



Sangfor KubeManager

Quick Deployment Guide

Product Version	6.0
Document Version	01
Released on	Dec. 12, 2021



Copyright © Sangfor Technologies Inc. 2021. All rights reserved.

Unless otherwise stated or authorized, Sangfor Technologies Inc. (hereinafter referred to as "Sangfor") and its affiliates reserve all intellectual property rights, including but not limited to copyrights, trademarks, patents, and trade secrets, and related rights to text, images, pictures, photographs, audio, videos, charts, colors, and layouts as presented in or concerning this document and content therein. Without prior written consent of Sangfor, this document and content therein must not be reproduced, forwarded, adapted, modified or displayed or distributed by any other means for any purpose.

Disclaimer

Products, services or features described in this document, whether wholly or in part, may be not within your purchase scope or usage scope. The products, services or features you purchase must be subject to the commercial contract and terms as agreed by you and Sangfor. Unless otherwise provided in the contract, Sangfor disclaims warranties of any kind, either express or implied, for the content of this document.

Due to product version upgrades or other reasons, the content of this document will be updated from time to time. Unless otherwise agreed, this document is used for reference only, and all statements, information, and recommendations therein do not constitute any express or implied warranties.

Technical Support

For technical support, please visit: <https://www.sangfor.com/en/about-us/contact-us/technical-support>

Send information about errors or any product related problem to tech.support@sangfor.com.

About This Document






This document describes the quick deployment guide of Sangfor KubeManager V6.0.

Intended Audience

This document is intended for:

- Cloud Engineer
- Operations Engineer

Note Icons

English Icon	Description
	Indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury.
	Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury.
	Indicates a hazardous situation, which if not avoided, could result in minor or moderate injury.
	Indicates a hazardous situation, which if not avoided, could result in settings failing to take effect, equipment damage, or data loss. NOTICE addresses practices not related to personal injury.
	Calls attention to important information, best practices, and tips. NOTE addresses information not related to personal injury or equipment damage.

Change Log

Date	Change Description
Dec.12, 2021	This is the first release of this document.

Contents

Technical Support	1
Change Log	2
1 Introduction	4
2 Overview	5
3 Prerequisites.....	6
3.1 Installation Files	6
3.2 Environment Requirements	7
3.2.1 Non-highly Available Environment	7
3.2.2 Highly Available Environment	7
3.3 Installation of KubeManager	9
3.3.1 Installation of KubeManager in HCI.....	9
3.3.2 Installation of User Cluster	13
4 Common Problems and Troubleshooting Steps	16
4.1 Network Problems	16
4.2 Other Problems.....	19

1 Introduction

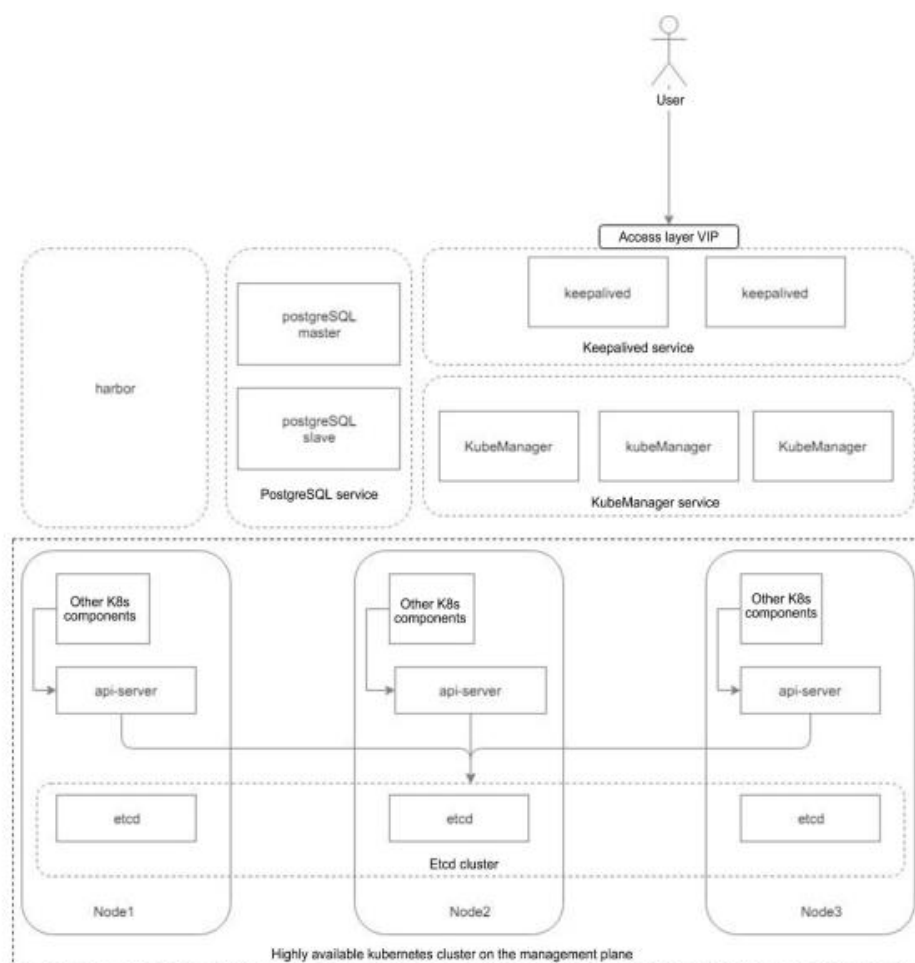
KubeManager is a simple, easy-to-use, and stable cloud-native multi-K8S management platform released by Sangfor. By applying cloud-native technology, KubeManager runs all system components on K8S. KubeManager provides default registry and app store, supports integration with the third-party registry and app store, and supports third-party account system.

KubeManager is a lightweight and stable commercial Kubernetes multi-cluster management system.

In this document, we will be introducing the deployment of KubeManager in the Sangfor HCI cluster.

2 Overview

The overview of the structure of Sangfor KubeManager is shown as below:



It can be seen from the figure above that the bottom layer of KubeManager is also a Kubernetes system, the KubeManager program runs as an application in Kubernetes, and at the same time, the entire system also provides services such as database and harbor.

- The keepalived service ensures the high availability of interfaces through floating IP addresses.
- The daemonset service ensures the access performance expansion and high availability of services.
- The postgreSQL service component ensures the high availability of databases. The operator component allows automatic configuration of

various database modes (single host, master-slave, and one-master-to-multiple-slave). It can be used to store audit logs or harbor user information.

- The etcd cluster ensures the high availability of configuration data. All configuration information of KubeManager is stored in K8S at the underlayer.
- KubeManger runs on Kubernetes as a service. Kubernetes guarantees the high availability of all components of KubeManager.

3 Prerequisites

3.1 Installation Files

Name	MD5	Function
CentOS-7-x86_64-Minimal-1908-user_cluster.vma	3f96c649a5ea26ff6a95d7cd9cb1b1af	VMA file of user cluster for HCL.
CentOS-7-x86_64-Minimal-1908-kubemanager.vma	d996199b7b04a444be25a9d3688d086f	VMA file of KubeManager for HCL.
Sangfor_KubeManager_6.0_20200930.run	682D598997057EE693460D70930FF875	Installation package of KubeManager.

Table 1: The required files for the installation of KubeManager in HCL.



For other installation methods (such as physical node installation), kindly consult Sangfor technical assistance for further assistance.

3.2 Environment Requirements

3.2.1 Non-highly Available Environment

Host	Hostname	CPU	Memory	Disk	Installation Component
Host 1	KubeManager	4C	8GB	80 GB system disk without partition, mount 50 GB.	Manage clusters, registries, and app stores.
Host 2	etcd, Controller, Worker.	8C	8GB	100 GB system disk without partition, mount 150 GB.	User cluster.
Host 3	etcd, controller, worker.	8C	8GB	100 GB system disk without partition, mount 150 GB.	User cluster.
Host 4	etcd, controller, worker.	8C	8GB	100 GB system disk without partition, mount 150 GB.	User cluster.

3.2.2 Highly Available Environment

Host	Hostname	CPU	Memory	Disk	Installation Component
Host 1	KubeManager	4C	8GB	80 GB system disk without partition, mount 50 GB.	Manage clusters, registries, and app stores.
Host 2	KubeManager	4C	8GB	80 GB system disk without partition, mount 50 GB.	Manage clusters, registries, and app stores.
Host 3	KubeManager	4C	8GB	80 GB system disk without partition, mount 50 GB.	Manage clusters, registries, and

					app stores.
Host 4	etcd, Controller, Worker.	8C	8GB	100 GB system disk without partition, mount 150 GB.	User cluster.
Host 5	etcd, controller, worker.	8C	8GB	100 GB system disk without partition, mount 150 GB.	User cluster.
Host 6	etcd, controller, worker.	8C	8GB	100 GB system disk without partition, mount 150 GB.	User cluster.
Host 7	worker	8C	8GB	100 GB system disk without partition, mount 150 GB.	User cluster.
Host 8	worker	8C	8GB	100 GB system disk without partition, mount 150 GB.	User cluster.



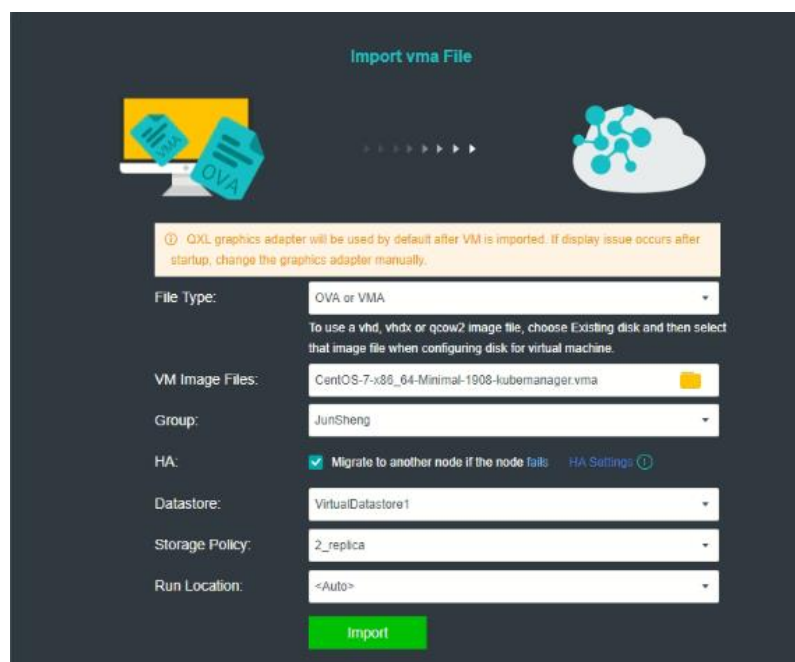
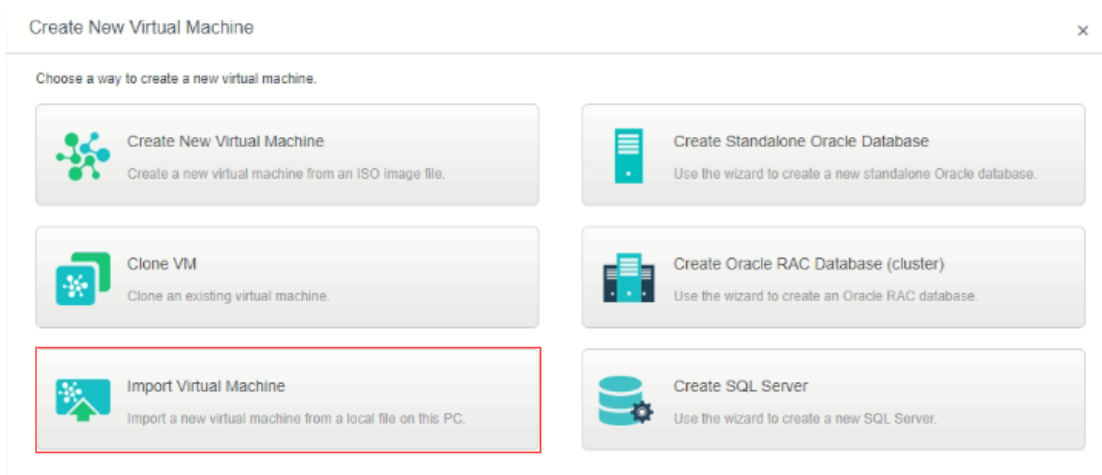
KubeManager can be installed on VM, bare metal, cloud server, other infrastructures and supports various network topologies. For example, the deployment in VPC, single NIC deployment in the classic network, multi-NIC deployment in the classic network.

3.3 Installation of KubeManager

In this chapter, we will mainly focus on the installation of KubeManager in the HCI classic network.

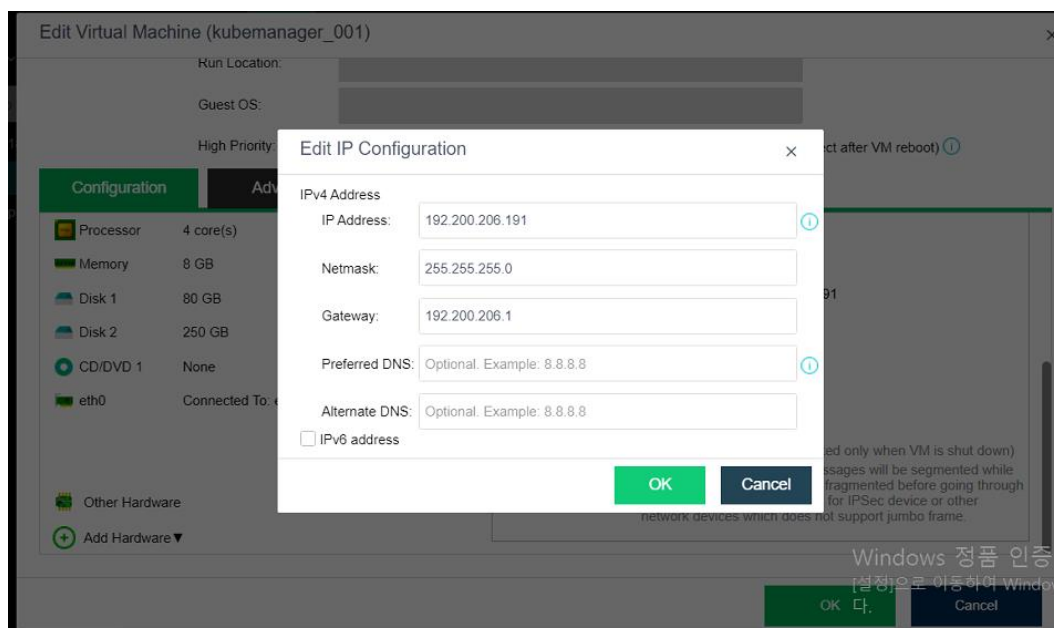
3.3.1 Installation of KubeManager in HCI

1. Import the related vma files in **Section 3.1** to the HCI cluster. It's advised to verify the MD5 checksum of the vma files to ensure integrity.



Installation files can obtain from the Sangfor representative.

2. Assign IP and configure the network to every KubeManager node and user cluster node.



3. SSH to one of the KubeManager Nodes with credentials (root/ Sangfor-paas.237). Upload the **Sangfor_KubeManager_6.0_20200930.run** file through sftp and grant the execute permission to the file. After this, execute the following commands to run the installation scripts.



Change the IP address according to your environment.

```
./Sangfor_KubeManager_6.0_20200930.run -- -e skm=10.113.81.161 \  
++ -e skm_api_server_virtual_ip=10.113.81.164 \  
-e harbor_ext_endpoint=10.113.81.165 \  
-e harbor_ext_vip=10.113.81.165 \  
-e ansible_ssh_port=22 \  
-e ansible_ssh_pass=password \  
-e remote_base_dir=/sf/data/pgsql/skm-deploy \  
-e kube_keepalived_vip_vrid=xxx
```

Parameter	Description
SKM	Nodes IP.
skm_api_server_virtual_ip	Kubemanager Virtual IP.
harbor_ext_vip	Harbor Virtual IP.
ansible_ssh_pass	Nodes SSH password, all Kubemanager nodes required to be same.
remote_base_dir	Temporary directory for deployment.
kube_keepalived_vip_vrid	Vrrp id for the cluster.

```
[root@localhost ~]# ./Sangfor_KubeManager_6.0_20200902_B.run -- -e skm=192.200.206.191,192.200.206.192,192.200.206.193 +
.206.190 -e harbor_ext_endpoint=192.200.206.194 -e harbor_ext_vip=192.200.206.194 -e ansible_ssh_pass=Sangfor-paas.237 -
eploy -e kube_keepalived_vip_vrid=255
Verifying archive integrity... 100% SHA256 checksums are OK. 100% MD5 checksums are OK. All good.
Uncompressing KubeManager installer x86_64 git b56d337 100%
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match '
localhost | CHANGED => {
  "changed": true,
  "checksum": "5ed4faf990c88455e5d556cc4f5d92746ed10731",
  "dest": "./other/install-scripts/../../other/install-scripts/hosts.ini",
  "gid": 0,
  "group": "root",
  "md5sum": "6e818c831d13a96598e01e9e95832994",
  "mode": "0600",
  "owner": "root",
  "secontext": "unconfined_u:object_r:admin_home_t:s0",
  "size": 246,
  "src": "/root/.ansible/tmp/ansible-tmp-1614746720.12-11406-133464945215445/source",
  "state": "file",
  "uid": 0
}
PLAY [localhost] *****
TASK [stat ssh key exists] *****
Wednesday 03 March 2021 12:45:21 +0800 (0:00:00.048) 0:00:00.048 *****
ok: [localhost]
```

- After the installation script is executed, you may access KubeManager using virtual IP and login Harbor using Harbor IP.

- Upon the first login, it will require the user to reset the password of the KubeManager.

Reset Password

Please input your account information.

Current Password:

New Password: ☒ Create my own password ☐ Use a randomly generated password

New Password

Confirm Password

Submit

- Login page of KubeManager.

KubeManager

KubeManager contributes to a more integrated Cloud IT architecture. It provides central management of multi-Kubernetes clusters featuring stability, usability and reliability. By virtue of multi-tenant management, applications, image registry and third-party services, KubeManager allows you to manage your own containers based on a variety of built-in infrastructures.

SANGFOR CLOUD

KubeManager

Version: v6.0

Username

Password

☐ Remember me

☒ I have read and accept the Terms of Use | Privacy Policy

Log In

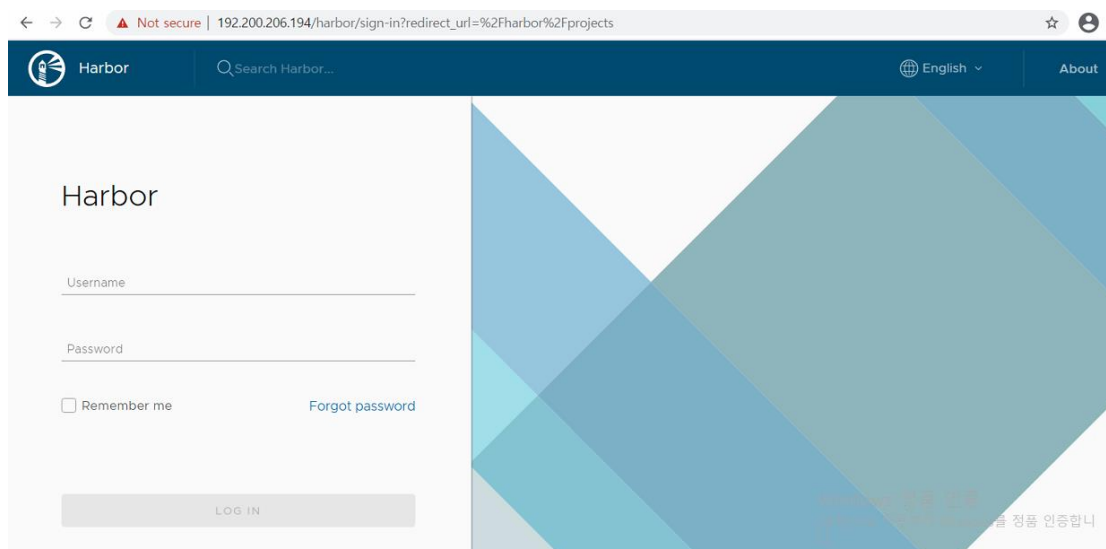
Windows 정품 인증
[설정]으로 이동하여 Windows를 정품 인증합니
다.

©2012-2021 Sangfor Technologies Inc. All rights reserved

7. Login page of Harbor, the default credential of Harbor:

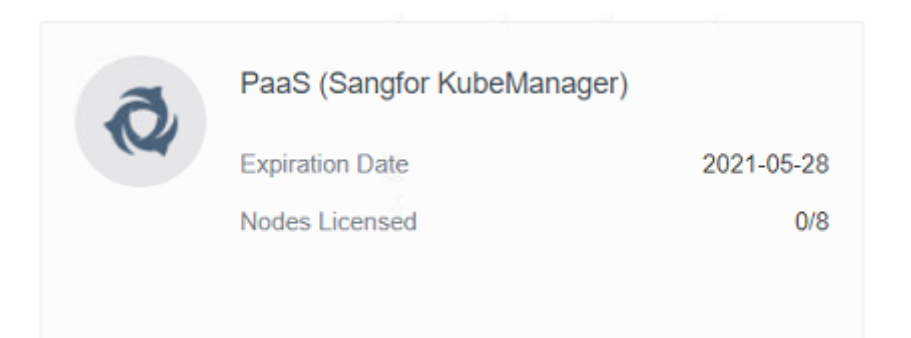
Username: admin

Password: Harbor12345



3.3.2 Installation of User Cluster

1. Before the installation, please log in to SCP and apply for a container license. Before this, it must make sure SCP does have a PaaS license.



Create Container License

Cluster Name:

192.200.206.160

IP Address:

192.200.206.160

Port:

443

Connectivity:

Start Test

Nodes Licensed:

8

OK

Cancel

Licensing

Platform License

Cluster License

Security Resource License

Container License

Services

New

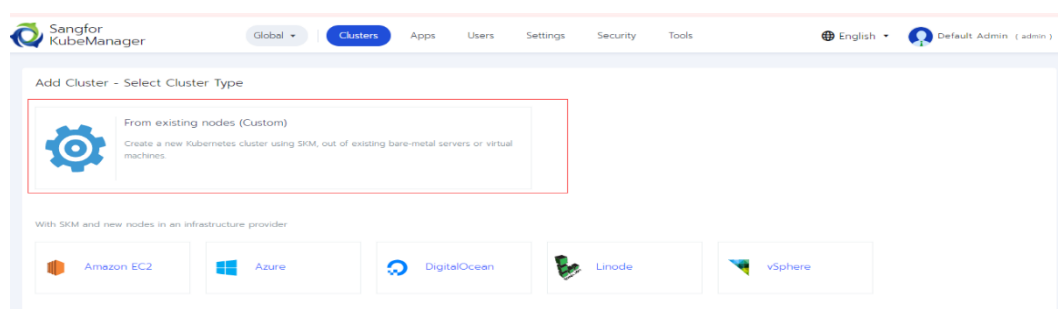
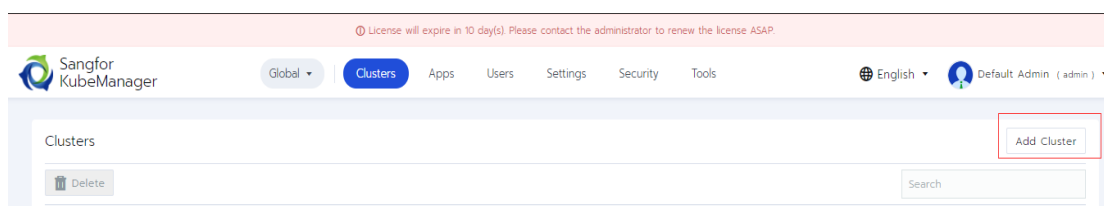
Refresh

Name

Status	Name	IP Address	Nodes Licensed	Operation
<div>✓ Licensed(expiration date: 2021-06-01)</div>	192.200.206.190	192.200.206.190	0/8	<div>Edit Deallocate</div>

0

- After having a PaaS license, we can log in to the KubeManager web page and install the user cluster. Then, under **Global** mode, select **Add Cluster** at the top right corner.



- Upon creating the user cluster, you will need to fill in the IP for each node and the login credentials. After this, assign the user cluster nodes' roles (etcd, Control Plane, Worker).

- Test the connectivity of the node and set the role of the node:

Hostname Prefix	IP	etcd	Control Plane	Worker	Labels	Taints	Connection Status
node1	Edit	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Labels	Taints	Normal
node2	Edit	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Labels	Taints	Normal
node3	Edit	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Labels	Taints	Normal
node4	Edit	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Labels	Taints	Normal
node5	Edit	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Labels	Taints	Normal

Number of nodes required: 1, 3, or 5 1 or more 1 or more

[+ Add Node](#) New Nodes Allowed: 8

- The function of the user cluster :

Etcd: The main function of the etcd node is data storage, which is responsible for storing kubemanager data and cluster status. The state of the Kubernetes cluster is stored in the etcd node, which runs the etcd database. It is recommended to run etcd on multiple nodes to ensure that

the backup cluster data can be obtained in case of a node failure.

Controller: The workloads running on the controlplane node include: Kubernetes API server, scheduler, and controller manager. These nodes perform daily tasks to ensure that your cluster status matches your cluster configuration. Although you can run a controlplane on a single node, we recommend running a controlplane on two or more nodes to ensure redundancy.

Worker: The work node runs the following applications.

Kubelet: An agent that monitors the status of nodes to ensure that your containers are in a healthy state.

Workload: The containers and pods that host your applications and other types of deployments.

The worker nodes also run storage and network drivers. There will also run an ingress controller if necessary. Kubemanager has no limit on the number of worker nodes. You can create them according to actual needs.

4 Common Problems and Troubleshooting Steps

4.1 Network Problems

Scenario 1: The deployment process exits due to network delays or other unknown reasons.

Possible reasons:

1. The management plane cluster deployment is deployed by multiple nodes

simultaneously, and the network communication needs to be maintained during the process. It causes the deployment to fail due to the network.

2. During the deployment process, the node disk or other reasons caused the management of the Kubernetes cluster to be abnormal. The process cannot update related resources, and the deployment fails.

Solution: Clean up the deployment and re-deploy.

Step 1. Clean up deployment.

```
./Sangfor_KubeManager_6.0_20200930.run -- --uninstall \  
-e skm=10.113.81.161,10.113.81.162,10.113.81.163 \  
++ -e ansible_ssh_port=22345 \  
-e ansible_ssh_pass=password
```

Clean up the execution background process, running docker ps and docker images. Both the mirror and the container are deleted, indicating that the cleanup is complete.

Step 2. Redeploy.

```
./Sangfor_KubeManager_6.0_20200930.run -- \  
-e skm=10.113.81.161,10.113.81.162,10.113.81.163 \  
++ -e skm_api_server_virtual_ip=10.113.81.164 \  
-e harbor_ext_endpoint=10.113.81.165 \  
-e harbor_ext_vip=10.113.81.165 \  
-e ansible_ssh_port=22345 \  
-e ansible_ssh_pass=password \  
-e remote_base_dir=/sf/data/pgsql/skm-deploy
```

Scenario 2: The deployment reports an error, prompting vrid conflict.

```

=====
TASK [check_network : Check Virtual Ntr ID] *****
Monday 27 July 2020 16:16:23 +0800 (0:00:33.703) 0:02:11.550 *****
fatal: [172.16.1.11]: FAILED! =>
  "changed": false
  "assertion": "\Virtual Ntr ID: 200" not in tshark_output.stdout,
  "evaluated": false
  "msg": "CheckKeepalived vrid 200 has conflict, please append -e kube_keeplived_vip_vrid=xxx command option and the vrid must be 0 - vrid = 255"

NO MORE HOSTS LEFT *****
to retry, use: --limit @/tmp/sgfpr1542719167/install.retry

PLAY RECAP *****
172.16.1.11 : ok=2 changed=0 unreachable=0 failed=0
172.16.1.12 : ok=0 changed=0 unreachable=0 failed=0
172.16.1.13 : ok=0 changed=0 unreachable=0 failed=0
localhost  : ok=1 changed=0 unreachable=0 failed=0

=====
Monday 27 July 2020 16:16:23 +0800 (0:00:00.125) 0:02:11.676 *****
install_docker : execute script install docker ..... 72.53s
check_network : Run tshark capture packet ..... 31.70s
check_network : copy netshoot images ..... 8.63s
install_docker : copy static docker binary ..... 6.30s
gather_facts : ..... 3.30s
install_docker : template manifests file ..... 1.42s
install_docker : create a directory if it doesn't exist ..... 1.05s
stat ssh key exists : ..... 0.78s
check_install : set architecture group ..... 0.25s
install_docker : import os specific tasks ..... 0.27s
check_install : check os resource ..... 0.26s
check_network : virtual ip should be defined ..... 0.25s
check_install : ansible architecture rename ..... 0.23s
task_defaults : Configure defaults ..... 0.23s
check_install : include tasks ..... 0.22s
install_docker : set os specific template directory ..... 0.21s
install_docker : setting registry url cert ..... 0.21s
check_install : Check times distribution support ..... 0.20s
[root@localhost ~]# ping 172.16.1.12

```

Possible reasons: The vrid of the management cluster conflicts with the vrid of the current network.

Solution :

Step 1. Packet capture by using the tcpdump command to look at the existing vrid. After this, plan a vrrp id and vrid range that can be used without conflict. It can only be between 1 and 255.

Command : tcpdump -i eth0 vrrp -nevl

Step 2. Add the command line parameter **-e kube_keeplived_vip_vrid=xxx** to re-run the deployment command. (No need to perform cleanup)

```

./Sangfor_KubeManager_6.0_20200930.run -- \
-e skm=10.113.81.161,10.113.81.162,10.113.81.163 \
++ -e skm_api_server_virtual_ip=10.113.81.164 \
-e harbor_ext_endpoint=10.113.81.165 \
-e harbor_ext_vip=10.113.81.165 \
-e ansible_ssh_port=22345 \
-e ansible_ssh_pass=password \
-e remote_base_dir=/sf/data/pgsql/skm-deploy \
-e kube_keeplived_vip_vrid=xxx

```

4.2 Other Problems

Scenario 3: Deployment stuck, stuck in Reconcile HA and patch KubeManager.

Possible reason: High-availability node deployment stuck during automatic HA configuration may be due to network jitter or other reasons. The node state is not within the expected range, so it's stuck.

```
TASK [deploy_k8s_post : Reconcile HA and patch KubeManager] *****
Saturday 08 August 2020 09:17:14 +0000 (0:00:02.889)    0:30:20.523 *****
skipping: [192.168.20.38]
skipping: [192.168.20.39]
```

Solution :

Step 1. Press **Ctrl+C** to interrupt the deployment to clean up the environment and redeploy.

Step 2. Clean up deployment.

```
./Sangfor_KubeManager_6.0_20200930.run -- --uninstall \
-e skm=10.113.81.161,10.113.81.162,10.113.81.163 \
++ -e ansible_ssh_port=22345 \
-e ansible_ssh_pass=password
```

Step 3. Re-deploy.

```
./Sangfor_KubeManager_6.0_20200930.run -- \
-e skm=10.113.81.161,10.113.81.162,10.113.81.163 \
++ -e skm_api_server_virtual_ip=10.113.81.164 \
-e harbor_ext_endpoint=10.113.81.165 \
-e harbor_ext_vip=10.113.81.165 \
-e ansible_ssh_port=22345 \
-e ansible_ssh_pass=password \
-e remote_base_dir=/sf/data/pgsql/skm-deploy
```

Scenario 4: A conflict between the user network and docker default docker0 bridge (172.17.0.0/16 or 172.18.0.0/16). It is resulting in deployment failure (interruption) suddenly.

```
TASK [install_docker : import os specific tasks] *****
Wednesday 26 August 2020 18:12:47 +0800 (0:00:00.058) 0:00:02.985 *****
included: /tmp/selfgz1685415775/roles/install_docker/tasks/centos-7.7.1908.yml for 172.17.25.31

TASK [install_docker : set role fact] *****
Wednesday 26 August 2020 18:12:47 +0800 (0:00:00.070) 0:00:03.055 *****
ok: [172.17.25.31]

TASK [install_docker : create a directory if it doesn't exist] *****
Wednesday 26 August 2020 18:12:47 +0800 (0:00:00.055) 0:00:03.111 *****
changed: [172.17.25.31]

TASK [install_docker : copy static docker binary] *****
Wednesday 26 August 2020 18:12:48 +0800 (0:00:00.443) 0:00:03.554 *****
changed: [172.17.25.31]

TASK [install_docker : set os specific template directory] *****
Wednesday 26 August 2020 18:12:51 +0800 (0:00:02.970) 0:00:06.524 *****
ok: [172.17.25.31]

TASK [install_docker : template manifests file] *****
Wednesday 26 August 2020 18:12:51 +0800 (0:00:00.094) 0:00:06.619 *****
changed: [172.17.25.31] => (item={'uid': 0, 'selevel': 'u's0', 'mtime': 1597201839.0, 'owner': 'u'root', 'path': 'u'install_offline_docker.sh',
'size': 1596, 'src': 'u'/tmp/selfgz1685415775/roles/install_docker/files/docker/centos-7.7.1908/install_offline_docker.sh', 'group': 'u'root',
'ctime': 1598436748.884, 'seuser': 'u'unconfined_u', 'serole': 'u'object_r', 'state': 'u'file', 'gid': 0, 'mode': 'u'0755', 'root': 'u'/tmp/selfgz
1685415775/roles/install_docker/files/docker/centos-7.7.1908', 'setype': 'u'user_tmp_t'})

TASK [install_docker : execute script install docker] *****
Wednesday 26 August 2020 18:12:51 +0800 (0:00:00.544) 0:00:07.164 *****

Session stopped
- Press <returns> to exit tab
- Press R to restart session
- Press S to save terminal output to file

Network error: Software caused connection abort
```

Possible reason:

Existing networks use 172.17.0.0 and 172.18.0.0 network segments, which conflict with docker's default docker0. If docker detects that the host belongs to the IP of the 172.17.0.0/16 network segment, it will use 172.18.0.0/16 instead, but at this time, it will not communicate with the existing 172.18 hosts.

Solutions :

1. Assign a subnet that does not conflict with the existing network as the subnet of the docker bridge, such as 10.41.0.0/16.
2. KubeManager deployment parameters add in **-e docker_bip=10.41.0.0/16** custom bip.

Scenario 5: User cluster's node disk check failed.

节点	角色	版本	代理器	内存	Pods
master1-96a6a5 192.200.244.24	Master	n/a	n/a	n/a	n/a
[Sangfor-node-en] diskCheck not pass: system disk need 100 GB					
master2-771504 192.200.244.25	Master	n/a	n/a	n/a	n/a
[Sangfor-node-en] diskCheck not pass: system disk need 100 GB					
master3-e33612 192.200.244.26	Master	n/a	n/a	n/a	n/a
[Sangfor-node-en] diskCheck not pass: system disk need 100 GB					
worker1-f523b2 192.200.244.112	Worker	n/a	n/a	n/a	n/a
Waiting to register with KubeMeters					
worker2-64d244 192.200.244.113	Worker	n/a	n/a	n/a	n/a
Waiting to register with KubeMeters					

Possible reason: The virtual machine system disk does not meet the requirement greater than 100GB. You may refer to **Section 3.2 Environment**

Requirements. If it is Sangfor aCloud platform, check whether the virtual machine system disk is greater than or equal to 100GB. If less than this value, check whether the imported vma is wrong.

Solution:

1. Delete the host or cluster, re-import the correct vma to create a virtual machine. Imported vma should be **CentOS-7-x86_64-Minimal-1908-user-cluster.vma**. After this, re-deploy the cluster.
2. Perform the expansion of the disk.



SANGFOR

