Continuous Validation on VMware Marketplace

Solution Test Packaging Guidelines for Publishers

vmware®

Table of contents

Overview

This document shows VMware Marketplace publishers how to submit their solution in a Kubernetes form-factor to VMware Marketplace Continuous Validation (CV) Pipeline Engine. This document also outlines a set of guidelines for publishers to follow when packaging their solution/application-specifics tests in 'Test Case Container image' and running that container image on the VMware Marketplace Continuous Validation (CV) Pipeline Engine.

Assumptions

- Each solution is packaged as Kubernetes form factor with artifact-type as Helm Chart
- All the Docker images referenced in the Helm Charts are stored in a public registry
- Each solution may optionally provide a single test case container image. This container image when launched will run a battery of tests against a previously deployed Helm chart
- The test case container image should also be stored in a public registry and accessible to VMware
- Multiple versions of the solution/application can choose to use the same test case container image or different version of test case container image
- Publisher test case container image is not accessible to Consumers
- Terms "Solution" & "Application" are used interchangeably in this document

Requirements

Publisher to provide two things:

1. (Must) Solution as Helm chart and any additional helm-install options
2. (Optional) Test Case Container image and associated parameters to launch the test cases

Helm Chart Requirements

The deployed application must have one and only one Kubernetes Service deployed as Load Balancer. If no services or more than one is set as Load Balancer, the CV test run will be aborted. Currently the VMware Marketplace CV Pipeline Engine does not support applications deployed with ingress.

Test Case Container Requirements

Publishers to provide a standalone self-contained test container image. Publishers are free to choose the base-image and test-framework.

The container **must:**

- Run the battery of tests. Test Cases must be packaged in the image such that a docker run should trigger execution of all test cases one after the other
- Return status code=0 when the tests pass or another value otherwise
- Expose logs and errors to STDOUT, STDERR respectively

The container optionally **can**:

- Leverage the Kubernetes config file that can be found at /tmp/kubeconfig. VMware Marketplace Continuous Pipeline Engine will mount the kubeconfig.yaml file of the pertaining cluster as /tmp/kubeconfig. With it, the Publisher can introspect the environment where the application is being run
- The container should use the networking directly from the host. The VMware Marketplace Continuous Pipeline Engine will add the "--network=host" option by default to the docker run command
- Any additional parameters that need to be added to docker run command, should be given in the "Test Case Launch Options" field of the VMware Marketplace UI. (see Figure-3)
- Place a single *xUnit* or any format report in /tmp/output/reports. VMware Marketplace Continuous Pipeline Engine will pick it up from this location & provide an option to download from VMware Marketplace (see Figure 4)

Examples

The following examples will show how to run the application and tests locally and how they will map to the continuous validations in VMware Marketplace. The examples used here are based on open source Bitnami packaged applications.

Example 1:

In the VMware Marketplace UI navigate to Publish->Solution workflow to create a new listing. Here select "Distributed Solution" to allow validation to select form-factor. Publishers can select the Kubernetes Form-Factor with Artifact as Helm-Chart. See Figure 1 below:

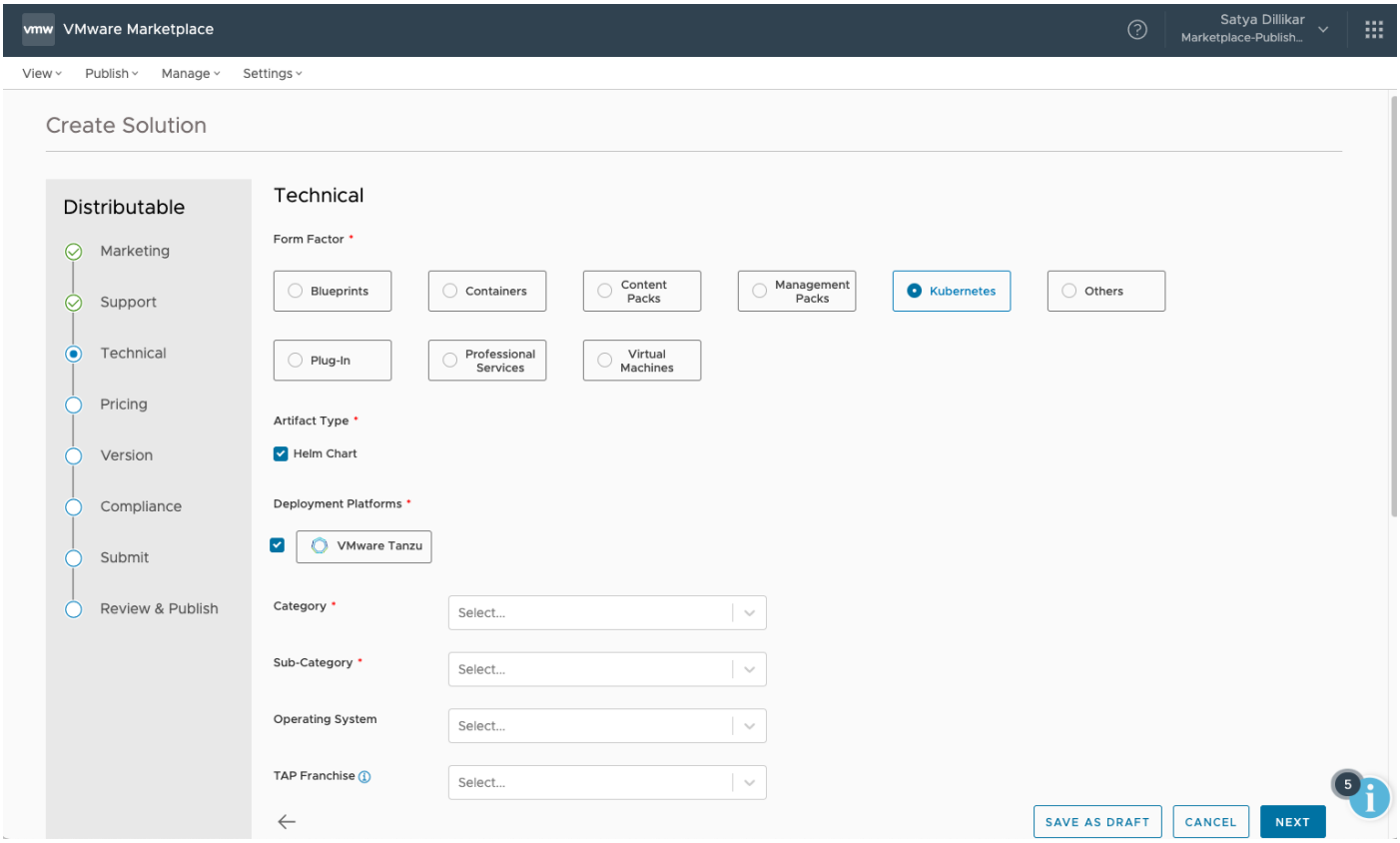

Figure 1: Form Factor & Artifact Type

Helm-Chart:

Chart Name	Redis-12.7.4
Chart Version	12.7.4
Public URL or Helm Chart tar file	redis-12.7.4.tgz
Chart Install Options	--set cluster.slaveCount=5

Table 1: Example-1 Helm Chart

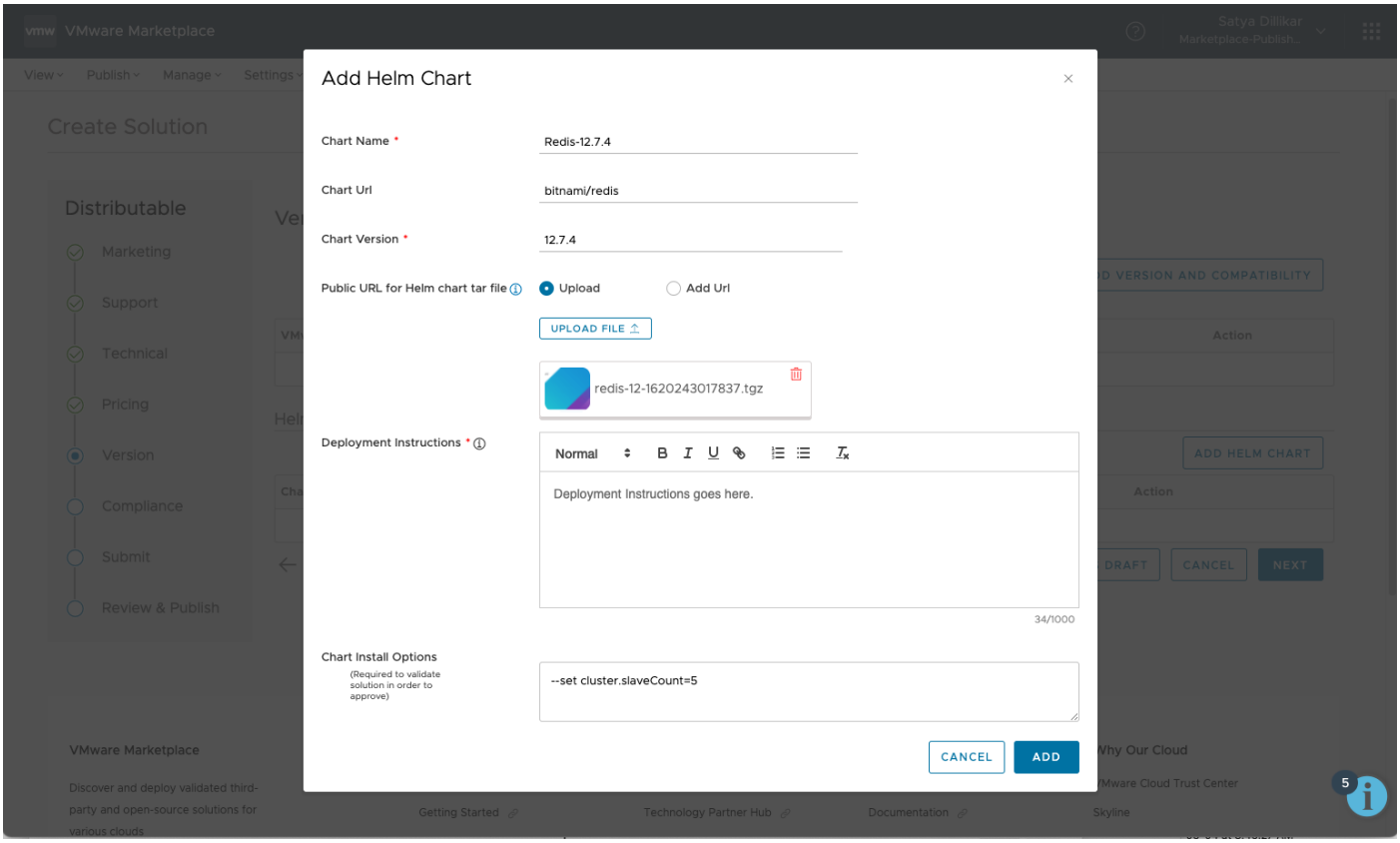

Figure 2: Add Helm Chart

Test Case Container

Test Case Container Image Public Container Registry	gcr.io/vmware-marketplace-bitnami/validationtest/bitnami-redis-verification-tests:0.2
Test Case Launch Options	--pod-selector role=master

Table 2: Example-1 Test Case Container

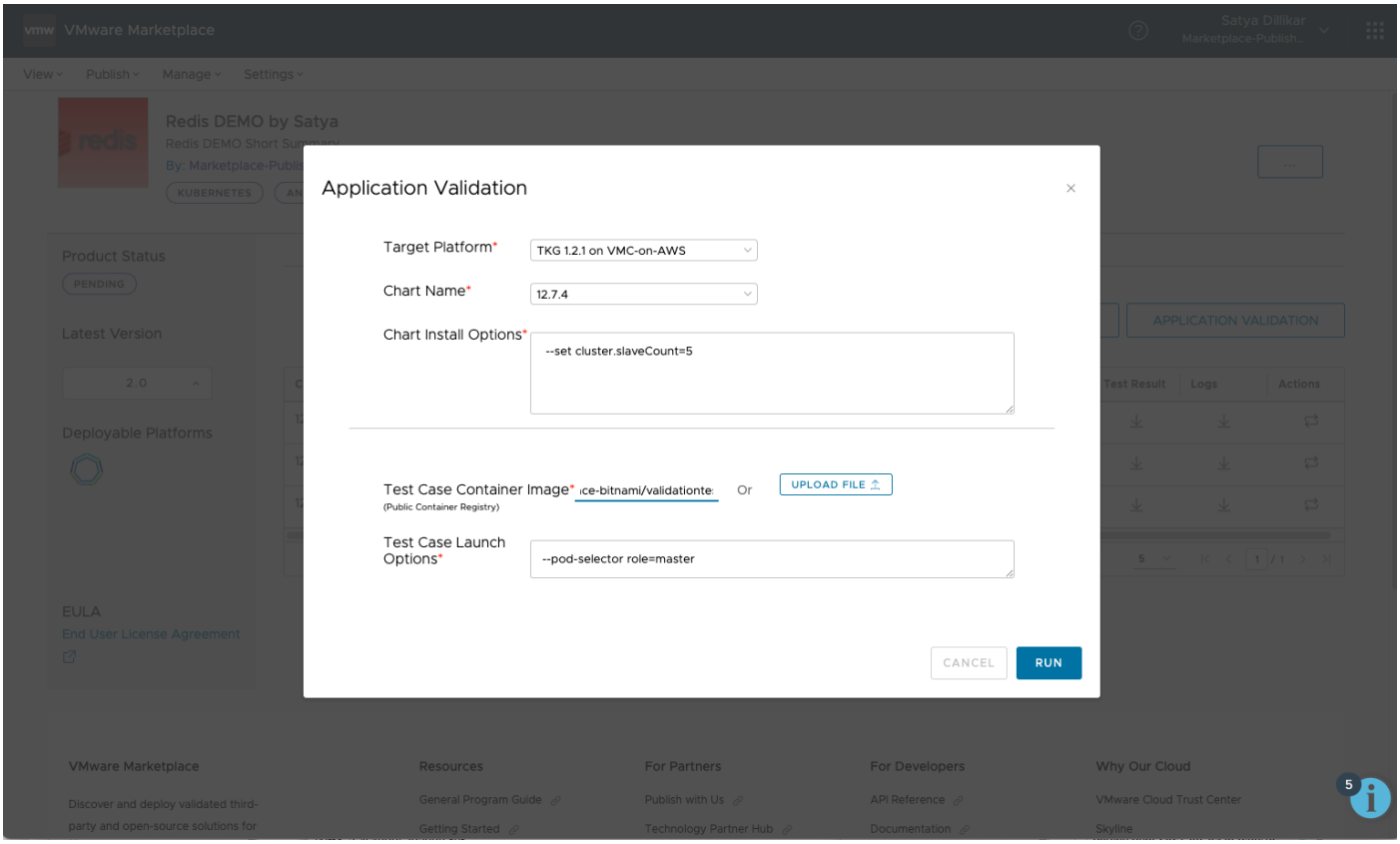

Figure 3: Application Specific Validation

Final Validation (Publisher Only) View

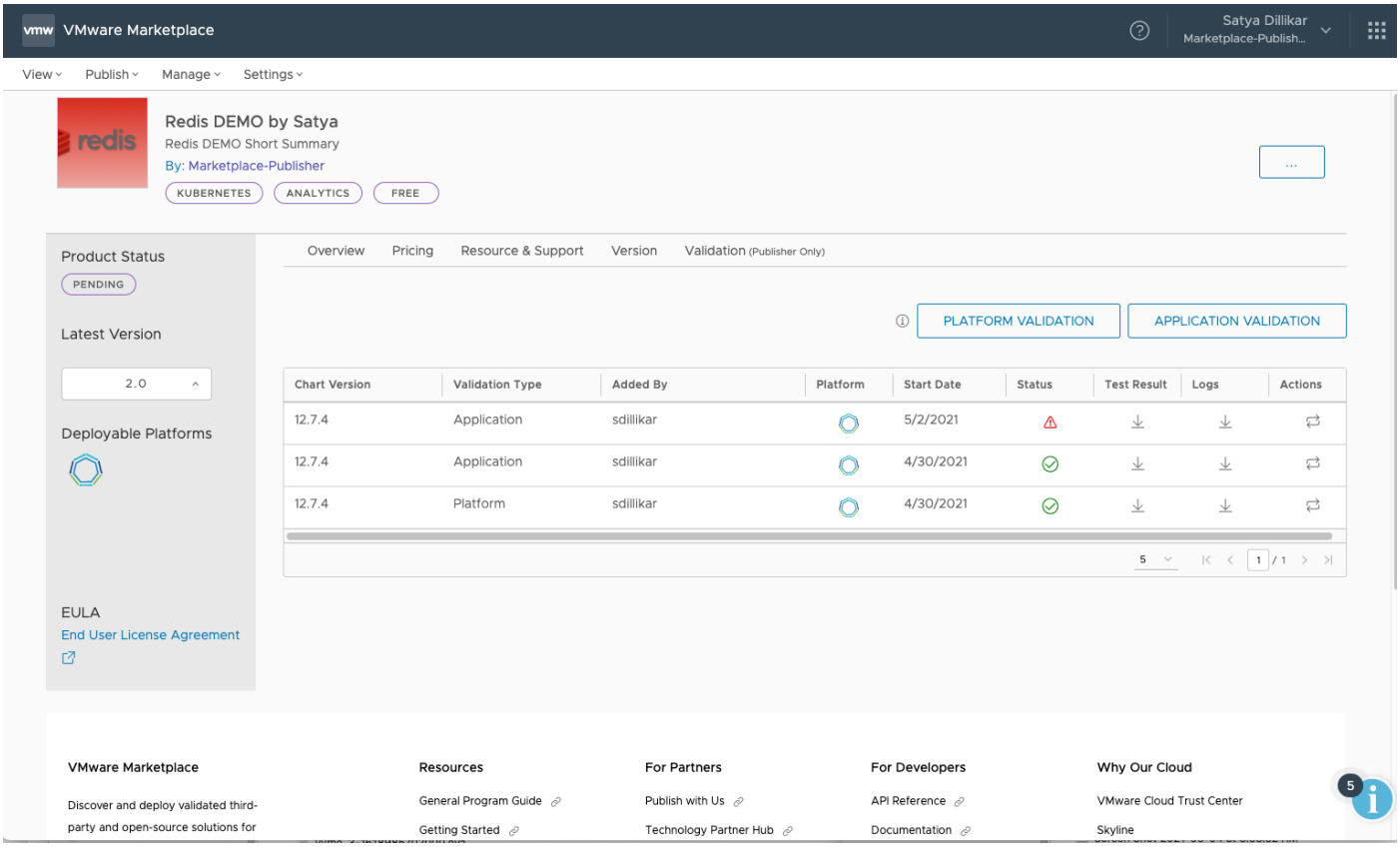

Figure 4: Validation Publisher Only View

The equivalent if manually run in a VMware Tanzu Kubernetes workload cluster:

```
helm install redis-12.7.4.tgz \
  --generate-name \
  --set cluster.slaveCount=5
# Run verification tests
docker run -it --network=host \
  -v /tmp/reports:/tmp/output/reports \
  -v ~/.kube/config:/tmp/kubeconfig \
  projects.registry.vmware.com/marketplacetest/bitnami-redis-verification-tests:0.2 \
  --pod-selector role=master
```

Table 3: Example-1 CLI Commands

Example 2:

Helm-Chart

Chart Name	wordpress-10.6.8
Chart Version	10.6.8
Public URL or Helm Chart tar file	wordpress-10.6.8.tgz
Chart Install Options	--set wordpressPassword=mytestpassword1 --set service.type=LoadBalancer

Table 4: Example-2 Helm Chart

Test Case Container

Test Case Container Image Public Container Registry	gcr.io/vmware-marketplace-bitnami/validationtest/bitnami-wordpress-functional-tests:0.2
Test Case Launch Options	--host $APPLICATION_HOST --port 80 --input password=mytestpassword1

Table 5: Example-2 Test Case Container

The equivalent manual commands:

```
# Install chart
helm install wordpress wordpress-10.6.8.tgz \
  --set wordpressPassword=mytestpassword1 \
  --set service.type=LoadBalancer
# Get hostname
APPLICATION_HOST=$(kubectl get svc wordpress -o jsonpath={.status.loadBalancer.ingress[0].ip})
# Run functional tests
docker run -it --network=host -v /tmp/reports:/tmp/output/reports \
  projects.registry.vmware.com/marketplacetest/bitnami-wordpress-functional-tests:0.2 \
  --host $APPLICATION_HOST --port 80 \
  --input password=mytestpassword1
```

Table 6: Example-2 CLI Commands

The VMware Marketplace Continuous Validation Pipeline will automatically fetch the application IP-address and populate the environment variable $APPLICATION_HOST

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 vmware.com Copyright © 2020 VMware, Inc.
All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at vmware.com/go/patents. VMware is a registered trademark or trademark of VMware, Inc. and its subsidiaries in the United States and other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies. Item No: vmw-wp-tech-temp-a4-word-101-proof 6/20