Running the Linux Azure Agent in the Cisco CSR 1000v

Information About the Linux Azure Agent

The primary requirement of a Linux-based virtual machine in the Azure cloud is to run the Microsoft Linux Azure Agent.

The Microsoft Azure Linux agent manages Linux provisioning and virtual machine interaction with the Azure Fabric Controller. It provides the following functionality for Linux deployments:

- Image Provisioning
- Networking
- Kernel
- Diagnostics
- System Center Virtual Machine Manager (SCVMM) Deployments
- VM Extensions

See the Microsoft Azure Linux VM Agent documentation or search for information about Microsoft Linux Azure Agent for more details.

Linux Azure Agent in the Cisco CSR 1000v

You can create a Cisco CSR 1000v in the Microsoft Azure cloud, running inside a Linux-based virtual machine. Although the Cisco CSR 1000v code runs on a Linux-based operating system, the Cisco CSR 1000v is not a generic Linux machine. Cisco IOS XE does not expose all the commands and functions of Linux to the user or to the Azure cloud.
In order for a Linux-based virtual machine to participate in various Azure management services, the virtual machine must support a minimal subset of Linux commands and functions.

Early in the initialization process for the Cisco CSR 1000v, it runs a minimalistic version of the Linux Azure Agent. This version only contains enough functionality to perform basic provisioning of the image. After completing this step, the Linux Agent loses its connectivity to the network, as ownership of all Cisco CSR 1000v interfaces are transferred to Cisco IOS XE.

This leaves the Cisco CSR 1000v in a state where it can no longer support the ongoing capabilities of the Linux Azure Agent. The Cisco CSR 1000v stops reporting its status on a periodic basis and cannot download/install any VM extensions. In order to restore this functionality, the Cisco CSR 1000v restarts the Linux Azure Agent in a guest shell container. The container provides an environment where all the Linux functions required by the agent are available.

Using the Guest Shell in the Azure Cloud

The guest shell container is the Linux host which represents the Cisco CSR 1000v virtual machine in the Azure cloud. When Azure servers and features interact with the Cisco CSR 1000v, they communicate with the Linux host in the guest shell container.

To install the Linux Azure Agent on a Cisco CSR 1000v, the guest shell container must be configured and enabled, see Information About the Guest Shell.

For Cisco IOS XE Fuji 16.8.1 and later, the guest shell container starts automatically during the initial configuration on the Cisco CSR 1000v. The Linux Azure Agent is downloaded, installed, and started.

For Cisco IOS XE Fuji 16.7 and Cisco IOS XE Everest 16.6, you must manually download and install the Linux Azure agent before starting the agent.

Manually Installing the Linux Azure Agent

Step 1  
guestshell

Enters the guest shell from privileged mode.

Example:

Router# guestshell
[guestshell@guestshell ~]$

Example:

Step 2  
sudo pip install csr_azure_guestshell

Download and install the package for Cisco CSR 1000v in the Azure cloud.

Example:

[guestshell@guestshell ~]$ sudo pip install csr_azure_guestshell
Collecting csr_azure_guestshell
/usr/lib/python2.7/site-packages/pip-8.1.2-py2.7.egg/pip/_vendor/requests/packages/urllib3/util/ssl_.py:318:

...%5c
/usr/lib/python2.7/site-packages/pip-8.1.2-py2.7.egg/pip/_vendor/requests/packages/urllib3/util/ssl_.py:122:
Downloading csr_azure_guestshell-0.0.1.dev70.tar.gz (274kB)
100% |################################| 276kB 3.1MB/s
Installing collected packages: csr-azure-guestshell
Running setup.py install for csr-azure-guestshell ... done
Successfully installed csr-azure-guestshell-1.0.0

Step 3
Verify that there is a new directory "azure" in directory /home/guestshell.

Example:
[guestshell@guestshell ~]$ ls
0_waagent.pid azure waagent.pid

Step 4
Display the two running processes.

The waagent daemon is the Linux Azure Agent.

Example:
In this example, apart from the waagent daemon, there is also a second process, which is also part of the agent and is used to download extensions to the virtual machine.

[guestshell@guestshell ~]$ ps -ef | grep waagent
root 110  1  0 13:32 ?  00:00:00 /usr/bin/python -u /usr/sbin/waagent -daemon
root 117 110  0 13:32 ?  00:00:00 python -u /usr/sbin/waagent -run-exthandlers

Step 5
Check on the status of the Linux Azure Agent (waagent.service).

Example:
[guestshell@guestshell]$ sudo systemctl status waagent.service
waagent.service - Azure Linux Agent
    Loaded: loaded (/usr/lib/systemd/system/waagent.service; disabled)
    Active: active (running) since Tue 2017-11-14 14:01:32 UTC; 15s ago
    Main PID: 161 (python)
    CGroup: /system.slice/libvirtd.service/system.slice/waagent.service
            └─161 /usr/bin/python -u /usr/sbin/waagent -daemon
            └─164 python -u /usr/sbin/waagent -run-exthandlers

Restarting the Linux Agent and Guest Shell

Step 1 is optional and shows how to reinstall the Linux agent package if the guest shell has previously been destroyed and then re-enabled.

Step 2 shows how to reinstall the Linux agent service.

Step 1
(Optional) sudo pip install csr_azure_guestshell

(Optional) Perform this step if the guest shell has been destroyed and then re-enabled. Reinstalls the Linux Azure Agent package.

Step 2
sudo systemctl start waagent.service

Restarts the Linux Azure Agent service. The service must be restarted if the guest shell has been re-enabled after having been disabled or destroyed.
Microsoft Azure Guest Shell Package Scripts

The following script is included in the csr_azure_guestshell package under the azure directory `get-metadata.py`. The script retrieves and prints instance metadata from Microsoft Azure.

```
[guestshell@guestshell azure]$ ./get-metadata.py
{
  "compute": {
    "sku": "",
    "publisher": "",
    "name": "r167-csr1",
    "offer": "",
    "vmSize": "Standard_D2_v2",
    "vmId": "5121eb3b-6503-486e-b93b-dbae5cf12fe9",
    "platformUpdateDomain": "0",
    "platformFaultDomain": "0",
    "version": "",
    "location": "eastus",
    "osType": "Linux"
  },
  "network": {
    "interface": [
      {
        "mac": "000D3A199E46",
        "ipv4": {
          "subnet": [
            {
              "prefix": "24",
              "dnsservers": [],
              "address": "192.168.35.0"
            }
          ],
          "ipaddress": [
            {
              "publicip": "13.92.177.219",
              "ipaddress": "192.168.35.12"
            }
          ],
          "ipv6": {
            "ipaddress": []
          }
        },
        {
          "mac": "000D3A1996E2",
          "ipv4": {
            "subnet": [
              {
                "prefix": "24",
                "dnsservers": [],
                "address": "192.168.35.0"
              }
            ],
            "ipaddress": [
              {
                "publicip": "",
                "ipaddress": "192.168.35.12"
              }
            ],
            "ipv6": {
              "ipaddress": []
            }
          }
        }
      }
    ]
  }
```

Running the Linux Azure Agent in the Cisco CSR 1000v
"ipaddress": []
}
}
}
}

Port 0
Mac is 00D3A199E46
Public ip is 13.92.177.219
Private ip is 192.168.35.12
subnet is 192.168.35.0/24

Port 1
Mac is 00D3A19962
Public ip is
Private ip is 192.168.36.12
subnet is 192.168.36.0/24