

Testbed-12 TEAM Engine Virtualization User Guide

Table of Contents

1. Introduction	3
2. VirtualBox	5
2.1. Overview	5
2.2. Install VirtualBox	5
2.3. Create and import the image.....	5
3. Amazon EC2	7
3.1. Overview	7
3.2. Preparation	7
3.3. Create and register the image	7
3.4. Launch the image.....	8

Publication Date: 2017-06-16

Approval Date: 2017-03-23

Posted Date: 2016-11-16

Reference number of this document: OGC 16-075r1

Reference URL for this document: <http://www.opengis.net/doc/bp/t12-A064>

Category: User Guide

Editor: R. Martell

Title: Testbed-12 TEAM Engine Virtualization User Guide

COPYRIGHT

Copyright © 2017 Open Geospatial Consortium. To obtain additional rights of use, visit <http://www.opengeospatial.org/>

IMPORTANT

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights. Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

NOTE

This document is a user guide created as a deliverable of an initiative from the OGC Innovation Program (formerly OGC Interoperability Program). It is not an OGC standard. It is not an official position of the OGC membership. There may be additional valid approaches beyond what is described in this user guide.

POINTS OF CONTACT

Name	Organization
R. Martell (ed.)	Galdos Systems, Inc.
Luis Bermudez	Open Geospatial Consortium

Chapter 1. Introduction

TEAM Engine is a test harness that can be used to validate web services, data sets, and client applications. The source code is available in the [teamengine](#) repository at GitHub.

This guide describes how to use [Packer](#) to create a virtual machine (VM) or container image that includes a complete TEAM Engine installation along with the latest OGC conformance test suites. The generated image will contain the software components shown in the following figure.

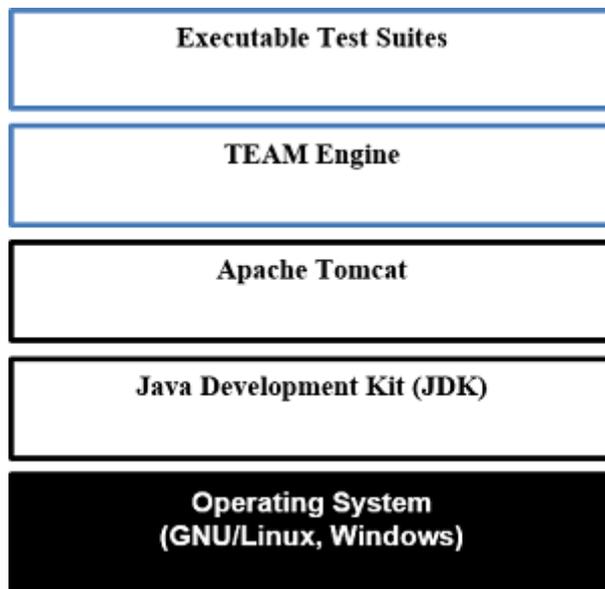


Figure 1. Software components

Packer is an open-source infrastructure definition tool used to create images for a variety of dynamic infrastructure platforms. It includes builders for many virtual environments, such as:

- VirtualBox
- Amazon EC2
- VMware vSphere
- Google Compute Engine
- Docker (container)

Virtual machine vs. container

A container is not a full-blown virtual machine—it doesn't require a hypervisor and runs in an isolated environment provided by a single host operating system. Since the OS kernel is shared, a container generally has less overhead than a virtual machine. Container technologies such as Docker, Warden, and Rocket offer a way to package and distribute a service or an application along with its dependencies.

The build process is driven by a template: a configuration file (in JSON format) that defines how a machine image is created and automatically provisioned.

To install Packer, simply download the appropriate [binary package](#) for your environment. Packages

are available for all major operating systems (Windows, macOS, and Linux). Unpack the archive in a suitable directory, and then set the following user environment variables:

PACKER_HOME

The directory where Packer is installed (Example: /usr/local/bin)

PACKER_CACHE_DIR

The location of the Packer cache; this is an optional setting (Example: /tmp)

Update the PATH environment variable to include PAKER_HOME. On Linux or macOS, add `export PATH=$PACKER_HOME:$PATH` to the shell environment. To do this on Windows, press **Win** > **Run...** and enter `sysdm.cpl`. On the *Advanced* tab select "Environment Variables". Edit the user-level PATH variable by appending `;%PACKER_HOME%` to the end (don't omit the semicolon).

Chapter 2. VirtualBox

2.1. Overview

VirtualBox is a freely available virtualization product that runs on Windows, Linux, and Mac hosts. It supports a wide variety of guest operating systems and is very useful in creating development or test environments. The Packer template will create a complete, ready-to-run TEAM Engine installation that contains the desired set of OGC test suites. The output is a "virtual appliance" (in the standard [OVF format](#)) that can be imported into VirtualBox.

2.2. Install VirtualBox

To install VirtualBox, download a suitable [installation package](#) for your host environment; see the user manual for [installation instructions](#). When installation is complete, set the following user-level environment variable:

`VBOX_USER_HOME`

The location where virtual machines will be created (Example: /srv/virtualbox)

By default virtual machines will be created in a subdirectory below the user home directory. The environment variable `VBOX_USER_HOME` overrides the default configuration to specify an alternative location.

2.3. Create and import the image

The Packer template makes use of several supporting shell scripts and files (see Note 1). In the 'files' subdirectory is a CSV file (ets-releases.csv) that contains a list of OGC test suite releases; each line has two fields: a Git repository URL, and a tag name indicating the release version. The test suites listed in this file will be added to the TEAM-engine instance. If a test suite is not wanted, just comment it out by inserting a number sign (or hash character, #) at the beginning of the appropriate line.

NOTE 1

The template file (teamengine-virtualbox.json) can be found in the src/main/config/packer/ directory in the TEAM Engine source tree, or wherever the Packer assembly was unpacked. To create the packer assembly, run this Maven command at the root of the TEAM Engine source tree: `mvn assembly:single`.

The template defines several user variables that can be overridden from the command line or a file; see <https://www.packer.io/docs/templates/user-variables.html>). These are listed in the table below, along with their default values.

Table 1. Template variables

Variable	Default value	Description
te_version	"4.9"	The desired version of the TEAM Engine web application available at Maven Central .

Variable	Default value	Description
ets_resources_release	""	The desired ets-resources release version ; this project aggregates all test suite dependencies. If not specified, the latest snapshot (tip of the master branch) will be used.
use_oracle_jdk	"false"	Whether or not to use the Oracle JDK (which is subject to a license agreement). If not, OpenJDK will be used.
iso_url	"http://pubmirrors.dal.corespace.com/centos/7/isos/x86_64/CentOS-7-x86_64-Minimal-1511.iso"	A URL referring to the base ISO installation image.

Any of the template variables can be overridden by a command-line assignment. For example, if a local copy of the base installation image exists it can be referenced using a file URL (the final image will be created more quickly if the base image doesn't have to be downloaded). To create the virtual appliance, simply change to the directory containing the Packer template and execute the `packer build` command in a shell console.

Listing 1. Example: Build appliance from a local ISO image

```
packer build -var iso_url=file:///tmp/CentOS-7-x86_64-Minimal-1511.iso teamengine-
virtualbox.json
```

The appliance will be created in the `output-centos72-virtualbox` subdirectory. The base operating system is [CentOS 7.2](#) (more specifically, the minimal distribution intended for "headless" operation without a GUI). To run it, start VirtualBox and perform the following steps:

1. Choose **File > Import Appliance...** and select the *.ovf file
2. When the import is completed, select the newly created virtual machine and press "Start"
3. When the boot sequence finishes, login using an SSH client to localhost:2222—or use the VirtualBox terminal—with the TEAM Engine user credentials (see Note 2)
4. To start or stop Tomcat, execute this command: `sudo systemctl {start|stop} tomcat-jsvc`
5. The main TEAM Engine web app is available at `http://localhost:8888/teamengine` (see Note 3)

NOTE 2

The user credentials are specified in the Packer template (`ssh_name`, `ssh_pass`). This is a normal user account with 'sudo' privileges.

NOTE 3

The VM runs on a private internal network using the default networking mode: network address translation (NAT); this prevents all direct inbound connections, but not outbound connections. Some port forwarding rules are created in order to allow remote access to the SSH server (via localhost:2222) and the Tomcat web container (via localhost:8888).

Chapter 3. Amazon EC2

3.1. Overview

Amazon Web Services (AWS) is a diverse platform of services that offer computing, storage, and networking capabilities in a public cloud environment. Amazon Elastic Compute Cloud (EC2) is a constituent service that provides dynamic infrastructure as a service (IaaS). Virtual servers can be created, launched, configured, and managed as needed.

3.2. Preparation

Before building an image, an Amazon Web Services (AWS) account must be available to use. If not, [sign up for an AWS account](#). The Packer template complies with the restrictions of the [free tier](#) so you will not be charged unless your usage exceeds the stipulated limits. It is strongly recommended to **not** use the root account (that is, the account owner). Instead, create a separate AWS Identity and Access Management (IAM) user to interact with AWS. For guidance about how to do this, see [Creating Your First IAM Admin User and Group](#).

NOTE 4

The AWS Free Tier offerings are available to new AWS customers for 12 months following the sign-up date. It allows up to 750 hours usage per month of a moderately small instance (instance type "t2.micro": 1 vCPU, 1 GiB memory). Note that the monthly allotment for Linux and Microsoft Windows instances is counted separately. Usage that exceeds the free tier limits is subject to billing. For more information, see [Using the Free Tier](#).

You will need the security credentials for the IAM user; specifically, the [access keys](#) required to send requests using various AWS APIs. Set them as the values of the environment variables shown below (which are also supported by the AWS command-line interface). The Packer template will obtain the credentials from these environment variables:

AWS_ACCESS_KEY_ID

Access Key ID (Example: AKIAIOSFODNN7EXAMPLE)

AWS_SECRET_ACCESS_KEY

Secret Access Key (Example: wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY)

The source image is the official [CentOS 7 HVM image](#), which is freely available from the AWS Marketplace. However, a subscription is required in order to access it. Visit the [AWS Marketplace page](#) for the official CentOS 7 image. Click "Continue" and select the "Manual Launch" tab. Then click "Accept Software Terms" in order to subscribe and enable access in any supported region.

3.3. Create and register the image

The same template variables as defined for the VirtualBox image apply here. To create and register the image, change to the directory containing the Packer template and execute this command in a shell console:

```
packer build teamengine-aws.json
```

When the process is completed, the image will appear in the AWS EC2 console—under IMAGES/AMIs—for the "N. Virginia" region (us-east-1). Note that the image will be marked as private, so it can only be launched by the owning account. If you have installed and configured the [AWS command-line tools](#), run the `describe-images` command:

```
aws ec2 describe-images --owners self --region us-east-1
```

3.4. Launch the image

An instance can be launched from the EC2 dashboard by selecting the image and clicking "Launch". Access via SSH is permitted from anywhere by default, but the source IP address can be restricted to a single address or an address range (in CIDR notation: 192.168.0.0/16). When the instance reaches the "running" state it will appear in the console under INSTANCES/Instances. The public hostname and IP address will be displayed on the "Description" tab. Connect via SSH and login as 'centos' using the key specified at launch. The server may be stopped and started as needed by selecting the appropriate action on the *Actions > Instance State* submenu.

Execute this command to start or stop the Tomcat web container:

```
sudo systemctl {start|stop} tomcat-jsvc
```

The main TEAM Engine web application will be available at `<http://{ip-address}:8080/teamengine/>`. However, no in-bound HTTP traffic is permitted by default. A firewall rule must be added either when launching the instance or by editing the security group that applies to the running instance. To do the latter, on the "Description" tab select the security group and choose the *Actions > Edit inbound rules* submenu; add a custom TCP rule for port 8080.