



An Introduction to Tanzu Kubernetes for VMC on AWS

VMware App Modernization

Table of contents

An Introduction to Tanzu Kubernetes for VMC on AWS	3
Overview	3
Scope	3
Tanzu Kubernetes Grid Architecture & Components	4
Tanzu Kubernetes Grid in VMC on AWS	4
Benefits of Running Tanzu Standard in VMware Cloud on AWS	6
TKG Compatibility Matrix	7
Tanzu Kubernetes Grid Deployment Workflow	8
Download Kubernetes Templates and TKG Tools	8
Configure Bootstrap Environment	8
Install the Tanzu CLI Binary and Plugins	8
Install Kubectl	9
Create an SSH Key Pair	9
Author and Contributors	10

An Introduction to Tanzu Kubernetes for VMC on AWS

Overview

VMware Tanzu Kubernetes Grid (TKG) is a multi-cloud Kubernetes footprint that you can run both on-premises in vSphere and the public cloud on Amazon EC2 and Microsoft Azure. TKG provides a consistent, upstream-compatible implementation of Kubernetes, that is tested, signed, and supported by VMware. TKG also includes signed and supported versions of open source applications to provide the registry, networking, monitoring, authentication, ingress control, and logging services required by a production Kubernetes environment.

Currently, VMware provides three variants of the TKG:

- **Tanzu Kubernetes Grid Multi-Cloud (TKGm):** Installer driven wizard to set up Kubernetes environment to run across on-prem software-defined datacenters (SDDC) and public cloud environments.
- **Tanzu Kubernetes grid Service (TKGS) aka vSphere With Tanzu:** It is a solution that has transformed vSphere into a platform for running Kubernetes workloads natively on the hypervisor layer. This is enabled on a vSphere cluster and provides the capability to run Kubernetes workloads directly on ESXi hosts and to create upstream Kubernetes clusters within dedicated resource pools.
- **Tanzu Kubernetes Grid Integrated Edition:** VMware Tanzu Kubernetes Grid Integrated Edition (formerly known as VMware Enterprise PKS) is a Kubernetes-based container solution with advanced networking, a private container registry, and life cycle management.

Scope

The scope of this document is limited to Tanzu Kubernetes Grid Multi-Cloud (TKGm) only.

This document briefly describes the architecture of the VMware Tanzu Standard platform when deployed into a VMware Cloud on AWS solution and offers a high-level overview of the different components.

Tanzu Kubernetes Grid Architecture & Components

The main components of the TKG are:

- **Management Cluster:** The management cluster is a Kubernetes cluster that performs the role of the primary management and operational center for the Tanzu Kubernetes Grid instance. This is the first element that you deploy when you create a Tanzu Kubernetes Grid Multi-Cloud instance.
- **Tanzu Kubernetes Cluster a.k.a Workload Cluster:** Tanzu Kubernetes clusters are the Kubernetes clusters in which your Kubernetes application run.
- **Tanzu CLI:** A command-line utility to deploy and manage the lifecycle of the Management & Tanzu Kubernetes Cluster.
- **Tanzu Kubernetes Grid Extensions:** A set of tools that provide in-cluster and shared services to the clusters running in your Tanzu Kubernetes Grid instance.

The following table provides information about the tools included in the TKG extension:

Function	Extension	Description
Ingress Control	Contour	Provides Layer 7 control to deployed HTTP(s) applications.
Log Forwarding	Fluent Bit	Provides export log streaming of TKG Management & Workload cluster to a wide range of supported aggregators provided in the extensions package for TKG.
Container Registry	Harbor	Provides a centralized location to push, pull, store, and scan container images used in Kubernetes workloads. It also supports storing many other artifacts such as Helm charts and includes enterprise-grade features such as RBAC, retention policies, automated garbage collection of stale images, etc.
Monitoring	Prometheus	Provides out-of-the box health monitoring of Kubernetes clusters.
Monitoring	Grafana	Provides monitoring dashboards for displaying key health metrics of Kubernetes clusters

To learn more about these components, refer to the VMware official [documentation](#).

Tanzu Kubernetes Grid in VMC on AWS

The following diagram shows the high-level architecture of the Tanzu Kubernetes Grid instance.

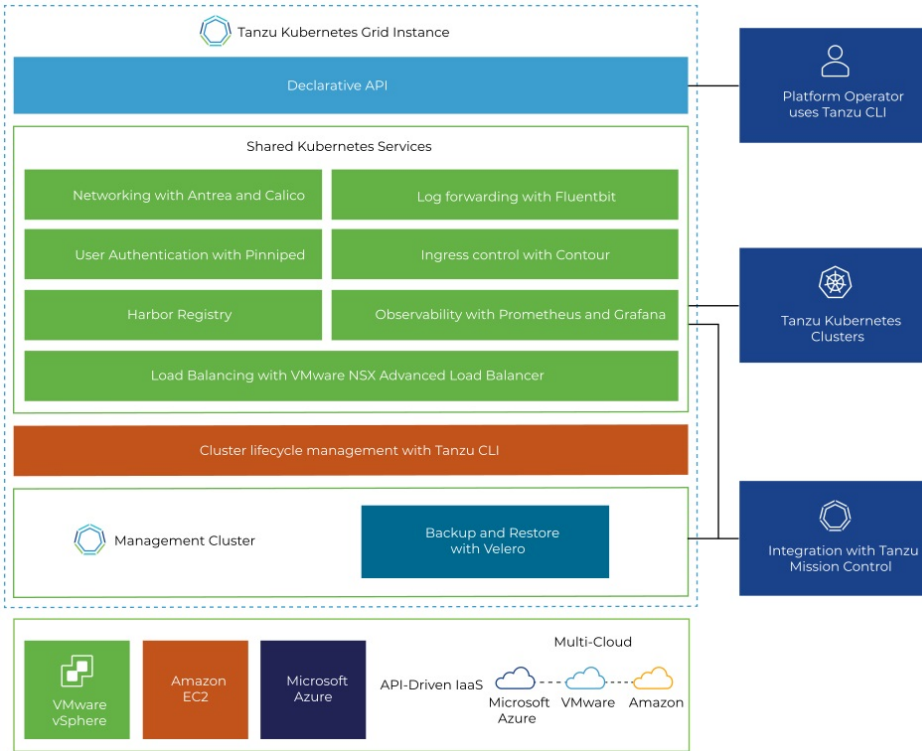


Figure 1 - Tanzu Kubernetes Grid Architecture

Tanzu Kubernetes Grid is deployed as a customer-managed solution in VMware Cloud on AWS. To provide load balancing services to deployments on vSphere, Tanzu Kubernetes Grid includes VMware NSX Advanced Load Balancer Essentials Edition.

Tanzu Kubernetes Grid components along with NSX ALB are collectively called VMware Tanzu Standard.

The following diagram shows the high-level connectivity of the TKG components with NSX ALB in VMC on AWS.

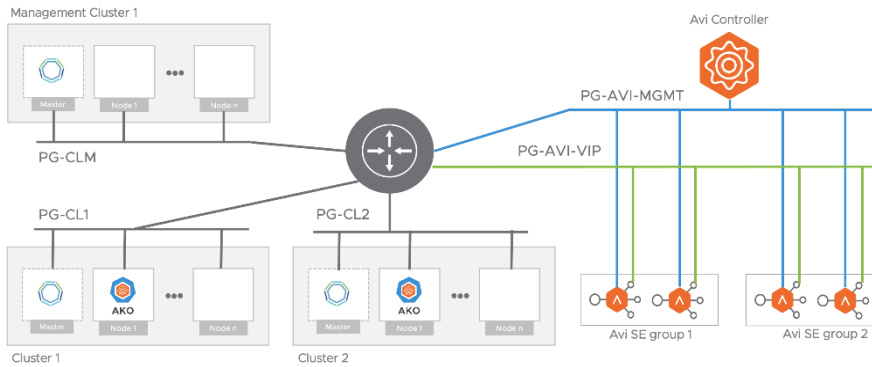


Figure 2 - TKG Interaction with NSX ALB

Benefits of Running Tanzu Standard in VMware Cloud on AWS

VMware Cloud on AWS provides a seamlessly integrated hybrid cloud offering to address use cases that align to a customer's cloud strategy. The service provides the following use cases:

1. Cloud Migrations
2. Data Center Extension
3. Disaster Recovery
4. Next-Generation Apps

By running Tanzu Standard within the same infrastructure as general virtual machine workloads enabled by the first three use cases above, organizations can start their next-generation application modernization strategy immediately without incurring additional costs.

TKG Compatibility Matrix

This table lists the software version of the components that are required to deploy TKG 1.3.1 in VMware Cloud on AWS.

Software Components	Version
Tanzu Kubernetes Grid	1.3.1
VMware Cloud on AWS SDDC Version	1.14
NSX Advanced Load Balancer	20.1.5

For the previous releases of the TKG, check the Interoperability Matrix [here](#).

Tanzu Kubernetes Grid Deployment Workflow

The steps for deploying TKG in VMC can be summarized as follows:

1. Setup TKG Bootstrapper machine.
2. Deploy TKG Management Cluster.
3. Deploy TKG Workload Cluster.

Note: This document only talks about setting up TKG bootstrapper as this step is common for TKG deployments across various supported platforms.

The deployment of the TKG management & workload cluster is facilitated by setting up a bootstrap machine where you install the Tanzu CLI and Kubectl utilities which are used to create and manage the TKG instance. This machine also keeps the TKG and Kubernetes configuration files of your deployments. The bootstrapper runs a local kind cluster when TKG management cluster deployment is triggered and once the kind cluster is fully initialized, the configuration is then used to deploy the actual management cluster on the backend infrastructure. Once the management cluster is fully configured, the local kind cluster is deleted and the future configurations are performed via Tanzu CLI.

Download Kubernetes Templates and TKG Tools

To deploy the TKG instance, you have to first import the supported version of the Kubernetes ova into your vCenter server and convert the imported ova into a template. This template will be used by the TKG installer to deploy the management and workload cluster.

To know more about the supported K8's version with TKG 1.3.1, see the [TKG Release Notes](#).

You can download the Kubernetes ova for TKG 1.3.1 deployment from [here](#).

TKG Management cluster makes use of "Photon v3 Kubernetes v1.20.5 vmware OVA".

Download the following items from the VMware portal:

- Photon v3 Kubernetes v1.20.5 OVA
- VMware Tanzu CLI
- Kubectl cluster cli v1.20.5

Configure Bootstrap Environment

In VMC on AWS environment, the bootstrap machine must be a cloud VM, not a local machine, and should meet the following prerequisites.

- A minimum of 6 GB of RAM and a 2-core CPU.
- System time is synchronized with a Network Time Protocol (NTP) server.
- Docker and containerd binaries installed. For instructions on how to install Docker, please refer to the [Docker official documentation](#)

Note: For the purpose of the demonstration, this article refers to a bootstrapper machine as an Ubuntu 20.4 instance deployed in VMC SDDC and attached to the TKG-Management logical segment.

To use the TKG installation binaries, upload the Tanzu CLI and Kubectl binary to the bootstrapper machine using WinSCP or a similar utility and unpack them using the system utilities like *tar/unzip/gunzip*.

Install the Tanzu CLI Binary and Plugins

After you unpack the Tanzu CLI bundle file, you will see a cli folder with multiple subfolders and files. Use the below command to install the Tanzu CLI.

```
# tar -xvf tanzu-cli-bundle-v1.3.1-linux-amd64.tar

# cd cli/

# sudo install core/v1.3.1/tanzu-core-linux_amd64 /usr/local/bin/tanzu

# chmod +x /usr/local/bin/tanzu
```


1. At the command line, run the Tanzu version command to check that the correct version of the CLI is properly installed.
2. After you have installed the Tanzu CLI, you must install the CLI plugins related to Tanzu Kubernetes cluster management and feature operations.
3. Navigate to the Tanzu folder that contains the cli folder and run the following command to install the plugins.

```
# tanzu plugin install --local cli all
```

4. Check plugin installation status.

```
# tanzu plugin list
```

After a successful installation, the following plugins are displayed:

```
root@ky-bootstrap:~# tanzu plugin list
NAME          LATEST VERSION  DESCRIPTION          REPOSITORY  VERSION  STATUS
alpha         v1.3.1          Alpha CLI commands  core        v1.3.1   not installed
cluster       v1.3.1          Kubernetes cluster operations  core        v1.3.1   installed
kubernetes-release v1.3.1        Kubernetes release operations  core        v1.3.1   installed
login         v1.3.1          Login to the platform  core        v1.3.1   installed
management-cluster v1.3.1        Kubernetes management cluster operations  core        v1.3.1   installed
plinkped-auth v1.3.1          Plinkped authentication operations (usually not directly invoked)  core        v1.3.1   installed
root@ky-bootstrap:~#
```

Figure 3 - Tanzu Plugins List

Install Kubectl

Run the following commands to install Kubectl utility:

```
# gunzip kubectl-linux-v1.20.5-vmware.1.gz

# mv kubectl-linux-v1.20.5-vmware.1 /usr/local/bin/kubectl

# chmod +x /usr/local/bin/kubectl
```

Create an SSH Key Pair

This is required for Tanzu CLI to connect to vSphere from the bootstrap machine. The public key part of the generated key will be passed during the TKG management cluster deployment.

1. To generate a new SSH key pair, execute the below command:

```
# ssh-keygen -t rsa -b 4096 -C "email@example.com"
```

2. You will be prompted to enter the file in which to save the key. Press Enter to accept the default.
3. Enter and repeat a password for the key pair.
4. Add the private key to the SSH agent running on your machine, and enter the password you created in the previous step.

```
# ssh-add ~/.ssh/id_rsa
```

5. If the above command fails, execute `eval $(ssh-agent)` and then re-run the command.
6. Make a note of the public key from the file `$home/.ssh/id_rsa.pub` and have it handy. You need this while creating a config file for deploying the TKG management cluster.

You are now ready to deploy the TKG Management & Workload cluster and run your Kubernetes workloads in the workload cluster.

Author and Contributors

[Manish Jha](#) has authored this article.

