFO`

Arguments

Name	Type	Description
Application Name	String	A substring that matches the application name. The substring can also be a regular expression. **Default:** Empty string (An empty string matches all names.)
Binding Type	String	Select the binding type.
Binding Name	String	A substring that matches the service or reference binding name. The substring can also be a regular expression. **Default:** Empty string (An empty string matches all names.)

Returns

Type: `COM.TIBCO.hawk.talon.CompositeData`

Name	Type	Description
AMXCE Host	String	Name of the AMXCE host.
AMXCE Node	String	Name of the AMXCE node.
Environment	String	Name of the environment.
Application	String	Name of the application.
Binding	String	Name of the binding.
Binding Type	String	Binding type (that is, service or reference).
Contract	String	Contract name (that is, service or reference name).
Contract Type	String	Contract type (that is, service or reference).
Binding URI	String	Service or reference binding URI.
Hits	Double	The number of execution responses for the last period.
Faults	Double	The number of execution faults for the last period.
Pending Responses	Double	The difference between the count of requests and responses at end of window.
Total Response Time	Double	Total response time over collection period.
Avg Response Time	Double	The average response time for collection period.
Max Response Time	Double	The maximum response time over collection period.
Min Response	Double	The minimum response time over collection period.

Name	Type	Description
Time		
Standard Deviation	Double	The standard deviation in response time over collection period.
Start Time	String	The start time of the collection period, in ISO-8601 format.
End Time	String	The end time of the collection period, in ISO-8601 format.
Total Time	Long	The collection period, in milliseconds.

getComponentStats

The getComponentStats method provides statistics of the component service and reference execution by component, component service or reference name, application name, and type. It serves as the regular expression used to filter the component services or references statistics that are returned.

Type

Open, Synchronous, `IMPACT_INFO`

Arguments

Name	Type	Description
Application Name	String	A substring that matches the application name. The substring can also be a regular expression. **Default:** Empty string (An empty string matches all names.)
Component Name	String	A substring that matches the component name. The substring can also be a regular expression. **Default:** Empty string (An empty string matches all names.)

Name	Type	Description
Contract Type	String	Select the contract type: • SERVICE • REFERENCE
Contract Name	String	A substring that matches the component service or reference name. The substring can also be a regular expression. **Default:** Empty string (An empty string matches all names.)

Returns

Type: `COM.TIBCO.hawk.talon.TabularData`

Name	Type	Description
Index	Integer	Count of the elements.
AMXCE Host	String	Name of the AMXCE host.
AMXCE Node	String	Name of the AMXCE node.
Environment	String	Name of the environment.
Application	String	Name of the application.
Component	String	Name of the component.
Contract	String	Name of the contract.
Contract Type	String	Contract type (that is, service or reference).
Component URI	String	Service or reference component URI.
Hits	Double	The number of execution responses for the last period.
Faults	Double	The number of execution faults for the last period.

Name	Type	Description
Pending Responses	Double	The difference between the count of requests and responses at end of window.
Total Response Time	Double	Total response time over collection period.
Avg Response Time	Double	The average response time for collection period.
Max Response Time	Double	The maximum response time over collection period.
Min Response Time	Double	The minimum response time over collection period.
Standard Deviation	Double	The standard deviation in response time over collection period.
Start Time	String	The start time of the collection period, in ISO-8601 format.
End Time	String	The end time of the collection period, in ISO-8601 format.
Total Time	Long	The collection period, in milliseconds.

getComponentInfo

The getComponentInfo method returns component information by the component and application name. It serves as a regular expression used to filter the components returned.

Type

Synchronous, `IMPACT_INFO`.

Arguments

Name	Type	Description
Application Name	String	A substring that matches the application name. The substring can also be a regular expression. **Default:** Empty string (An empty string matches all names.)
Component Name	String	A substring that matches the component name. The substring can also be a regular expression. **Default:** Empty string (An empty string matches all names.)
Runtime State	String	Select a runtime state. **Possible values:** • RUNNING • NOT_RUNNING • STOPPED • LOST_CONTACT • INSTALLED • UNINSTALLED • START_FAILED

Returns

Type: `COM.TIBCO.hawk.talon.TabularData`

Name	Type	Description
AMXCE Host	String	Name of the AMXCE host associated with the node.
AMXCE Node	String	Name of the node.
Environment	String	Name of the environment.

Name	Type	Description
Application	String	Name of the application.
Component	String	Name of the component.
Runtime State	String	Runtime state of the component. **Possible values:** • RUNNING • NOT_RUNNING • STOPPED • LOST_CONTACT • INSTALLED • UNINSTALLED • START_FAILED
Version	String	Version of the component.
Component URI	String	URI of the component.
Time Stamp	String	Timestamp of the changed runtime state of the component.

getComponentOperationStats

The getComponentOperationStats method provides statistics of component service and reference operation execution by operation name, component service or reference name, application name, and type. It serves as the regular expression used to filter the operations statistics returned.

Type

Synchronous, `IMPACT_INFO`.

Arguments

Name	Type	Description
Operation Name	String	A substring that matches the operation name. The substring can also be a regular expression. **Default:** Empty string (An empty string matches all names.)
Contract Name	String	A substring that matches the component service or reference name. The substring can also be a regular expression. **Default:** Empty string (An empty string matches all names.)
Component Name	String	A substring that matches the component name. The substring can also be a regular expression. **Default:** Empty string (An empty string matches all names.)
Application Name	String	A substring that matches the application name. The substring can also be a regular expression. **Default:** Empty string (An empty string matches all names.)
Contract Type	String	Select type. **Possible values:** • **<none>:** All service and reference bindings. • **SERVICE:** Only component services. • **REFERENCE:** Only component references.

Return

Type: `COM.TIBCO.hawk.talon.TabularData`

Name	Type	Description
AMXCE Host	String	Name of the AMXCE host.

Name	Type	Description
AMXCE Node	String	Name of the node.
Environment	String	Name of the environment.
Application	String	Name of the application.
Component	String	Name of the component.
Contract	String	Name of the contract.
Contract Type	String	Contract type (service or reference).
Operation	String	Operation name.
URI	String	Component URI.

getConfig

The getConfig method provides monitoring configuration details.

Type

Synchronous, `IMPACT_INFO`.

Arguments

None.

Returns

Name	Type	Description
AMXCE Host	String	Name of the AMXCE host associated with the node.
Environment	String	Name of the environment.

Name	Type	Description
AMXCE Node	String	Name of the node.
Monitoring Application State	String	Runtime state of the monitoring application. **Note:** To get service execution statistics, monitoring application must be in RUNNING state.
Notification Enabled	Long	Returns true or false for statistics notification enable or disable.
Notification Interval	Long	Notification interval, in milliseconds.
Rolling Window Length	Long	Rolling window length, in milliseconds.
Number of Buckets	Long	Number of time intervals in rolling window.
Statistics Gathering Interval	Long	Statistics gathering interval time, in milliseconds.
Is Partial Window	Boolean	Indicates whether the sliding window statistics includes the current working bucket.

getJVMArgs

The getJVMArgs method returns JVM arguments of ActiveMatrix Service Grid - Container Edition node JVM.

Type

Open, Synchronous, `IMPACT_INFO`.

Arguments

None.

Returns

Type: `COM.TIBCO.hawk.talon.TabularData`

Name	Type	Description
jvmargs	String	JVM arguments.
envVar	String	Environment variable.

getMemoryPoolInfo

The getMemoryPoolInfo method returns details of memory pools of the JVM process started by the same user on the local machine.

Type

Open, Synchronous, `IMPACT_INFO`.

Arguments

None.

Returns

Type: `COM.TIBCO.hawk.talon.TabularData`

Name	Type	Description
Memory Pool Name	String	Name of the memory pool.
Peak Usage Committed	Long	Peak usage committed.
Peak Usage Init	Long	Initial peak usage.
Peak Usage Max	Long	Maximum peak usage.
Peak Usage Used	Long	Peak usage used.

Name	Type	Description
Memory Pool Type	String	Type of the memory pool.
Current Usage Committed	Long	Current usage committed.
Current Usage Init	Long	Initial current usage.
Current Usage Max	Long	Maximum current usage.
Current Usage Used	Long	Current usage used.

getNodeInfo

The getNodeInfo method returns node information by node name. If the node name argument is blank, then all nodes are returned. If a node name argument is provided, then it serves as a regular expression used to filter the nodes that are returned.

Type

Open, Synchronous, `IMPACT_INFO`

Arguments

None

Returns

Type: `COM.TIBCO.hawk.talon.CompositeData`

Name	Type	Description
Node Name	String	The name of the node.
Host Name	String	The name of the host associated with the node.
Runtime State	String	The actual state of the node as reported by the host.
Up Time	String	The uptime of the node in days, hours, minutes, and seconds.

Name	Type	Description
Process Id	String	The process id of the TIBCO host (OS process id).
Type	String	The type of the ActiveMatrix host: TIBCO Host.
Version	String	Version of the ActiveMatrix host.
Binding Status	String	The status of the host binding.
Host Platform Version	String	The host platform version.
Internet Host Name	String	The internet host name.
OS Name	String	The operating system name of the host.
OS Version	String	The operating system version of the host.
System Arch	String	The system architecture of the host.

getThreadInfo

The getThreadInfo method returns general thread information, execution information, and synchronization statistics of a specific thread or all threads of a particular JVM.

Type

Open, Synchronous, `IMPACT_INFO`.

Arguments

Name	Type	Description
Thread Name	String	Returns the name of the thread.

Returns

Type: `COM.TIBCO.hawk.talon.TabularData`

Name	Type	Description
Thread Name	String	The name of the thread
Thread Id	Long	The ID of the thread
Thread State	String	The state of the thread
User Time	Long	CPU time spent by the thread in user mode in nanoseconds
isInNative	Boolean	Specifies whether it is running native code through the Java Native Interface (JNI)
Suspended	Boolean	Specifies whether the thread is suspended
Blocked Count	Long	The total number of attempts that the thread is blocked to enter or re-enter a monitor
Blocked Time	String	The approximate accumulated elapsed time (in milliseconds) that the thread has blocked to enter or re-enter a monitor since the enabling of thread contention monitoring
Lock Name	String	The string representation of the monitor lock that the thread is blocked to enter or waiting to be notified
Lock Owner Name	String	The thread name that holds the monitor lock of an object on which the thread is blocked
Lock Owner Id	Long	The thread ID that holds the monitor lock of an object on which the thread is blocked
Waited Count	Long	The total count of the number of times the thread waited for notification
Waited Time	Long	The approximate accumulated elapsed time (in milliseconds) that the thread has waited for notification since the enabling of thread

Name	Type	Description
		contention monitoring
Uncaught Exception	String	Uncaught exception of the thread
Stack Trace	String	Displays stack trace when an exception was thrown

getVirtualMachineInfo

The getVirtualMachineInfo method discovers and provides detailed information of all JVM started by the same user on the local machine.

Type

Open, Synchronous, `IMPACT_INFO`.

Arguments

None.

Returns

Type: `COM.TIBCO.hawk.talon.CompositeData`

Name	Type	Description
Name	String	The name representing the running JVM.
Start Time	String	Start time of the JVM in yyyy-MM-dd'T'HH:mm:ssZ format.
Up Time	String	Uptime of the JVM in day, hour, minutes, and seconds.
VM Name	String	JVM implementation name.
VM Vendor	String	JVM implementation vendor.

Name	Type	Description
Version	String	JVM implementation version.
Current Heap Size	Long	Size of used memory, in bytes.
Committed Memory	Long	Size of memory in bytes that is committed for the JVM to use.
Maximum Heap Size	Long	Maximum size of memory in bytes that can be used for memory management.
Free Heap Size	Long	Amount of free memory, in bytes.
Non Heap Memory Used	Long	Amount of used non heap memory, in bytes.
Live Threads	Integer	Current number of live threads including both daemon and non-daemon threads.
Peak Threads	Integer	Peak live thread count since the JVM was started or peak was reset.
Daemon Threads	Integer	Current number of live daemon threads.
Total Threads Started	Long	Total number of threads created and started since the start of JVM.
Current Classes Loaded	Integer	Number of classes currently loaded into JVM.
Total Classes Loaded	Long	Total number of classes loaded since the start of JVM execution.
Total Classes Unloaded	Long	Total number of classes unloaded since the start of JVM execution.

Analyzing and Visualizing Application Metrics Using Prometheus and Grafana

ActiveMatrix Service Grid - Container Edition supports integration with Prometheus to export data from it. Prometheus is a monitoring tool which helps in analyzing the app metrics. Prometheus integrates with Grafana which provides better visual analytics. Data similar to SPM Server and Dashboard can now be exported by using Prometheus exporter.

Prometheus Exporter

ActiveMatrix Service Grid - Container Edition node provides the exporter which shows component, binding, and JVM information. This information is accessible at: `http://<localhost>:9998/metrics`.

This URL provides the following types of metrics:

- JVM related data which includes memory, thread, and so on. This is the default Prometheus JVM exporter.
- ActiveMatrix Service Grid - Container Edition component and binding operation metrics.

Prometheus servers scrape data from the HTTP /metrics endpoint of the applications. For more information about how to configure Prometheus to analyze ActiveMatrix Service Grid - Container Edition application metrics, see Using Prometheus to Analyze TIBCO ActiveMatrix Application Metrics. The Prometheus dashboard is available at `https://`*`<host>`*`:`*`<nodePort>`*. The default URL is `http://localhost:9090/`

It shows the following parameters for a component or binding. Here, the Prometheus metric type is gauge.

Label	Description
hits	Number of hits during the collection period.
minResponseTime	Minimum response time for a request during collection period.
maxResponseTime	Maximum response time for a request during collection period.

avgResponseTime	Average response time for a request during collection period.
totalResponseTime	Total response time for a request during collection period.
pendingResponse	Number of pending responses on the component.
faults	Number of faults on the component.
collectionPeriod	Collection period is the difference between the start time and end time.

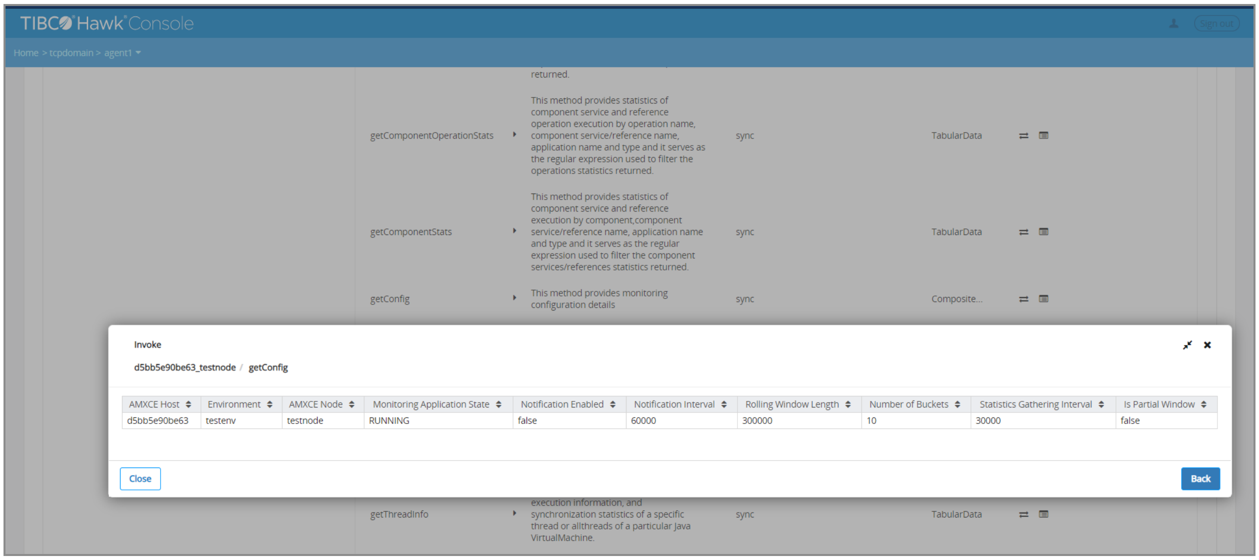
The default status gathering configuration is as follows:

- Status Gathering Interval - 30 seconds.
- Notification Interval - 60 seconds.

Therefore, set the Prometheus exporter polling interval to 30 seconds.

Controlling the Status Gathering Interval

You can control the status gathering interval by using the `changeMonitoringConfig` Hawk microagent method. To get the current configuration, invoke the `getconfig` method of an AMXCE node microagent.



returned.

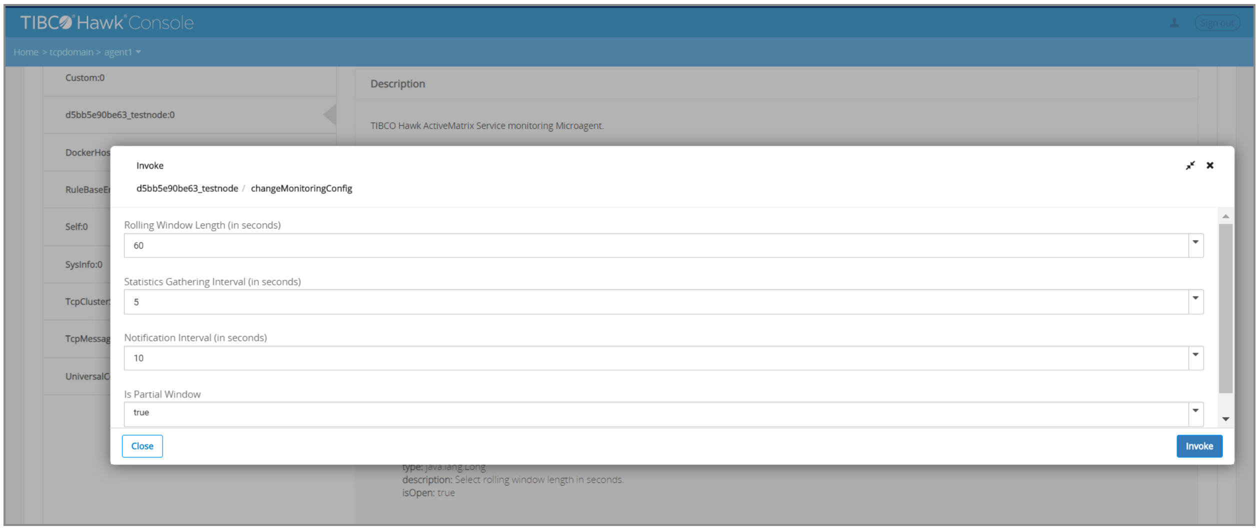
getComponentOperationStats	This method provides statistics of component service and reference operation execution by operation name, component service/reference name, application name and type and it serves as the regular expression used to filter the operations statistics returned.	sync	TabularData
getComponentStats	This method provides statistics of component service and reference execution by component,component service/reference name, application name and type and it serves as the regular expression used to filter the component services/references statistics returned.	sync	TabularData
getConfig	This method provides monitoring configuration details	sync	Composite...
getThreadInfo	execution information, and synchronization statistics of a specific thread or allthreads of a particular Java VirtualMachine.	sync	TabularData

Invoke

d5bb5e90be63_testnode / getConfig

AMXCE Host	Environment	AMXCE Node	Monitoring Application State	Notification Enabled	Notification Interval	Rolling Window Length	Number of Buckets	Statistics Gathering Interval	Is Partial Window
d5bb5e90be63	testenv	testnode	RUNNING	false	60000	300000	10	30000	false

To change this configuration, invoke the `changeMonitoringConfig` method. AMXCE component or binding related data will be shown if there is any hit on that entity.



TIBCO Hawk ActiveMatrix Service monitoring Microagent.

Rolling Window Length (in seconds): 60

Statistics Gathering Interval (in seconds): 5

Notification Interval (in seconds): 10

Is Partial Window: true

type: java.lang.Long
description: Select rolling window length in seconds.
isOpen: true

To disable the default JVM exporter, set the application level property, `initializeDefaultExports`, to false. The default value is true.

Using Prometheus to Analyze TIBCO ActiveMatrix Application Metrics

Sample YAML files to deploy Prometheus in Kubernetes are provided at `amsgce-runtime-<version>\samples\kubernetes\prometheus`.

Before you begin

To monitor an application with Prometheus, provide the following configuration details in the application YAML file:

The following annotations are mandatory:

- `prometheus.io/scrape`: `'true'`: Setting this annotation to true registers the application with the monitoring application and enables the metrics collection on the application. Setting the annotation to false deregisters it from the monitoring application and turns off the metrics collection.
- `prometheus.io/port: '9998'`: This is the monitoring application port.

Procedure

For complete instructions, see the file `amsgce-runtime-<version>\samples\kubernetes\prometheus\readme.md`.

Grafana Dashboards

Grafana is an open source application to visualize large-scale measurement data. You can use Grafana to display the monitoring data exported by using Prometheus.

The sample files to deploy Grafana in Kubernetes are provided in the folder `amsgce-runtime-<version>\samples\kubernetes\grafana`.

The default URL of the Grafana dashboard is `https://<IP>:3000/`

For complete instructions, see the file `amsgce-runtime-<version>\samples\kubernetes\grafana\readme.md`.

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

SOME TIBCO SOFTWARE EMBEDS OR BUNDLES OTHER TIBCO SOFTWARE. USE OF SUCH EMBEDDED OR BUNDLED TIBCO SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED TIBCO SOFTWARE. THE EMBEDDED OR BUNDLED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER TIBCO SOFTWARE OR FOR ANY OTHER PURPOSE.

USE OF TIBCO SOFTWARE AND THIS DOCUMENT IS SUBJECT TO THE TERMS AND CONDITIONS OF A LICENSE AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED SOFTWARE LICENSE AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER LICENSE AGREEMENT WHICH IS DISPLAYED DURING DOWNLOAD OR INSTALLATION OF THE SOFTWARE (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH SOFTWARE LICENSE AGREEMENT OR CLICKWRAP END USER LICENSE AGREEMENT, THE LICENSE(S) LOCATED IN THE “LICENSE” FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

TIBCO, the TIBCO logo, the TIBCO O logo, ActiveMatrix, Business Studio, TIBCO Business Studio, Enterprise Message Service, and Hawk are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

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

This document includes fonts that are licensed under the SIL Open Font License, Version 1.1, which is available at: https://scripts.sil.org/OFL

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

This software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the readme file for the availability of this software version on a specific operating system platform.

THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

This and other products of TIBCO Software Inc. may be covered by registered patents. Please refer to TIBCO's Virtual Patent Marking document (https://www.tibco.com/patents) for details.