Connected Grid Router IOT Applications Infrastructure

Guest OS Configuration Guide

This document describes procedures to install, start and configure the Linux Guest OS on Cisco's Connected Grid Routers (CGR 1120 and CGR 1240).

Cisco Systems
Tasman Drive, San Jose
CA, United States

Table of Contents

1. Introduction

Fog computing extends the cloud computing paradigm to the edge of the network. Instead of hosting applications in a remote data center, these applications can now be hosted in the Field Area Routers, close to smart sensors and meters. Many utility applications, which need to be run in the field close to sensors and meters, can now be hosted in the Field Area Routers (FANs) rather than in a remote data center. Cisco's Connected Grid Routers (CGR 1120 and CGR 1240), which connect to sensors and meters, can also be used to run existing utility management applications with minimal effort. This document outlines the simple steps to run these applications on the CGRs.

2. Virtual Machines

The CGR now implements a hypervisor architecture in which Cisco's IOS software runs as a Virtual Machine (VM). The hypervisor can support other VMs that users can choose. Currently, a stock Linux OS is packaged with the CGR image. When this image is installed, a Linux VM is automatically created on the CGR. Support is planned for other OSs in the future.

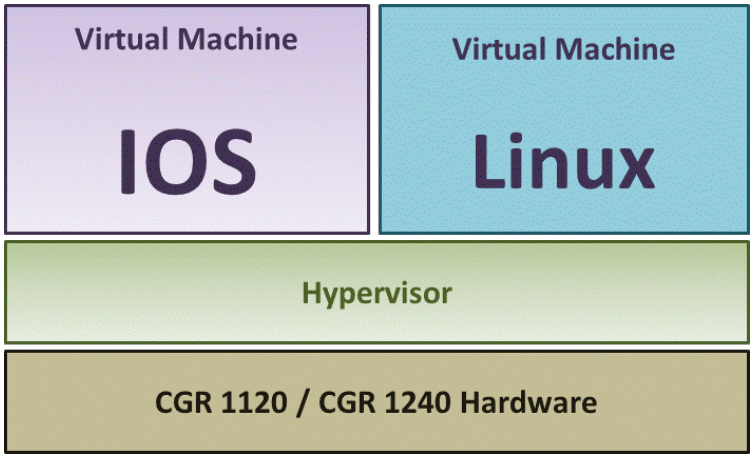

Figure-1 Connected Grid Router Software Architecture

The Linux VM on the CGR is like a regular Linux host, and can be used to install and run utility applications. The following sections give details on how to start and manage the Linux Guest OS. Details of using the Guest OS and IOS to manage applications are described separately.

3. Image Installation

1. Connect Gig 2/2 of CGR to a network from which TFTP server can be reached.
2. Uninstall Hypervisor – this will allow CGR to stop @rommon-1.

```
CGR1000# hypervisor uninstall
```

3. Reload CGR

```
CGR1000# reload
```

4. On rommon-1 prompt, set IP and Gateway

```
rommon-1> set ip 10.106.224.34 255.255.255.128
rommon-1> set gw 10.106.224.1
```

5. Boot hypervisor. After booting HV, SD card will be formatted and images will be wiped.

```
rommon-1> boot tftp:// 10.106.224.100/cgr1000-hv.srp.SSA.0.26
```

6. At rommon-2 prompt, set IP and gateway.

```
set ip 10.106.224.34 255.255.255.128
set gw 10.106.224.1
```

7. Boot IOS Image. This will upgrade the BIOS and will come back to rommon-1 again.

```
rommon-2> boot tftp://10.106.224.100/cgr1000-universalk9-mz.SSA.154-0.26.22
```

8. Repeat steps 4, 5, 6 and 7. IOS is up, HV- .26, IOS images can be copied to flash.

9. Copy Guest OS image to flash

```
CGR1000# copy scp://username@x.x.x.x//full-path/cgr1k-ref-
gos.img.1.9.gz flash:
```

10. Installing G-OS

```
CGR1000# guest-os 1 image install flash:cgr1k-ref-gos.img.1.9.gz
```

4. Configuring IOS and Starting Linux VM

This section describes how to configure IOS and start the Linux VM from IOS.

4.1. Configure DHCP Pool

To assign an IP address to the Guest OS, configure a Local DHCP pool:

```
CGR1000#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
CGR1000(config)#
CGR1000(config)# ip dhcp pool gospool
CGR1000(dhcp-config)# network 9.1.2.0 255.255.255.0
CGR1000(dhcp-config)# default-router 9.1.2.1
CGR1000(dhcp-config)# domain-name utility.com
CGR1000(dhcp-config)# dns-server address 9.1.1.1
```

```
CGR1000(dhcp-config)# lease 5
```

4.2. Configuring an IOS Interface to Connect to the VM

The VM eth0 Ethernet Interface connects to GigabitEthernet 0/1. This port must be configure before starting the Linux VM. To configure the GigabitEthernet 0/1interface with the default gateway address of the DHCP pool:

```
CGR1000# interface GigabitEthernet 0/1
CGR1000(config-if)# ip address 9.1.2.1 255.255.255.0
CGR1000(config-if)# no shutdown
```

4.3. Starting Linux VM

```
CGR1000# guest-os 1 start
Start Guest OS: ....Done!
CGR1000#
```

During bootup, the Linux VM sends a DHCP request and is assigned a password from the local pool. The Linux VM is also configured with a hostname and sync time from IOS.

4.4. Configuring CGR Access

The CGR is automatically configured with a routable IP address, if you are using Cisco Connected Grid NMS (CG-NMS) and the Provisioning Server. However, if you are using a CGR without CG-NMS, then you should configure routing and interfaces for external access. Details of this configuration are described in the Cisco Connected Grid Router 2010 Software Configuration Guide for CGR. This example assumes that you configured GigabitEthernet 2/2 with the IPv4 address 9.1.1.1.

For IPv6 configuration, please refer to **Appendix A**.

4.5. Configuring the Linux Console

You can connect to the Linux console using the IOS IP address and port 2070.

```
CGR1000#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
CGR1000(config)#line 1/4
CGR1000(config-line)#transport input all
```

4.6. Accessing Linux Console

You can access the Linux console through IOS, which provides a console server on port 70.

```
CGR1000# telnet 9.1.1.1 2070
```

```
Poky 9.0 (Yocto Project 1.4 Reference Distro) 1.4 qemux86 ttyS0

qemux86 login: root
root@qemux86:~#
```

5. Configuring the Linux VM

Before you start using the Guest OS, perform some basic configuration on the newly launched VM.

5.1. Setting a Root Password

By default, Linux does not have a root password. Set the root password before turning on SSH access:

```
[GOS] # passwd
Changing password for user root.
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
[GOS]#
```

5.2. Enabling Remote SSH Access

By default, all SSH access is disabled to prevent unauthorized access to Linux until the user properly configures the host through the console.

1. To enable root access, enter:

vi /etc/ssh/sshd_config

2. Set the “PermitRootLogin” and “PasswordAuthentication” parameters to “yes”.
3. Ensure that the “PermitEmptyPasswords” parameter is set to “no”.

```
PermitRootLogin yes
PasswordAuthentication yes
```

4. Restart SSHD.

```
[GOS]# /etc/init.d/sshd stop
Stopping sshd:                                             [  OK  ]
[GOS]# /etc/init.d/sshd start
Starting sshd:                                             [  OK  ]
[GOS]#
```

You should now be able to access the host remotely through SSH.

5. Networking Configuration on Guest OS

The Guest OS is auto-configured with an interface IP address on eth0 and a default router. For further networking configuration, please refer to **Appendix B**.

6. Troubleshooting

This section presents ways to determine common causes of configuration failure

6.1. Check the Host IP Address

Check if the host was assigned an IP address.

```
[GOS]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 02:00:03:f1:cd:05
          inet addr:9.1.2.2  Bcast:0.0.0.0  Mask:255.255.255.248
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:684 (684.0 B)  TX bytes:894 (894.0 B)
[GOS]#
```

6.2. Check the Host Route Table

```
[GOS]# netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
default         9.1.2.1         0.0.0.0         UG        0 0          0 eth0
9.1.2.0         *               255.255.255.0   U         0 0          0 eth0
[GOS]#
```

6.3. IOS Side Debugging

Verify that IOS has learned the Guest OS's ARP mapping.

```
CGR1000#show ip arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  9.1.1.1                -   0022.bdef.c562  ARPA   GigabitEthernet2/2
Internet  9.1.2.1                -   0022.bdef.c569  ARPA   GigabitEthernet0/1
Internet  9.1.2.2              112   0022.bdef.c56d  ARPA   GigabitEthernet0/1
CGR1000#
```

7. APPENDIX-A: IPv6 Configuration

This is an example configuration for using IPv6 on IOS:

```
interface GigabitEthernet 2/2
      no switchport
      ipv6 address autoconfig default
      ipv6 enable
      ipv6 dhcp client pd prefix-from-provider
      end

interface GigabitEthernet 0/1
      duplex auto
      speed auto
      ipv6 address prefix-from-provider ::2:0:0:0:1/64
      ipv6 enable
      end
```

8. APPENDIX-B: Guest OS Bridge Configuration

This example creates a bridge on Linux:

```
root@CGR1000-GOS-1:/# brctl addbr gosbr
Bridge firewalling registered
root@CGR1000-GOS-1:/#
```

This example adds interfaces to the bridge:

```
root@CGR1000-GOS-1:/# brctl addif gosbr eth0
device eth0 entered promiscuous mode
root@CGR1000-GOS-1:/#
```

This example removes the IP configuration on the interface and sets up the IP on the bridge:

```
root@CGR1000-GOS-1:/# ifconfig eth0 0.0.0.0
root@CGR1000-GOS-1:/# ifconfig gosbr 9.1.2.2 netmask 255.255.255.0 up
gosbr: port 1(eth0) entered forwarding state
gosbr: port 1(eth0) entered forwarding state
```

```
root@CGR1000-GOS-1:/#
```

This example turns on Spanning-Tree on the bridge:

```
root@CGR1000-GOS-1:/# brctl stp gosbr on
root@CGR1000-GOS-1:/#
```

This example verifies bridge configuration:

```
root@CGR1000-GOS-1:/# brctl show
bridge name     bridge id               STP enabled     interfaces
gosbr           8000.0022bdefc56d       yes             eth0
root@CGR1000-GOS-1:/#
root@CGR1000-GOS-1:/# brctl showstp gosbr
gosbr
 bridge id              8000.0022bdefc56d
 designated root        8000.0022bdefc56d
 root port                 0                    path cost                  0
 max age                  20.00                 bridge max age            20.00
 hello time                2.00                 bridge hello time          2.00
 forward delay            15.00                 bridge forward delay      15.00
 ageing time             300.00
 hello timer               0.82                 tcn timer                  0.00
 topology change timer     0.00                 gc timer                 216.90
 flags


eth0 (1)
 port id                8001                    state                forwarding
 designated root        8000.0022bdefc56d       path cost                   4
 designated bridge      8000.0022bdefc56d       message age timer          0.00
 designated port        8001                    forward delay timer        0.00
 designated cost           0                    hold timer                 0.00
 flags
```

This example verifies that the bridge is in the interface list:

```
root@CGR1000-GOS-1:/# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:22:bd:ef:c5:6d
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:19527 errors:1333 dropped:0 overruns:0 frame:1333
          TX packets:35097 errors:0 dropped:0 overruns:0 carrier:0
```

```
          collisions:0 txqueuelen:1000
          RX bytes:1674774 (1.5 MiB)  TX bytes:3208129 (3.0 MiB)

gosbr     Link encap:Ethernet  HWaddr 00:22:bd:ef:c5:6d
          inet addr:9.1.2.2  Bcast:9.1.2.255  Mask:255.255.255.0
          inet6 addr: fe80::222:bdff:feef:c56d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:902 (902.0 B)  TX bytes:468 (468.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:4348 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4348 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:313088 (305.7 KiB)  TX bytes:313088 (305.7 KiB)

root@CGR1000-GOS-1:/#
```

http://www.cisco.com/go/smartgrid