

Tanzu Mission Control Self-Managed Tech Preview for Cloud Service Providers

Providers Guide to Preview Tanzu Mission Control Self-Managed VCD Integration

Table of contents

Version History.....	4
Introduction	4
Architecture	4
Bill of Material	5
Prerequisites	5
Onboard the TMC-SM Solution Add-On	6
Prepare VCD	7
Deploy Installer VM	8
Create a self-signed certificate authority	10
Create Clusters	11
Deploy Harbor	13
Deploy TMC-SM for VCD	16
Accessing services	24
Configure Solution Org Cluster Backup	24
Publish the Solution Add-On to Tenants	25
Publish a TMC-SM rights bundle to tenants	25
Publish TMC-SM roles to tenants	25
Configure a TMC branding link	25
Configure Tenant Users	25
Share the TMC-SM details with tenants	26
Manage Tenant Clusters with TMC-SM	26
Configure Tenant Cluster Backup and Restore to OSE	28
Troubleshooting	31
Kubernetes Load Balancer comes up with the wrong IP	31
errcode: 3012 errmsg: Forbidden	31
errcode: 3001 errmsg: Unauthorized	31
TMC UI displays 403 Forbidden	31
TMC UI doesn't display any cluster groups in the attach cluster UI	31
Solution Add-On installation is stuck	31
An error occurred during login. Please, contact your administrator.	31
Solution Add-On Instance stuck with IN_PROGRESS/PENDING state	31
Removing a Solution Add-On Instance	32
Known Behavior	32
Sharing a browser tab with multiple identities can cause mixed results.	32
The Solution Add-On instance screen reports that global roles are missing.	32

Reference	32
Network Requirements	32
DNS Entries	33
Glossary	33

Version History

Date	Description
June 16, 2023	Initial version
June 21, 2023	<ul style="list-style-type: none"> Add version history Add storage policy specification to CSE cluster definitions
June 30, 2023	<ul style="list-style-type: none"> Update organization network requirements
July 17, 2023	<ul style="list-style-type: none"> Update Harbor deployment to use Tanzu packages

Introduction

One of the key growth areas for VMware Sovereign Clouds is to bring Sovereign-ready SaaS offerings into the global markets. Sovereign-compliant Tanzu Mission Control Self-Managed (TMC-SM) is the first VMware SaaS offering that is purpose-built and designed for highly-regulated and sovereign environments without any Hyperscaler or SaaS dependencies. Cloud Services Providers who offer Kubernetes Infrastructure as a Service to run container workloads in a multi-tenant environment using VMware Cloud Director (VCD) Container Service Extension can now centrally manage their multi-cluster Kubernetes and apply IT policies seamlessly using this new Tanzu Mission Control Self-Managed offering.

The TMC-SM VCD Integration expands the TMC-SM functionality with:

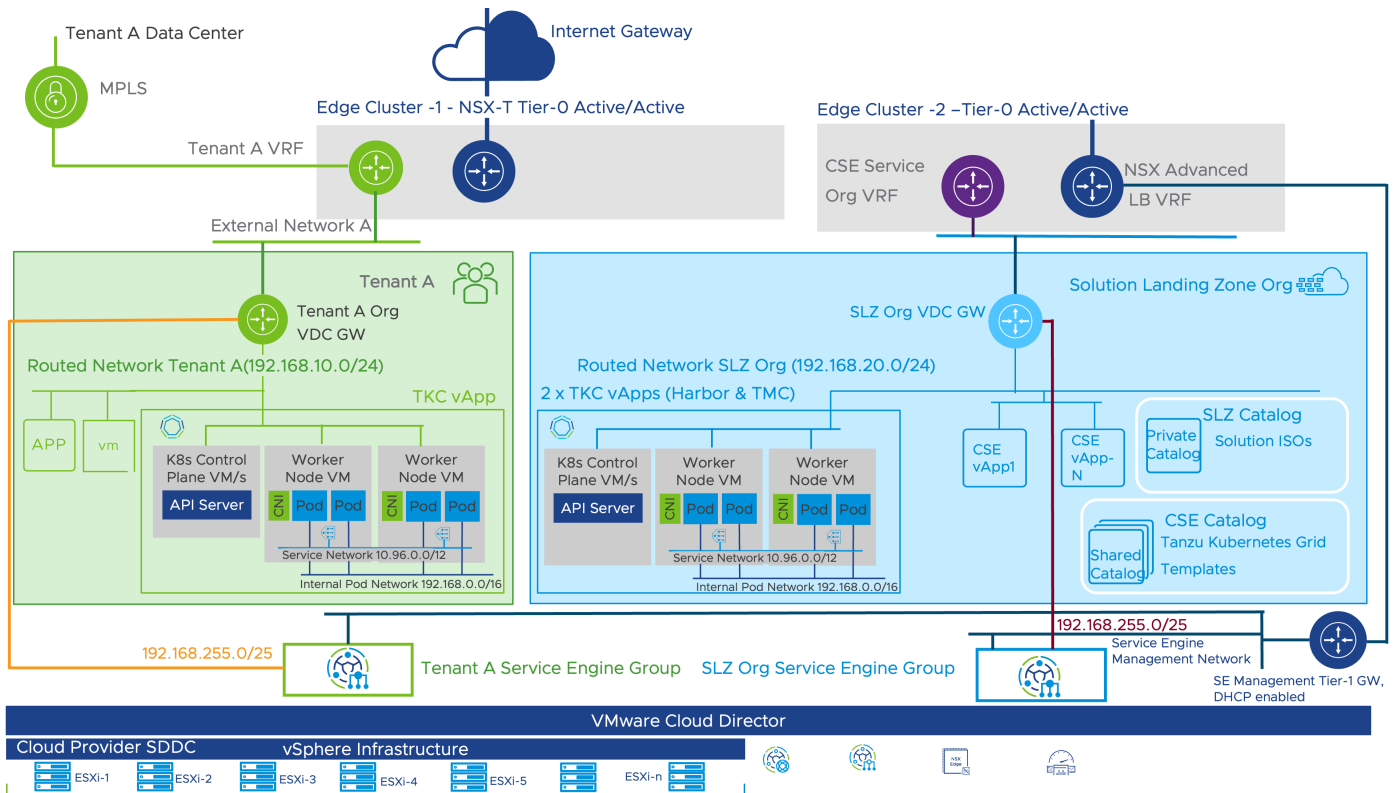
- Authentication through the VCD organization identity provider (IDP).
- Integration with the VCD Themes to provide direct links to TMC-SM.
- Cluster backup and restore to the Object Storage Extension (OSE).

Note: TMC-SM VCD Integration can be used to manage Container Service Extension (CSE) deployed clusters, but it does not include LCM for those clusters.

Architecture

TMC-SM VCD Integration is packaged and deployed via [VCD Solution Add-Ons](#). The add-on requires a Solution Landing Zone (SLZ) to be configured before installation. This process will identify an organization to be used as the Solution Org and a catalog to hold Solution Add-On ISOs.

The tech preview will deploy two Kubernetes clusters to the Solution Org through CSE. The first cluster will host a Harbor Registry to locally host TMC-SM images. The second cluster will host the TMC-SM services. Access to Harbor and TMC will be routed through Load Balancers on the Solution Org edge gateway.



Bill of Material

Software	Release
VMware Cloud Director	10.4.2 https://customerconnect.vmware.com/downloads/get-download?downloadGroup=VSPP_VCD1042
Container Service Extension	4.0.3 https://customerconnect.vmware.com/downloads/get-download?downloadGroup=VCD-CSE-4.0.3
Container Service Extension UI	4.0.400 https://customerconnect.vmware.com/downloads/get-download?downloadGroup=VCD-CSE-PLUGIN-4.0.400
Tanzu Kubernetes Grid	TKG 1.6.1. OVAs loaded into the CSE catalog.
Object Storage Extension (Optional)	2.2.1 https://customerconnect.vmware.com/downloads/get-download?downloadGroup=OSE221-TMC-VCD-TP
NSX-T	4.x
NSX Advance Load balancer aka AVI	Compatible with 10.4.2
TMC-SM for VCD Solution ISO	https://customerconnect.vmware.com/downloads/get-download?downloadGroup=TMC-VCD-S-ADD-ON

Prerequisites

Provision non-production environment based on the above BOM.

Solution Org

- This will be used to configure the Solution Landing Zone (SLZ)
- It will usually be the same organization used to host the CSE VM
- Quota to allocate 96 VCPU and 384 GB RAM
- Edge gateway with access to VCD and external networks
- External Traffic – Typically called North-South Traffic
 - Required set of IPs (Two for cluster control planes and one each for Installer VM, Harbor, and TMC service)
- Organization network
 - Scoped to the Org VDC. The SLZ does not support networks scoped to a data center group.
 - Bound to the edge gateway.
 - Available static IP pool of at least 64 IPs.
- See Network Requirements for more details.

Tenant Org

- Quota to allocate 32 VCPU and 128 GB RAM
- Edge gateway with access to VCD and external networks
- External Traffic – Typically called North-South Traffic
 - Required set of IPs (One for cluster control plane and one for application load balancer)
- Organization network
 - Bound to the edge gateway.
 - Available static IP pool of at least 64 IPs.
- See Network Requirements for more details.

DNS Entries

- Harbor hostname (e.g., harbor.slz.vcd.local) -> Harbor external IP
- TMC services DNS zone (e.g., tmc.slz.vcd.local and *.tmc.slz.vcd.local) -> TMC services external IP
- See DNS Entries for more details.

GitHub Personal API Token (Requires an account on GitHub)

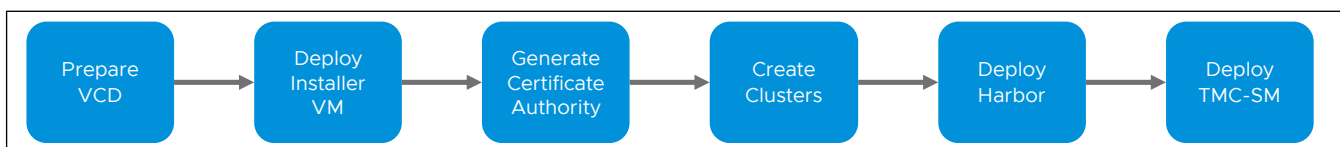
- <https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/managing-your-personal-access-tokens#creating-a-fine-grained-personal-access-token>
- The token only needs access to “Public Repositories (read-only)”

Onboard the TMC-SM Solution Add-On

This section will walk you through the steps to update VCD and CSE for the tech preview and deploy TMC Self-Managed for consumption by tenants.

We only support **CLI** installation during the tech preview. The process requires mounting the Solution Add-On ISO and executing commands to create the solution instance. We have documented a process that uses a temporary Installer VM in the Solution Org to execute the installation to CSE clusters. The Installer VM will not be required for installation from the UI.

The commands use environment variables for any configuration values which you are likely to override. You should create a local copy of these statements to easily copy and paste your values into the terminal session. You may also prefer to save them to a file on the Installer VM and load them into your environment with the **source** command. Values in **red** should be replaced with appropriate values for your environment. The examples have used nip.io hostnames as a default but you can fill in hostnames from your domain if you have configured the appropriate DNS entries from the prerequisites.



Prepare VCD

Update CSE

This tech preview includes an updated CSE UI that allows you to specify certificates that the cluster nodes should trust. The new UI must be loaded before creating clusters so the CA can be loaded into the cluster.

1. Open the Cloud Director Provider UI (e.g., <https://vcd.local/provider>)
2. Browse to More -> Customize Portal
3. Disable any old versions of the “Container UI Plugin” or “Kubernetes Container Clusters” plugins.
4. Upload the new UI plugin archive and publish it to all tenants.
5. Refresh the browser.
6. Browse to More -> Kubernetes Container Clusters
7. Click on “CSE Management”
8. Click on “Server Details”
9. Click on “Update Server”
10. Update the CSE server configuration to match these settings.
 - o CAPVCD - 1.0.2
 - o CPI - 1.3.0
 - o CSI - 1.3.2
 - o Github Personal API Token

CSE Server Components

CAPVCD Version	1.0.2
Cloud Provider Interface (CPI) Version	1.3.0 <small>This version will be used for TKG clusters</small>
Container Storage Interface (CSI) Version	1.3.2 <small>This version will be used for TKG clusters</small>
Github Personal Access Token (Optional)	<u>github_pat_ABCDEFGHIJKLM</u> <small>Prevents potential github rate limiting errors during cluster creation and deletion</small>

11. Click on “Submit Changes”
12. Restart the CSE VM to ensure the new configuration settings are reloaded.
 - o Open the Cloud Director Tenant UI for the Solutions Org
 - o Browse to Applications
 - o Click on “Actions” for the CSE vApp
 - o Select Power -> Reset

Configure Solution Landing Zone

1. Open the Cloud Director Tenant UI for the Solutions Org
2. Browse to Libraries -> Catalogs
3. Create a Catalog named “Solution Add-Ons” to hold Solution Add-On ISO files. Be sure to configure a storage policy for the catalog files.

Create Catalog

Name this Catalog

You can use a catalog for sharing vApp templates and media with other users in your organization. You can also have a private catalog for vApp templates and media that you frequently use.

Name *

Description

Pre-provision on specific storage policy

Org VDC

Storage Policy

4. Access or return to the Cloud Director Provider UI
5. Browse to More -> Solution Add-On Management
6. Click “Configure Landing Zone”

Follow the prompts to complete the process. You will need to configure the selected organization VDC before continuing the next step. Click the three vertical dots next to the name and select a default entry for network, compute policy and storage policy. These selections are not used by TMC-SM but they will be used by future solution add-ons. The SLZ does not support networks which are scoped to a data center group. You must decrease the scope of the network or create a new one if this applies to your environment.

Name	vCenter Name	Network (Required)	Compute Policies	Storage Policies (Required)
⋮ △ solutions-OVDC1 Default	vc.0	192.168.20.1/24 Default	System Default Default	vSAN Default Storage Policy Default
10 1 - 1 of undefined organization VDC(s)				

7. Upload the Solution Add-On ISO. Disable the “Create add-on instance” checkbox before uploading the ISO.

Deploy Installer VM

1. Browse to the Cloud Director organization for the SLZ.
2. Identify a catalog to hold the Installer VM image or create a new catalog.
3. Import the Photon 5 template into the catalog. Use the default name or give it a custom name if you prefer. https://packages.vmware.com/photon/5.0/GA/ovf/photon-hw15-5.0-dde71ec57.x86_64/photon-hw15-5.0-dde71ec57.x86_64.ovf
4. Create a vApp using the imported template from the catalog
 - a. Sizing Policy: TKG Large
5. Browse to the new vApp.
6. Attach the vApp to a routed network.
7. Browse to the Photon OS VM in the vApp.
8. Configure the NIC to connect to the routed network.
 - a. IP Mode: Static - IP Pool

9. Modify the Guest OS Customization settings to specify a root password.

Edit Guest Properties

General

Enable guest customization

The computer name and network settings configured for this VM are applied to its Guest OS when the VM is powered on. The following settings are only applied the 1st time the VM is powered on or if "Power on and Force Recustomization" is performed: Change SID, Password Reset, Join Domain and Customization Script. Guest customization should not be enabled if the VM uses Guest Properties for customization.

Password Reset

Allow local administrator password

Require Administrator to change password on first login

Auto generate password

Specify password

RootPassword

10. Attach the Solution Add-On ISO to the VM
 - a. Find the All Actions menu in the top right of the screen
 - b. Browse to All Actions->Media->Insert Media
 - c. Select the uploaded vmware-vcd-tmc-0.1.0-21897297.iso file
 - d. Click "Insert"
11. Start the vApp.
 - a. Return to the vApp configuration screen
 - b. Click "Start"
12. The Photon OS VM will start with SSH enabled. Configure a DNAT rule on the edge gateway for SSH traffic into the Photon OS VM.
 - a. Retrieve the IP address for the Photon OS VM

[All vApps](#) > [Installer VM](#) > Photon OS

Photon OS
Powered on

POWER ON POWER OFF LAUNCH WEB CONSOLE LAUNCH REMOTE CONSOLE ALL ACTIONS ▾

General

EDIT

Security Tags

Hardware

Removable Media

Hard Disks

Compute

Advanced

NICs

Guest OS

Customization

Guest Properties

Primary NIC	NIC	Connected	Network Adapter Type	Network	IP Mode	IP Address	External IP Address	MAC Address
Yes	0		VMXNET3	routed-192.168.20.1/24	Static - IP Pool	192.168.20.101	-	00:50:56

- b. Browse to the edge gateway for the network connected to the Photon OS VM

- c. Configure a DNAT rule for port 22 from an available external IP to the IP address assigned to the Photon OS VM

The screenshot shows a dialog box titled "Edit NAT Rule" with a close button (X) in the top right corner. The form contains the following fields:

- Name:** Installer VM SSH
- Description:** (empty text area)
- Interface Type:** DNAT (dropdown menu)
- External IP:** 192.168.116.130 (with an information icon)
- External Port:** 22
- Internal IP:** 192.168.20.101
- Application:** - (with an edit icon)

Below the Application field, there is a "Translated Port" label. At the bottom left, there is a link for "Advanced Settings". At the bottom right, there are two buttons: "DISCARD" and "SAVE".

13. Confirm you can SSH into the VM as the root user. The root password will be the one you assigned in the Guest OS Customization screen.
14. Prepare the Installer VM to run installation commands

Install kubectl, kctrl and required utilities. The openssl package gives us any easy way to inject the root CA into the system trust store.

```
# tdnf install -y git jq openssl-c_rehash tar unzip
# curl -L --output /usr/local/bin/kubectl \
  https://dl.k8s.io/release/v1.23.10/bin/linux/amd64/kubectl && chmod +x /usr/local/bin/kubectl
# curl -L https://github.com/carvel-dev/kapp-controller/releases/download/v0.46.1/kctrl-linux-
amd64 -o /tmp/kctrl && install /tmp/kctrl /usr/local/bin && rm /tmp/kctrl
```

Increase the capacity of /tmp to hold images prior to upload

```
# umount /tmp && mount -t tmpfs -o size=10G tmpfs /tmp
```

Mount the solution ISO to the Installer VM

```
# sed -i '/\mnt\cdrom/d' /etc/fstab
# mount /dev/sr0 /mnt/cdrom -t udf -o ro
```

Create a self-signed certificate authority


The next step is to create a self-signed certificate authority (CA) to sign certificates for Harbor and TMC-SM services. The steps below will generate a rootCA.key and rootCA.crt file. The contents of rootCA.crt will be provided as a trusted certificate during cluster creation so all cluster nodes trust certificates signed by the CA. Both files will be used to configure cert-manager so the CA can sign the certificates for Harbor and TMC.

```
# openssl req -x509 -sha256 -days 1825 -newkey rsa:2048 \
-keyout $HOME/rootCA.key -out $HOME/rootCA.crt \
-nodes -extensions v3_ca \
-subj "/C=US/ST=CA/L=Palo Alto/O=CompanyName/OU=OrgName/CN=TMC-SM VCD Tech Preview Issuing CA"
```


- Name: tmc
- Version: 1.6.1
- Control Plane
 - Number of nodes: 3
 - Sizing Policy: TKG Large
- Worker Pool
 - Number of Nodes: 4
 - Sizing Policy: TKG Extra-Large
- Kubernetes Storage: Configure the storage class to use the preferred storage policy for persistent volumes.
- Certificates: Paste the contents of rootCA.crt.

Store cluster KUBECONFIG files

Wait for the node pools of each cluster to be complete.

 harbor


RESIZE DOWNLOAD KUBE CONFIG UPGRADE DELETE SETTINGS

Overview **Node Pools** Kubernetes Storage Events

CREATE NEW WORKER NODE POOLS

Name	Node Count ⓘ	Sizing Policy	Storage Profile	GPU Activated	Placement/vGPU Policy
⋮ harbor-control-plane-node-po...	3 / 3	TKG mediu...		No	
⋮ harbor-worker-node-pool-1	3 / 3	TKG large		No	

1 - 2 of 2 Node Pools

 tmc

RESIZE DOWNLOAD KUBE CONFIG UPGRADE DELETE SETTINGS

Overview **Node Pools** Kubernetes Storage Events

CREATE NEW WORKER NODE POOLS

Name	Node Count ⓘ	Sizing Policy	Storage Profile	GPU Activated	Placement/vGPU Policy
⋮ tmc-control-plane-node-po...	3 / 3	TKG large		No	
⋮ tmc-worker-node-pool-1	4 / 4	TKG extra-lar...		No	

1 - 2 of 2 Node Pools

Download the KUBECONFIG file for each cluster and copy them to the Installer VM. These files should be set so only your user has read/write access to them because they include cluster credentials.

```
# chmod 600 kubeconfig-*
# ls -al kubeconfig-*
```

```
root@PhotonOS-001 [ ~ ]# chmod 600 kubeconfig-*
root@PhotonOS-001 [ ~ ]# ls -al kubeconfig-*
-rw----- 1 root root 5540 Jun  9 16:31 kubeconfig-harbor.txt
-rw----- 1 root root 5520 Jun  9 16:31 kubeconfig-tmc.txt
```

Deploy Harbor

The TMC-SM Solution requires a local instance of Harbor to host images. There are many ways to deploy Harbor. We are providing this mechanism to streamline the tech preview process. This deployment should provide enough capacity for the tech preview but does not reflect the appropriate sizing for large production environments.

Configure certificates

The Harbor services will use cert-manager to create certificates signed by the CA that you created. These steps load the CA into the cluster and configure a ClusterIssuer resource that will be specified in the Harbor configuration.

```
# export KUBECONFIG=$PWD/kubeconfig-harbor.txt
# kubectl create secret tls -n cert-manager selfsigned-ca-pair \
--cert=$HOME/rootCA.crt --key=$HOME/rootCA.key
# cat <<EOF | kubectl apply -f -

{
  "apiVersion": "cert-manager.io/v1",
  "kind": "ClusterIssuer",
  "metadata": {
    "name": "selfsigned-ca-clusterissuer"
  },
  "spec": {
    "ca": {
      "secretName": "selfsigned-ca-pair"
    }
  }
}
EOF
```

```

root@PhotonOS-001 [ ~ ]# export KUBECONFIG=$PWD/kubeconfig-harbor.txt
root@PhotonOS-001 [ ~ ]# kubectl create secret tls -n cert-manager selfsigned-ca-pair \
--cert=$HOME/rootCA.crt --key=$HOME/rootCA.key
secret/selfsigned-ca-pair created
root@PhotonOS-001 [ ~ ]# cat <<EOF | kubectl apply -f -
{
"apiVersion": "cert-manager.io/v1", "kind": "ClusterIssuer", "metadata": {
"name": "selfsigned-ca-clusterissuer" },
"spec": {
"ca": {
"secretName": "selfsigned-ca-pair" }
} }
EOF
clusterissuer.cert-manager.io/selfsigned-ca-clusterissuer created

```

Deploy Contour and Harbor

- Set environment variables with configuration values.


```

# IP address to associate with the Load Balancer for Harbor
export HARBOR_LOAD_BALANCER_IP="10.11.12.13"

# Desired hostname for the Harbor service. This must be configured to point to the IP
# address above.
export HARBOR_HOSTNAME="harbor.${HARBOR_LOAD_BALANCER_IP}.nip.io"

# This will be used as the initial password for the "admin" user
export HARBOR_ADMIN_PASSWORD="AdminPassword"

```
- Prepare a values file for the Contour installation


```

# cat <<EOF > contour-packageinstall-values.yaml
envoy:
  service:
    type: LoadBalancer
    loadBalancerIP: ${HARBOR_LOAD_BALANCER_IP}
EOF

```
- Deploy Contour using the Tanzu package


```

# kctrl package install \
-i contour \
-n tanzu-system \
--package contour.tanzu.vmware.com \
--version 1.20.2+vmware.2-tkg.1 \
--values-file contour-packageinstall-values.yaml

```

The kctrl command will wait until all resources are ready before returning. Check the cloud manager logs if there is an error or the service doesn't get assigned the correct External IP.

```
# kubectl -n kube-system logs -f deploy/vmware-cloud-director-ccm
```

4. Create a certificate for the Harbor services using the ClusterIssuer resource

```
# kubectl create ns tanzu-system-registry
```

```
# cat <<EOF | kubectl apply -f -
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
  name: ${HARBOR_HOSTNAME}
  namespace: tanzu-system-registry
spec:
  secretName: ${HARBOR_HOSTNAME}-tls

  duration: 8760h # 365d
  renewBefore: 720h # 30d
  subject:
    organizations:
      - MyOrgName
  isCA: false
  privateKey:
    algorithm: RSA
    encoding: PKCS1
    size: 2048
  usages:
    - server auth
    - client auth
  dnsNames:
    - ${HARBOR_HOSTNAME}
  ipAddresses:
    - ${HARBOR_LOAD_BALANCER_IP}
  issuerRef:
    name: selfsigned-ca-clusterissuer
    kind: ClusterIssuer
    group: cert-manager.io
EOF
```

5. Prepare a values file for the Harbor installation. This includes the hostname, certificate and default “admin” user password. It also defines the encryption keys for each Harbor component.

```
# cat <<EOF > harbor-packageinstall-values.yaml
secretKey: $(head -1 /dev/random | base64 | head -c 16)
core:
  secret: $(head -1 /dev/random | base64 | head -c 16)
  xsrfKey: $(head -1 /dev/random | base64 | head -c 32)
jobservice:
  secret: $(head -1 /dev/random | base64 | head -c 16)
registry:
  secret: $(head -1 /dev/random | base64 | head -c 16)
database:
  password: $(head -1 /dev/random | base64 | head -c 16)
hostname: ${HARBOR_HOSTNAME}
harborAdminPassword: ${HARBOR_ADMIN_PASSWORD}
tlsCertificateSecretName: ${HARBOR_HOSTNAME}-tls
notary:
  enabled: false
persistence:
  persistentVolumeClaim:
    registry:
      size: 128Gi
EOF
```

6. Deploy Harbor using the Tanzu package

```
# kctrl package install \
-i harbor \
-n tanzu-system \
--package harbor.tanzu.vmware.com \
--version 2.6.1+vmware.1-tkg.1 \
--values-file harbor-packageinstall-values.yaml
```

The kctrl command will wait until all resources are ready before returning.

```
# kubectl -n tanzu-system-registry get httpproxy
```

Deploy TMC-SM for VCD

Prepare Harbor

1. Login to <https://harbor.slz.vcd.local> as the admin user. The service is secured by a certificate signed by the certificate authority we created earlier. You will need to trust the certificate each time you visit a new address. You can optionally add the certificate authority to your operating system or browser trust store and the certificates will be verified against it.
2. Create a project to hold the TMC-SM images.
 - a. Browse to Projects
 - b. Click 'New Project'
 - c. Complete the form.
Project Name: tmc

Access Level: Public

The screenshot shows a configuration form for a project. It has a dark background with white text. The fields are: 'Project Name' with the value 'tmc'; 'Access Level' with a checked checkbox and the label 'Public'; 'Storage Quota' with the value '-1' and a unit dropdown set to 'GiB'; and 'Proxy Cache' with a disabled toggle switch.

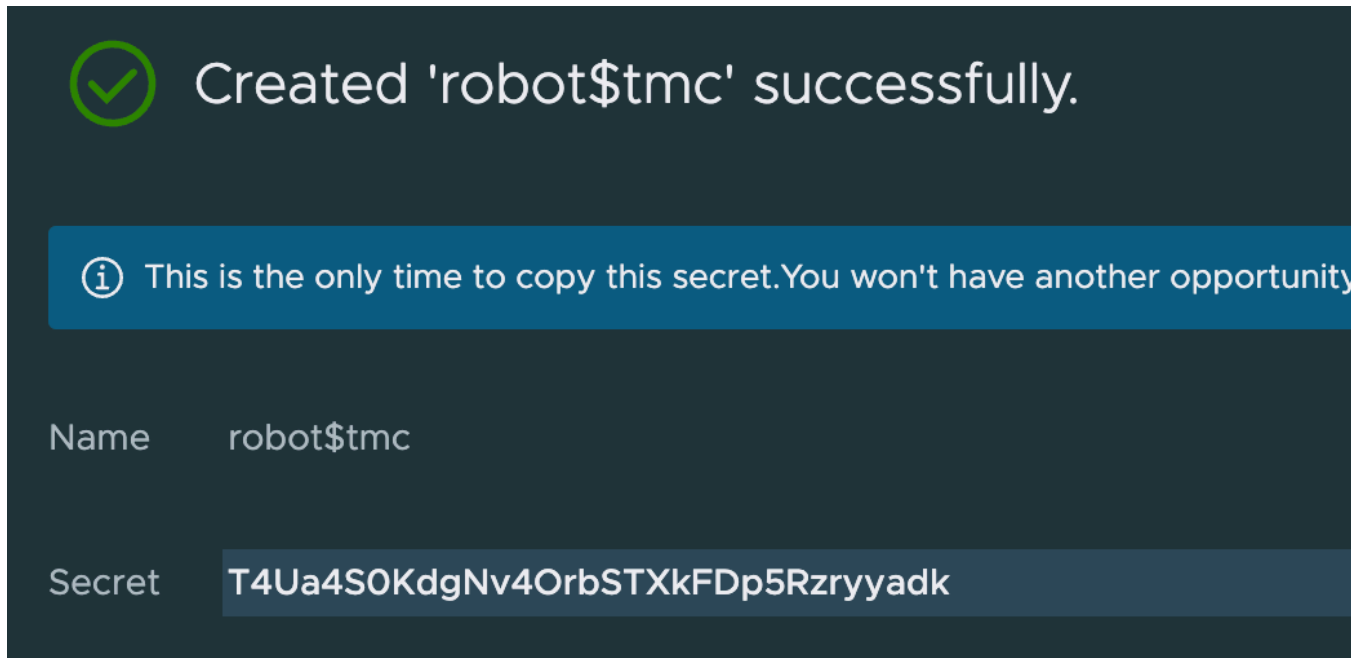
Project Name *	tmc
Access Level ⓘ	<input checked="" type="checkbox"/> Public
Storage Quota ⓘ *	-1 GiB ▾
Proxy Cache ⓘ	<input type="checkbox"/>

3. Create a Robot Account
 - a. Browse to Administration -> Robot Accounts
 - b. Click 'New Robot Account'
 - c. Complete the form.
Name: tmc
Expiration time: Never
Cover all projects: Checked

The screenshot shows a configuration form for a robot account. It has a dark background with white text. The fields are: 'Name' with the value 'tmc'; 'Expiration time' with a dropdown set to 'Never' and a value '-1'; 'Description' with an empty text area; and 'Cover all projects' with a checked checkbox and the text '19 PERMISSION(S) ▾'.

Name ⓘ *	tmc
Expiration time ⓘ *	Never ▾ -1
Description	<input type="text"/>
Cover all projects	<input checked="" type="checkbox"/> ⓘ 19 PERMISSION(S) ▾

- d. Save the account credentials shown on the next screen. You cannot retrieve this secret later.



Created 'robot\$tmc' successfully.

i This is the only time to copy this secret. You won't have another opportunity.

Name	robot\$tmc
Secret	T4Ua4S0KdgNv4OrbSTXkFDp5Rzryyadk

Configure certificates

The TMC services will use cert-manager to create certificates signed by the CA that you created. These steps load the CA into the cluster and configure a ClusterIssuer resource that will be specified in the TMC configuration.

```
# export KUBECONFIG=$PWD/kubeconfig-tmc.txt
# kubectl create secret tls -n cert-manager selfsigned-ca-pair \
--cert=$HOME/rootCA.crt --key=$HOME/rootCA.key
# cat <<EOF | kubectl apply -f -

{
  "apiVersion": "cert-manager.io/v1",
  "kind": "ClusterIssuer",
  "metadata": {
    "name": "selfsigned-ca-clusterissuer"
  },
  "spec": {
    "ca": {
      "secretName": "selfsigned-ca-pair"
    }
  }
}
EOF
```

```

root@PhotonOS-001 [ ~ ]# export KUBECONFIG=$PWD/kubeconfig-tmc.txt
root@PhotonOS-001 [ ~ ]# kubectl create secret tls -n cert-manager selfsigned-ca-pair \
--cert=$HOME/rootCA.crt --key=$HOME/rootCA.key
secret/selfsigned-ca-pair created
root@PhotonOS-001 [ ~ ]# cat <<EOF | kubectl apply -f -
{
"apiVersion": "cert-manager.io/v1", "kind": "ClusterIssuer", "metadata": {
"name": "selfsigned-ca-clusterissuer" },
"spec": {
"ca": {
"secretName": "selfsigned-ca-pair" }
} }
EOF
clusterissuer.cert-manager.io/selfsigned-ca-clusterissuer created

```

Install the Solution Add-On

1. Set environment variables with the desired configuration settings. These will be referenced in later commands or consumed by the Solution Add-On installation process. The environment variables prefixed with `VCD_EXT_` will be loaded into command-line options with the same name.

```

export VCD_HOSTNAME=vcd.example.com
export VCD_USERNAME=administrator
export VCD_EXT_PASSWORD=password

export TMC_SM_INSTANCE_NAME=VALUE_REQUIRED
export TMC_SM_ENCRYPTION_KEY=MySuperSecretKeyThatIRemember

# Provide the Kubernetes cluster name for TMC deployment,
# e.g., tkgm-tmc-cluster
export TMC_SM_KUBE_CLUSTER_NAME=VALUE_REQUIRED

# Provide DNS zone to configure TMC endpoints, i.e., tmc.mydomain.com
export TMC_SM_DNS_ZONE=VALUE_REQUIRED

# Provide the Load balancer IP of Contour Envoy, i.e., 10.11.12.23. TMC DNS
# Zone should be mapped to this IP.
export TMC_SM_LOAD_BALANCER_IP=VALUE_REQUIRED

# Provide Harbor project path for pushing/pulling TMC packages during
# installation, i.e., harbor.mydomain.com/myproject
export TMC_SM_HARBOR_URL=harbor.slz.vcd.local/tmc

# Provide Harbor username for Basic authentication
export TMC_SM_HARBOR_USERNAME=robot\${tmc}

# Provide Harbor password for Basic authentication
export VCD_EXT_INPUT_HARBOR_PASSWORD=VALUE_REQUIRED

# Provide the base64 encoded CA bundle in PEM format of the Harbor server.
# It is required if the Harbor server certificate is not signed by a

```

```

# well-known certificate authority.
export VCD_EXT_INPUT_HARBOR_CA_BUNDLE=$(cat $HOME/rootCA.crt | base64 -w0)

#####
# Optional Settings
#####

# Set MinIO root user name. Defaults to minioadmin.
export VCD_EXT_INPUT_MINIO_ROOT_USERNAME=

# Set MinIO root user password. If left blank, a random password will be
# generated. Format: no less than 8 chars, at least 1 digit, at least 1
# special char(@$!%*#. ,_-=*), at least 1 letter, i.e., P@ssw0rd
export VCD_EXT_INPUT_MINIO_ROOT_PASSWORD=

# Set TMC's PostgreSQL password. If left blank, a random password will be
# generated. Format: no less than 8 chars, at least 1 digit, at least 1
# special char(@$!%*#. ,_-=*), at least 1 letter, i.e., P@ssw0rd
export VCD_EXT_INPUT_POSTGRES_PASSWORD=S3cretPGP@ssw0rd

# Set the default Grafana admin user name. Defaults to admin.
export VCD_EXT_INPUT_GRAFANA_ADMIN_USERNAME=

# Set the default Grafana admin user password. If left blank, a random
# password will be generated. Format: no less than 8 chars, at least 1 digit,
# at least 1 special char(@$!%*#. ,_-=*), at least 1 letter, i.e., P@ssw0rd
export VCD_EXT_INPUT_GRAFANA_ADMIN_PASSWORD=

# Sets the timeout in seconds for TMC installation. Defaults to 3600.
export VCD_EXT_INPUT_DEPLOY_TIMEOUT=3600

```

- The Harbor server is using a certificate that is signed by the self-signed CA we created. The CA bundle needs to be added to the system certificates to trust the connection when images are uploaded during the installation. Use the `openssl` command to verify the certificate can be verified.

```

# cp $HOME/rootCA.crt /etc/ssl/certs/harbor.pem && rehash_ca_certificates.sh
# timeout 1 openssl s_client -quiet -verify_return_error ${HARBOR_HOSTNAME}:443

```

```

root@PhotonOS-001 [ ~ ]# cp $HOME/rootCA.crt /etc/ssl/certs/harbor.pem && rehash_ca_certificates.sh
root@PhotonOS-001 [ ~ ]# timeout 1 openssl s_client -quiet -verify_return_error ${HARBOR_HOSTNAME}:443
depth=1 C = US, ST = CA, L = Palo Alto, O = CompanyName, OU = OrgName, CN = TMC-SM VCD Tech Preview Issuing CA
verify return:1
depth=0
verify return:1

```

- Download the VCD certificate to a file. It will be used to trust the connection in later commands. The `linux.run` command validates the contents of all files in the solution when it is executed. This can take several minutes the first time it runs but will be shorter during subsequent commands.

```

# /mnt/cdrom/linux.run get certificates --host $VCD_HOSTNAME \
--output /tmp/vcd.pem \
--accept

```

```

root@PhotonOS-001 [ ~ ]# /mnt/cdrom/linux.run get certificates --host $VCD_HOSTNAME \
--output /tmp/vcd.pem \
--accept
-----BEGIN CERTIFICATE-----
MIIDkDCCAnigAwIBAgIIMIiFvLxURgRcwDQYJKoZIhvcNAQELBQAwNDEyMDA1UE
Awwdpnh5YW4tdm0tMTEuLTIwMC5uaW1idXMtdGIuZW5nLnZtd2FyZS5jb20wHhcN
MjMwNjA5MTQxMTM0WmcNMjQwNjA4MTQxMTM0WjA0MTIwMAYDVQQDDC12eGxhbi12
bS0xMTEtMjAwLm5pbWJ1cy10Yi5lYm9udm13YXJLLmNvbTCCASIwDQYJKoZIhvcN
AQEBBQADggEPADCCAQoCggEBAJ4ph0/qf4GxuWjinvfVUrtJKIRpz0k8CI7ZLIBi
yDnvNEYZR4gc+SNVIm5AF3413rP0l0Wb0w01SDa1UfLLvNoH0zkJFUrPxiR+0LLQ
9fcKxkJq0T21ziDm2Mo35/6dNHLmWd6A+WkQMunZjNcxqtYam0CAycpCjEiJTZhW
TvTd2JJ7zUt4EZ3YBFTJ+CiK5EgVEPJy3YL6STDSfSvxV3zSjSTBrzpLHEAa9AP
K7DbAYMj+qP/TzTQ+r5dFPRkv5z5RyA1kxpF+qMj1Bhn2GjImnUoDRUPBr4gTSy
51cFeyF9xIJ2LkLrw/xLsI8wKSRGmHr+G/g9gkbZEfdyPUSCAwEAa0BpTCBojAd
BgNVHQ4EFgQUKsaB3m3dTbFs0qxPF0zIa03bk7QwDgYDVR0PAQH/BAQDAgGOMFIG
A1UdEQRLMEcKXZ4bGFuLXZtLTEXMS0yMDAubmltYnVzLXRiLmVuZy52bXdhcmUu
Y29tghB2eGxhbi12bS0xMTEtMjAwLm5pbWJ1cy10Yi5lYm9udm13YXJLLmNvbTCC
AQUBwMBBggrBgEFBQcDAjANBgkqhkiG9w0BAQsFAAOCAQEAJXPHz4QkMvIBq/NM
r17QQY5FV0JHFrxHu7h0LQ6F/p1Mf0E014/wEG+HJuaF52wD+vfQGRHtV7Q0XGiC
a143BcV3auQ/avN8SVFt6XciFbFETNMdK+18i0GJHCaBWMNnBWYfuDXeeCLYU5aD
lgzxCtJKTxMNIr8nnM6PhAJRhh1NsAA7LXr1XJpuLuY14ij4Swhz+bktrnz2p6s5
tV5Kuoxp47bxVBrcxQoXRtP53ZsiTfj987oYqpqbxEiMtCOX1k3tT8Jo0nR8643+
N7Tjnnu5Etek0iZFnZLojnX0GJTwgZDKSETD4kNxI/CGLPgprQgwKAORxW4Iadoa
GuHNmW==
-----END CERTIFICATE-----

```

4. Configure VCD to trust the TMC-SM VCD Integration Solution Add-On.

```

# /mnt/cdrom/linux.run trust --host $VCD_HOSTNAME \
--username $VCD_USERNAME \
--certificate-file /tmp/vcd.pem \
--accept

root@PhotonOS-001 [ ~ ]# /mnt/cdrom/linux.run trust --host $VCD_HOSTNAME \
--username $VCD_USERNAME \
--certificate-file /tmp/vcd.pem \
--accept

```

5. Create the solution add-on instance. This will publish images to Harbor and deploy TMC on the cluster. The process can take up to an hour.

```
# /mnt/cdrom/linux.run create instance --name $TMC_SM_INSTANCE_NAME \
--host $VCD_HOSTNAME \
--username $VCD_USERNAME \
--certificate-file /tmp/vcd.pem \
--encryption-key ${TMC_SM_ENCRYPTION_KEY} \
--input-kube-cluster-name=${TMC_SM_KUBE_CLUSTER_NAME} \
--input-cert-provider=cluster-issuer \
--input-cert-cluster-issuer-name=selfsigned-ca-clusterissuer \
--input-dns-zone=${TMC_SM_DNS_ZONE} \
--input-contour-envoy-load-balancer-ip=${TMC_SM_LOAD_BALANCER_IP} \
--input-harbor-url=${TMC_SM_HARBOR_URL} \
--input-harbor-username=${TMC_SM_HARBOR_USERNAME} \
--accept
```

```
root@PhotonOS-001 [ ~ ]# /mnt/cdrom/linux.run create instance --name $TMC_SM_INSTANCE_NAME \
--host $VCD_HOSTNAME \
--username $VCD_USERNAME \
--certificate-file /tmp/vcd.pem \
--encryption-key ${TMC_SM_ENCRYPTION_KEY} \
--input-kube-cluster-name=${TMC_SM_KUBE_CLUSTER_NAME} \
--input-cert-provider=cluster-issuer \
--input-cert-cluster-issuer-name=selfsigned-ca-clusterissuer \
--input-dns-zone=${TMC_SM_DNS_ZONE} \
--input-contour-envoy-load-balancer-ip=${TMC_SM_LOAD_BALANCER_IP} \
--input-harbor-url=${TMC_SM_HARBOR_URL} \
--input-harbor-username=${TMC_SM_HARBOR_USERNAME} \
--accept

INFO [0026] Creating Solution instance entity           instance=vmware.vcd-tmc-0.1.0-21897297-tmcsm01
INFO [0027] Triggering action                               action=hook event=PreCreate
INFO [0028] Run EventPreCreate Hook                       action=hook event=PreCreate
INFO [0028] Run EventPreCreate Hook successfully             action=hook event=PreCreate
INFO [0028] Creating element                                   name=rde
INFO [0029] Creating element                                   name=tmc-admin-global-role
INFO [0030] Creating element                                   name=tmc-member-global-role
INFO [0030] Creating element                                   name=rights-bundle
INFO [0031] Triggering action                               action=hook event=PostCreate
INFO [0032] Run EventPostCreate Hook                       action=hook event=PostCreate
```

```

INFO [0436] 2023/06/09 17:37:33 harbor.192.168.116.140.nip.io/tmc/498533941640.dkr.ecr.us-west-2.amazonaws.com/policy-engine/patchdb:82ab85a55b14fe35e746e05d7e4a6fcd531d2a18: digest: sha256:d5af8a0eee96fc831823960fa89e290481718a00e6c161b5ce2abc4825baf89b size: 3241 action=hook event=PostCreate
INFO [0436] harbor.192.168.116.140.nip.io/tmc/498533941640.dkr.ecr.us-west-2.amazonaws.com/policy-engine/patchdb@sha256:d5af8a0eee96fc831823960fa89e290481718a00e6c161b5ce2abc4825baf89b action=hook event=PostCreate
INFO [0436] time="2023-06-09T17:37:33Z" level=info msg="Pushing image" progress=129/149 uri="harbor.192.168.116.140.nip.io/tmc/498533941640.dkr.ecr.us-west-2.amazonaws.com/policy-view-service/server:f600d63c33b53385b0a810307c6a8229c0b50cd3" action=hook event=PostCreate
INFO [0436] 2023/06/09 17:37:33 pushed blob: sha256:b245e3b6c425527cfbf23d12e8865d0bf77c52a1c22e6c2dac4244a90a4cedf action=hook event=PostCreate
INFO [0437] 2023/06/09 17:37:34 pushed blob: sha256:82ccfc419f6fbccddeab18ffebae8a9c6bba09d04f04769fb4c07b4903549fb3 action=hook event=PostCreate
INFO [0437] 2023/06/09 17:37:34 pushed blob: sha256:d3c894b5b2b0fa857549aeb6cbc38b038b5b2828736be37b6d9fff0b886f12fd action=hook event=PostCreate

INFO [0847] Release "tmc-local-stack" does not exist. Installing it now. action=hook event=PostCreate
INFO [0847] NAME: tmc-local-stack action=hook event=PostCreate
INFO [0847] LAST DEPLOYED: Fri Jun 9 17:44:08 2023 action=hook event=PostCreate
INFO [0847] NAMESPACE: tmc-local action=hook event=PostCreate
INFO [0847] STATUS: deployed action=hook event=PostCreate
INFO [0847] REVISION: 1 action=hook event=PostCreate
INFO [0847] TEST SUITE: None action=hook event=PostCreate
INFO [0847] tmc-local-stack tmc-local 1 2023-06-09 17:44:08.178267831 +0000 UTC deployed t
mc-local-stack-20230519.164404.175660443+97ea225 dev action=hook event=PostCreate
INFO [1155] Release "monitoring" does not exist. Installing it now. action=hook event=PostCreate
INFO [1155] NAME: monitoring action=hook event=PostCreate
INFO [1155] LAST DEPLOYED: Fri Jun 9 17:44:30 2023 action=hook event=PostCreate
INFO [1155] NAMESPACE: tmc-local action=hook event=PostCreate
INFO [1155] STATUS: deployed action=hook event=PostCreate
INFO [1155] REVISION: 1 action=hook event=PostCreate
INFO [1155] TEST SUITE: None action=hook event=PostCreate
INFO [1155] monitoring tmc-local 1 2023-06-09 17:44:30.470439131 +0000 UTC deployed t
mc-local-monitoring-0.0.13 0.0.1 action=hook event=PostCreate
INFO [1155] Finish running cmd action=hook event=PostCreate
INFO [1155] Start to install grafana action=hook event=PostCreate
INFO [1155] Prepare for grafana installation action=hook event=PostCreate
INFO [1155] Install grafana package action=hook event=PostCreate
INFO [1156] Create the grafana PackageInstall action=hook event=PostCreate
INFO [1156] Wait for the grafana PackageInstall to be reconciled action=hook event=PostCreate
INFO [1369] Create grafana dashboards action=hook event=PostCreate
INFO [1369] Grafana was installed successfully action=hook event=PostCreate
INFO [1371] Successfully created instance 'tmcsm01' name=tmcsm01

```

6. Check on TMC pod status

```
# kubectl config set-context --current --namespace=tmc-local
```

```
# kubectl get pods
```

```
root@PhotonOS-001 [ ~ ]# kubectl config set-context --current --namespace=tmc-local
Context "tmc-admin@tmc" modified.
root@PhotonOS-001 [ ~ ]# kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
account-manager-server-656ddb8dff-g8lxc	1/1	Running	0	10m
account-manager-server-656ddb8dff-tkw95	1/1	Running	0	10m
agent-gateway-server-6fcb7f8d6b-4mcnj	1/1	Running	4 (6m8s ago)	10m
agent-gateway-server-6fcb7f8d6b-jr8h5	1/1	Running	2 (6m30s ago)	10m
alertmanager-monitoring-tmc-local-0	2/2	Running	0	10m
api-gateway-server-5464c6bc8-bpzlk	1/1	Running	1 (7m46s ago)	10m
api-gateway-server-5464c6bc8-zck8d	1/1	Running	4 (6m11s ago)	10m
audit-service-consumer-7b84cdd8cb-58xdv	1/1	Running	0	10m
audit-service-consumer-7b84cdd8cb-vxz4s	1/1	Running	0	10m
audit-service-server-7fccd8d849-npxqc	1/1	Running	0	10m
audit-service-server-7fccd8d849-q6hk6	1/1	Running	0	10m
auth-manager-server-599754ffdc-8gjx8	1/1	Running	0	10m
auth-manager-server-599754ffdc-vdvlw	1/1	Running	0	10m
auth-manager-server-599754ffdc-z64ml	1/1	Running	0	10m
authentication-server-58fcf7bf6d-dyrzc	1/1	Running	0	10m

Accessing services

The TMC UI is now available at <https://tmc.slz.vcd.local>. Accessing this URL will not work until you have a TMC role associated with your user.

The TMC-SM Solution includes an instance of Grafana to monitor the health of TMC-SM services. The UI is available at <https://grafana.tmc.slz.vcd.local>. You can retrieve the “admin” user password if you did not configure it as part of the solution installation.

```
# kubectl -n tmc-local get secrets/grafana -o jsonpath='{.data.admin-password}' | base64 -d
```

```
root@PhotonOS-001 [ ~ ]# kubectl -n tmc-local get secrets/grafana -o jsonpath='{.data.admin-password}' | base64 -d
20g%u8jtr
root@PhotonOS-001 [ ~ ]# █
```

Configure Solution Org Cluster Backup

Note: This section is specific to the Object Storage Extension. Skip to the next section if you do not have it installed for this environment.

The Object Storage Extension supports backup and restore for Kubernetes clusters. This process will create backups of the complete cluster hosting Harbor or TMC services.

1. Open the Cloud Director Tenant UI for the Solution Org
2. Browse to More -> Object Storage
3. Browse to “Kubernetes Clusters”
4. Expand the “Unprotected Clusters” section
5. Click “Start Protection” for the cluster hosting Harbor or TMC services
6. Complete the form with your desired backup settings.
7. Submit the form and wait for the process to complete. This may take several minutes to complete.
8. Browse to “Kubernetes Clusters”
9. Expand the “Protected Clusters” section

10. See that the cluster is listed with a “Protection Preparing” badge. The badge should eventually change to “In Protection”. This may take up to 15 minutes to complete.

Publish the Solution Add-On to Tenants

Publish a TMC-SM rights bundle to tenants

1. Browse to Administration -> Rights Bundles
2. Select “vmware:tmc_tenant Rights Bundle”
3. Click “Publish”
4. Publish to all tenants or select the tenants that will be allowed to use TMC-SM.

Publish TMC-SM roles to tenants

1. Browse to Administration -> Global Roles
2. Select “tmc:admin”
3. Click “Publish”
4. Publish to all tenants or select the tenants that will be allowed to use TMC-SM.
5. Select “tmc:member”
6. Click “Publish”
7. Publish to all tenants or select the tenants that will be allowed to use TMC-SM.

Configure a TMC branding link

1. Open the Cloud Director Provider UI
2. Browse to Administration -> Feature Flags
3. Ensure the “Branding API” flag is set to “Enabled”
4. Refresh the browser
5. Browse to More -> Customize Portal -> Themes
6. If the active theme is one of the base themes, you will not be able to make changes.
 - a. Create a clone of the default theme
 - b. Click Actions -> Make Default for the cloned version of the theme
 - c. Reload your browser
7. Click “Details” for the active theme
8. Update the name if you had to clone a base theme
9. Browse to “Links”
10. Click “(+) Menu Items” -> Link
 - a. Link text: Tanzu Mission Control
 - b. URL: <https://tmc.slz.vcd.local>
This is the base DNS Zone you used when deploying the solution add-on.
11. Click “Save”
12. Reload your browser

Configure Tenant Users

Note: TMC-SM requires a “Full name” for all users.

1. Open the Cloud Director Tenant UI for a tenant organization
2. Browse to Administration -> Users
3. Create a local user to demonstrate TMC administration
 - a. Name: tmc-admin
 - b. Role: tmc:admin
 - c. Full name: TMC Administrator
4. Create another local user to demonstrate basic TMC usage
 - a. Name: tmc-member
 - b. Role: tmc:member
 - c. Full name: TMC Member

Share the TMC-SM details with tenants

Tenant users need to use the certificate authority contents when they attach clusters or connect to TMC-SM. This will also allow them to establish the necessary trust relationship for the tmc CLI access to work. Give tenant users the contents of the rootCA.crt file or retrieve it from the cluster.

```
# kubectl -n cert-manager get secret/selfsigned-ca-pair -o=jsonpath='{.data.tls\.crt}' | base64 -d
```

Users will need additional details if they want to use the tmc CLI.

TMC Hostname: This is the base DNS Zone you used when deploying the solution add-on.

VCD OIDC Issuer URL: Append '/oidc' to the base VCD URL. (e.g.; <https://vcd.local/oidc>)

VCD OIDC Client ID: This can be retrieved from the Cloud Director Provider UI

- Open the Cloud Director Provider UI.
- Browse to Administration -> OIDC Proxy.
- Identify the entry which includes the TMC-SM DNS Zone in the Redirect URIs.
- Save the "Relying Party ID" value.

Manage Tenant Clusters with TMC-SM

This section covers the tenant process to create and manage a new CSE cluster. These steps should not be executed from the Installer VM. They can be executed from any machine with the kubectl CLI.

1. Login as the tmc-admin user
2. Create cluster with the content from rootCA.crt
3. Download the KUBECONFIG file for the new cluster
4. Open the branding link created above by browsing to the top-right menu (3 dots) -> Tanzu Mission Control
5. Browse to Clusters
6. Click "Attach Cluster"
7. Fill out the form with the appropriate details
8. The TMC-SM UI will display a **kubectl create** command. This command can be used to create resources on the cluster that will attach it to the TMC-SM services. Some configuration is needed for the command to work because the TMC server uses a certificate signed with the CA we generated earlier.
 - a. Add the TMC-SM certificate authority to the operating system's trusted certificates. The process to do this will vary depending on your platform.

Alternatively, you can store the TMC-SM certificate authority content in a local file. The kubectl CLI will refer to the SSL_CERT_FILE environment when it runs.

```
# vi $HOME/tmcCA.pem
```

```
# export SSL_CERT_FILE=$HOME/tmcCA.pem
```

- b. Run the command against the cluster to be managed.

```
# export KUBECONFIG=$PWD/kubeconfig-managed01.txt
```

```
# kubectl create -f
```

```
"https://tmc.192.168.116.141.nip.io/installer?id=836c1528fa6a24c0fb7ca2795e33524c7eeb5eed03"
```

```
ea075200401ebd2e2f3c40&source=attach"
```

```
kubo@ovIT9fcubUUY1:~$ export KUBECONFIG=$PWD/kubeconfig-harbor.txt
kubo@ovIT9fcubUUY1:~$ kubectl create -f "https://tmc.192.168.116.141.nip.io/installer?id=836c1528fa6a24c0fb7ca27c7eeb5eed03ea075200401ebd2e2f3c40&source=attach"
namespace/vmware-system-tmc created
configmap/stack-config created
secret/tmc-access-secret created
serviceaccount/extension-updater-serviceaccount created
Warning: policy/v1beta1 PodSecurityPolicy is deprecated in v1.21+, unavailable in v1.25+
podsecuritypolicy.policy/vmware-system-tmc-agent-restricted created
clusterrole.rbac.authorization.k8s.io/extension-updater-clusterrole created
clusterrole.rbac.authorization.k8s.io/vmware-system-tmc-ppsp-agent-restricted created
clusterrolebinding.rbac.authorization.k8s.io/extension-updater-clusterrolebinding created
clusterrolebinding.rbac.authorization.k8s.io/vmware-system-tmc-ppsp-agent-restricted created
service/extension-updater created
deployment.apps/extension-updater created
customresourcedefinition.apiextensions.k8s.io/agents.clusters.tmc.cloud.vmware.com created
customresourcedefinition.apiextensions.k8s.io/detachconfigs.intents.tmc.cloud.vmware.com created
customresourcedefinition.apiextensions.k8s.io/extensionconfigs.intents.tmc.cloud.vmware.com created
customresourcedefinition.apiextensions.k8s.io/extensionintegrations.clusters.tmc.cloud.vmware.com created
customresourcedefinition.apiextensions.k8s.io/extensionresourceowners.clusters.tmc.cloud.vmware.com created
customresourcedefinition.apiextensions.k8s.io/extensions.clusters.tmc.cloud.vmware.com created
serviceaccount/extension-manager created
clusterrole.rbac.authorization.k8s.io/extension-manager-role created
clusterrolebinding.rbac.authorization.k8s.io/extension-manager-rolebinding created
service/extension-manager-service created
deployment.apps/extension-manager created
serviceaccount/agent-updater created
clusterrole.rbac.authorization.k8s.io/agent-updater-role created
clusterrolebinding.rbac.authorization.k8s.io/agent-updater-rolebinding created
deployment.apps/agent-updater created
Warning: batch/v1beta1 CronJob is deprecated in v1.21+, unavailable in v1.25+; use batch/v1 CronJob
cronjob.batch/agentupdater-workload created
```

- c. Wait for all pods to start and you see the agentupdater-workload jobs run every minute. This should take 5-10 minutes.

```
# kubectl get pods -n vmware-system-tmc -w
```

```
agentupdater-workload-28105621-86xqr      0/1      Pending      0          0s
agentupdater-workload-28105621-86xqr      0/1      Pending      0          0s
agentupdater-workload-28105621-86xqr      0/1      ContainerCreating  0          1s
agentupdater-workload-28105621-86xqr      1/1      Running      0          5s
agentupdater-workload-28105620-714z4      0/1      Terminating  0          70s
agentupdater-workload-28105620-714z4      0/1      Terminating  0          70s
agentupdater-workload-28105621-86xqr      0/1      Completed    0          15s
agentupdater-workload-28105621-86xqr      0/1      Completed    0          17s
```

- d. Try the “Verify Connection” button to ensure the connection has been made. Retry this step if the verification fails or not all pods are running.

9. Return to the TMC-SM UI
10. Browse to “Clusters”
11. Click on the newly attached cluster name.
12. The cluster details will appear in the TMC-SM UI after TMC-SM has collected cluster details.

Configure Tenant Cluster Backup and Restore to OSE

Note: This section is specific to the Object Storage Extension. Skip to the next section if you do not have it installed for this environment.

TMC provides backup and restore functionality in the same portal used for other management activities. The backup artifacts are stored in an S3-compatible object storage service. This process creates a new OSE storage bucket and configures TMC to store backup artifacts into that bucket.

1. Login to the Cloud Director Tenant UI as a tenant administrator
2. Browse to More -> Object Storage
3. Click on "Buckets"
4. Click on "New Bucket"
5. Choose the desired region for the bucket and give it a name. The bucket can be used to backup multiple clusters.

Create Bucket

Asterisk(*) denotes required

Region *

Name *

Activate Versioning

Object Lock

6. Click on the name of the newly created bucket
7. Click on "Properties"
8. Record the name, region, and S3 URL for the bucket
9. Click on "Security Credentials"
10. Click on "Application Credentials"
11. Create a new application credential that can access the new bucket

Create Application Credential

Asterisk(*) denotes required

Application Name *

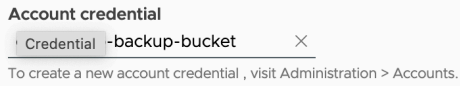
Apply the credential to all buckets

Apply the credential to selected buckets

Region *

Buckets

19. Browse to Administration -> Target locations
20. Click on “Create Target Location”
21. Click on “AWS S3 or S3-compatible”
22. Select the previously created account credential



23. Configure the location with the saved details. The URL should be in the form of <https://OSE-hostname> without the port or subpath.

AWS S3 or S3-compatible URL
<https://ose-tmc-sm.eng.vmware.co>
Input must use http:// or https:// as prefix for the URL

Bucket
 tmc-tp-backup-bucket

Region ⓘ
 us-north-1

Custom root/CA certificate (optional)
 -----BEGIN CERTIFICATE-----
 OPGXSq
 W87447Uc6W7j4pQ0Vfbj9ruuOHwrANO9w0NJ/Vu/tYO.TikoRTJEYg
 rrBGZKBxmm
 4nEFJwREfsmUUQBOBEb0OKIPvx2m0H0DA4ZotHwZuGnrD6eha+/g
 5kyBl0eDTc
 Mweyhavsmfx9Kg7ZlxU/BXiE2sldUnP4GL6j7CJKEw==
 -----END CERTIFICATE-----

24. Select the cluster groups or clusters you would like to have access to this bucket.

Select clusters ✕

1 Clusters selected

Name Health Cluster group Management cluster Show selected clusters

Name	Health	Cluster group	Provider	Management cluster	Provisioner
<input checked="" type="checkbox"/> managed01	✔ Healthy	default	Unknown	attached	attached

1 1 to 1 of 1 Cluster

25. Give the target location a name and submit the form.
26. Browse to “Clusters”
27. Open a cluster that you would like to backup
28. Click on “Enable Data Protection”
29. Click “Enable” to allow TMC to install the Velero operator to the cluster
30. Wait for the operator to become available. This may take several minutes.
31. Click on “Create Backup”

32. Complete the form and select the newly created target location to store the backup.

All locations ⓘ
 Currently available locations ⓘ

Target location
 tmc-tp-backup-bucket × [VIEW DETAILS](#)

To create a new target location, visit [Administration > Target locations](#).

[NEXT](#)

33. Wait for the backup process to be complete. This should take a few minutes for an empty cluster or longer for a cluster with a lot of data.

The TMC-SM tenant is now connected to OSE to provide storage for cluster backup and restore operations. The OSE portal can be used to view backup artifacts. TMC-SM supports much more functionality than is shown here. The TMC documentation can provide further details on restore and cross-cluster operations.

Troubleshooting

Kubernetes Load Balancer comes up with the wrong IP

```
# kubectl -n kube-system set image deploy/vmware-cloud-director-ccm vmware-cloud-director-ccm=projects.registry.vmware.com/vmware-cloud-director/cloud-provider-for-cloud-director:1.3.0
```

```
# kubectl -n kube-system get pods -w
```

errcode: 3012 errmsg: Forbidden

This error message can appear if the VCD user does not has not been assigned one of the “tmc:*” roles.

errcode: 3001 errmsg: Unauthorized

This error message can appear if the VCD user does not have a Full Name set for their user. Update the user record with a Full Name and attempt TMC-SM login again.

TMC UI displays 403 Forbidden

This error message can appear if the VCD user does not has not been assigned one of the “tmc:*” roles.

TMC UI doesn't display any cluster groups in the attach cluster UI

This error message can appear if the VCD user does not has not been assigned one of the “tmc:*” roles.

Solution Add-On installation is stuck

```
INFO [0072] tmc-local-stack-secrets      tmc-local      1      2023-06-05 13:28:04.283020194 +0000 UTC
           deployed      tmc-local-stack-secrets-0.0.1      action=hook event=PostCreate
```

This can happen if the target cluster is not configured with the appropriate certificate. Create a new cluster and ensure the root CA is included as a trusted certificate.

An error occurred during login. Please, contact your administrator.

This can occur for a couple of reasons:

- The rights bundle has not been published to the tenant attempting to log in.
- The VCD user does not been assigned one of the “tmc:*” roles

Solution Add-On Instance stuck with IN_PROGRESS/PENDING state

The TMC-SM solution add-on only allows a single instance. This instance can become stuck in an IN_PROGRESS state if you experience an issue while running the **create instance** command.

- Generate and store a VCD access token to authenticate the following API calls.

```
# export VCLOUD_ACCESS_TOKEN=`curl -ksSL -D - -X POST
https://${VCD_HOSTNAME}/cloudapi/1.0.0/sessions/provider -H "Accept:
application/json;version=37.0" -u "${VCD_USERNAME}@system:${VCD_EXT_PASSWORD}" | grep X-VMWARE-
VCLOUD-ACCESS-TOKEN | sed 's/.*: //' | tr -cd '[:alnum:]._-'`
```
- Retrieve the current state of the TMC-SM entity

```
# curl -ks -H "Accept: application/json;version=37.0" -H "Authorization: Bearer
${VCLOUD_ACCESS_TOKEN}" -X GET
"https://${VCD_HOSTNAME}/cloudapi/1.0.0/entities/types/vmware/solutions_add_on_instance/1.0.0" |
jq --arg name ${TMC_SM_INSTANCE_NAME} '.values[] | select((.name | startswith("vmware.vcd-tmc"))
and (.entity.name == $name))' > tmc-entity.json
```
- Modify the tmc-entity.json file to set entity.status to "FAILED".
- Update the state of the TMC-SM entity with the new status

```
# curl -ks -H "Accept: application/json;version=37.0" -H "Content-Type: application/json" -H
"Authorization: Bearer ${VCLOUD_ACCESS_TOKEN}" -X PUT -d @tmc-entity.json
"https://${VCD_HOSTNAME}/cloudapi/1.0.0/entities/${cat tmc-entity.json | jq -r .id}" | jq .
```

Removing a Solution Add-On Instance

Use this command to delete a Solution Add-On Instance. It is not possible to delete an instance with a status of IN_PROGRESS or PENDING. See the section above for steps to override the instance status.

Note: Be sure to set the required environment variables before attempting to run the command.

```
# /mnt/cdrom/linux.run delete instance --name $TMC_SM_INSTANCE_NAME \
--host $VCD_HOSTNAME \
--username $VCD_USERNAME \
--certificate-file /tmp/vcd.pem \
--encryption-key ${TMC_SM_ENCRYPTION_KEY} \
--accept
```

Known Behavior

Sharing a browser tab with multiple identities can cause mixed results.

Be sure to logout of TMC-SM and VCD before attempting to use another identity. Clear the browser cache if you have inconsistent behavior.

The Solution Add-On instance screen reports that global roles are missing.

This will be resolved in GA.

Reference

Network Requirements

ALLOW FOLLOWING PORT/PROTOCOLS FOR CSE, HARBOR AND TMC SELF-MANAGED COMMUNICATION		
SOURCE	PORT/PROTOCOL	DESTINATION ENDPOINT
Solution Org Network	443	Cloud Director
Solution Org Network	443	Internet

Tenant Org Network	443	Cloud Director
Tenant Org Network	443	Internet
Tenant Org Network	443	Harbor Load Balancer IP
Tenant Org Network	443	TMC-SM Load Balancer IP
Solution Add-On Admin	22	Installer VM IP on the Solution Org Edge Network
Provider Users	443	Cloud Director
Provider Users	443	TMC-SM Load Balancer IP
Tenant Users	443	Cloud Director
Tenant Users	443	TMC-SM Load Balancer IP

DNS Entries

DNS LIST FOR TMC LOCAL	
Harbor	A single record is needed pointing the Harbor hostname (e.g., harbor.slz.vcd.local) used in the “Deploy Contour and Harbor” section to the Load Balancer IP used in the same section.
TMC-SM DNS Zone	<p>TMC-SM is configured with a DNS Zone (e.g., tmc.slz.vcd.local) that it uses as a base for the microservices. The DNS Zone is provided in the “Install the Solution Add-On” section. The following records are needed for each of these services to work. Each record should point to the Load Balancer IP provided in the “Install the Solution Add-On” section.</p> <ul style="list-style-type: none"> • alertmanager.<zone> • auth.<zone> • blob.<zone> • console.s3.<zone> • grafana.<zone> • gts-rest.<zone> • gts.<zone> • landing.<zone> • pinniped-supervisor.<zone> • prometheus.<zone> • s3.<zone> • tmc-local.s3.<zone> • <zone>

Glossary

cert-manager

This component is added to each CSE cluster as it is created to sign certificate requests via custom resources on the cluster. See <https://cert-manager.io> for more details.

Certificate Authority (CA)	Certificate Authorities (CA) sign certificates. The tech preview process creates a single CA and configures our components to use it for all certificates for Harbor and TMC-SM. The CA certificate may be trusted by intermediate hosts and end users to trust all certificates signed by the CA.
ClusterIssuer	This is a custom resource created on Kubernetes clusters to configure how cert-manager can sign certificate requests.
Contour	Contour is an open-source ingress controller for Kubernetes. See https://projectcontour.io for more details.
Harbor	Harbor is an open-source registry for artifacts like Docker images. See https://goharbor.io for more details.
Harbor Robot User	This is the term Harbor uses for service accounts. They authenticate with a system-generated password. The administrator can control what privileges the robot user will have.
Solution Add-On	A solution add-on is the representation of a solution that is custom built for VMware Cloud Director in the VMware Cloud Director extensibility ecosystem. See https://docs.vmware.com/en/VMware-Cloud-Director/10.4/VMware-Cloud-Director-Service-Provider-Admin-Portal-Guide/GUID-4F12C8F7-7CD3-44E8-9711-A5F43F8DCEB5.html for more details.
Solution Landing Zone (SLZ)	The Solution Add-On Landing Zone is a part of the provider management plane that represents a pool of compute, storage and networking resources dedicated to hosting, managing, and running solution add-ons on behalf of the cloud provider.
Solution Org	The Solution Org is a tenant organization used by the provider to host solution add-ons. It is configured in the solution landing zone as the source of compute, storage and networking resources.

About the Authors

Jeff Mace is a Staff 2 Engineer at VMware in the Cloud Infrastructure Business Group.

Rohan Mukesh is a Sr. Staff Solutions Architect at VMware in the Modern Application and Management Business unit.

