TIBCO BusinessWorks™ Container Edition and Plug-ins for AWS Marketplace User's Guide

Software Release 2.3.1
November 2017
Document Update: November 2017

Two-Second Advantage®

TIBCO®

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.

THE SOFTWARE ITEMS IDENTIFIED BELOW ARE AVAILABLE UNDER SEPARATE SOFTWARE LICENSE TERMS AND ARE NOT PART OF A TIBCO PRODUCT. AS SUCH, THEY ARE NOT COVERED BY THE TERMS OF YOUR AGREEMENT WITH TIBCO, INCLUDING ANY TERMS CONCERNING SUPPORT, MAINTENANCE, WARRANTIES, AND INDEMNITIES. DOWNLOAD AND USE THESE ITEMS IS SOLELY AT YOUR OWN DISCRETION AND SUBJECT TO THE LICENSE TERMS APPLICABLE TO THEM. BY PROCEEDING TO DOWNLOAD, INSTALL OR USE ANY OF THESE ITEMS, YOU ACKNOWLEDGE THE FOREGOING DISTINCTIONS BETWEEN THESE ITEMS AND TIBCO PRODUCTS.

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, the TIBCO logo, Two-Second Advantage, TIB, Information Bus, Rendezvous, TIBCO Rendezvous, and TIBCO Messaging Appliance 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.

Copyright © 2017 TIBCO Software Inc. All rights reserved.
TIBCO Software Inc. Confidential Information

Contents

TIBCO Documentation and Support Services 5

Overview 6

Subscribing to TIBCO BusinessWorks™ Container Edition and Plug-ins for AWS Marketplace 6

Metering Workflow for Flexible Consumption Pricing (FCP) 8

Flexible Consumption Pricing Model 8

Supported Plug-ins 10

Cancellation and Termination 10

Launching TIBCO BusinessWorks™ Container Edition and Plug-ins for AWS Marketplace 11

Launching Single Amazon Machine Image (AMI) 11

Launching CloudFormation Template to Set up AWS ECS Cluster 15

Launching CloudFormation Template to create and extend TIBCO BusinessWorks™ Container Edition Base Docker Image 22

Cloud Formation Template Behavior 28

Application Development for Docker 29

Switching the Container Platform 29

Starting Studio in the Docker Mode 29

Environment Variables 30

Using Configurations from Configuration Management Services 32

Deploying an application on ECS 34

Amazon Terminology 37

TIBCO Documentation and Support Services

Documentation for this and other TIBCO products is available on the TIBCO Documentation site. This site is updated more frequently than any documentation that might be included with the product. To ensure that you are accessing the latest available help topics, visit:

https://docs.tibco.com

Product-Specific Documentation

The following documents for this product can be found on the TIBCO Documentation site:

- *BusinessWorks Container Edition and Plug-Ins for AWS Marketplace User's Guide*

How to Contact TIBCO Support

For comments or problems with this manual or the software it addresses, contact TIBCO Support:

- 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.

How to Join TIBCO Community

TIBCO Community is an online destination for TIBCO customers, partners, and resident experts. It is a place to share and access the collective experience of the TIBCO community. TIBCO Community offers forums, blogs, and access to a variety of resources including product wikis that provide in-depth information, white papers, and video tutorials. In addition, users can submit and vote on feature requests via the Ideas portal. For a free registration, go to https://community.tibco.com.

Overview

Using TIBCO BusinessWorks™ Container Edition and Plug-ins for AWS, you can quickly and easily connect API's, microservices and backend systems. With easy-to-use drag-and-drop graphical development environment, graphical data mapper and a vast library of connectors, you can quickly and easily create cloud-native integration applications and deploy them on AWS leveraging native features of AWS Elastic Container Service or your choice of Docker-based PaaS build on AWS for container management.

The following are products available for TIBCO BusinessWorks™ Container Edition and Plug-ins:

- **Flexible Consumption Pricing (FCP)**- Consumption-based pricing model helps you to pay only for the number of containers running per hour. This gives you the flexibility to scale on demand and manage software cost as you go.
- **Bring Your Own License (BYOL**) - With BYOL model, you can use your existing TIBCO BusinessWorks Container Edition licenses and run applications on AWS, leveraging the AMI (Amazon Machine Image) or CloudFormation templates provided on AWS Marketplace.

You must have basic knowledge of TIBCO BusinessWorks Container Edition to use TIBCO BusinessWorks™ Container Edition and Plug-ins for AWS. For more information, See the TIBCO BusinessWorks Container Edition documentation.

This chapter contains the following sections:

- Subscribing to TIBCO BusinessWorks Container Edition and Plug-ins for AWS Marketplace
- Metering Workflow for Flexible Consumption Pricing (FCP)
- Flexible Consumption Pricing Model
- Supported Plug-ins
- Cancellation and Termination

Subscribing to TIBCO BusinessWorks™ Container Edition and Plug-ins for AWS Marketplace

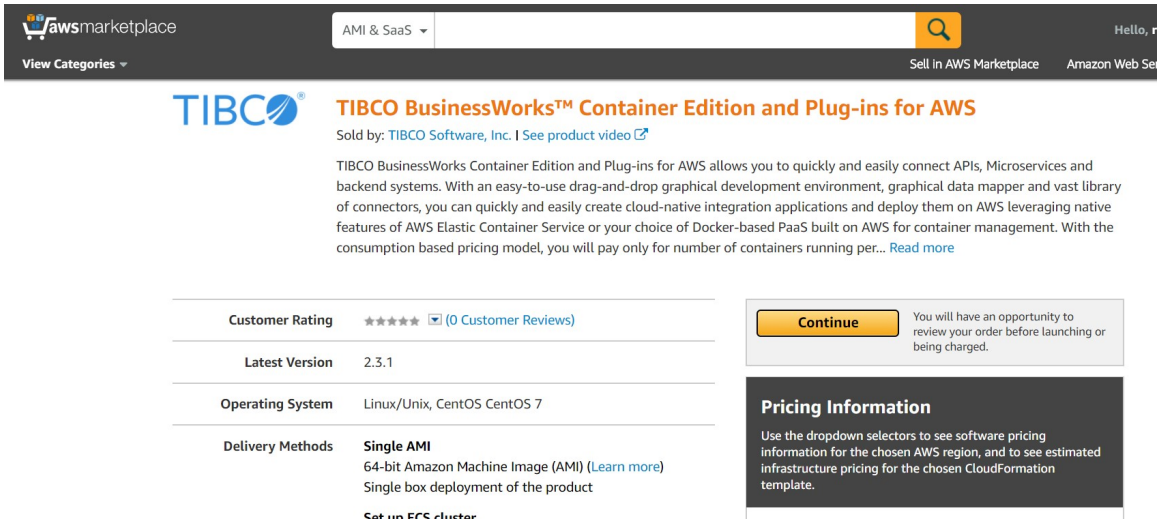
To get started, you must first access AWS marketplace and subscribe to TIBCO BusinessWorks™ Container Edition and Plug-ins for AWS Marketplace.

Prerequisites

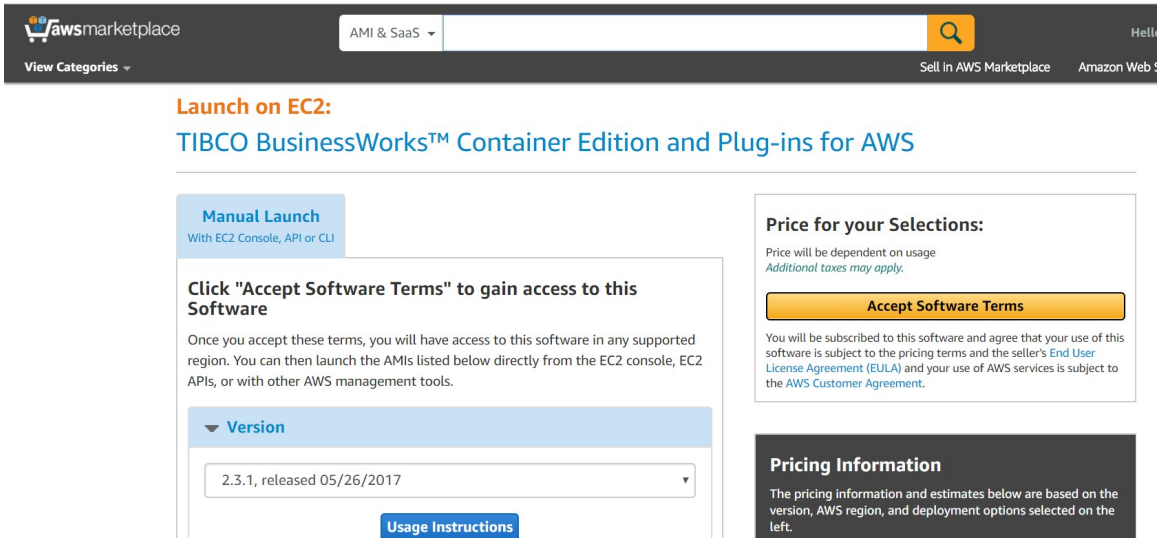
You'll need a few things before you install and run TIBCO BusinessWorks Container Edition for AWS Marketplace on Amazon Web Services. Ensure that you have an AWS account. To create an AWS account, go to the Amazon Web Services sign up page, click the Sign Up button, and follow the instructions.

Procedure

1. Log in to Amazon Web Services Marketplace account.
2. Use the search option to locate TIBCO BusinessWorks Container Edition and Plug-ins for AWS on the AWS Marketplace.
3. Click **Continue** to go to the Launch page.

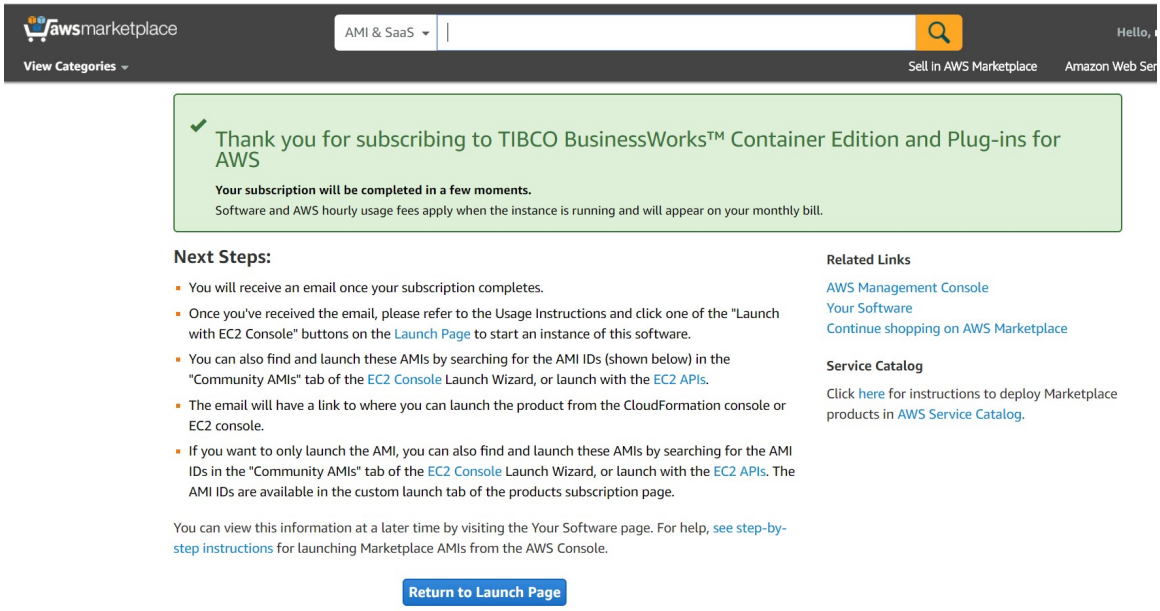


4. To subscribe to TIBCO BusinessWorks Container Edition and Plug-ins for AWS, verify the information on this page and click **Accept Software terms**.



After subscribing to the TIBCO BusinessWorks Container Edition and Plug-ins for AWS, you can launch the Single AMI or AWS CloudFormation Stack using CloudFormation template. For more information, see Launching TIBCO BusinessWorks™ Container Edition and Plug-ins for AWS for AWS Marketplace

5. Click **Return to Launch Page** to start using TIBCO BusinessWorks Container Edition software.



Metering Workflow for Flexible Consumption Pricing (FCP)

TIBCO BusinessWorks™ Container Edition and Plug-ins for AWS Marketplace starts an aggregator service on every EC2 instance, launched through Single AMI or CloudFormation Template option. This service MUST be running on the EC2 instance. Any TIBCO BusinessWorks Container Edition application started in a Docker container on the EC2 instance, it automatically connects to the aggregator and sends its metering units every hour. The aggregator service aggregates metering units from all the containers running on the EC2 instance and invokes AWS Marketplace Metering Service to send metering record every hour.

Scenario 1

If aggregator service is down,

- TIBCO BusinessWorks Container Edition application tries to send metering units to aggregator hourly.
- If the hourly invocation fails due to TIBCO BusinessWorks Container Edition application not receiving any response from aggregator, the application tries to send the metering units to aggregator in the next hour.
- If 2 retry attempt fails, then TIBCO BusinessWorks Container Edition application will be stopped.

Scenario 2

If aggregator service fails to send metering records to AWS metering service,

- TIBCO BusinessWorks Container Edition application tries to send metering units to aggregator hourly.
- If the hourly invocation fails due to TIBCO BusinessWorks Container Edition application receiving any status code other than 200 from aggregator, the application tries to send the metering units to aggregator in the next hour.
- If 3 retry attempts fail, then TIBCO BusinessWorks Container Edition is stopped.

Flexible Consumption Pricing Model

TIBCO BusinessWorks Container Edition and Plug-ins for AWS Marketplace uses consumption-based pricing model to calculate the cost of running TIBCO BusinessWorks Container Edition application on

AWS Marketplace Metering Service. You pay only for the number of containers running per hour. This gives you flexibility to scale on demand and manage software cost on the go. All charges are incurred hourly by the customer. Usage is calculated and billed monthly.

Software Cost for FCP

TIBCO BusinessWorks Container Edition and Plug-ins for AWS Marketplace uses "Consumption Unit" as a Flexible Consumption Pricing (FCP) dimension when sending metering records to AWS Marketplace Metering Service. The metering service uses this dimension to meter the TIBCO BusinessWorks Container Edition and Plug-ins consumption per hour and charges the customer based on the consumption.

One TIBCO BusinessWorks Container Edition application docker container uses 5 TIBCO BusinessWorks consumption units per hour and one ActiveMatrix BusinessWorks plug-in uses 2 TIBCO BusinessWorks consumption units per hour.

For Example,

Assume that 1 TIBCO BusinessWorks Container Edition application container is running for 10 hours and the TIBCO BusinessWorks Container Edition application uses 3 BusinessWorks Plug-ins supported with TIBCO BusinessWorks Container Edition for AWS marketplace:

1 TIBCO BusinessWorks Container Edition App Container = 5 TIBCO BusinessWorks Consumption Units per hour

1 BW Plug-in = 2 TIBCO BusinessWorks consumption units per hour

Total units consumed per hour = 1*5 + 3*2 = 11 TIBCO BusinessWorks Consumption Units

Total units consumed in 10 hours = 10*11 = 110 TIBCO BusinessWorks ConsumptionUnit Hrs

Total Price for the app container = 110 * Price for 1 TIBCO BusinessWorks Consumption Unit

Price per consumption unit is listed on the AWS Marketplace page for TIBCO BusinessWorks Container Edition and Plug-ins. On Billing dashboard for your AWS account, you can see the total ConsumptionUnitHrs for TIBCO BusinessWorks Container Edition and Plug-ins and the software cost is calculated based on consumption.

For BYOL, software cost is not paid through AWS Marketplace. Users can use their existing TIBCO BusinessWorks Container Edition software licenses purchased from TIBCO.

AWS Infrastructure Cost

Users incur associated AWS infrastructure charges depending on the services and infrastructure used such as AWS EC2, S3, EBS and so on. These rates and fees are defined and controlled by AWS and can vary between regions.

We do not recommend user to stop and restart same container (same container id) within one hour of billing. You will be charged on the basis of restart attempts made within an hour. Instead of restarting the same container, you can terminate the existing container and start another container from same docker image.

If the target group is not configured correctly for an application, the health check fails and the application is stopped. I this scenario, the new container spins up and you are charged for the each new container spinned up automatically till the time you fixes the target group or deletes the application service.

Annual Premium Subscription

Users can contact TIBCO Sales to purchase Premium Subscription for TIBCO BusinessWorks Container Edition and Plug-ins for AWS Marketplace. This subscription includes right to open up to 12 or unlimited Support Requests expiring one year from the date of purchase.

Supported Plug-ins

The following plug-ins are supported by TIBCO BusinessWorks™ Container Edition and Plug-ins on Amazon Web Services Marketplace.

Sr. No	ID	Plug-In Name	Version
1	DC	TIBCO ActiveMatrix BusinessWorks™ Plug-in for Data Conversion	4.5.1
2	DCRM	TIBCO ActiveMatrix BusinessWorks™ Plug-in for Microsoft Dynamics CRM	6.4.0
3	MongoDB	TIBCO ActiveMatrix BusinessWorks™ Plug-in for MongoDB	6.1.1
4	SFDC	TIBCO ActiveMatrix BusinessWorks™ Plug-in for Salesforce.com	6.2.1
5	ServiceNow	TIBCO ActiveMatrix BusinessWorks™ Plug-in for ServiceNow	6.0.0
6	MQ	TIBCO ActiveMatrix BusinessWorks™ Plug-in for WebSphere MQ	8.5.1
7	OData	TIBCO ActiveMatrix BusinessWorks™ Plug-in for OData	6.0.1
8	AMQP	TIBCO ActiveMatrix BusinessWorks™ Plug-in for AMQP	6.0.1
9	Kafka	TIBCO ActiveMatrix BusinessWorks™ Plug-in for Apache Kafka	6.0.0
10	S3	TIBCO ActiveMatrix BusinessWorks™ Plug-in for Amazon S3	6.0.0

Cancellation and Termination

Users have the right to cancel their subscription for TIBCO BusinessWorks™ Container Edition and Plug-ins for AWS Marketplace at any time.

TIBCO may remove the listing from the AWS Marketplace at any time with prior notice in accordance with the AWS Marketplace Listing Guidelines. AWS may also remove the listing from the AWS Marketplace at any time for any reason. Upon any removal by TIBCO or AWS of the listing from the AWS Marketplace, existing users can continue to exercise their rights in Marketplace Listing for the remainder of the term the user has acquired rights to use Marketplace Content or, if longer, 90 days after removal. Upon termination of the Agreement, AWS may terminate all access to the Marketplace Content. We strongly advise existing users that they back up their data prior to expiration or cancellation, as TIBCO will not provide access to the customer's data after termination or cancellation.

Launching TIBCO BusinessWorks™ Container Edition and Plug-ins for AWS Marketplace

TIBCO BusinessWorks Container Edition provide the following deployment options on AWS Marketplace.

- Single AMI
- CloudFormation Template to setup AWS ECS Cluster.
- CloudFormation Template to create and extend TIBCO BusinessWorks Container Edition base docker image and push installers to S3

Launching Single Amazon Machine Image (AMI)

You can create an EC2 instance from Amazon Machine Image (AMI). After you subscribe to TIBCO BusinessWorks Container Edition and Plug-ins on AWS marketplace, you can launch the AMI from the marketplace page.
For more details about AMI, Refer AWS Documentation.

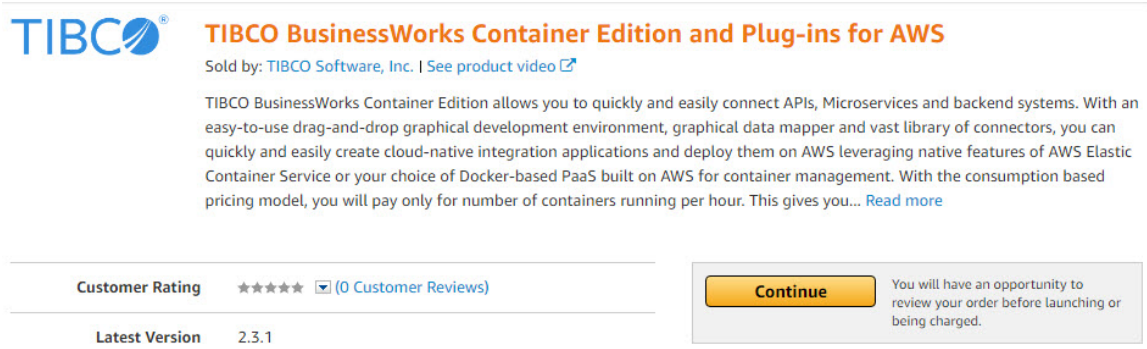
Prerequisites

Ensure that you have subscribed to TIBCO BusinessWorks Container Edition and Plug-ins. See Subscribing TIBCO BusinessWorks Container Edition and Plug-ins on AWS Marketplace.

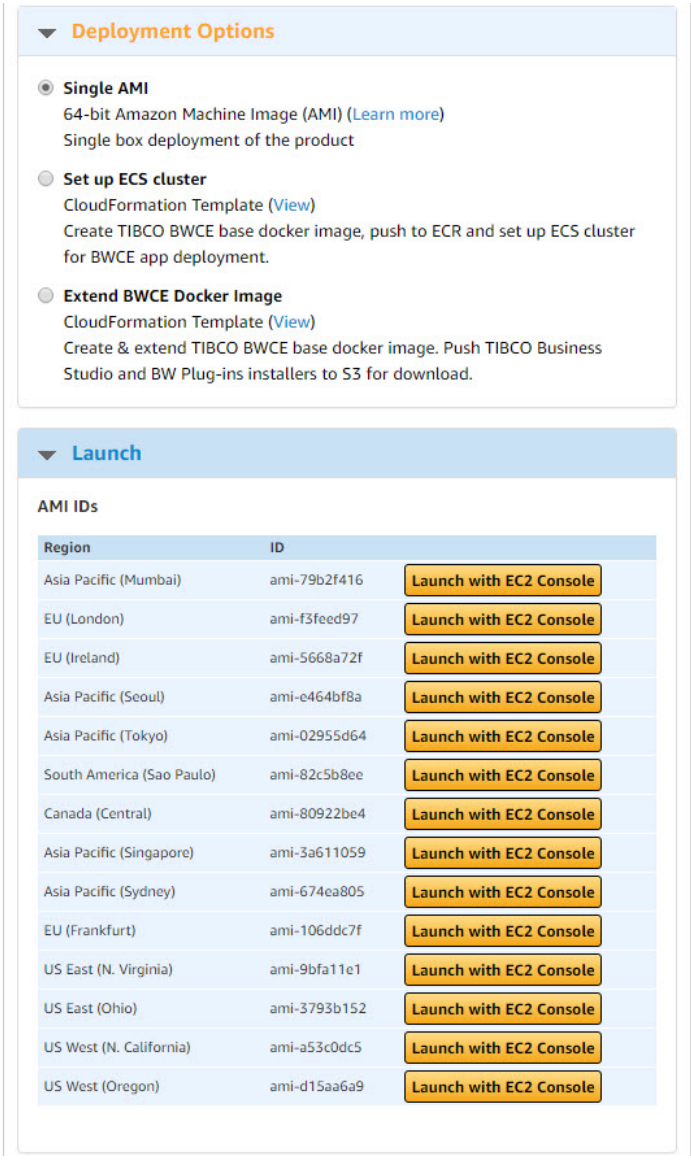
Procedure

1. Open the TIBCO BusinessWorks Container Edition and Plug-ins for AWS web page in AWS Marketplace.
2. In the **Pricing Information** section, select the region where you want to make the software available and specify the delivery method as **Single AMI**.

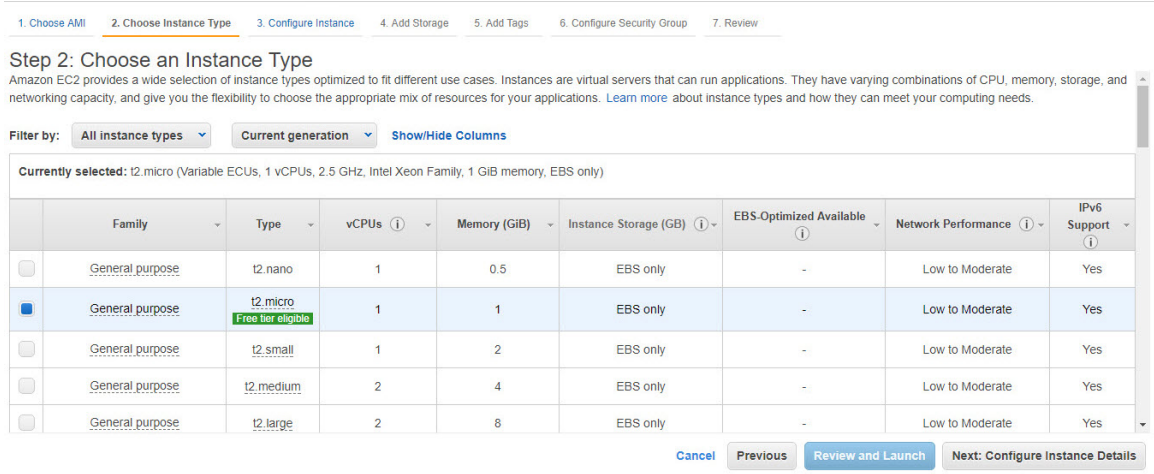
3. Click **Continue** to go to the Launch page.



4. Click the **Launch with EC2 Console** button for you region to launch the EC2 instance from the AMI.

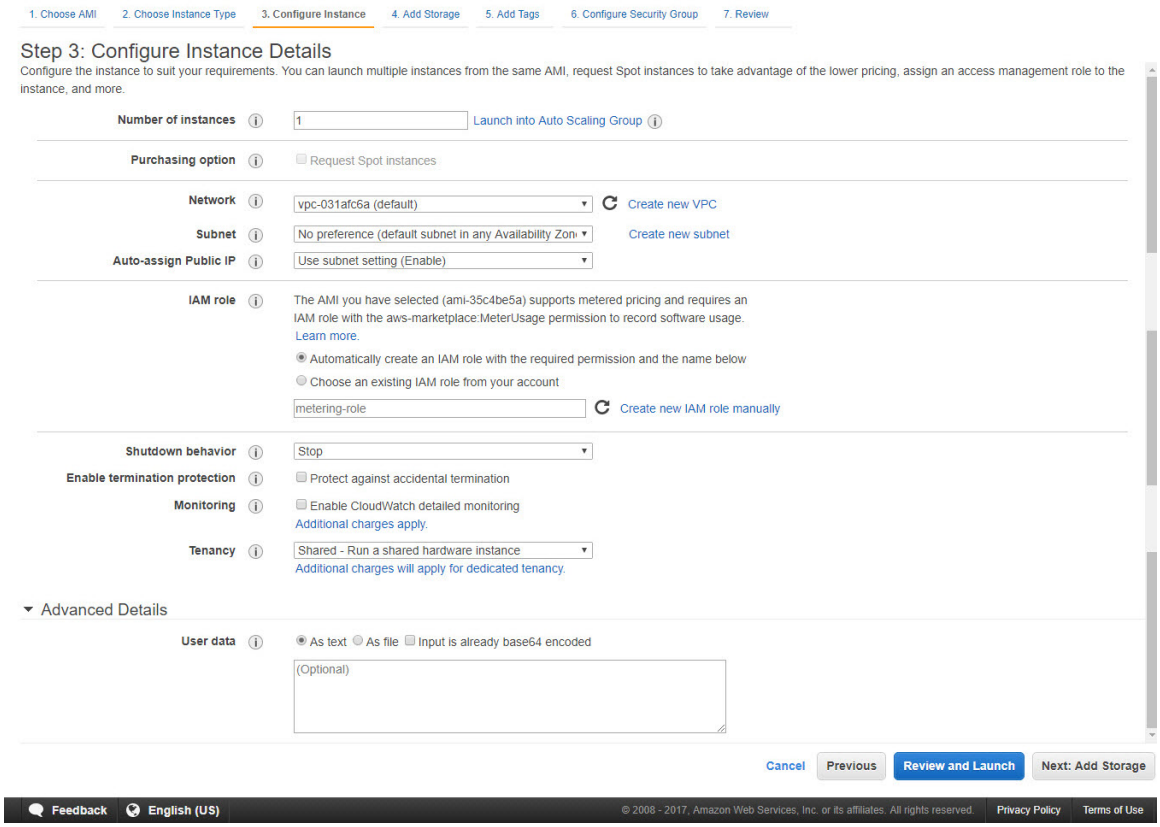


5. From **All instance types** list, select an instance type. For more information, see Amazon EC2 Instance Types

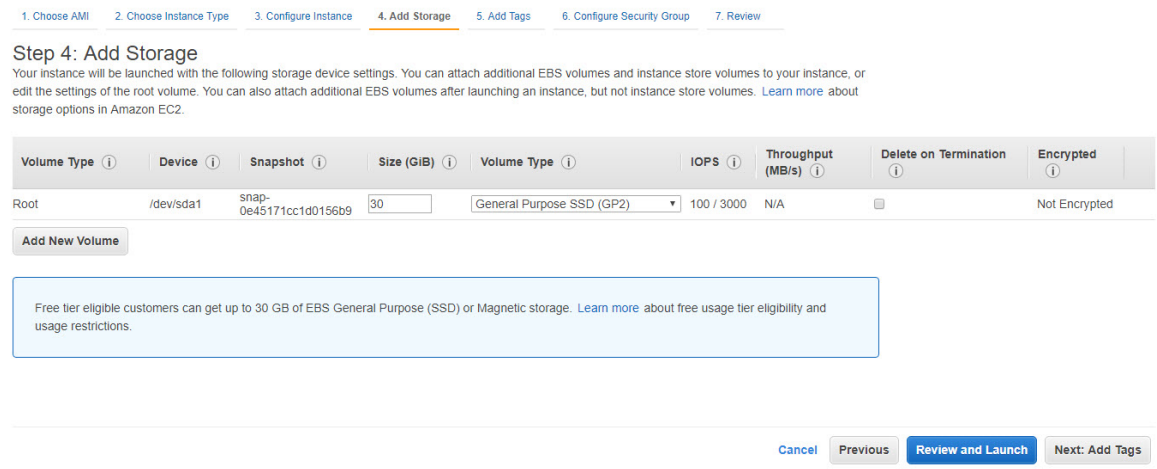


6. Click **Next: Configure Instance Details** at the bottom of the page, and provide the following details as per requirement. See the ⓘ icon for descriptions of each item.
   - Number of instances
   - Purchasing option
   - Network
   - Availability Zone (if using EC2) or Subnet (if using VPC)
   - Auto-assign Public IP
   - IAM role

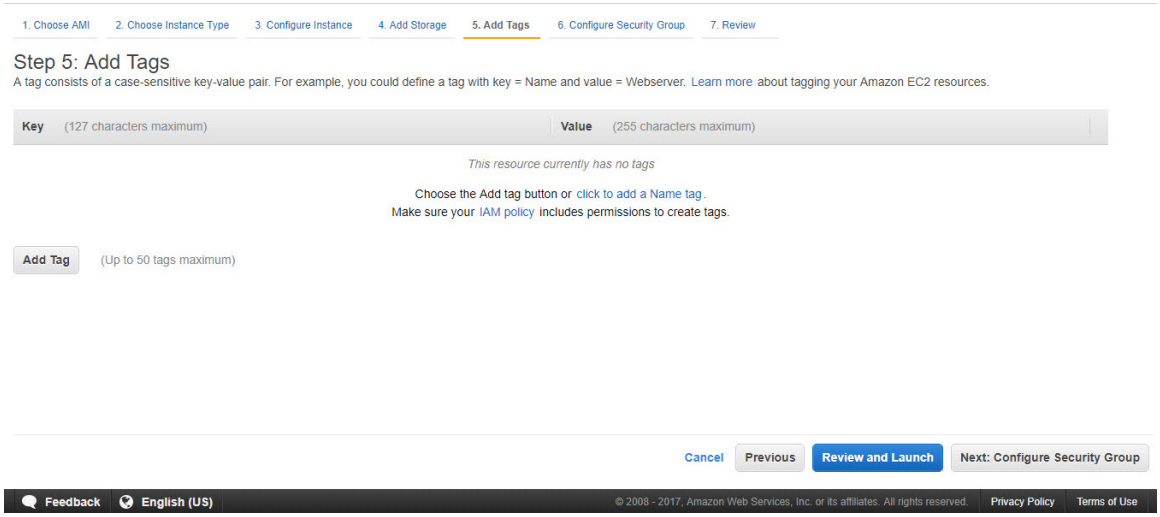
     IAM role must have permission for action `aws-marketplace:MeterUsuage`.
   - Shutdown behavior
   - Enable termination protection
   - Monitoring
   - Tenancy
7. Expand the **Advanced Details** section and fill out the **User data** field if required.



8. Click **Next: Add Storage** and review the storage details. You can add Elastic Block Store and instance store volumes by clicking the **Add New Volume** button.

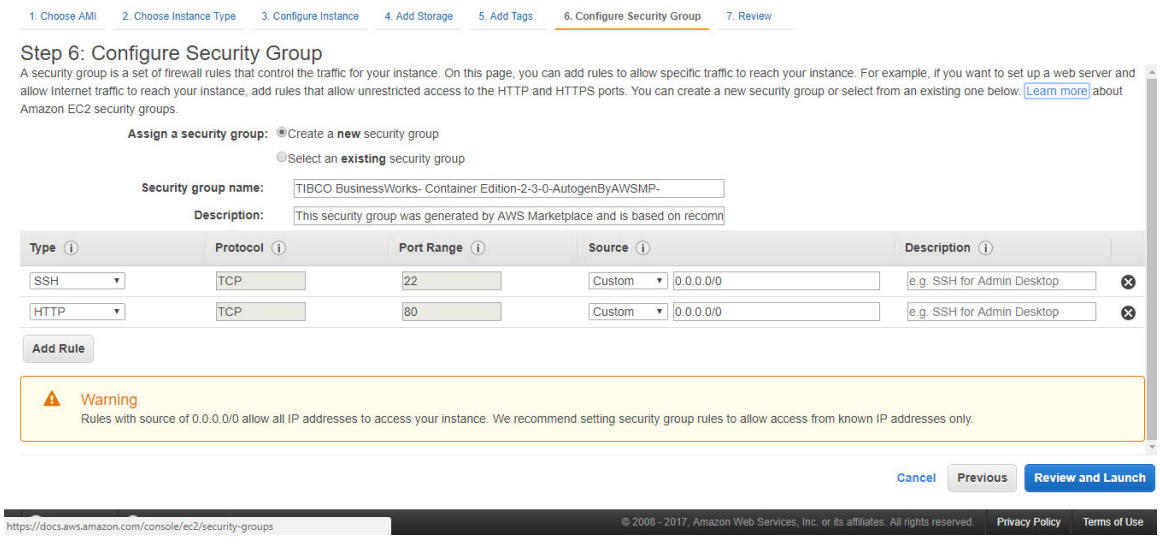


9. Click **Next: Tag Instance** and add optional tags. To add more tags, click the **Create Tag** button.



A tag is a key or value pair that flows to resources inside your stack. You can add up to 10 unique tags to each instance.

10. Click **Next: Configure Security Group** and configure your firewall rules. Choose an existing security group or create a new security group.



11. Click **Review and Launch**. Review your information and edit if necessary.
12. Click **Launch**.

    A message is displayed, informing you that the instance is now launching. Your instances might take a few minutes to launch, depending on the software you are running.
13. Click **Close**.

Launching CloudFormation Template to Set up AWS ECS Cluster

CloudFormation template creates a CloudFormation Stack which builds a base TIBCO BusinessWorks™ Container Edition and Plug-ins docker image, creates an ECR repository with name 'tibco-bwce' and pushes base docker image into the ECR repository. It then deploys a VPC, with a pair of public and private subnets spread across two availability zones. It deploys an Internet Gateway, with a default route on the public subnets. It deploys a pair of NAT Gateways (one in each AZ) and default routes for them in the private subnets. It then deploys a highly available ECS cluster of EC2 instances (ECS Hosts) launched from the TIBCO BWCE AMI in an AutoScaling Group, with the EC2 instances distributed

across multiple Availability Zones. This template also provides you an option to create an Amazon S3 bucket and folder structure inside the bucket required for base docker image customization.

Prerequisites

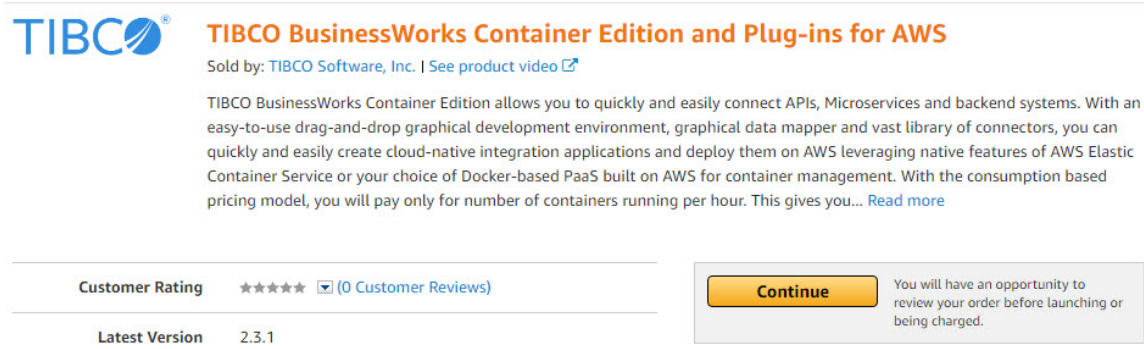
Ensure that you have subscribed to TIBCO BusinessWorks Container Edition and Plug-ins on AWS Marketplace. For more information, see Subscribing TIBCO BusinessWorks Container Edition and Plug-ins on AWS Marketplace.

Procedure

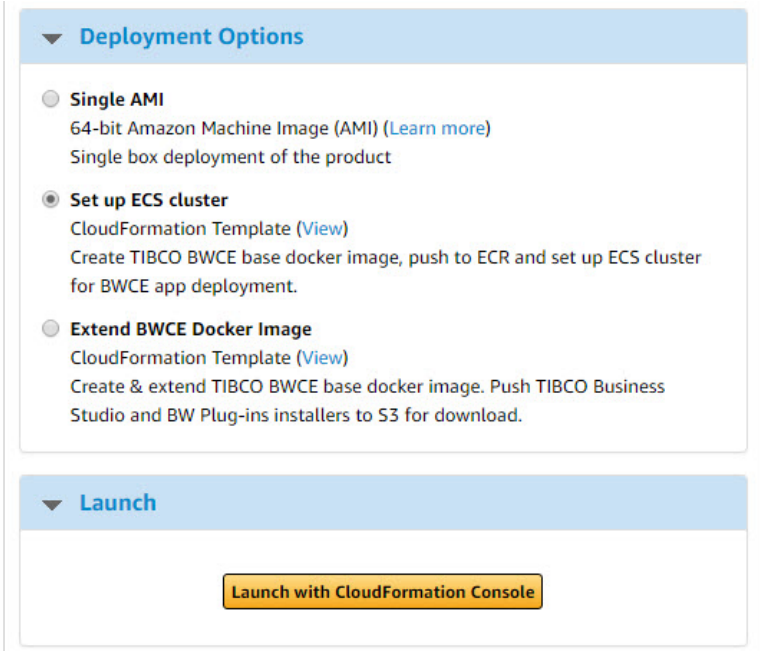
1. Open the TIBCO BusinessWorks Container Edition and Plug-ins for AWS web page.
2. In the **Pricing Information** section, select the region where you want to make the software available and specify the delivery method as **Setup ECS Cluster**.

Ensure that the ECS is supported in some regions only. For more information, see Region Table.

3. Click **Continue** to go to the Launch page.

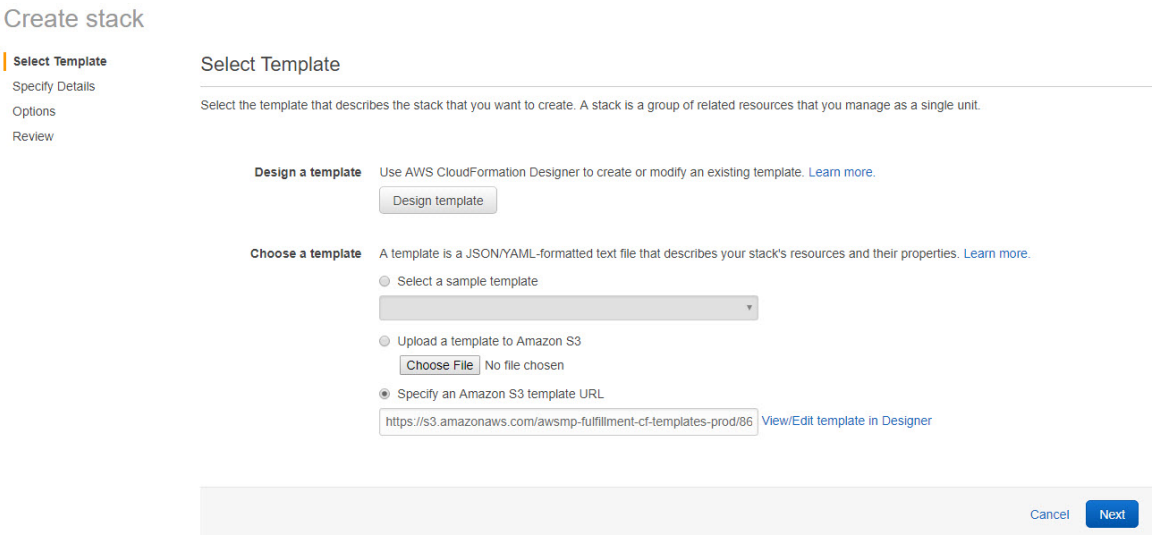


4. Click the **Launch with CloudFormation Console** button.

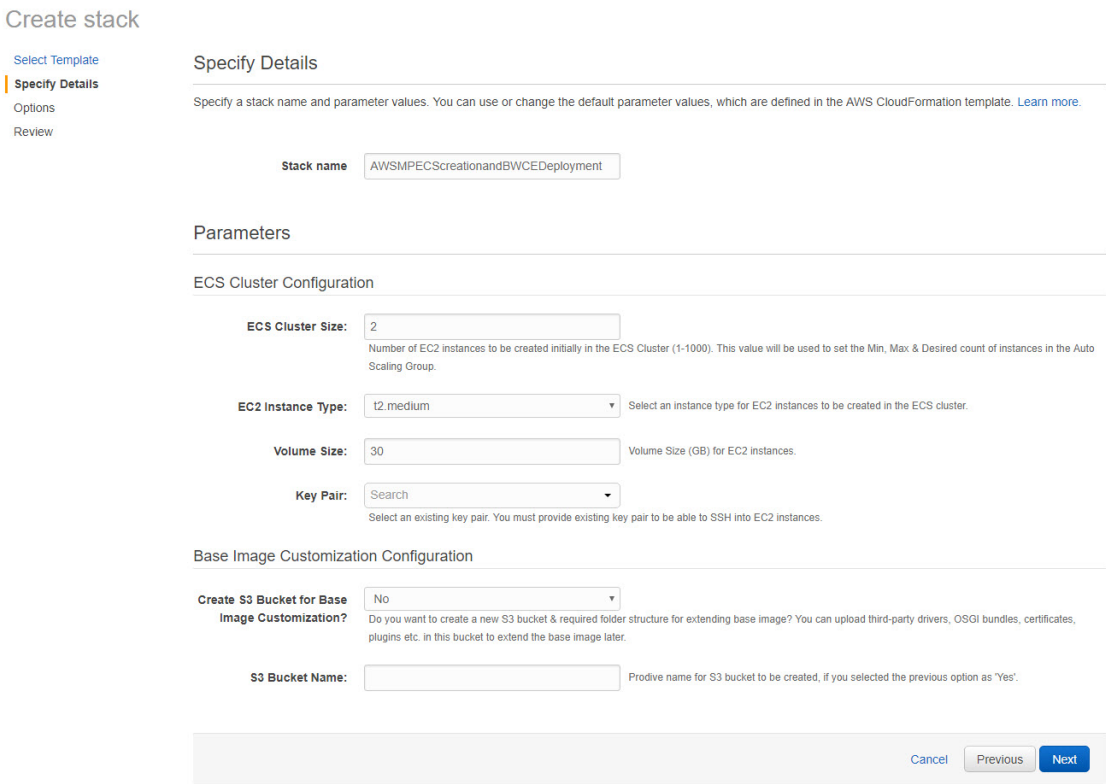


The Select Template page is displayed.

By default, AWS provides a template source URL. Do not change this.



5. Click **Next**. The Specify Details page is displayed.



6. In the **Stack Name** field, give a unique name to your CloudFormation stack .

> Stack name must not exceed 24 characters.

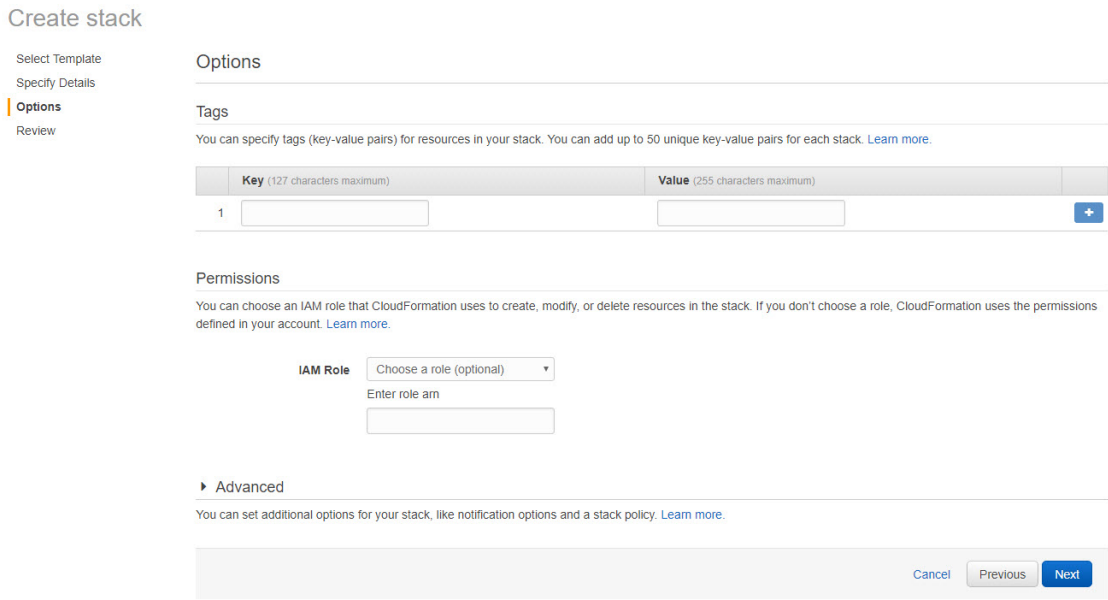
7. In the **ECS Cluster Size** field, enter the number of ECS hosts you want to deploy.
8. Select an appropriate **EC2 InstanceType** for the ECS Cluster from the drop-down.

> Performance might vary based on system attributes such as network bandwidth, memory requirements for a given use case and query requirements. For more information about EC2 instance types, see the AWS documentation: http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instance-types.html

9. In the **Key Pair** field (optional), enter an existing key pair name for SSH access to the ECS instance.
10. Select **Yes** to **Create S3 Bucket for Base Image Customization** and specify **S3 Bucket Name**.
    a) After the Stack is in `CREATE_COMPLETE` state, an S3 Bucket is created with the user provided name.
    b) The S3 Bucket consist of following folders:
        - /certs
        - /jars
        - /lib
        - /monitor-agents
        - /plugins
        - /thirdparty-installs
    c) Before running the Extend TIBCO BusinessWorks Container Edition Docker Image CloudFormation template, user must copy required artifacts (optional) in respective folders in the S3 bucket.

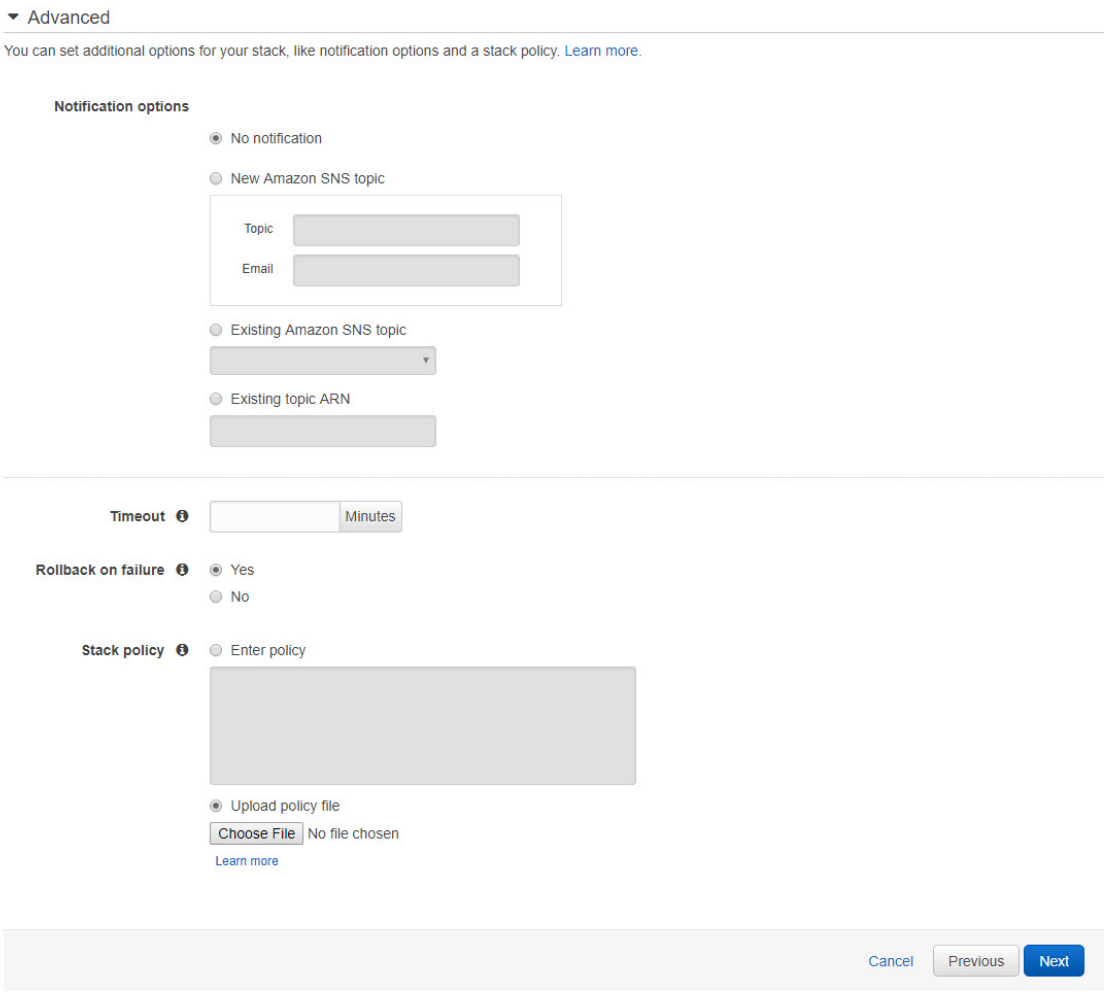
- Copy certificates to /certs folder.
- Copy additional OSGi bundle jars (jars required by plugins, DB drivers, MQ jars etc.) to /jars folder.
- Copy additional libraries to /lib folder.
- Copy TIBCO BusinessWorks Container Edition monitoring agent jar to /monitor-agents folder.
- Copy TIBCO BusinessWorks Container Edition supported plug-in runtime zip files to /plugins folder.
- Copy thirdparty installation zip to /thirdparty-installs folder.

11. Click **Next**. The Options page is displayed.
12. Specify tags (key-value pairs) for resources in your stack.

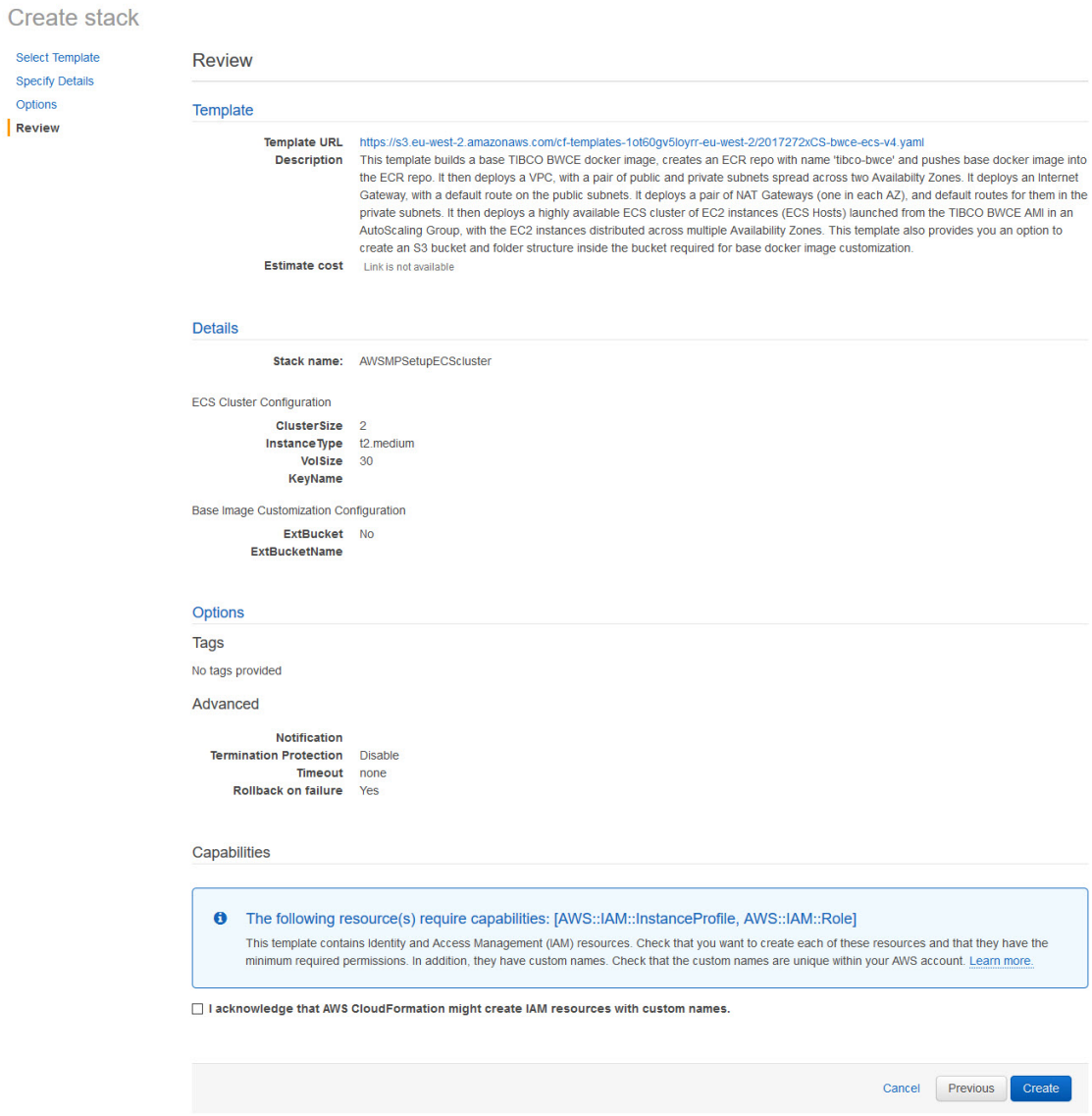


A tag consists of a key or value pair that flows to resources inside your stack. You can add up to 10 unique tags for each instance.

13. Expand the **Advanced** section of the Options page and set your notification, timeout, and other options, if required. Click the Learn more link for an explanation of these options.

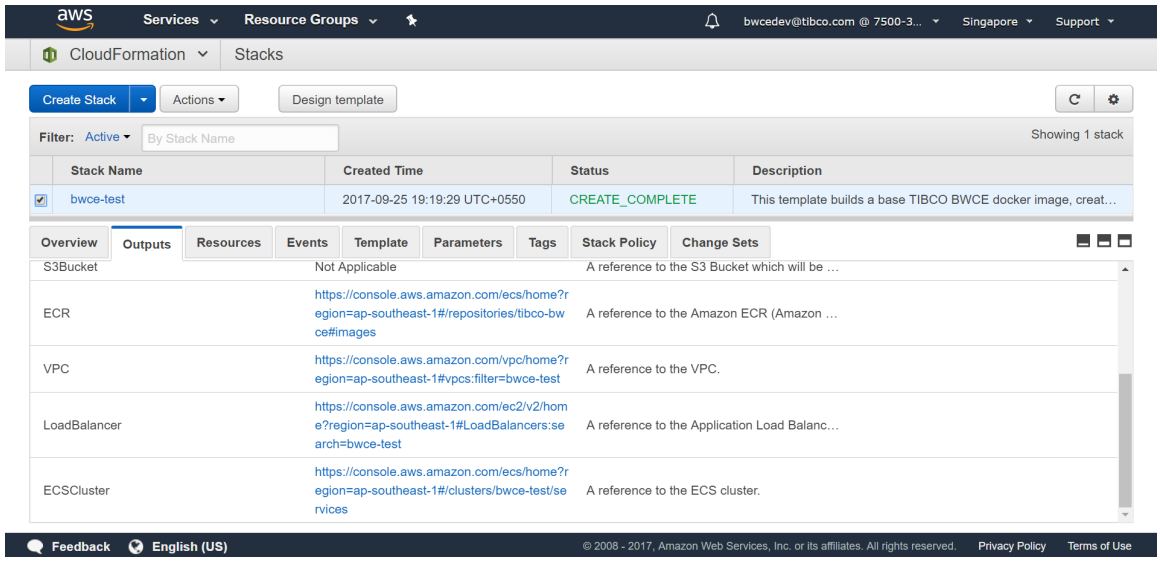


14. Click **Next**. The Review page is displayed.
15. Click the **acknowledgment** check box, then click **Create**.



Result

You see your Stack Name listed in a table. While it is being created, the Status column is displays the status as `CREATE_IN_PROGRESS`. After a few minutes the status should change to `CREATE_COMPLETE`. After the Stack is in `CREAT_COMPLETE` state, the **Outputs tab** contains the URL for various AWS resources created as part of the Stack.



Launching CloudFormation Template to create and extend TIBCO BusinessWorks™ Container Edition Base Docker Image

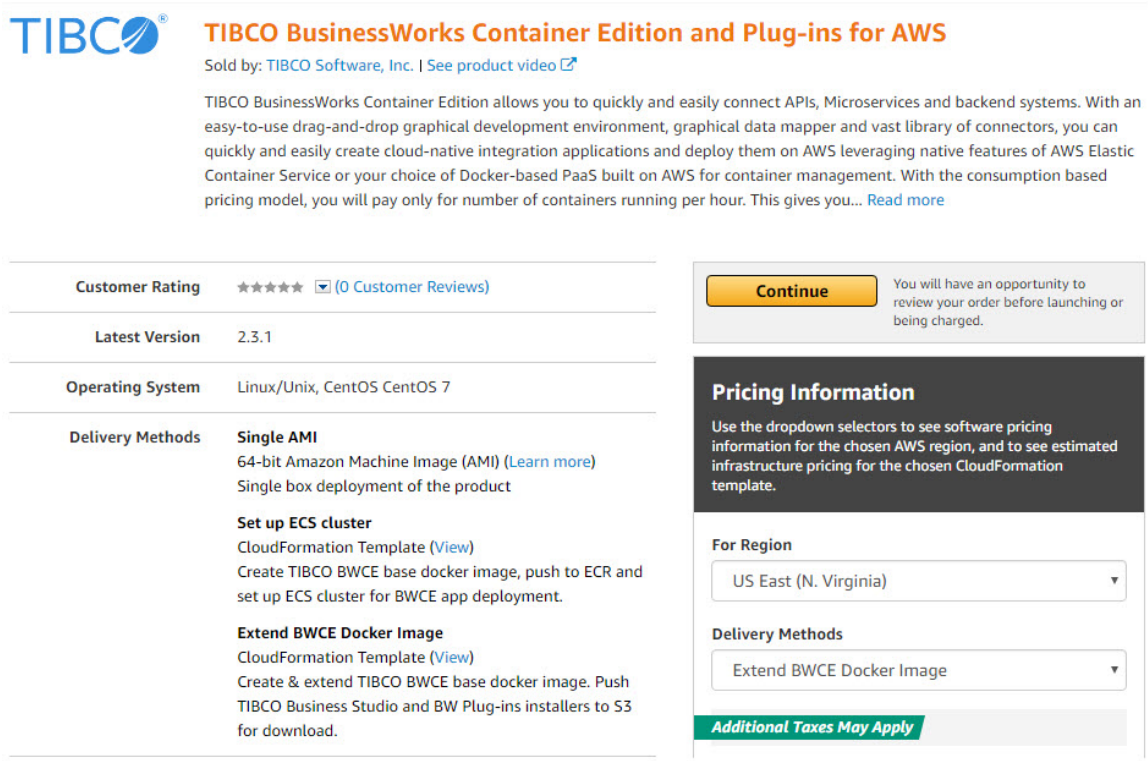
The CloudFormation template creates new ECR repository and pushes TIBCO BusinessWorks Container Edition base docker image into the repository. You can use it to extend the base docker image. If you provide S3 Bucket Name, the template adds all resources uploaded in the S3 bucket to the base image. If you provide List of Plug-ins, the template adds specified BusinessWorks Plug-ins runtime artifacts to the base image. In case, you select **Do you want to push Business Studio and BusinessWorks Plug-ins installer to S3** as **Yes** and provide **S3 Bucket Name**, the TIBCO Business Studio™ Container Edition and Business works Plug-ins installer is copied to S3 bucket.

Prerequisites

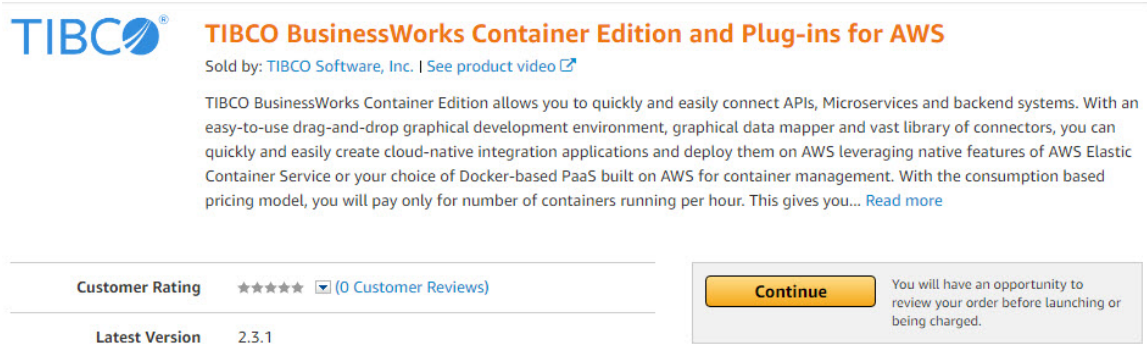
Ensure that you have subscribed to TIBCO BusinessWorks Container Edition on AWS Marketplace. Refer Subscribing to TIBCO BusinessWorks Container Edition and Plug-ins on AWS Marketplace.

Procedure

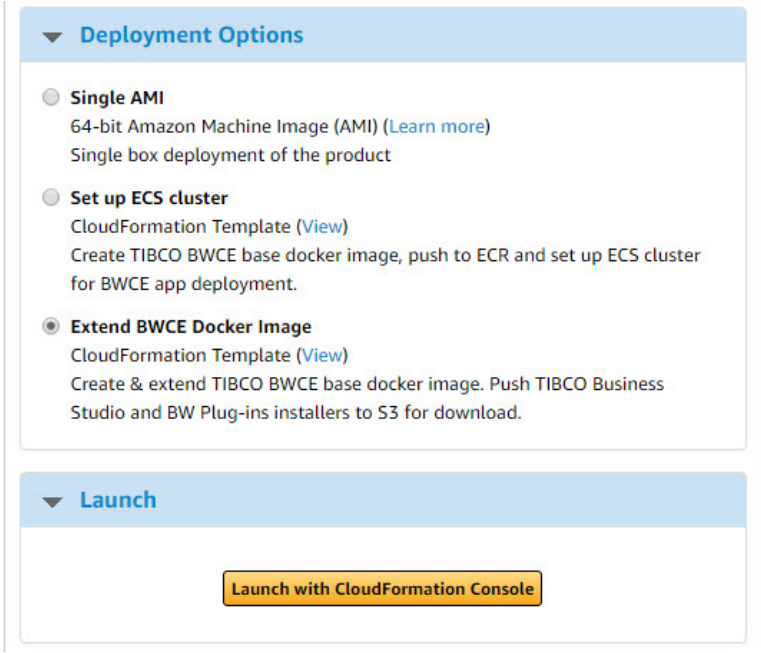
1. Open the TIBCO BusinessWorks Container Edition and Plug-ins for AWS webpage.
2. Select the region where you want to make the software available and select the delivery method as **Extend BWCE Docker Image** from the drop-down list.



3. Click **Continue** to go to the Launch page.

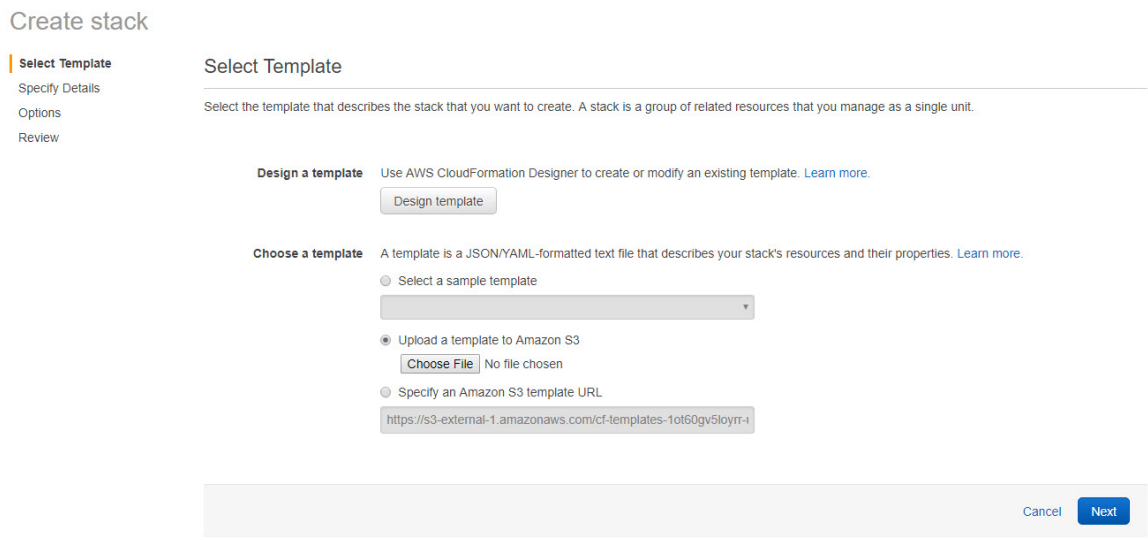


4. Click the **Launch with CloudFormation Console** button.

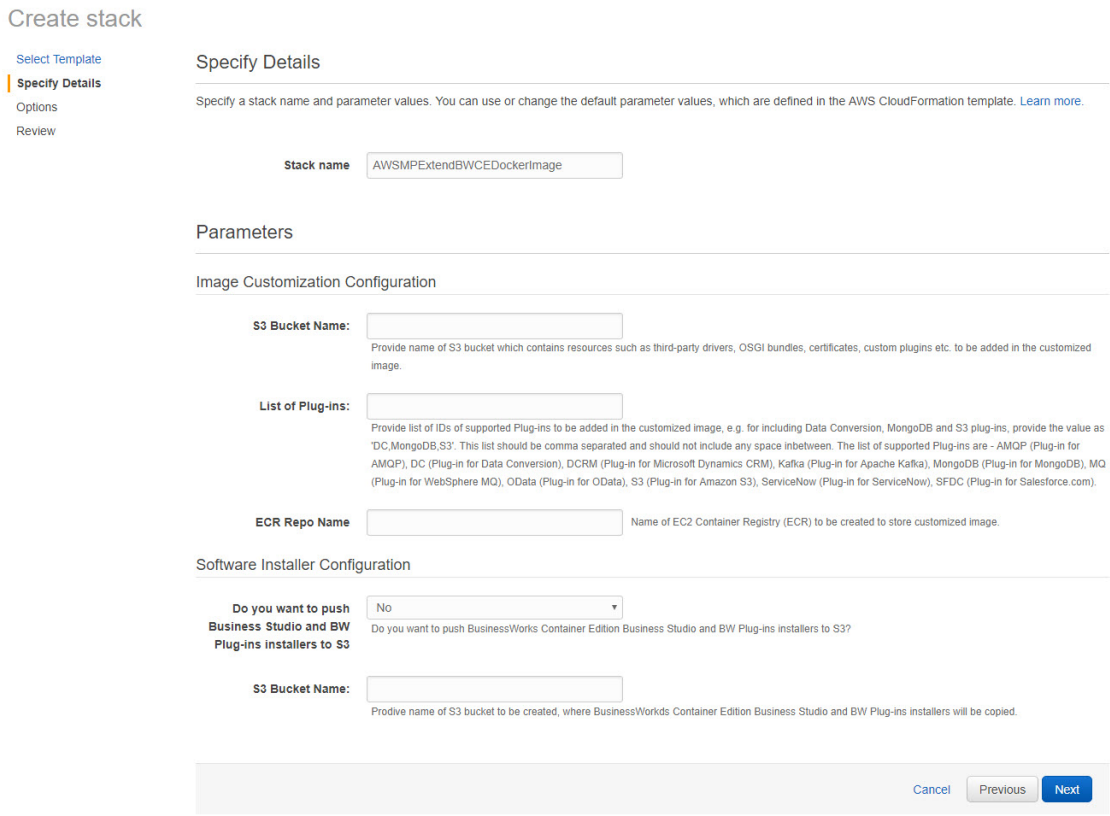


The Select Template page is displayed.

By default, AWS provides a template source URL. Do not change this.



5. Click **Next**. The Specify Details page is displayed.



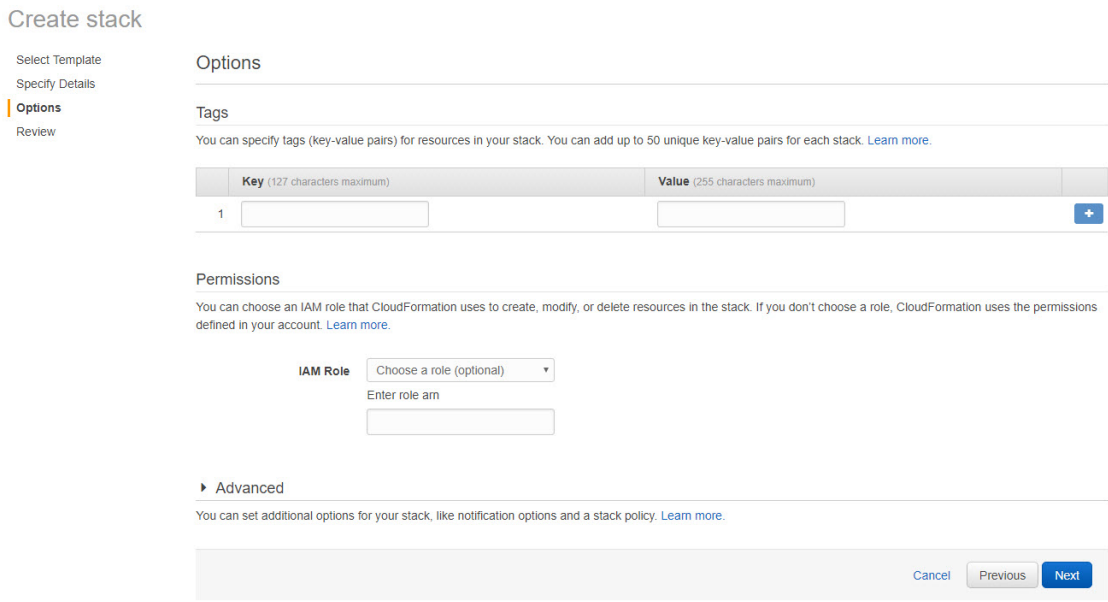
6. In the **Stack Name** field, give a unique name to your CloudFormation stack .
7. In **S3 Bucket Name** field, provide the bucket name from where the artifacts are included in the customized base image.
8. Provide list of IDs of supported Plug-ins to be added in the customized image. For more information, refer Supported Plug-ins.

   List of Plug-ins is case sensitive and must be comma separated with no space in between. This field is not available for TIBCO BusinessWorks Container Edition and Plug-ins for AWS (BYOL). You need to download the plug-in installer from edelivery site and upload it to the plug-in S3 bucket.

9. In the **ECR Repo Name** field, enter the name of EC2 Container Registry (ECR) to be created to store customized image.
10. Select **Yes** to push Business Studio and BW Plug-ins installers to S3 and specify **S3 Bucket Name**.

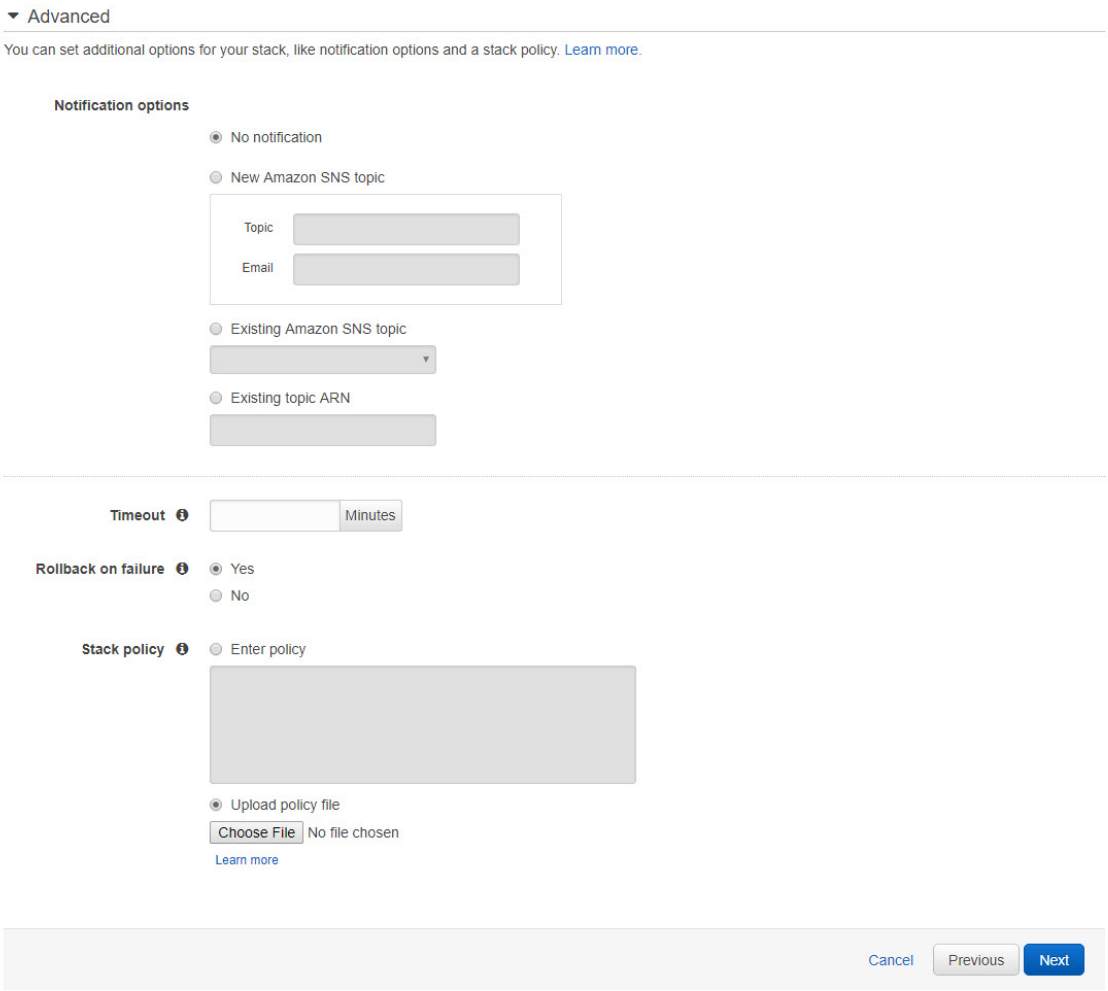
    This field is not available for TIBCO BusinessWorks Container Edition and Plug-ins for AWS (BYOL).

11. Click **Next**. The Options page is displayed.
12. Add any tags you want to simplify administration of your infrastructure.

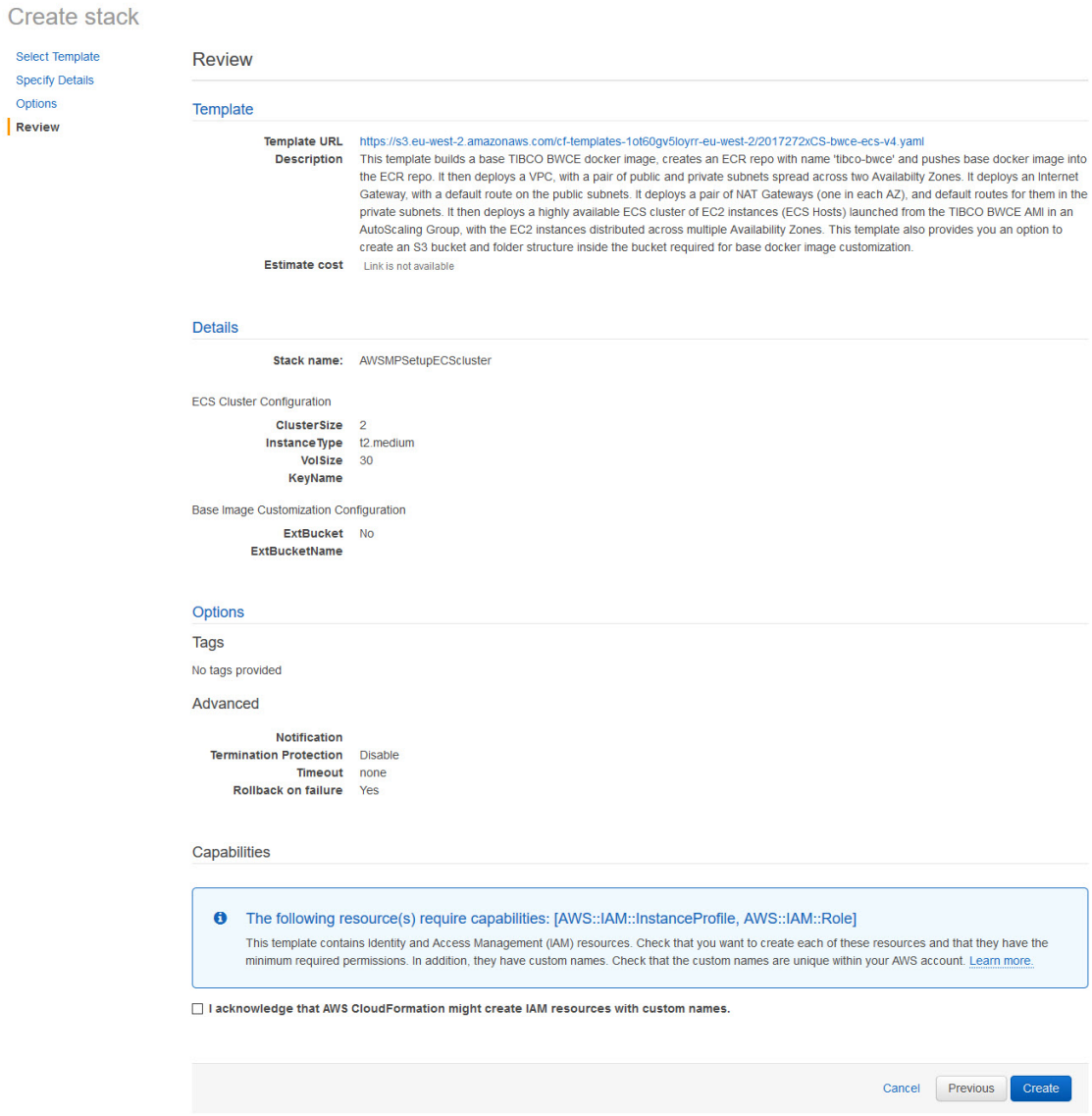


A tag consists of a key-value pair and flows to resources inside your stack. You can add up to 10 unique tags for each instance.

13. Expand the **Advanced** section of the Options page and set your notification, timeout, and other options. Click the **Learn more** link for an explanation of these options if required.



14. Click **Next**. The Review page is displayed.
15. Select the **acknowledgment** check box and click **Create**.
    You see your Stack Name listed in a table. While it's being created the Status column displays `CREATE_IN_PROGRESS`. After a few minutes, the status should change to `CREATE_COMPLETE`.



Cloud Formation Template Behavior

This section describe the creation and deletion behavior of the cloud formation templates.

- **Set up ECS cluster:**

  The following AWS resources are created when ECS Cloud Formation template is used to create a stack.

```
AWS::EC2::VPC
AWS::EC2::Subnet (4)
AWS::EC2::SubnetRouteTableAssociation (4)
AWS::EC2::Route (3)
AWS::EC2::RouteTable (2)
AWS::EC2::EIP (2)
AWS::EC2::NatGateway (2)
AWS::EC2::SecurityGroup (2)
AWS::EC2::VPCGatewayAttachment
AWS::EC2::InternetGateway
AWS::ECS::Cluster
AWS::ElasticLoadBalancingV2::LoadBalancer
AWS::ElasticLoadBalancingV2::TargetGroup
AWS::ElasticLoadBalancingV2::Listener
AWS::AutoScaling::LaunchConfiguration
AWS::AutoScaling::AutoScalingGroup
AWS::IAM::InstanceProfile
AWS::IAM::Role
AWS::ECR::Repository
AWS::S3::Bucket
```

  When you delete the stack, the AWS resources are deleted except for the following resources.

```
AWS::ECR::Repository
AWS::S3::Bucket
```

- **"Extend BusinessWorks Container Edition base Docker image" template:**

  The following AWS resources are created when EXT Cloud Formation template is used to create a stack.

```
AWS::EC2::VPC
AWS::EC2::InternetGateway
AWS::EC2::VPCGatewayAttachment
AWS::EC2::Subnet
AWS::EC2::RouteTable
AWS::EC2::Route
AWS::EC2::SubnetRouteTableAssociation
AWS::EC2::Instance
AWS::IAM::InstanceProfile
AWS::IAM::Role
AWS::ECR::Repository
AWS::S3::Bucket
```

  When you delete the stack, the AWS resources are deleted except for the following resources.

```
AWS::IAM::Role
AWS::ECR::Repository
AWS::S3::Bucket
```

Application Development for Docker

The following section provides information about system module properties and environment variables as they apply to TIBCO BusinessWorks™ Container Edition.

Switching the Container Platform

Procedure

1. In TIBCO Business Studio™ Container Edition, click **Window** > **Preferences** .
2. In the Preferences dialog click **BusinessWorks Container Edition** > **Container Platform**.
3. Choose **Docker**.
4. Click **Apply** and then **OK**.

What to do next

TIBCO Business Studio™ Container Edition has to be restarted for the changes to take effect.

TIBCO recommends that you clean your workspace after TIBCO Business Studio™ Container Edition is restarted.

This option can be accessed from the TIBCO Business Studio™ Container Edition menu **Project** > **Clean**.

Starting Studio in the Docker Mode

To start TIBCO Business Studio™ Container Edition in the Docker mode add the property `ContainerTarget` and set the value to `Docker` in the following file:

- *TIBCO_HOME*`\studio\`*<version>*`\eclipse\configuration\config.ini`

Here is a snippet of a sample `config.ini` file:

```
eclipse.application=org.eclipse.ui.ide.workbench
eclipse.p2.data.area=@config.dir/../p2
osgi.bundles.defaultStartLevel=4
ContainerTarget=Docker
```

The preference is now set to `Docker` for every new workspace.

Environment Variables

This section lists the environment variables that can be used for TIBCO BusinessWorks™ Container Edition application deployment on Docker and Docker based platforms.

Environment Variable	Default Values	Description
BW_LOGLEVEL	ERROR	Used to set a log level for the TIBCO BusinessWorks™ Container Edition application. The default value is ERROR. Supported values are: • INFO • DEBUG • WARN • ERROR
BW_ENGINE_THREADCOUNT	8	Used to set engine thread count for the TIBCO BusinessWorks™ Container Edition application.
BW_ENGINE_STEPCOUNT	-1	Used to set engine step count for the TIBCO BusinessWorks™ Container Edition application.
BW_APPLICATION_JOB_FLOWLIMIT	n/a	Used to set flow limit for TIBCO BusinessWorks™ Container Edition application.
APP_CONFIG_PROFILE	n/a	Name of the application profile that is to be used from a configuration management system such as ZUUL, Spring Cloud Config etc.
BW_PROFILE	n/a	Used to set the name of the BusinessWorks profile from the application.
BW_JAVA_OPTS	n/a	Used to set Java properties that are used at run time. The properties are specified using name-value pairs and are separated by spaces. For example, `BW_JAVA_OPTS="-Dname=value -Dname=value"`

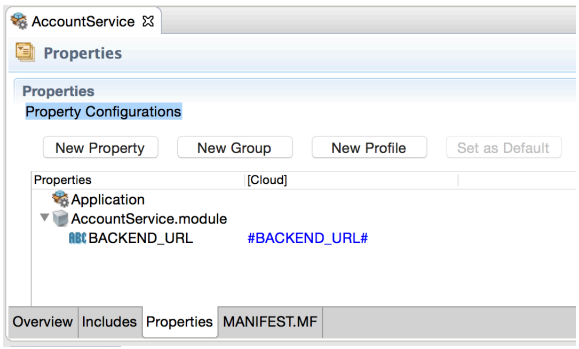
Environment Variable	Default Values	Description
MASHERY_SERVICE_CONFIG	n/a	Applications can pass TIBCO Mashery configuration information using the `MASHERY_SERVICE_CONFIG` environment variable. The value of the environment variable is a JSON string with the required TIBCO Mashery configuration. See Integrating with TIBCO Mashery for more information.
CONSUL_SERVER_URL	n/a	Used to set Consul server configuration. For example, `CONSUL_SERVER_URL=http://127.0.0.1:8085` This must be set if you intend to use Consul for application configuration or for service registration and discovery.
EUREKA_SERVER_URL		Used to set Eureka server configuration. For example, `EUREKA_SERVER_URL=http://127.1.0.1:8080/eureka` This must be set if you intend to use Eureka for service registration and discovery.
MEMORY_LIMIT	1024M	To optimize memory usage at runtime, set this environment variable to the configured memory. For example, `MEMORY_LIMIT=512M` when the Docker container is launched with `512M`.
BW_JMX_CONFIG	n/a	Used to set JMX configuration (RMI host and JMX port) for monitoring TIBCO BusinessWorks™ Container Edition application. The value should be provided in RMI_HOST:JMX_PORT format. For example, `BW_JMX_CONFIG=192.168.99.100:8050`
BW_JAVA_GC_OPTS	-XX:+UseG1GC	Used to set JAVA GC configuration. The value should be one of the standard Java GC VM Options. For example: `BW_JAVA_GC_OPTS=-XX:+UseParallelGC`

Environment Variable	Default Values	Description
-e DISABLE_BWCE_EAR_VALIDATION=true	None	Used to deploy the ActiveMatrix BusinessWorks 6.x application EAR file on TIBCO BusinessWorks Container Edition without converting project to Container Edition and rebuilding EAR file from TIBCO Business Studio Container Edition Ensure that the ActiveMatrix BusinessWorks 6.x EAR file is exported. ActiveMatrix BusinessWorks 6.x EAR file should only have TIBCO BusinessWorks Container Edition supported activities and features.

Using Configurations from Configuration Management Services

You can use configurations from the configuration management services such as Consul by defining a token such as *#<property name>#* in the application properties, where *<property name>* is the name of the configuration parameter.

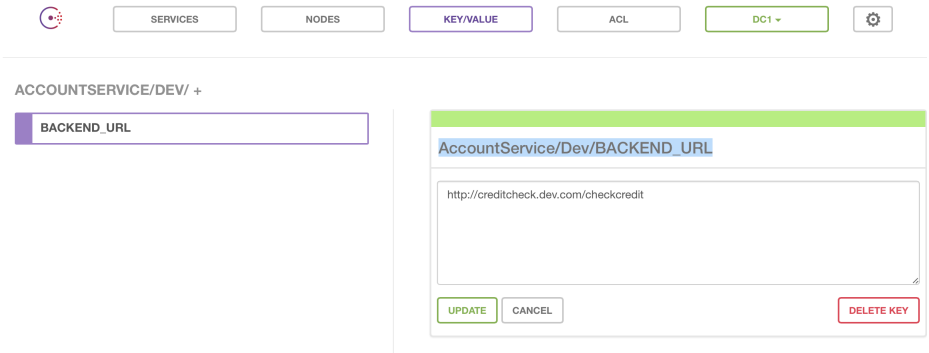
For example, `#BACKEND_URL#`.



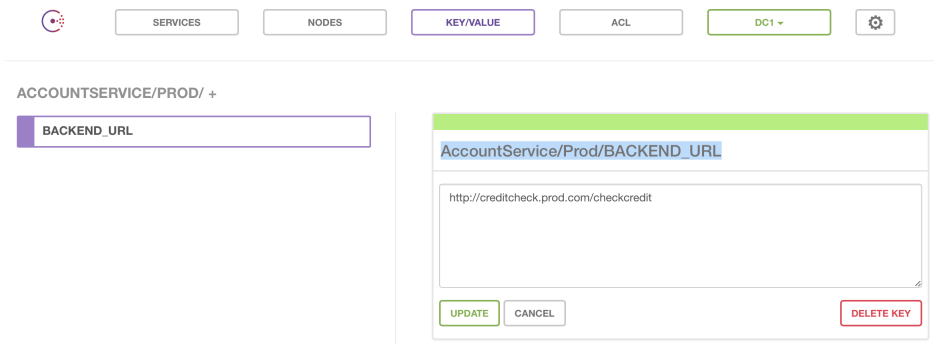
Follow these steps to use configurations from Consul:

1. Set the environment variable `CONSUL_SERVER_URL`. See Environment Variables
2. In your Consul service define the keys using the format `<BWCE_APP_NAME>/<PROFILE NAME>/<KEY Name>`.

   For example, `AccountService/Dev/BACKEND_URL`



`AccountService/Prod/BACKEND_URL`



For more information, see Launching CloudFormation Template to create and extend TIBCO BusinessWorks™ Container Edition Base Docker Image.

Deploying an application on ECS

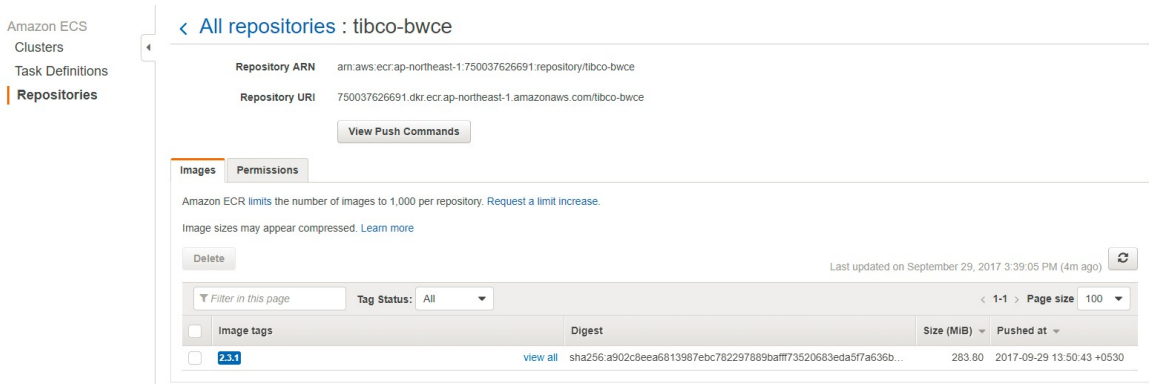
You can deploy an application on ECS and monitor the application logs by configuring the container with CloudWatch.

Prerequisites

- Ensure that you have TIBCO BusinessWorks™ Container Edition application EAR and Docker files in the same directory.
- In the Docker file, ensure that the EAR file name and path is correct.
- In the Docker file, ensure that the base image points to the TIBCO BusinessWorks™ Container Edition runtime base image, which is created by the CloudFormation template.

Procedure

1. To view the runtime base image, go to **AWS Console** > **Services** > **EC2 Container Service** > **Repositories** > **tibco-bwce**.
   The Repository URI is shown as
   ```
   <AWS_account_id>.dkr.ecr.<region_name>.amazonaws.com/tibco-bwce
   ```



To install the AWS CLI and Docker and for more information, see Amazon ECR Documentation.

2. Retrieve the docker login command that is used to authenticate your Docker client with your registry.
   ```
   aws ecr get-login --no-include-email --region <region_name>
   ```

   If you receive an `Unknown options: --no-include-email` error, install the latest version of the AWS CLI. For more details, see Installing the AWS Command Line Interface.

3. Run the docker login command that was returned in the previous step.
   If you are using Windows PowerShell, run the following command.
   ```
   Invoke-Expression -Command (aws ecr get-login --no-include-email --region
   <region_name>)
   ```

4. To generate the application image, navigate to the folder where the EAR and Docker files are stored and run the following command. For information about building a Docker file, see Docker Basics.
   ```
   docker build -t <application_name> .
   ```

   You can skip this step if your image is already built.

5. After the build is ready, tag the image to push it to the repository.
   ```
   docker tag application_name:latest
   <AWS_account_id>.dkr.ecr.<region_name>.amazonaws.com/application_name:latest
   ```

6. To push the image to your newly created AWS repository, run the following command:

```
docker push <AWS_account_id>.dkr.ecr.<region_name>.amazonaws.com/
application_name:latest
```

Ensure that you replace the *<region_name>* with your region such as ap-northeast-1 and Repository URI with the URI.

7. Create the services and task definition in ECS. A task definition is required to run Docker containers on Amazon ECS. Create task definition for TIBCO BusinessWorks Container Edition application. For more details, see Amazon ECS Task Definitions.
The following is a sample taskdef.json file for reference.

```
{
    "family": "launch-test-app",
    "containerDefinitions": [
        {
            "image": "<AWS-account_id>.dkr.ecr.<region_name>.amazonaws.com/
<repo_name>:latest",
            "name": "bwce-test-app",
            "cpu": 10,
            "memory": 512,
            "essential": true,
            "portMappings": [
                {
                    "containerPort": 8080,
                    "hostPort": 8080
                }
            ],
            "environment": [
                {
                    "name": "BW_LOGLEVEL",
                    "value": "DEBUG"
                }
            ],
            "logConfiguration": {
                "logDriver": "awslogs",
                "options": {
                    "awslogs-group": "bwce-app-log",
                    "awslogs-region": "<region_name>",
                    "awslogs-stream-prefix": "bwce"
                }
        }
        }
    ]
}
```

8. Register the task definition in the repository by running the following command.

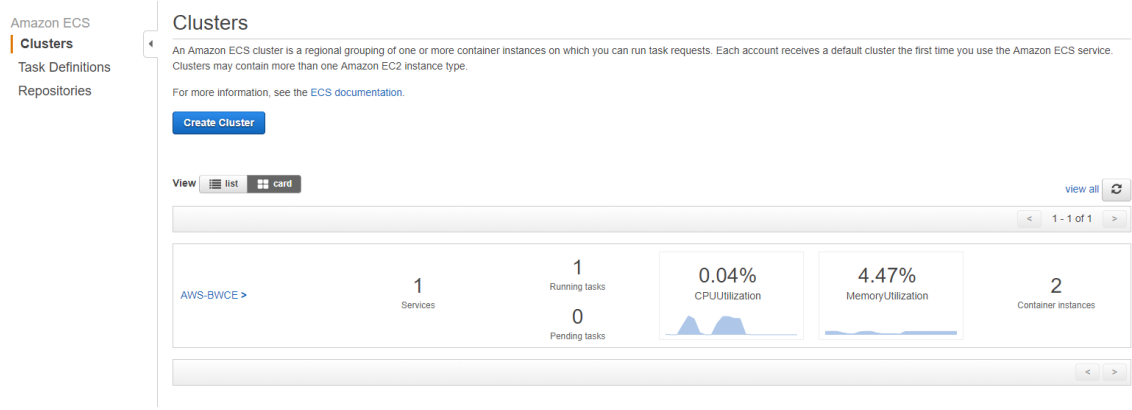
```
aws ecs register-task-definition --family <family_name_for_your_app> --cli-input-
json file://<taskdef_path>.json/> --region <aws_region>
```

9. To create a new service for your application, run the following command using the task definition. You can optionally configure your service to use load balancer to distribute traffic evenly across tasks in your service. For more details, see Services in Amazon ECS Documentation.

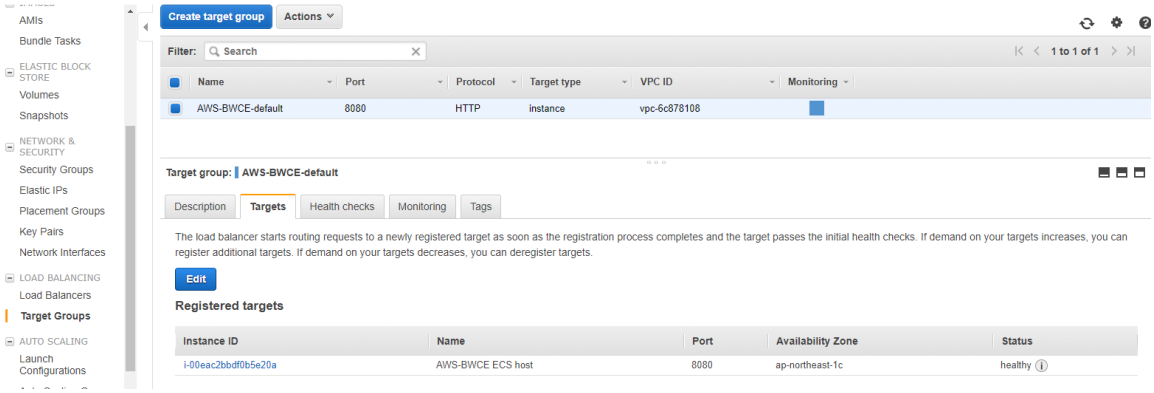
```
aws ecs create-service --service-name <Your_service_name> --desired-count 1 --
task-definition <your_family_name> --load-balancers
targetGroupArn=<your_target_group_arn>,containerName=<container_name>,containerPo
rt=<container_Port> --role <your_IAM_role_Arn> --cluster <your_cluster_name>
```

After the container is automatically configured with CloudWatch logs in taskdef, you can check your application logs in the service created in the ECS cluster.

You can also manually configure the CloudWatch. For more details, see Amazon Cloud watch Logs.



After the service is successfully running and the target group status is healthy, you can hit the load balancer URL according to the listener and target group configured in load balancers.



Amazon Terminology

- Amazon Machine Image (AMI)

  A supported and maintained Linux provided by Amazon Web Services for use on Amazon Elastic Compute Cloud (Amazon EC2). It is designed to provide a stable, secure, and high-performance execution environment for applications running on Amazon EC2. It also includes several packages that enable easy integration with AWS, including launch configuration tools and many popular AWS libraries and tools. Amazon Web Services also provides ongoing security and maintenance updates to all instances running the Amazon AMI.

- Amazon Web Services (AWS)

  Cloud platform, used to provide and host a family of services, such as RDS, S3, EC2, DynamoDB.

- AWS Console

  The user interface Amazon has built around the available services offered. Within the AWS Console, there are sub-consoles for individual services (EC2, S3, RDS, CloudFront, DynamoDB, etc.)

- AWS Marketplace

  Storefront for commercial AMIs provided and managed by Amazon, which bills customer for usage and keeps a percentage of sales proceeds.

- AWS Identity and Access Management (IAM)

  AWS Identity and Access Management (IAM) enables you to create multiple users and manage the permissions for each of these users within your AWS Account. A user is an identity within your AWS account with unique security credentials that can be used to access AWS Services. IAM eliminates the need to share passwords or access keys, and makes it easy to enable or disable a user's access as appropriate.

- CloudFormation (CF)AWS CloudFormation gives developers and systems administrators an easy way to create and manage a collection of related AWS resources, provisioning and updating them in an orderly and predictable fashion.

- Marketplace AMI

  An AMI that is distributed through the AWS Marketplace.

- Public AMI

  AMI configured as public by any Amazon user, and listed in everyone's AWS EC2 console AMI area.

- RDS

  Amazon Relational Database Service, which makes it easy to run MySQL, Oracle, or SQL Server database servers in the cloud. Amazon manages, upgrades, and backs up the server.

- Stack

  A collection of AWS resources you create and delete as a single unit.

- CloudWatch

  Amazon CloudWatch is a monitoring service for AWS cloud resources and the applications you run on AWS. You can use Amazon CloudWatch to collect and track metrics, collect and monitor log files, set alarms, and automatically react to changes in your AWS resources. Amazon CloudWatch can

monitor AWS resources such as Amazon EC2 instances, Amazon DynamoDB tables, and Amazon RDS DB instances, as well as custom metrics generated by your applications and services, and any log files your applications generate. You can use Amazon CloudWatch to gain system-wide visibility into resource utilization, application performance, and operational health. You can use these insights to react and keep your application running smoothly.

- ECS

  Amazon EC2 Container Service (Amazon ECS) is a highly scalable, fast, container management service that makes it easy to run, stop, and manage Docker containers on a cluster of Amazon Elastic Compute Cloud (Amazon EC2) instances. Amazon ECS lets you launch and stop container-based applications with simple API calls, allows you to get the state of your cluster from a centralized service, and gives you access to many familiar Amazon EC2 features.