

Installer - Internal Suite Docker Range Conflicts with Private Network

Contents

[Introduction](#)

[Problem](#)

[Prerequisites](#)

[Solution](#)

Introduction

This document describes the process in order to switch on the changing the default network bridge in the installer so that it does not conflict with a private network and the install can be completed.

Problem

The default suite docker bridge uses the IP range of 172.18.0.0/16 and this can interfere with a private network. When the installer loads, things might become unroutable from installer to private network. You can see this bridge via:

```
ip address show
```

When logged into the installer VM. You can also see this bridge in docker via:

```
docker network ls
```

It should look similar to this:

```
[root@rcdn-ccs-repo ~]# docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
eadee20d76ce       bridge             bridge              local
3a009a3d7747       host              host                local
f4ca595dfac5       none              null                local
5b0dbc510082       suite             bridge              local
```

Prerequisites

1. When you deploy installer VM (ova or qcow2), you must be able to login to it, so if you use VMware, you must change the default-instance-id and the hostname when you deploy the OVA and add a public key or password.

2. If you use Openstack, just pass in a key.

Solution

Step 1. Log in to the installer.

Step 2. Disconnect the suite installer containers from the current bridge via:

```
docker network disconnect suite suite-prod-mgmt
docker network disconnect suite suite-nginx
docker network disconnect suite suite-installer-ui
docker network disconnect suite suite-k8s-mgmt
```

Step 3. If there are any other containers connected to the suite bridge, you must disconnect them too, in a similar fashion.

Step 4. After you disconnect all containers from the bridge, you must remove the bridge via:

```
docker network rm suite
```

Step 5. You now must create a new docker bridge with the same name but a different /16 network as to not conflict with the private network. In this command example, 192.168.0.0/16 is used:

```
docker network create --driver=bridge --subnet=192.168.0.0/16 --gateway=192.168.0.1 suite
```

Note: It must be a /16 network and must be named suite.

Step 6. Reboot the installer VM via:

```
shutdown -r now
```

Step 7. Log back into the VM.

Step 8. Verify that the suite installer's containers are up and running via:

```
docker ps
```

It should look like this:

```
[root@rcdn-ccs-repo ~]# docker ps
CONTAINER ID   IMAGE
COMMAND       CREATED          STATUS          PORTS
NAMES
8d437dab65b9   devhub-docker.cisco.com/multicloudsuite-release/suite-nginx:5.0.0-RC2.8
"/entrypoint.sh" 13 days ago     Up 13 days     0.0.0.0:80->80/tcp, 0.0.0.0:443-
>443/tcp      suite-nginx
6bbffff3c248   devhub-docker.cisco.com/multicloudsuite-release/suite-prod-mgmt:5.0.0-RC2.8
"/entrypoint.sh" 13 days ago     Up 13 days     8080/tcp
suite-prod-mgmt
515754611a28   devhub-docker.cisco.com/multicloudsuite-release/suite-k8s-mgmt:5.0.0-RC2.8
"/entrypoint.sh" 13 days ago     Up 13 days     8080/tcp
suite-k8s-mgmt
8cd9d0c6ddb4   devhub-docker.cisco.com/multicloudsuite-release/suite-installer-ui:5.0.0-
RC2.8  "npm start"    13 days ago     Up 13 days     8080/tcp
suite-installer-ui
```

Step 9. You can then verify that the *NEW* suite bridge works fine with:

docker network inspect suite

It should look similar to this with your *NEW* ip range as the IPv4 addresses:

```
[root@rcdn-ccs-repo ~]# docker network inspect suite
[
  {
    "Name": "suite",
    "Id": "5b0dbc51008296cccbc766cfd44da1ce95e2cc803fd208e28af60b7931737485",
    "Created": "2019-02-14T20:41:25.323694587Z",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": {},
      "Config": [
        {
          "Subnet": "192.168.0.0/16",
          "Gateway": "192.168.0.1"
        }
      ]
    },
    "Internal": false,
    "Attachable": false,
    "Containers": {
      "515754611a28d25ad43cb028ca133755fe7fe97609efe62f142c77607167844b": {
        "Name": "suite-k8s-mgmt",
        "EndpointID":
"9e9e3860e284548a8634eef0dfc208da26532a0d5f696ad60aba0edf9a05ba73",
        "MacAddress": "02:42:ac:12:00:03",
        "IPv4Address": "192.168.0.3/16",
        "IPv6Address": ""
      },
      "6bbffff3c24848028c0ea9b4294dcf47e78486acdaa0895747adbaed93aee83a": {
        "Name": "suite-prod-mgmt",
        "EndpointID":
"d3f7b14f0c5101df9a17ca857d7b6fb1bbbd47063d3538cdfcad982c01f120e0",
        "MacAddress": "02:42:ac:12:00:04",
        "IPv4Address": "192.168.0.4/16",
        "IPv6Address": ""
      },
      "8cd9d0c6ddb4424af25898131b68d22ca531c954c032d345981a25c90e9a3777": {
        "Name": "suite-installer-ui",
        "EndpointID":
"47bf559917bb044b4a06a1010685b2388a670c29979fa4201d922445f31da63e",
        "MacAddress": "02:42:ac:12:00:02",
        "IPv4Address": "192.168.0.2/16",
        "IPv6Address": ""
      },
      "8d437dab65b9ec19307f194ed50201163c486f5f2aff5133d8ecd4a7c014a656": {
        "Name": "suite-nginx",
        "EndpointID":
"1a445b1b0d7208921e29fd64605992b259c5d1071513434768ec39ca68509cd9",
        "MacAddress": "02:42:ac:12:00:05",
        "IPv4Address": "192.168.0.5/16",
        "IPv6Address": ""
      }
    },
    "Options": {},
    "Labels": {}
  }
]
```

]

Step 10. From there you should be able to go to **https://<INSTALLER_VM_IP>** and load the installer and have no private/internal network conflicts.