

TIBCO Silver[®] Fabric

Enabler for Docker Guide

Software Release 1.0
April 2016

Important Information

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.

This document contains confidential information that is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of TIBCO Software Inc.

TIBCO, Two-Second Advantage, GridServer, FabricServer, GridClient, GridBroker, FabricBroker, LiveCluster, VersaUtility, VersaVision, SpeedLink, Federator, and RTI Design are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

Enterprise Java Beans (EJB), Java Platform Enterprise Edition (Java EE), Java 2 Platform Enterprise Edition (J2EE), and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle Corporation in the U.S. and other countries.

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.

TIBCO products may include some or all of the following:

Software developed by Terence Parr.

Software developed by the Apache Software Foundation (<http://www.apache.org/>).

This product uses c3p0. c3p0 is distributed pursuant to the terms of the Lesser General Public License. The source code for c3p0 may be obtained from <http://sourceforge.net/projects/c3p0/>. For a period of time not to exceed three years from the Purchase Date, TIBCO also offers to provide Customer, upon written request of Customer, a copy of the source code for c3p0.

Software developed by MetaStuff, Ltd.

Software licensed under the Eclipse Public License. The source code for such software licensed under the Eclipse Public License is available upon request to TIBCO and additionally may be obtained from <http://eclipse.org/>.
Software developed by Info-ZIP.

This product includes Javassist licensed under the Mozilla Public License, v1.1. You may obtain a copy of the source code from <http://www.jboss.org/javassist/>

This product includes software licensed under the Common Development and Distribution License (CDDL) version 1.0. The source code for such software licensed under the Common Development and Distribution License (CDDL) version 1.0 is available upon request to TIBCO.

Software developed by Jason Hunter & Brett McLaughlin.

Software developed by JSON.org.

Software developed by QOS.ch.

Software developed by the OpenSymphony Group (<http://www.opensymphony.com/>).

This product includes WSDL4J software which is licensed under the Common Public License, v1.0. The source code for this software may be obtained from TIBCO's software distribution site.

Software developed by the Indiana University Extreme! Lab (<http://www.extreme.indiana.edu/>).

Software developed by Jean-loup Gailly and Mark Adler.

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 Product is covered by U.S. Patent No. 6,757,730, 7,093,004, 7,093,004, and patents pending.

Copyright © 1999-2016 TIBCO Software Inc. All rights reserved.

TIBCO Software Inc. Confidential Information

Contents

Preface	vii
Related Documentation	viii
TIBCO Silver Fabric Documentation	viii
Other Documentation and Help	ix
Typographical Conventions	x
Connecting with TIBCO Resources	xiii
How to Join TIBCOCommunity	xiii
How to Access All TIBCO Documentation	xiii
How to Contact TIBCO Support	xiii
Chapter 1 Introduction	1
Before Beginning	2
Requirements	2
Chapter 2 Installation	3
Required Grid Libraries	4
Installing the TIBCO Silver® Fabric Enabler for Docker	5
Installing TIBCO Silver® Fabric Enabler for Docker on the Silver Fabric Engine	6
Chapter 3 Configuration	7
Before Beginning	8
Running a Docker Component	9
Docker Container Set	11
Archive Management	12
Chapter 4 Running Your Component	13
Verifying Your Component Configuration	14
Load Balancing	15
VirtualRouter	15
Custom External Load Balancer	15
Chapter 5 Statistics and Variables	17
Statistics	18

Runtime Context Variables 19

Preface

TIBCO Silver[®] Fabric Enabler for Docker combines the flexibility and scalability of the public cloud with the security and control of your own data center. It brings the elasticity of cloud computing to your organization – supporting existing solutions within your current infrastructure while automatically scaling resources to meet demand.

Topics

- [Related Documentation, page viii](#)
- [Typographical Conventions, page x](#)
- [Connecting with TIBCO Resources, page xiii](#)

Related Documentation

This section lists documentation resources you may find useful.

For the latest version of documentation, including any changes or additions made since the last product release, please visit <http://docs.tibco.com>.

TIBCO Silver Fabric Documentation

The following documentation is included with Silver Fabric in Adobe Acrobat (PDF) format. To view the guides, log in to the Administration Tool and go to **Admin > Documentation**. The PDF files are also on the Broker at `SF_HOME/webapps/livecluster/admin/docs`. The following documents form the Silver Fabric documentation set:

- *Silver Fabric Concepts* Contains an introduction to Silver Fabric, including definitions of key concepts and terms, such as Enablers, Stacks, Components, Engines, and Brokers. Read this first if you are new to Silver Fabric.
- *Silver Fabric Installation Guide* Covers installation of Silver Fabric for Windows and Unix, including Brokers, Engines, and pre-installation planning.
- *Silver Fabric Cloud Administration Guide* Covers Silver Fabric cloud administration, configuration of Engines, Enablers, and Components, and configuration and use of Skyway. Also covers security, general maintenance, performance tuning, and database administration.
- *Silver Fabric Developer's Guide* Developer-related topics such as logging and debugging, using the Admin API, and the Enabler SDK.
- *Silver Fabric Developer's Tutorial* Tutorials for developers, such as how to write Enablers and Asset Managers.
- *Silver Fabric User's Guide* Covers Silver Fabric use and operation, including management of Engines, Enablers, Components, and Stacks.
- *Silver Fabric Skyway User's Guide* Covers usage of Skyway, which enables users to quickly and easily provision and manage their Silver Fabric Stacks.
- *Silver Fabric Tomcat Enabler Guide* Covers installation and configuration of applications run on the Tomcat Enabler.
- *Silver Fabric Command Line Enabler Guide* Covers installation and configuration of applications run on the Command Line Enabler.

Other Documentation and Help

Additional help and information is available from the following sources:

- *Silver Fabric Administration Tool Help* Context-sensitive help is provided throughout the Silver Fabric Administration Tool by clicking the Page Help button located on any page.
- *API Reference* Silver Fabric API reference information is available in the Silver Fabric SDK in the `api` directory in JavaDoc format. You can also view and search them from the Silver Fabric Administration Tool; log in to the Administration Tool and go to **Admin > Documentation**.

Typographical Conventions

The following typographical conventions are used in this manual.

Table 1 General Typographical Conventions

Convention	Use
<i>TIBCO_HOME</i>	Many TIBCO products must be installed within the same home directory. This directory is referenced in documentation as <i>TIBCO_HOME</i> . The default value of <i>TIBCO_HOME</i> depends on the operating system. For example, on Windows systems, the default value is C:\tibco.
<i>SF_HOME</i>	TIBCO Silver [®] Fabric installs into a directory within <i>TIBCO_HOME</i> . This directory is referenced in documentation as <i>SF_HOME</i> . The default value of <i>SF_HOME</i> depends on the operating system. For example on Windows systems, the default value is C:\tibco\fabric.
code font	Code font identifies commands, code examples, filenames, pathnames, and output displayed in a command window. For example: Use MyCommand to start the foo process.
bold code font	Bold code font is used in the following ways: <ul style="list-style-type: none">• In procedures, to indicate what a user types. For example: Type admin.• In large code samples, to indicate the parts of the sample that are of particular interest.• In command syntax, to indicate the default parameter for a command. For example, if no parameter is specified, MyCommand is enabled: MyCommand [enable disable]
italic font	Italic font is used in the following ways: <ul style="list-style-type: none">• To indicate a document title. For example: See <i>TIBCO ActiveMatrix BusinessWorks Concepts</i>.• To introduce new terms. For example: A portal page may contain several portlets. <i>Portlets</i> are mini-applications that run in a portal.• To indicate a variable in a command or code syntax that you must replace. For example: MyCommand <i>PathName</i>

Table 1 General Typographical Conventions (Continued)




Convention	Use
Key combinations	Key names separated by a plus sign indicates keys pressed simultaneously. For example: Ctrl+C. Key names separated by a comma and space indicate keys pressed one after the other. For example: Esc, Ctrl+Q.
	The note icon indicates information that is of special interest or importance, for example, an additional action required only in certain circumstances.
	The tip icon indicates an idea that could be useful, for example, a way to apply the information provided in the current section to achieve a specific result.
	The warning icon indicates the potential for a damaging situation, for example, data loss or corruption if certain steps are taken or not taken.

Table 2 Syntax Typographical Conventions

Convention	Use
[]	An optional item in a command or code syntax. For example: <code>MyCommand [optional_parameter] required_parameter</code>
	A logical OR that separates multiple items of which only one may be chosen. For example, you can select only one of the following parameters: <code>MyCommand param1 param2 param3</code>

Table 2 Syntax Typographical Conventions (Continued)

Convention	Use
{ }	<p>A logical group of items in a command. Other syntax notations may appear within each logical group.</p> <p>For example, the following command requires two parameters, which can be either the pair param1 and param2, or the pair param3 and param4.</p> <pre>MyCommand {param1 param2} {param3 param4}</pre> <p>In the next example, the command requires two parameters. The first parameter can be either param1 or param2 and the second can be either param3 or param4:</p> <pre>MyCommand {param1 param2} {param3 param4}</pre> <p>In the next example, the command can accept either two or three parameters. The first parameter must be param1. You can optionally include param2 as the second parameter. And the last parameter is either param3 or param4.</p> <pre>MyCommand param1 [param2] {param3 param4}</pre>

Connecting with TIBCO Resources

How to Join TIBCOCommunity

TIBCOCommunity is an online destination for TIBCO customers, partners, and resident experts, a place to share and access the collective experience of the TIBCO community. TIBCOCommunity offers forums, blogs, and access to a variety of resources. To register, go to <http://www.tibcommunity.com>.

How to Access All TIBCO Documentation

After you join TIBCOCommunity, you can access the documentation for all supported product versions here:

<http://docs.tibco.com>

How to Contact TIBCO Support

For comments or problems with this manual or the software it addresses, please contact TIBCO Support as follows.

- For an overview of TIBCO Support, and information about getting started with TIBCO Support, visit this site:

<http://www.tibco.com/services/support>

- If you already have a valid maintenance or support contract, visit this site:

<https://support.tibco.com>

Entry to this site requires a user name and password. If you do not have a user name, you can request one.

Chapter 1 Introduction

A Silver Fabric Enabler allows an external application or application platform to run in a TIBCO Silver[®] Fabric software environment. The TIBCO Silver[®] Fabric Enabler for Docker provides integration between Silver Fabric and Docker.

With the TIBCO Silver[®] Fabric Enabler for Docker, you can:

- Start and stop a Docker container from a Docker registry.
- Pull images from a Docker registry.
- Locally build a Docker image if the image is not found on the Docker registry as long a Docker file exists.

Topics

- [Before Beginning, page 2](#)

Before Beginning

This guide provides the instructions for installing and configuring the TIBCO Silver[®] Fabric Enabler for Docker. This guide assumes a Silver Fabric Broker is running with at least one Engine installed, and that you have the Broker's hostname, a username, and password for the Silver Fabric Administration Tool. If this isn't true, see the *Silver Fabric Installation Guide*, or contact your administrator.

This guide presumes a strong familiarity with Docker. The instructions provided for accomplishing tasks are not version-specific. If you are uncertain of how to achieve a particular task, consult your version-specific Docker documentation.

Requirements

Please see the included Silver Fabric Readme for the latest prerequisites required for this Enabler.

Chapter 2 **Installation**

This chapter provides information on installation of the TIBCO Silver[®] Fabric Enabler for Docker.

Topics

- [Required Grid Libraries, page 4](#)
- [Installing the TIBCO Silver[®] Fabric Enabler for Docker, page 5](#)
- [Installing TIBCO Silver[®] Fabric Enabler for Docker on the Silver Fabric Engine, page 6](#)

Required Grid Libraries

The TIBCO Silver[®] Fabric Enabler for Docker consists of an Enabler Runtime Grid Library, and a Component Type Grid Library.

The required Grid Libraries for each version of the Enabler are listed below.

Docker Enabler Version	Required Grid Libraries
1.0.0	SilverFabric_docker_container_1.0.0_gridlib.zip SilverFabric_docker_componentType_1.0.0.zip

Installing the TIBCO Silver® Fabric Enabler for Docker

To install the TIBCO Silver® Fabric Enabler for Docker on the Broker:

1. If you are upgrading, ensure the respective Component and Enabler are not part of an active Stack.
2. Copy the desired Enabler version Grid Library files to the `SF_HOME/webapps/livecluster/deploy/resources/gridlib` directory.



Copying the Grid Libraries files to this directory also extracts them to the deployed directory `SF_HOME/webapps/livecluster/deploy/expanded/`. This overwrites any changes to the existing Grid Library in the staging directory.

3. Verify successful installation by selecting **Stacks > Enablers** in the Silver Fabric Administration Tool and ensuring that the Enabler appears in the list.

Installing TIBCO Silver® Fabric Enabler for Docker on the Silver Fabric Engine

Before using the TIBCO Silver® Fabric Enabler for Docker, you must prepare Engines to run Docker containers:

1. Install Docker 1.9.0 or greater on each Engine machine. See <https://docs.docker.com/engine/installation/> for more details.
2. Configure Docker to enable the Remote API to run on a TCP port.
3. Configure passwordless sudo on the OS user running the Engine, so that Docker CLI commands can be run without being prompted for a password.
4. Configure the Docker daemon storage-driver option to use a non-loopback driver. See <https://docs.docker.com/engine/reference/commandline/daemon/#daemon-storage-driver-option> for details.
5. Configure the Docker daemon selinux-enabled option.
6. In the Silver Fabric Administration Tool, go to **Engines > Daemons** and in the Actions list, select **Restart Daemon**.

Chapter 3 **Configuration**

This chapter provides information on configuring the TIBCO Silver[®] Fabric Enabler for Docker.

Topics

- [Before Beginning, page 8](#)
- [Running a Docker Component, page 9](#)
- [Archive Management on page 12](#)

Before Beginning

These instructions presume a strong familiarity with your particular version of Docker. If you are uncertain of how to achieve a particular task, consult your version-specific Docker documentation.

Running a Docker Component

The following section describes how to run a Component in Docker on an Engine.

To define the Component in Silver Fabric:

1. Enter your Silver Fabric host and port in your browser and log in to the Silver Fabric Administration Tool.
2. Select **Stacks > Components**.
3. Select **Create New Docker Enabler Component** from the Global Actions list.
4. Enter a Component name in the **Name** field.
5. Click **Next**. The Docker Configuration screen appears.
6. Enter values for the following properties:
 - **Docker Image** — The name of the Docker image. This is required
 - **Docker Image Tag** — A tag for the image, such as `latest`.
 - **Docker Registry** — The hostname and port of the Docker registry, such as `dockerregistry.example.com:5000`.
7. Click **Next**. The Docker Basic Network Configuration screen appears.
8. Enter values for **Docker Publish Ports** and **Docker Expose Ports**.
9. Click **Next**. The **Docker Basic Volumes Configuration** screen appears.
10. Enter values for **Docker Working Directory**, **Docker Mount Volumes**, **Docker Volumes From**, and **Docker Add Host**, as needed.
11. Click **Next**. The **Docker Extra Basic Configuration** screen appears.

12. Enter values for the following properties:
 - Docker Extra Run Options (Docker run options (--detach=false option is not supported)) --detach=true
 - Docker Environment Variables (Docker environment variables --env var=value)
 - Docker Environment File Path (Docker environment file --env-file file)
 - Docker Extra Build Options (Docker build options) --quiet=false --no-cache=true --rm=true
 - Reuse Docker Image (Reuse existing local Docker image if it exists.)
 - Reuse Docker Container (Reuse Docker container if it exists)
 - Remove Docker Image (Remove local Docker image on component shutdown)
 - Remove Docker Container (Remove Docker container on component shutdown)
13. Click **Next**. The **Configure Component Features** screen appears. HTTP Support, Application Logging Support, and Archive Management Support are added to the feature list by default. You can further customize Component features as needed by selecting the feature and clicking **Edit**.
14. Configure the options on the remaining screens as needed. Additional options are also configurable through context variables.
15. Optionally, on the **Add/edit default rule settings** screen, add a rule to limit this Component to Engines with Docker:
 - a. Click **Add Rule**.
 - b. Select **Resource Preference**.
 - c. For **Property Name**, select **Docker Enabled**.
 - d. For **Property Value**, enter `true`.
 - e. For **Preference**, select **Required**.
 - f. Click **OK**.
16. Click **Finish**.
17. Select **Stacks > Components** in the Silver Fabric Administration Tool.
18. Select **Publish Component** from the corresponding Actions list of the Component you created.
19. Create a Stack and add the Component to the Stack as described in the *Silver Fabric Cloud Administration Guide*. Start the Stack when desired.

Docker Container Set

The TIBCO Silver[®] Fabric Enabler for Docker supports two mutually exclusive approaches for specifying the Docker container set managed by the Component using this Enabler

Docker containers may be specified in a Docker Compose file included within the Component. Under this approach, the variable `DOCKER_COMPOSE_FILE` must be specified, and other `DOCKER_COMPOSE` variables may be used to specify appropriate values.

Alternatively, the Docker container set maybe specified using Enabler runtime context variables without the `DOCKER_COMPOSE` prefix. Under this approach, `DOCKER_IMAGE` must be specified.

If both approaches are specified, Docker Compose approach takes precedence. The relative folder containing Docker compose file within the Component must contain all the Docker compose context files.

See [Runtime Context Variables on page 19](#) for more information on Runtime Context Variables.

Archive Management

The TIBCO Silver[®] Fabric Enabler for Docker does not implement any of the methods required for archive management. Archive Management must be implemented by the end-user as Docker can run any type of application. It is assumed that the application archives are managed directly through the image or are managed by using a Dockerfile.

Chapter 4 **Running Your Component**

After configuring and activating your Silver Fabric Docker Component, you can access the Components running on the Engine. To further verify a successful configuration, you can access the a running Web application through the VirtualRouter Status page as described below.

Topics

- [Verifying Your Component Configuration, page 14](#)
- [Load Balancing, page 15](#)

Verifying Your Component Configuration

To verify your Component configuration:

1. Go to **Engines > Engines** in the Silver Fabric Administration Tool and ensure that at least one instance of your Component is running on an Engine.
2. Go to **Admin > VirtualRouter**. The VirtualRouter page contains a table for each VirtualRouter client that is currently running. Select the **VirtualRouter Properties** action next to each client to show the Component each host is running.
3. When you find the client running your Component, select the **Status Page** action. The VirtualRouter Status page is shown:
4. In the **Relative URLs** column, entries for each web application are listed. Click a URL to connect to the web application. HTTP requests made when the URL is clicked in the Component-level **Relative URLs** column, are sent to VirtualRouter, which forwards the request to an Engine running the Component. HTTP requests made when the URL is clicked on the Engine-level Relative URLs column are sent directly to that Engine.
 - Alternatively, you can mimic the behavior of the VirtualRouter Status page by directing your browser to the address/port of your Silver Fabric Broker with a relative URL from your Component, such as `http://myserver:8080/myapp/index.html` (where myapp is one of the deployed Component). If you cannot access your Web application, reverify your Component configuration based on the steps in [Chapter 3, Configuration, on page 7](#).

Load Balancing

Load balancing is recommended for proxying requests from clients and routing them to Engines running the appropriate Component. Load balancing is achieved through mappings between relative URLs and Components.

VirtualRouter

VirtualRouter is Silver Fabric's load balancer for HTTP enabled Components. An instance of VirtualRouter runs by default on each Silver Fabric Broker, and can also run externally. For more information about using VirtualRouter, see the *Silver Fabric Cloud Administration Guide*.

Custom External Load Balancer

You can create a custom load balancer to distribute requests across instances of your Component. Accommodating the dynamic nature of Silver Fabric allocation requires a mechanism to update the routing list of the custom balancer when the Silver Fabric allocation changes. Notification of activation/deactivation events can occur in the following ways:

- ServerHook events
- SNMP traps
- Web Service calls

Refer to the *Silver Fabric Cloud Administration Guide* for more information on configuring and using these methods.

Chapter 5 **Statistics and Variables**

This chapter provides information on statistics and runtime context variables available in the TIBCO Silver[®] Fabric Enabler for Docker.

Topics

- [Statistics, page 18](#)
- [Runtime Context Variables, page 19](#)

Statistics

The following are the default statistics supported by the Enabler. You can select and track these statistics from the Component Wizard. Tracked statistics are available for report output. You can also create Policy rules based on any tracked statistic.

Table 3 TIBCO Silver® Fabric Enabler for Docker Statistics

Name	Description
Docker CPU Usage %	Docker CPU usage percentage
Docker Memory Usage %	Docker memory usage percentage
Docker Memory Usage (MB)	Docker memory usage (MB)
Docker Memory Limit (MB)	Docker Memory Limit (MB)
Docker Network Input (MB)	Docker network input (MB)
Docker Network Output (MB)	Docker network output (MB)
Docker Block Output (MB)	Docker block device output (MB)
Docker Block Input (MB)	Docker block device input (MB)

Runtime Context Variables

The following tables are a comprehensive list of all runtime context variables used by the Enabler.

Table 4 TIBCO Silver® Fabric Enabler for Docker Runtime Context Variables

Variable	Type	Description
DOCKER_CONTAINER_NAME	String	Leave this blank if you want unique name to be auto-generated
DOCKER_REGISTRY	String	[DEFAULT] Docker registry for fetching image. For example, <code>https://registryhost:5000/</code> .
DOCKER_IMAGE	String	Docker image name. For example, <code>centos:latest</code>
DOCKER_CONTEXT	String	Docker context path or URL used for building new image
DOCKER_COMMAND	String	Docker command executed in Docker container at startup
DOCKER_COMMAND_ARGS	String	Docker command args
DOCKER_CONTAINER_HOSTNAME	String	Docker container hostname
DOCKER_CONTAINER_WORK_DIR	String	Docker container work directory
DOCKER_CONTAINER_LOGS	String	Docker container logs file. The default value is <code>\${CONTAINER_WORK_DIR}/docker.log</code> .
DOCKER_CONTAINER_NETWORK_MODE	String	[DEFAULT] Docker container network mode. The default value is <code>--net=bridge</code> .
DOCKER_EXTRA_RUN_OPTIONS	String	[DEFAULT] Docker run options (<code>--detach=false</code> option is not supported). The default value is <code>--detach=true</code> .
DOCKER_PUBLISH_PORTS	String	Docker publish ports <code>--publish \${LISTEN_ADDRESS}:hostPort:containerPort</code>

Table 4 TIBCO Silver® Fabric Enabler for Docker Runtime Context Variables (Continued)

Variable	Type	Description
DOCKER_EXPOSE_PORTS	String	Docker expose ports --expose port
DOCKER_MOUNT_VOLUMES	String	Docker mount volumes --volume hostdir:containerdir
DOCKER_VOLUMES_FROM	String	Docker volumes from --volume-from foo
DOCKER_ADD_HOST	String	Docker add host --add-host=host:ip
DOCKER_ENV_VARIABLES	String	Docker environment variables --env var=value
DOCKER_ENV_FILE	String	Docker environment file --env-file=file
DOCKER_LINK	String	Docker container links, e.g --link foo
DNS_SEARCH_DOMAINS	String	DNS search domains format: --dns-search=
DOCKER_EXTRA_BUILD_OPTIONS	String	[DEFAULT] Docker build options. The default value is --quiet=false --no-cache=true --rm=true.
DOCKER_STOP_OPTIONS	String	Docker stop options. The default value is --time=30.
DOCKER_REMOVE_OPTIONS	String	Docker remove container options. The default value is --force=true --volumes=true.
DOCKER_REMOVE_IMAGE_OPTIONS-	String	Docker remove image options. The default value is --force=true.
APP_RUNNING_PORTS	String	Docker container port used to check if container app is running. The default value is --force=true.
USE_SUDO	String	Run Docker with sudo. The sudo must not prompt for password. The default is false.
REUSE_DOCKER_IMAGE	String	Reuse existing local Docker image if it exists. The default is true.

Table 4 TIBCO Silver® Fabric Enabler for Docker Runtime Context Variables (Continued)

Variable	Type	Description
REUSE_DOCKER_CONTAINER	String	Reuse existing local Docker container if it exists. The default is false.
REMOVE_DOCKER_CONTAINER	String	Remove Docker container on Component shutdown. The default is true.
REMOVE_DOCKER_IMAGE	String	Remove Docker image on Component shutdown. The default is false.
HTTP_STATIC_ROUTES	String	A space-separated list of HTTP static routes, such as <code>ContextUrl:http://\${LISTEN_ADDRESS}:port</code> .
BIND_ON_ALL_LOCAL_ADDRESSES	Environment	If all network interfaces should be bounded for all public port access. The default is false.
LISTEN_ADDRESS_NET_MASK	Environment	A comma delimited list of net masks inCIDR notation. The first IP address found that matches one of the net masks is used as the listen address. Note that <code>BIND_ON_ALL_LOCAL_ADDRESSES</code> overrides this setting.

