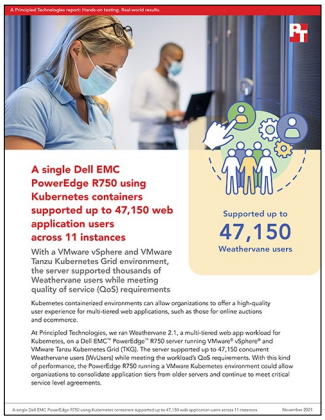

The science behind the report:

A single Dell EMC PowerEdge R750 using Kubernetes containers supported up to 47,150 web application users across 11 instances

This document describes what we tested, how we tested, and what we found. To learn how these facts translate into real-world benefits, read the report A single Dell EMC PowerEdge R750 using Kubernetes containers supported up to 47,150 web application users across 11 instances.

We concluded our hands-on testing on September 16, 2021. During testing, we determined the appropriate hardware and software configurations and applied updates as they became available. The results in this report reflect configurations that we finalized on August 25, 2021 or earlier. Unavoidably, these configurations may not represent the latest versions available when this report appears.

Our results

To learn more about how we have calculated the wins in this report, go to http://facts.pt/calculating-and-highlighting-wins. Unless we state otherwise, we have followed the rules and principles we outline in that document.

Table 1: Results of our Weathervane 2.0 testing.

	App cluster worker-node VM count	Driver cluster worker -node VM count	App instances	Total Weathervane users
Run 1	18	10	11	47,950
Run 2	18	10	11	47,125
Run 3	18	10	11	47,150
Median	18	10	11	47,150

Figures 1 through 5 show system-level outputs for the Dell EMC PowerEdge R750 server used in our testing.

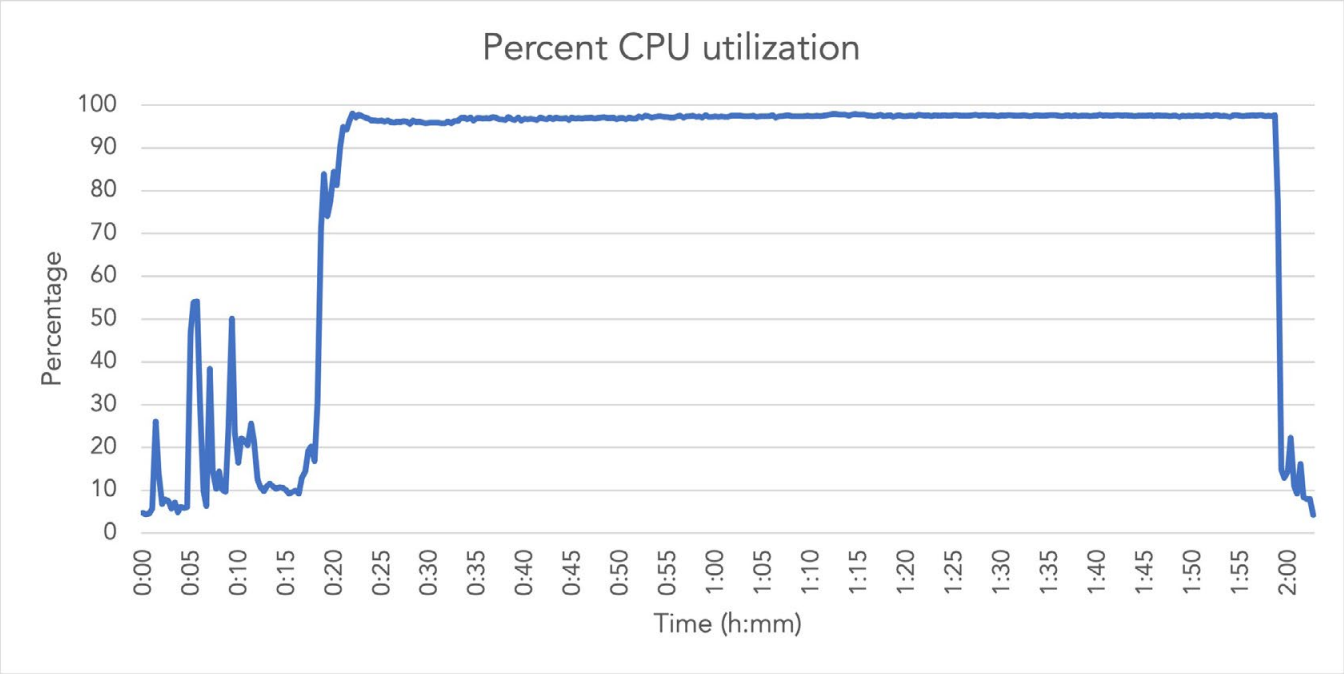

Figure 1: CPU utilization for the Dell EMC PowerEdge R750 server used in our testing. Source: Principled Technologies.

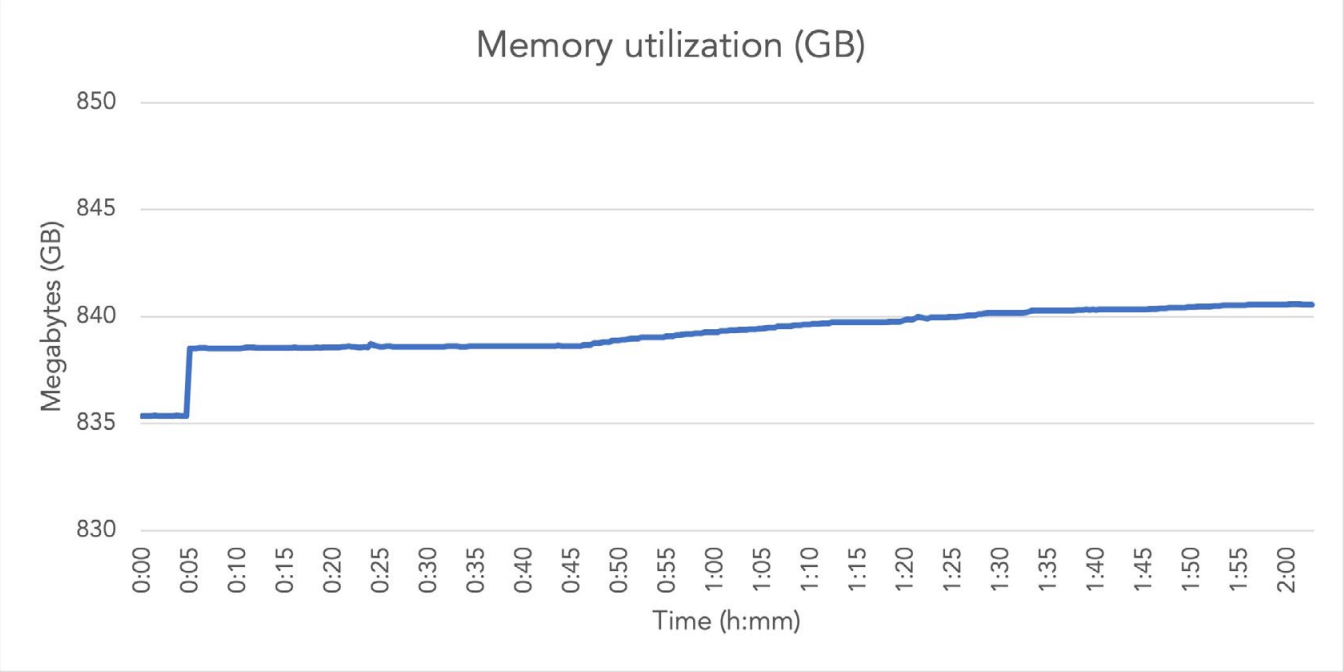

Figure 2: Memory utilization for the Dell EMC PowerEdge R750 server used in our testing. Source: Principled Technologies.

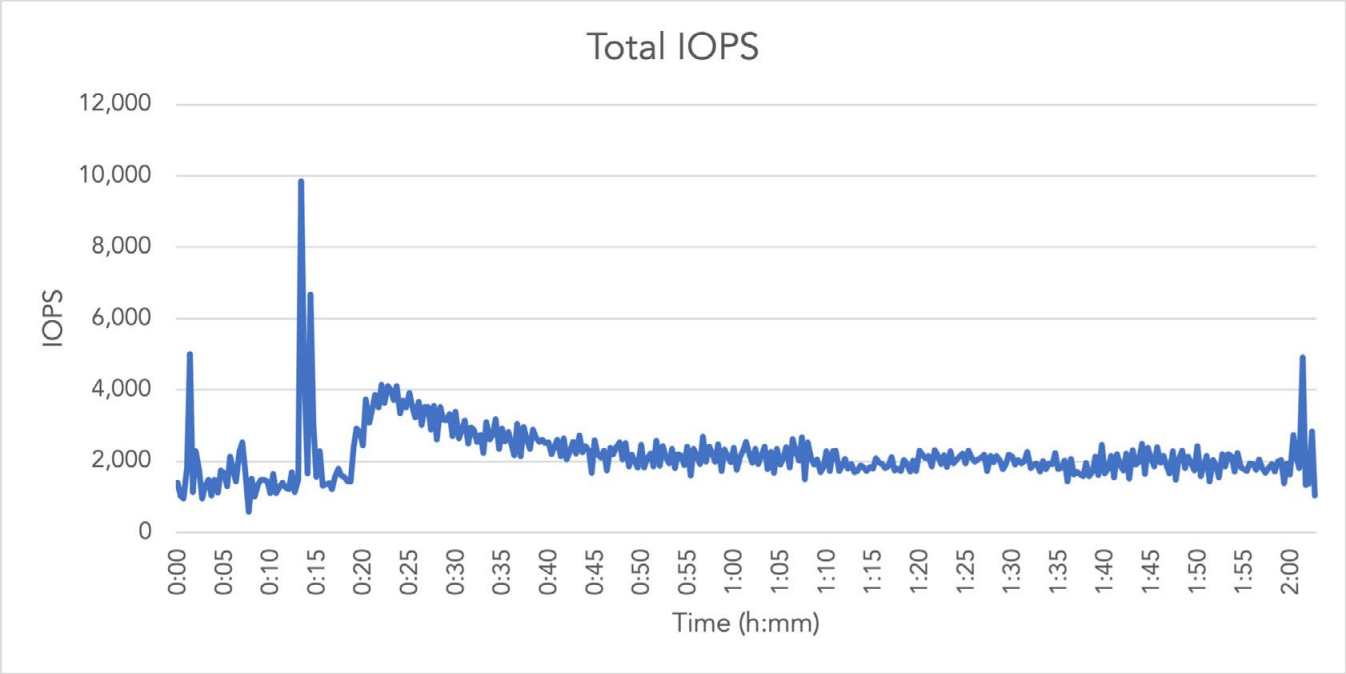

Figure 3: The total IOPS processed by the Dell EMC PowerEdge R750 server used in our testing.
Source: Principled Technologies.

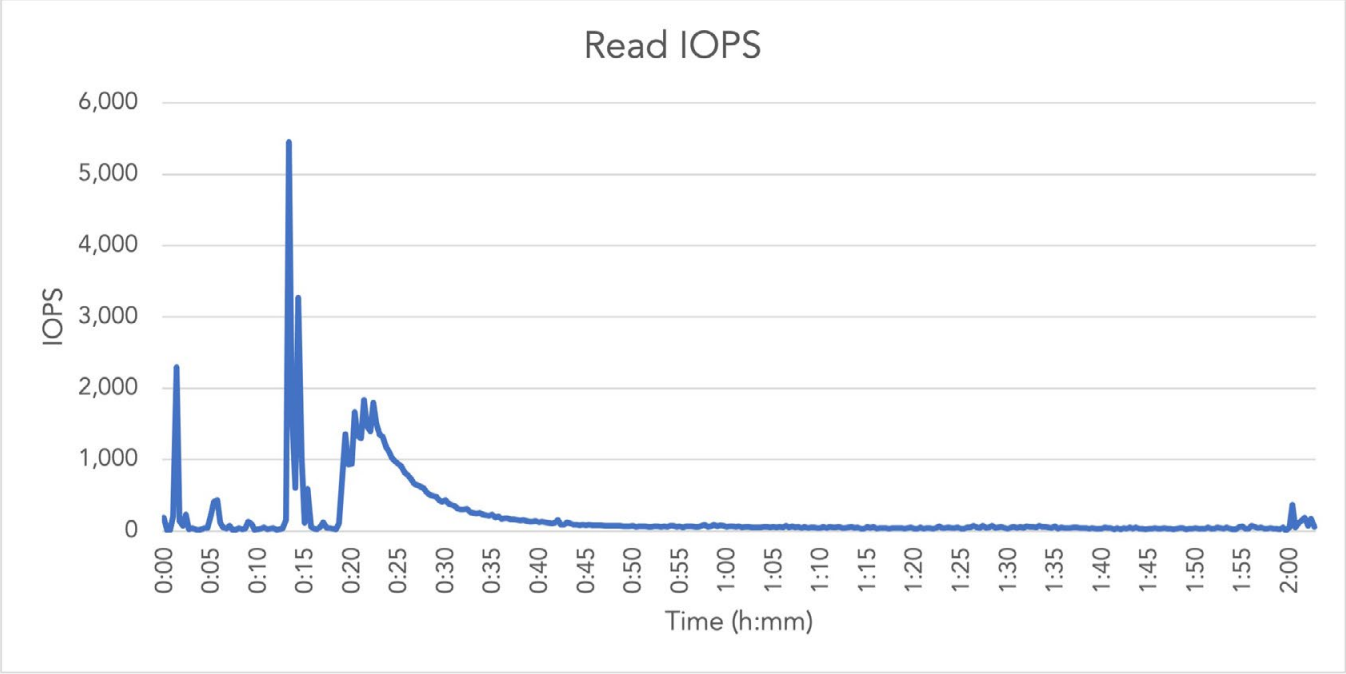

Figure 4: The total read IOPS processed by the Dell EMC PowerEdge R750 server used in our testing.
Source: Principled Technologies.

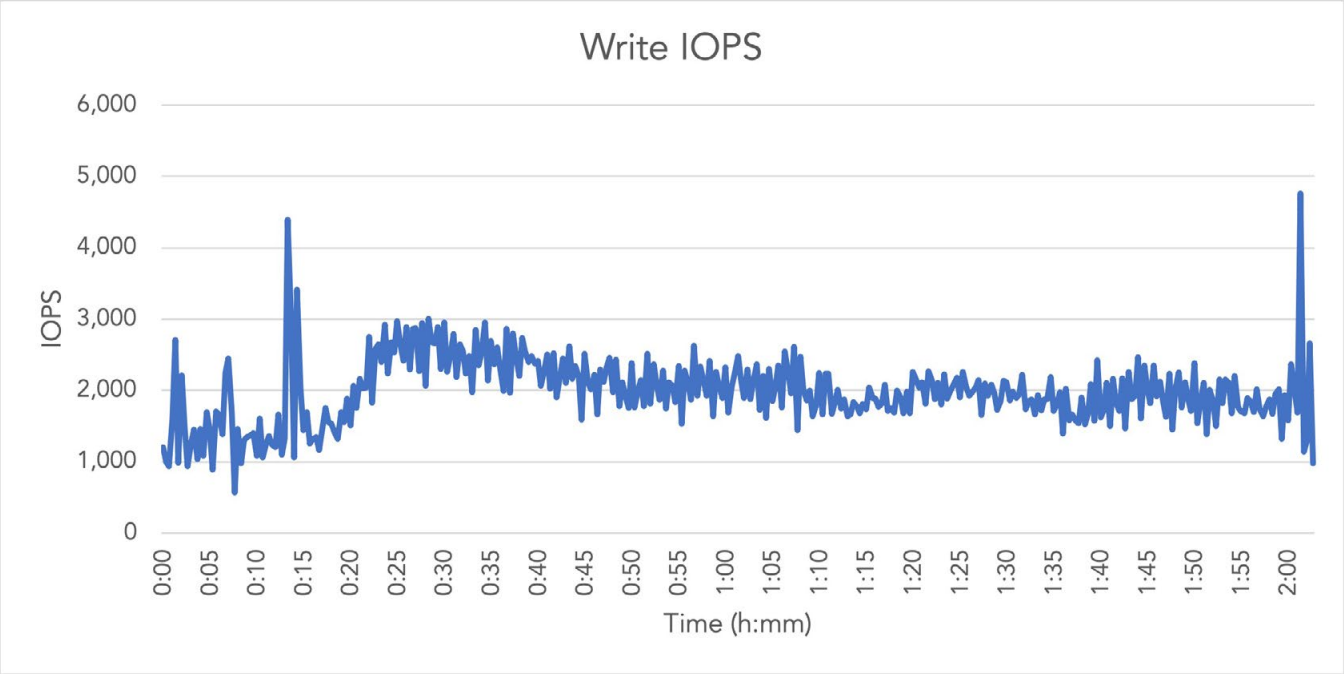

Figure 5: The total write IOPS processed by the Dell EMC PowerEdge R750 server used in our testing. Source: Principled Technologies.

System configuration information

Table 2: Detailed information on the system we tested.

System configuration information	Dell EMC™ PowerEdge™ R750
BIOS name and version	Dell 1.2.4
Non-default BIOS settings	N/A
Operating system name and version/build number	VMware® ESXi® 7.0.2 (Build 17867351)
Date of last OS updates/patches applied	7/27/2021
System profile settings	Performance
Processor	
Number of processors	2
Vendor and model	Intel® Xeon® Gold 6330
Core count (per processor)	28
Core frequency (GHz)	2
Stepping	Model 106 Stepping 6
Memory module(s)	
Total memory in system (GB)	1,024
Number of memory modules	16
Vendor and model	Samsung® M393A8G40AB2-CWE, Hynix® HMAA8GR7AJR4N-XN
Size (GB)	64
Type	PC4-3200
Speed (MHz)	3,200
Speed running in the server (MHz)	3,200
Storage controller	
Vendor and model	Dell PERC H755N Front
Cache size (GB)	8
Firmware version	52.14.0-3901
Driver version	7.716.03.00
Local storage	
Number of drives	8
Drive vendor and model	Samsung MZ-WLJ1T60
Drive size (GB)	1,600
Drive information (speed, interface, type)	NVMe™, SSD, 2.5

System configuration information	Dell EMC™ PowerEdge™ R750
Network adapter	
Vendor and model	Intel Ethernet 25G 2P E810-XXV
Number and type of ports	2 x 25G
Driver version	1.5.74.0
Cooling fans	
Vendor and model	Foxconn PIA060K12Q
Number of cooling fans	6
Power supplies	
Vendor and model	PWR SPLY, 1400W, RDNT, LTON
Number of power supplies	2
Wattage of each (W)	1,400

How we tested

We deployed two on-premises Tanzu Kubernetes Grid (TKG) clusters to a Dell EMC PowerEdge R750 to run the Weathervane 2.1 application-level performance benchmark and determine the number of simulated users (WvUsers) that the hardware can support without violating the built-in Weathervane quality-of-service (QoS) requirements. One cluster hosted the Weathervane benchmark application instances, and the other hosted the Weathervane workload drivers. We sized the two clusters based on the Weathervane user guide's application resource requirements configuration sizing tables for the small2 application instance size. We also deployed a third TKG management cluster as part of the TKG setup process. We deployed VMware vCenter Server® 7.0 U2 and a run-harness auxiliary VM to a separate infrastructure host to run the benchmark and manage the testbed.

Our application instance cluster consisted of a control plane VM with two vCPUs, 4 GB of RAM, and 20 GB of local storage, as well as 18 worker-node VMs with four vCPUs, 32 GB of RAM, and 200 GB of local storage each. Our workload driver cluster consisted of a control plane VM with two vCPUs, 4 GB of RAM, and 20 GB of local storage, as well as 10 worker-node VMs with four vCPUs, 24 GB of RAM, and 50 GB of local storage each. The required TKG management cluster consisted of a control plane VM with two vCPUs, 4 GB of RAM, and 20 GB of local storage, and a single worker VM of the same size. These cluster sizes allowed us to host 11 simultaneous Weathervane small2 application instances on the server to demonstrate the maximum number of simulated WvUsers the system could support across all application instances.

Installing VMware ESXi vSphere 7.0 U2 on the Dell EMC PowerEdge R750

1. Attach the installation media.
2. Boot the server.
3. Select the installation media.
4. At the VMware Installer screen, press Enter.
5. At the EULA screen, press F11 to Accept and Continue.
6. Under Storage Devices, select the appropriate virtual disk, and press Enter.
7. Select US as the keyboard layout, and press Enter.
8. Enter the root password twice, and press Enter.
9. Press F11 to start the installation.
10. After the server reboots, press F2, and enter root credentials.
11. Select Configure Management Network, and press Enter.
12. Select the appropriate network adapter, and click OK.
13. Select IPv4 settings, and enter the desired IP address, subnet mask, and gateway for the server.
14. Select OK, and restart the management network.

Deploying VMware vCenter Server 7.0 U2

We deployed a VMware vCenter Server 7.0 U2 virtual appliance on a separate infrastructure server and added both the PowerEdge R750 system under test (SUT) and the additional infrastructure to the vCenter.

1. On a Windows machine or VM, locate the VMware-VCSA installer image.
2. Mount the image, navigate to the vcsa-ui-installer folder, and double-click win32.
3. Double-click installer.exe
4. Click Install.
5. Click Next.
6. Accept the terms of the license agreement, and click Next.
7. Leave the default (vCenter Server with an Embedded Platform Services Controller) selected, and click Next.
8. Enter the FQDN or IP address of the host onto which the vCenter Server Appliance will deploy.
9. Provide the server's credentials, and click Next.
10. At the Configure Network Settings page, configure the network settings for your environment, and click Next.
11. Review your settings, and click Finish.
12. When the deployment completes, click Next.
13. At the Introduction page, click Next.
14. At the Appliance configuration page, select the time synchronization mode and SSH access settings, and click Next. We used a local NTP server and enabled SSH.
15. Select Create a new SSO domain.
16. Provide a password, and confirm it.
17. Provide an SSO Domain name and SSO Site name, and click Next.
18. At the CEIP page, click Next.

19. At the Ready to complete page, click Finish.
20. When the installation completes, click Close.
21. Using the vSphere web client, log into the vCenter server using the credentials you previously provided.

Creating a data center in VMware vCenter and adding hosts

Creating the data center

1. After logging into VMware vCenter®, navigate to Hosts and Clusters.
2. Select the primary site management vCenter.
3. Right-click the vCenter object, and select New Datacenter.
4. Enter a name for the new data center, and click OK.

Adding the hosts

1. Right-click the data center, and click Add Host.
2. Enter the FQDN or IP address of the PowerEdge R750 host SUT, and click Next.
3. Enter the root credentials for the server, and click Next.
4. To accept the server's certificate, click Yes.
5. Review the server details, and click Next.
6. Assign the desired license, and click Next.
7. Disable Lockdown mode, and click Next.
8. Click Finish.
9. Repeat steps 1 through 8 to add the infrastructure host.

Installing VMware Tanzu Kubernetes Grid (TKG) Standalone version 1.3.1

We installed TKG Standalone version 1.3.1 by following the installation steps in the VMware documentation: https://docs.vmware.com/en/VMware-Tanzu-Kubernetes-Grid/1.3/vmware-tanzu-kubernetes-grid-13/GUID-install-cli.html.

We installed ESXi 7 on the SUT and added it to an existing VMware vSphere® environment under the control of a vCenter appliance. The vSphere cluster uses one distributed switch with one physical NICS connected to a TOR switch running DHCP. We created a data center and resource pool on vCenter for the test.

1. Install Ubuntu 20.04 LTS on an auxiliary VM running on the additional infrastructure server in the vSphere environment, and create a local user account on it. We named our account `ubuntu`. The VM should have sufficient space on its root files system (30GB suffices) to hold several Docker images.
2. Log onto the VM as the local user.
3. Install Docker 20 on Ubuntu by following the instructions at https://docs.docker.com/engine/install/ubuntu/:

```
sudo apt-get remove docker docker-engine docker.io containerd runc
    sudo apt-get update
    sudo apt-get install apt-transport-https ca-certificates curl gnupg lsb-release
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | \
      sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] \
      https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | \
      sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
    sudo apt-get install docker-ce docker-ce-cli containerd.io
```

4. Add the local user to the Docker group:

```
sudo usermod -a -G docker ubuntu
```

5. Log out and log back in as the local user `ubuntu`
6. Start Docker.

```
sudo service docker start
```

7. Create SSH keys:

```
ssh-keygen -t rsa -b 4096 -C ubuntu@ubu01
```

8. Download the packages for TKG CLI version 1.31 (tanzu-cli-bundle-v1.3.1-linux-amd64.tar), TKG kubectl version1.20.5 (kubectl-linux-v1.20.5-vmware.1.gz), and one Ubuntu OS images (ubuntu-2004-kube-v1.20.5-vmware.2-tkg.1-6700972457122900687.ova) from the VMware website: https://www.vmware.com/go/get-tkg.

9. Install the TKG CLI, TKG kubectl, and assorted tools on the VM:

```
tar -xf tanzu-cli-bundle-v1.3.1-linux-amd64.tar
    sudo install cli/core/v1.3.1/tanzu-core-linux_amd64 /usr/local/bin/tanzu
gunzip kubectl-linux-v1.20.5-vmware.1.gz
    sudo install kubectl-linux-v1.20.5-vmware.1 /usr/local/bin/kubectl
gunzip cli/{imgpkg,kapp,kbld,ytt}-*.1.gz
    for i in imgpkg kapp kbld ytt ; do
      sudo install cli/${i}-linux-amd64-*vmware.1 /usr/local/bin/${i}
    done
```

10. Install the Tanzu cli plugins:

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

11. Create a VM folder with the vSphere client. Right-click the data center icon, and select→New Folder→New VM and template folder. In the pop-up window, enter the name of the folder.
12. To upload a base image for the nodes, right-click the data center, and select Deploy an OVF Template.
13. Select Local file.
14. Click Upload files, and select the file ubuntu-2004-kube-v1.20.5-vmware.2-tkg.1-6700972457122900687.ova
15. In the Select a name and folder window, click the folder you created in step 11, and click Next.
16. Select the test's compute resource pool, and click Next.
17. In the Review details window, click Next.
18. Accept the license agreement, and click Next.
19. Select the datastore, and click Next.
20. Select the cluster's network, and click Next.
21. In the Ready to complete window, click Finish.
22. After the upload completes, right-click the base-image VM, and select Template→Convert to template. In the pop-up window, click Yes.
23. Type the following script to apply a workaround for the latest kernel:

```
sudo sysctl net/netfilter/nf_conntrack_max=131072
```

24. Initialize the TKG environment for the local user on the VM (Note: The third command may display the error message "Error: unable to parse provider name: invalid provider name "". Provider name should be in the form name[:version] and name cannot be empty", which you can ignore):

```
rm -rf ~/.tanzu/tkg/bom
    export TKG_BOM_CUSTOM_IMAGE_TAG="v1.3.1-patch1"
    tanzu management-cluster create --ui
```

25. Open a browser, and navigate to https://127.0.0.1:8080/
26. In the VMware vSphere area, select Deploy.
27. On the vSphere 7.0.1 Environment Detected screen, click Deploy TKG management cluster.
28. Enter the following information:
    a. IP address for the vCenter Server
    b. The administrator account username
    c. The administrator account password
29. Click Connect.
30. On the Verify SSL Thumbprint screen, click Continue.
31. Enter or select the following information:
    a. Data center for the cluster
    b. The local account's SSH public key (`cat ~/.ssh/id_rsa.pub`)
32. Click Next.
33. In the Development window, enter the following information:
    a. Select the small instance type
    b. Type `mgmt` for the management cluster name
    c. Enter a local, non-DHCP IP address for the control plane endpoint, e.g., `10.220.156.165`
    d. Select the small worker node instance type
34. Click Next.
35. To bypass the VMware NSX Advanced Load Balancer screen, click Next.
36. To bypass the Metadata screen, click Next.
37. Select the VM folder, datastore, and resource pool, and click Next.

38. Select the VM network name, and click Next.
39. Disable Identity Management Settings, and click Next.
40. Select the OS image, and click Next.
41. To bypass the register screen, click Next.
42. Unselect Customer Experience Improvement Program, and click Next.
43. Click Review configuration.
44. Click Deploy management cluster.
45. After management cluster deploys, switch back to the deployment VM's shell, and query the cluster:

```
tanzu management-cluster  get mgmt
    kubectl get nodes
```

Creating the application and driver Kubernetes clusters

1. On the deployment VM, make two copies of the YAML cluster configuration file (which has a randomly generated name, such as 3nu4utlz5u.yaml), and name them `app.yaml` and drv.yaml:

```
cp ~/.tanzu/tkg/clusterconfigs/3nu4utlz5u.yaml ~/.tanzu/tkg/clusterconfigs/app.yaml
    cp ~/.tanzu/tkg/clusterconfigs/3nu4utlz5u.yaml ~/.tanzu/tkg/clusterconfigs/drv.yaml
```

2. Modify the new cluster files to change the cluster names, change their endpoint IP addresses, change the VM sizes (CPUs, memory, and local storage) for the worker nodes, and change the number of worker nodes. For example,

```
sed -i -e 's/^CLUSTER_NAME:.*/CLUSTER_NAME: app/' \
       -e 's/^VSPHERE_CONTROL_PLANE_ENDPOINT:.*/VSPHERE_CONTROL_PLANE_ENDPOINT: 10.220.55.61/' \
      ~/.tanzu/tkg/clusterconfigs/app.yaml
sed -i -e 's/^CLUSTER_NAME:.*/CLUSTER_NAME: drv/' \
       -e 's/^VSPHERE_CONTROL_PLANE_ENDPOINT:.*/VSPHERE_CONTROL_PLANE_ENDPOINT: 10.220.55.62/' \
      ~/.tanzu/tkg/clusterconfigs/drv.yaml
```

3. Create each Kubernetes cluster from its configuration file:

```
tanzu cluster create --file ~/.tanzu/tkg/clusterconfigs/app.yaml -v 6
    tanzu cluster create --file ~/.tanzu/tkg/clusterconfigs/drv.yaml -v 6
```

4. Add the cluster-access credentials to the Kubernetes configuration, and set the Kubernetes context to that of the cluster:

```
tanzu cluster kubeconfig get app --admin
    tanzu cluster kubeconfig get drv --admin
    kubectl config use-context app-admin@app
```

Creating the back-end storage for persistent volumes

Select the application cluster and add the storage class defined in the file ps.yaml (see below) to it.

```
kubectl config use-context app-admin@app
kubectl apply -f ps.yaml
```

{File ps.yaml}

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: wv-01-csi
  annotations:
    storageclass.kubernetes.io/is-default-class: "true"
provisioner: csi.vsphere.vmware.com
parameters:
  datastoreurl: "ds:///vmfs/volumes/610189a6-fa8d93ff-8fce-70b5e8d0c96a/"
```

Running the tests

In this section, we list the steps to run the VMware Weathervane benchmark on the SUT. We used the findMaximumSingleRun test type to determine the largest number of Weathervane users that our TKG clusters could sustain while still maintaining the quality of standards required to pass the benchmark.

1. Log into the run-harness instance.
2. Navigate to the Weathervane folder.
3. Open the run configuration file with the following command:

   ```
   nano weathervane.config.k8s.quickstart
   ```

4. Make sure the following parameters are filled out correctly (see our complete config file below):

   - description: A description of the run
   - configurationSize: `small2`
   - runStrategy: `findMaxSingleRun`
   - kubeconfigClusters: Kubernetes cluster names, kube config files, and kube cluster contexts for Weathervane application and driver clusters
   - appIngressMethod: `nodeport`
   - driverCluster: Weathervane driver cluster name
   - appInstanceCluster: Weathervane application cluster name
   - StorageClasses: Storage class name you created in previous section (“Creating the back-end storage for persistent volumes”)
   - numAppInstance: Number of app instances
   - reloadOnFailure: `true`

5. Exit the text editor.
6. Run the test with the following command:

   ```
   sudo ./runWeathervane.pl –configFile=weathervane.config.k8s.quickstart
   ```

7. Complete the test two more times for a total of three runs, and record the median run.

File weathervane.config.k8s.quickstart:

```
{
  "description" : "small2",
  "configurationSize": "small2",
  "runStrategy" : "findMaxSingleRun",
  "reloadOnFailure" : true,
  "numAppInstances" : 11,
  "dockerNamespace" : "joohny7long",
  "kubernetesClusters" : [
    {
      "name" : "appCluster",
      "kubeconfigFile" : "/home/ptuser/.kube/config",
      "kubeconfigContext" : "cluster01-admin@cluster01",
    },
    {
      "name" : "driverCluster",
      "kubeconfigFile" : "/home/ptuser/.kube/config",
      "kubeconfigContext" : "cluster02-admin@cluster02",
    },
  ],
  "driverCluster" : "driverCluster",
  "appInstanceCluster" : "appCluster",
  "appIngressMethod" : "nodeport",
  "cassandraDataStorageClass" : "wv-01-csi",
  "postgresqlStorageClass" : "wv-01-csi",
  "nginxCacheStorageClass" : "wv-01-csi",
}
```

Read the report at http://facts.pt/bhZYLHe ▶

This project was commissioned by Dell Technologies.