



# Cisco CGR 1000 Compute Module Installation and Configuration Guide

Last Updated: 2018-03-23

First Published: 2017-06-19

Copyright 2017-2018



# Preface

## Conventions

This document uses the following conventions.

Conventions	Indication
<b>bold font</b>	Commands and keywords and user-entered text appear in <b>bold font</b> .
<i>italic font</i>	Document titles, new or emphasized terms, and arguments for which you supply values are in <i>italic font</i> .
[ ]	Elements in square brackets are optional.
{x   y   z }	Required alternative keywords are grouped in braces and separated by vertical bars.
[ x   y   z ]	Optional alternative keywords are grouped in brackets and separated by vertical bars.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.
courier font	Terminal sessions and information the system displays appear in courier font.
< >	Nonprinting characters such as passwords are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

**Note:** Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.

**Caution:** Means *reader be careful*. In this situation, you might perform an action that could result in equipment damage or loss of data.

**Warning:** IMPORTANT SAFETY INSTRUCTIONS

Means danger. You are in a situation that could cause bodily injury. Before you work on any equipment, be aware of the hazards involved with electrical circuitry and be familiar with standard practices for preventing accidents. Use the statement number provided at the end of each warning to locate its translation in the translated safety warnings that accompanied this device.

SAVE THESE INSTRUCTIONS

**Regulatory:** Provided for additional information and to comply with regulatory and customer requirements.





# Product Overview

This section provides an overview of the CGR Compute Module.

## General Description

The CGR Compute Module is a modular, hardened server that can be inserted into CGR 1000 routers. The compute module functions as a fog computing node at the edge of the network.

The compute module runs Cisco IOx, a hypervisor architecture. Customers can run Internet of Things (IoT) applications on a Virtual Machine (VM) (see [Table 4 on page 4](#)) and connect legacy networking technologies through a network interface card. For example, applications hosted on the router can collect system information from Cisco IOS using APIs. By acquiring this information, the applications can use it to control and query other network devices using IoT protocols.

## Hardware Overview

The x86-based multi-core compute module hardware incorporates the industry standard COM Express module, referred to in this document as COMe sub-module. The sub-module fits inside the compute module enclosure, which occupies one slot of the host CGR. The CGR1120 and CGR1240 can each host one compute module. See the “Installing the Compute Module” chapter for slot restrictions on the CGR1240.

## SKU Information

[Table 1 on page 1](#) lists the SKUs available for the compute module.

**Table 1** Compute Module Product SKUs

Product ID	Bulk Memory	DDR Memory	CPU on COM-E
CGM-SRV-64	50 GB	4 GB	AMD GX-410VC. 4-Core 800Mhz
CGM-SRV-128	100 GB	4 GB	AMD GX-410VC. 4-Core 800Mhz

## Hardware Features

The compute module provides these hardware features:

- IoT hardware environmental features:
  - Operational temperature range from -40C to +50C, no fan.
  - Meets IP67 standard when inserted in a CGR1240 chassis.
  - Meets IEC 61850-3 and IEEE1613 standards.
- Compliant to Field Area Network (FAN) module specifications:
  - Takes +12V power input at 6-8 Watts (Max 10 Watts).
  - Dimensions (H x W X D): 1.50” x 3.5” x 4.46” (3.81 cm x 8.89 cm x 11.32 cm).

- Backplane/host connector is a PCIe 98-pin connector with custom pinout.
- Backplane data-path through single USB 2.0.
- Module control through backplane I2C bus.
- Acts as a carrier board for COM Express module:
  - Accepts a COM Express Type 10 Mini (84mm x 55mm) module.
  - Complies with COM Express Carrier Board specification from PICMG Open Modular Computing Standards.
  - COMe Pinout is compliant to Rev 2.1 Type 10 specification.
- Uses the following interfaces on COMe:
  - Two USB 2.0 ports for front-panel data communication.
  - One copper GE port for front-panel data communication.
  - One SATA port for connection to mSATA bulk storage.
  - One USB 2.0 port for backplane data path.
  - One Serial UART port for host monitoring.
  - SPI port for alternate BIOS on the compute module—alternate BIOS Flash with Cisco custom BIOS+Bootloader code.
- COM Express (COMe) module features:
  - Advantech SOM-9751 module.
  - CPU = AMD GX-410VC. 4-Core at 800Mhz.
  - Main DRAM Memory = 4Gbyte DDR3L-1066.
  - Power consumption < 6.1 Watts (Typical), < 8.5W (Max)
  - BIOS flash depopulates as alternate BIOS flash on carrier board is used.
  - One GE Mac/Phy, Intel i210.
- mSATA bulk storage slot:
  - Compliant to mSATA specifications.
  - Interface compliant to SATA 3.0 (6.0Gbps, Data rate 4800Mbit/sec).
  - Allows future size upgrade feature. Currently 50GB and 110GB are possible.

**Note:** Shielded cables are required for USB ports. The GE port does not require shielded cables.

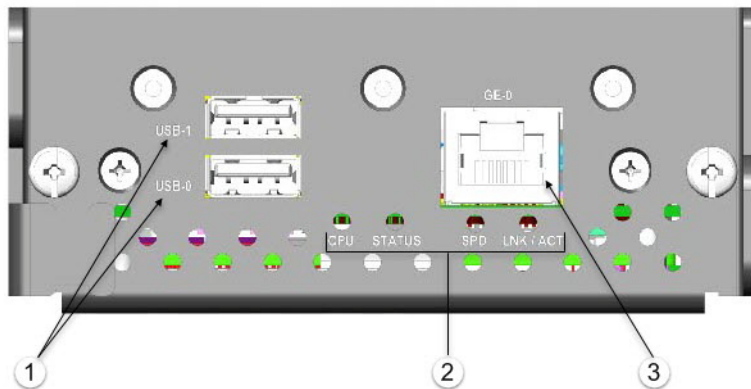
**Note** For slot requirements for the module, please refer to the Pre-Installation notes in the [1 0: Module Overview and Installation, page 1](#) chapter in this guide.

## Front Panel

The compute module has two USB 2.0 ports and one copper Gigabit Ethernet Interface. These ports go directly to the COMe sub-module as a data-path into the fog computing facility of the compute module. Packets or data needing network routing services are forwarded towards the backplane/Host through a USB 2.0 port (on COMe) via a USB-to-FE device that acts like any another network port for the FOG Software running on the COMe sub-module.

You can monitor details on the USB ports within the Host-OS and VM applications.

**Figure 1 Compute Module Front Panel**



1	USB 2.0 ports (USB-0 and USB-1)
2	LEDs
3	Gigabit Ethernet Copper RJ45 port

### USB Ports

USB ports provide storage and support file transfer. Hot swap of USB module is supported.

**Note:** The compute module **does not** support USB to Ethernet traffic; and, cannot be used as a power source.

Each USB port is limited to supporting Low-Power and Self-power devices with capability to supply 100mA. Warning messages will be displayed if more than 200mA aggregate is being drawn by the two ports.

### USB Cable Considerations

USB cables should be limited to use in protected areas, such as a control room. We recommend limiting USB cables to Classes 0, 1, and 2. USB cables must be at least 2 meters and are limited to be no longer than 4.3 meters (15 feet) by specification. (IEC61850-3:2013, 61000-4-5:2006).

If you connect a cable greater than 2 meters to a USB port, a ferrite clamp will need to be installed. You can order ferrite clamps from Cisco (PID: FERRITE-CLAMP=). Refer to the following Cisco document for additional details on the ferrite clamp:

[Attaching Ferrite Clamp on CGR 1000 Series Alarm Cables for Noise Suppression](#)

### LEDs

Table 2 on page 4 describes the LEDs on the front panel of the compute module.

**Table 2 LEDs**

LED	Activity	Description
CPU (of module)	Off	No power.
	Yellow	Disabled.
	Steady green	Active.
Status (Operational status of module)	Off	No power.
	Yellow	Power available. Module not yet initialized.
	Steady green	Module is operating.
Gigabit Ethernet Port Link Speed	Off	100/10.
	Steady green	1000.
Gigabit Ethernet Link and Activity	Off	No link.
	Steady green	Link.
	Blinking green	Activity.

## Memory

The compute module provides a single mSATA module for bulk storage. The mSATA module allows for configurable storage options, shown in [Table 3 on page 4](#).

**Table 3 Memory Options (Hardware)**

Memory	Size
mSATA 64 GB with 14 GB reserved for wear-leveling	50 GB
mSATA 128 GB with 28 GB reserved for wear-leveling	100 GB

## Software Overview

Using either the command line interface (CLI) or NMS, the user can install and run a virtual machine (VM), which is assigned CPU, memory and IO resources of the compute module (as per virtual PC model).

[Table 4 on page 4](#) lists the supported Virtual Machines (VM) and the corresponding Operating Systems (OS) supported on those VMs.

**Table 4 Supported Operating Systems and Virtual Machines**

Operating System	Virtual Machine
Ubuntu versions 14.04 and 15.10	Linux
Windows 7	Windows

Two emulated Ethernet interfaces are available at the VM level: the first one is connected via backplane to router's GigabitEthernet <slot>/1 interface and provides the VM a means to access networks connected through the router. The second Ethernet interface is connected to Ethernet interface on the front panel of the module. VM can also access USB interfaces on the module's front panel.

A VM serial console can be accessed from router CLI to allow VM debugging.

A Host-OS serial console can be accessed from router CLI to allow Host-OS debugging.





# Installing the Compute Module

**Note:** In this document, the Compute Module is identified as a server-module in Cisco IOS commands.

## Table of Contents

- [1 0: Module Overview and Installation, page 1](#)
- [2.0 IOx Software Workflow on the Compute Module, page 5](#)
- [3.0 Configuring the Compute Module Using CGR IOS, page 8](#)
- [4.0 Monitoring and Verifying Server-Module Interfaces, page 13](#)
- [5.0 Application Hosting on Server Module, page 14](#)

## 1 0: Module Overview and Installation

This section provides the following information:

- [1.1 Module PIDs and Install Preparation, page 1](#)
- [1.2 Installing Hypervisor and Cisco IOS Images \(Bundle Image\), page 3](#)
- [1.3 Inserting the Compute Module, page 3](#)
- [1.4 Installing the Server-Module Image, page 5](#)

### 1.1 Module PIDs and Install Preparation

#### Before You Begin

Verify the following before downloading any software or installing or configuring the CGR 1000, compute module or software:

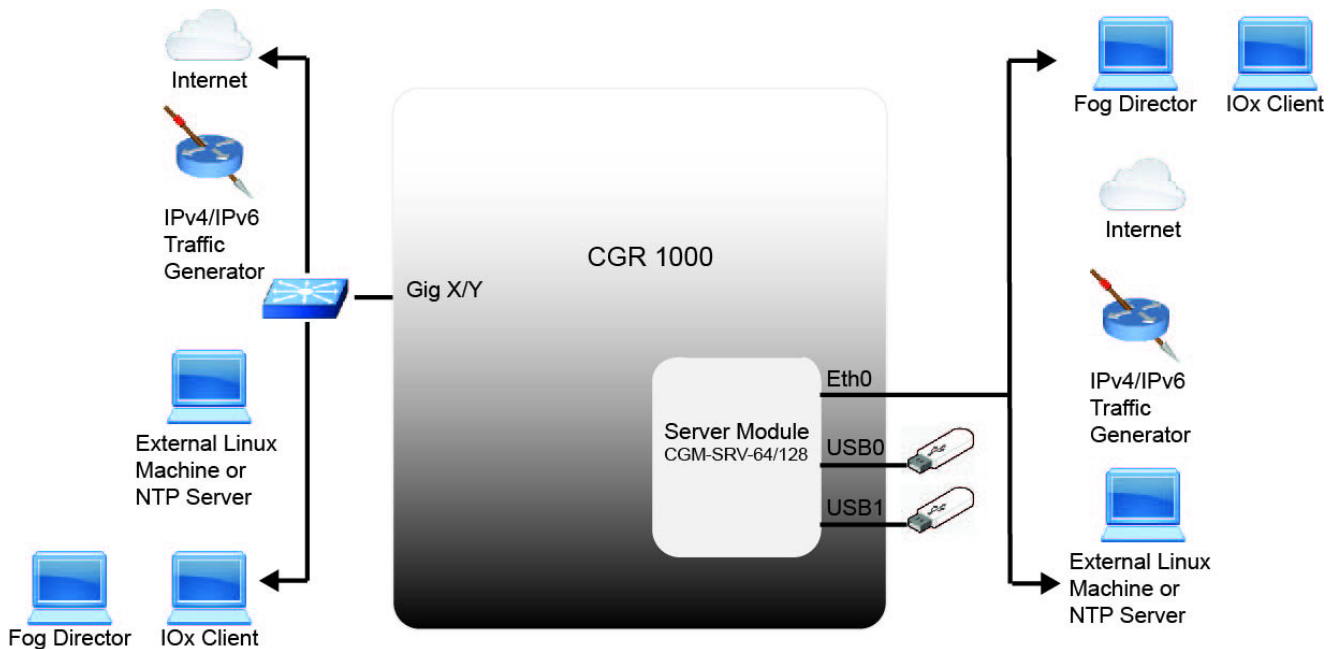
- Your network setup matches that shown in [Figure 1 Network Setup, page 2](#).
- You have an Ubuntu14.04 system for installation (Recommended). The Ubuntu system **must be** on the same network as the devices to which you will push applications.
- Ensure that you download the latest images from Cisco.com.
- Ensure that you have one of the compute modules referenced in [Table 1 Module Part Number and Specifications, page 2](#)
- You have reviewed all of the [Pre-Installation Notes, page 2](#)

**Table 1 Module Part Number and Specifications**

Product PID	Bulk Memory	DDR Memory	CPU on COM-E
CGM-SRV-64	50 GByte (approximate)	4 GByte	AMD GX-410VC, 4 core 800 MHz
CGM-SRV-128	100 GByte (approximate)	4 GByte	AMD GX-410VC, 4 core 800 MHz

## Pre-Installation Notes

- Before you install the compute module, please note the following information:
  - You can only install one compute module in a CGR 1120 and CGR 1240.
  - For a CGR 1240, you must install the compute module in Slot 5. If a cellular module is installed in Slot 6 the maximum system operating temperature is derated to 50°C
  - For a CGR 1120, you can install the compute module in any slot.
  - Install only one Application hosting image per compute module. Recommended virtual machines: Windows 7 and Linux VM  
We recommend Windows VM.
- When installing a 3G/4G module and/or WPAN module in the CGR 1240, reserve the following slots:
  - Populate slot 3 with either the 3G or 4G module
  - Populate slot 4 with the WPAN module.

**Figure 1 Network Setup**

## 1.2 Installing Hypervisor and Cisco IOS Images (Bundle Image)

On the CGR 1000 Series, upgrade the Hypervisor and Cisco IOS images using the **bundle install** command:

**Note:** We exclude the Guest OS (GOS) image during the **bundle install** because it is already resident on the CGR 1000 and no update is required.

```
cgr1120# bundle install flash:<bundle_image> exclude GOS<--Wait approximately 10 minutes for completion.
:
cgr1120# wr mem
cgr1120# reload
```

**Note:** The server-module Image for the compute module is **not** included in the CGR bundle image. You must install the server-image **after you install the Module**. Proceed to [1.3 Inserting the Compute Module, page 3](#).

## 1.3 Inserting the Compute Module

Before you physically install the compute module into an available CGR slot, review both of the possible options and associated commands required:

**Option 1:** If the target CGR slot is empty or previously hosted a compute module, do the following:

1. Power off the slot:

```
configuration terminal
hw-module poweroff 3 <---In this example, the module is installed in slot 3.
```

2. Insert the compute module.

3. Power on the slot:

```
configuration terminal
no hw-module poweroff 3
```

**Option 2:** If inserting the compute module into a CGR slot that previously hosted a different interface module:

1. Insert the compute module.
2. Save the configuration, and reboot the CGR.

```
configuration terminal
no hw-module poweroff slot-number
end
write
reload
```

## Verify Module Status

Enter the following **show** commands to verify the module status:

```
show module
Mod  Ports  Module-Type                Model                Status
---  ---  -
3    1      CGR1000 Server Module 64GB Disk  CGM-SRV-64          ok

show inventory
```

## 1 0: Module Overview and Installation

```

NAME: "CGR1000 Third Party Module",
DESCR: "CGR1000 Third Party Module"
PID: CGM-SRV-64
VID: 0.2,
SN: FOC20201YWB

```

**show diag | begin CGM**

```

Product Number   : CGM-SRV-64
Serial Number    : FOC20201YWB
Part Number      : 73-100967-02
Part Revision    : 08
Mfg Deviation    :
H/W Version     : 0.2
Mfg Bits        : 0
Engineer Use    : 0
snmpOID        : 9.12.3.1.9.91.28
Power Consump   : 49
CLEI Code       :
VID            : V00

```

## Linecard Module specific block:

```

Block Signature  : 0x6003
Block Version    : 2
Block Length     : 103
Block Checksum   : 0xC7A
Feature Bits     : 0x0
HW Changes Bits  : 0x0
Card Index       : 30027
MAC Addresses    : CC-16-7E-EF-8E-0C
Number of MACs   : 4
Number of EOBC links : 0
Number of EPLD   : 0
Port Type-Num    : 2-1
SRAM size        : 0
Sensor #1        : 107,92
Sensor #2        : -128,-128
Sensor #3        : -128,-128
Sensor #4        : -128,-128
Sensor #5        : -128,-128
Sensor #6        : -128,-128
Sensor #7        : -128,-128
Sensor #8        : -128,-128
Max Connector Power: 0
Cooling Requirement: 0
Ambient Temperature: 55

```

## Top Assembly Number(TAN) block:

```

Block Signature  : 0x600A
Block Version    : 1
Block Length     : 46
Block Checksum   : 0x5C3
TAN serial number: FOC20201YWB
TAN 68-level part number: 68-5973-02
TAN 68-level rev number: 09

```

## SPROM contents (hex):

```

0x00: AB AB 03 A0 12 C5 2C 00 00 04 60 03 00 00 43 69
0x10: 73 63 6F 20 53 79 73 74 65 6D 73 2C 20 49 6E 63
0x20: 2E 00 43 47 4D 2D 53 52 56 2D 36 34 00 00 00 00
0x30: 00 00 00 00 00 00 46 4F 43 32 30 32 30 31 59 57
0x40: 42 00 00 00 00 00 00 00 00 00 37 33 2D 31 30 30
0x50: 39 36 37 2D 30 32 00 00 00 00 30 38 00 00 00 00
0x60: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x70: 00 00 00 00 00 02 00 00 00 00 00 09 00 0C 00 03
0x80: 00 01 00 09 00 5B 00 1C 00 00 00 31 00 00 00 00
0x90: 00 00 00 00 00 00 00 00 00 00 00 00 56 30 30 00
0xA0: 60 03 02 67 0C 7A 00 00 00 00 00 00 00 00 00 00

```

## 2.0 IOx Software Workflow on the Compute Module

```

0xB0: 00 00 00 00 00 00 75 4B CC 16 7E EF 8E 0C 00 04
0xC0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0xD0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0xE0: 02 01 00 00 00 00 00 00 00 00 00 00 00 00 00
0xF0: 00 00 6B 5C 80 80 80 80 80 80 80 80 80 80 80
0x100: 80 80 00 00 00 00 37 60 0A 01 2E 05 C3 46 4F 43
0x110: 32 30 32 30 31 59 57 42 00 00 00 00 00 00 00
0x120: 00 36 38 2D 35 39 37 33 2D 30 32 00 00 00 00
0x130: 00 30 39 00 00 60 0C 01 2E 06 11 46 4F 43 32 30
0x140: 32 30 31 59 58 37 00 00 00 00 00 00 00 00 37
0x150: 33 2D 31 30 30 39 36 38 2D 30 32 00 00 00 30
0x160: 33 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x170: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x180: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x190: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x1A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x1B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x1C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x1D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x1E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x1F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```

## 1.4 Installing the Server-Module Image

After physically installing the compute module, install the server-module image on the module:

```
cgr1120#server-module <slot-number> install flash: <image>
```

Operation requires module reload, do you want to continue? [yes]:**yes** *!--Completes in about 5 minutes.*

## 2.0 IOx Software Workflow on the Compute Module

The Compute module has the following interfaces that **must** be configured:

- **Internal Interfaces** (see [Figure 2 Internal Interfaces - Config 01, page 6](#) and [Figure 3 Internal Interfaces - Config 02 with NAT, page 7](#))
  - **Server-module Console:** Virtual console line to Host-OS of the module that provides login using the Cisco IOS command-line interface without requiring any IOS networking.
  - **Internal Management Interface, gig<slot>/1:** You can network via virtual IOS interfaces > VDS > ethernet\_management interface on Host-OS. Interface provides basic reachability as well as moderate throughput to Host-OS. Used for copying application to IOx.
  - **Internal Data Interface, gig<slot>/2:** We can network via physical IOS interfaces > VDS > CAF Virtual Switch > NAT/Bridged Interfaces to Application. In the examples that follow in section 3.0, we only conform to bridged interfaces. Purpose: Basic reachability as well as moderate throughput to Application and VM.
- **Server-Module External Interfaces** (see [Figure 4 Server Module, page 8](#))
  - **USB0 and USB1:** USB2.0 ports, 5V, for storage. Hot swap of USBs supported on Windows VM only. The USB ports **cannot be used** as a power source.
  - **Front Panel RJ45 Gigabit Ethernet Interface:** Provides direct access to external network

Figure 2 Internal Interfaces - Config 01

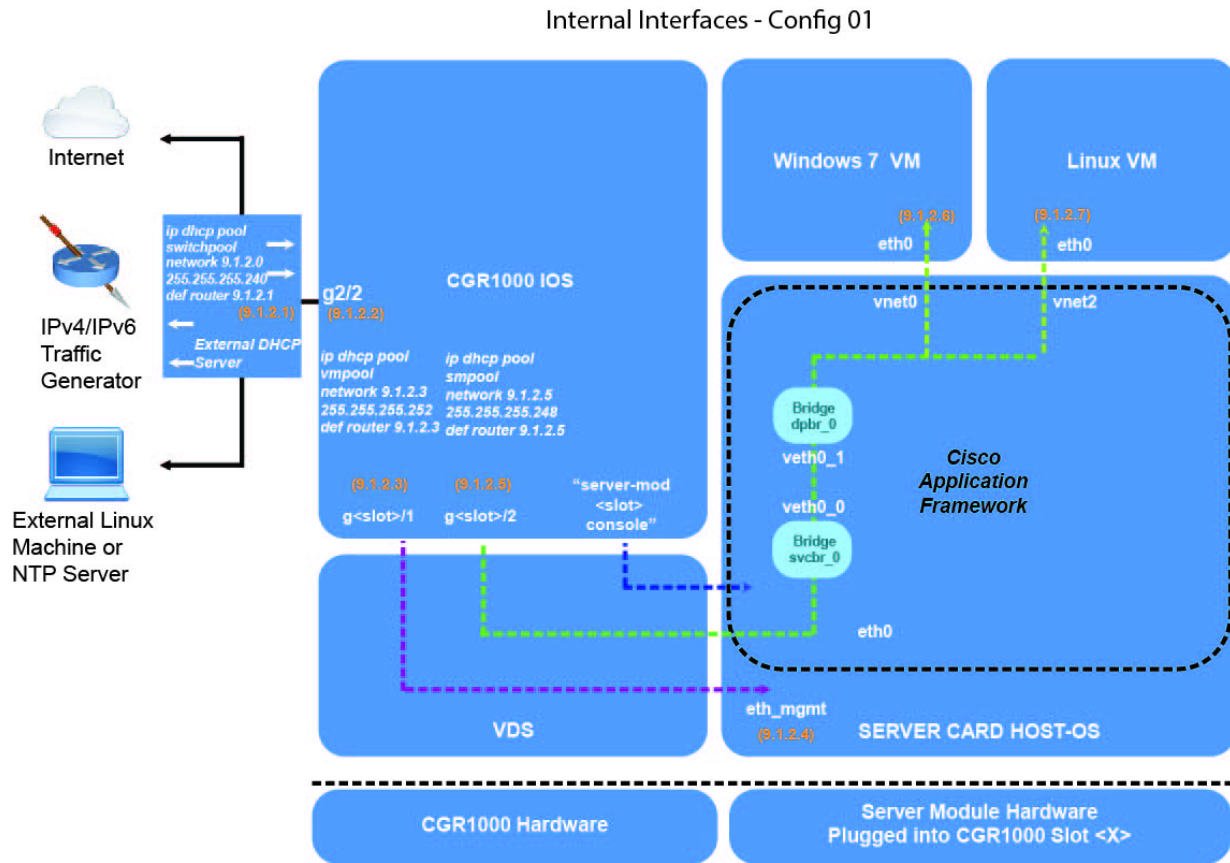


Figure 3 Internal Interfaces - Config 02 with NAT

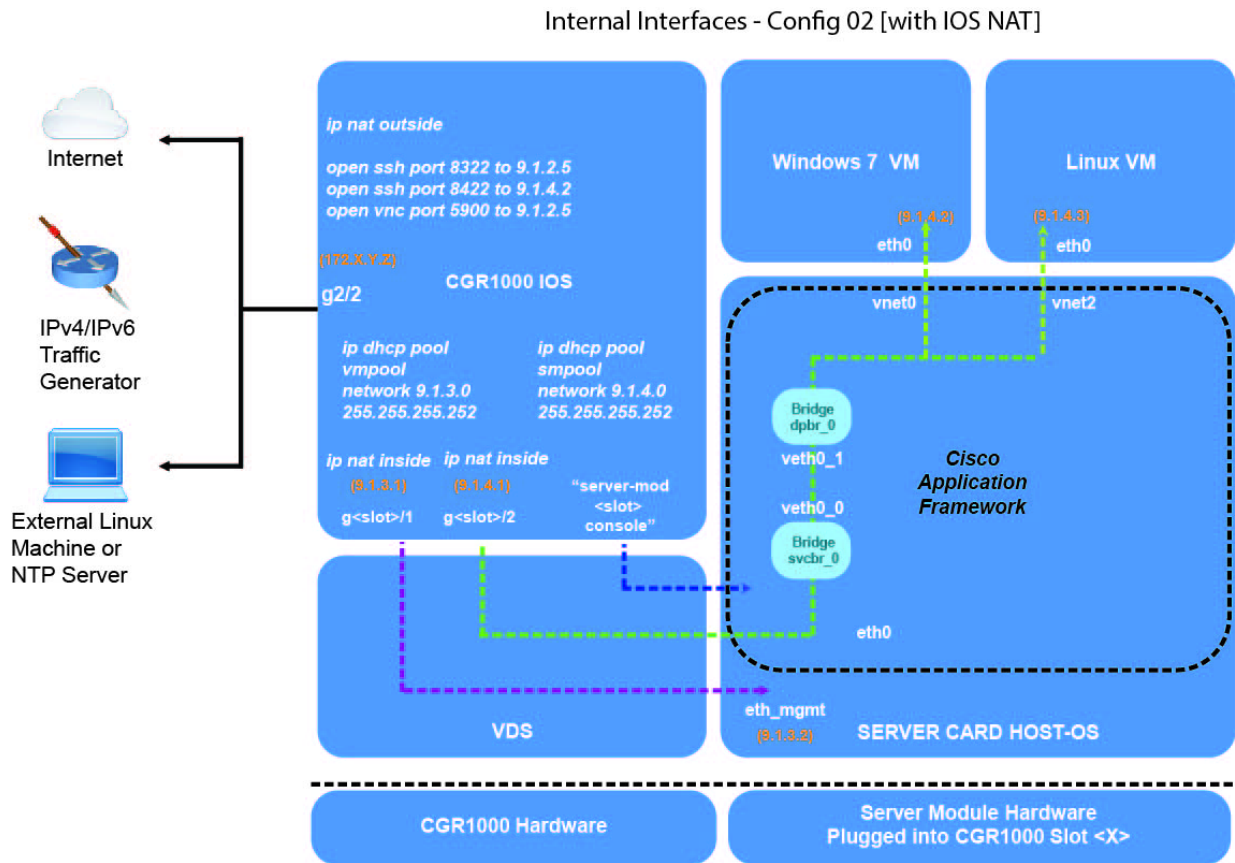
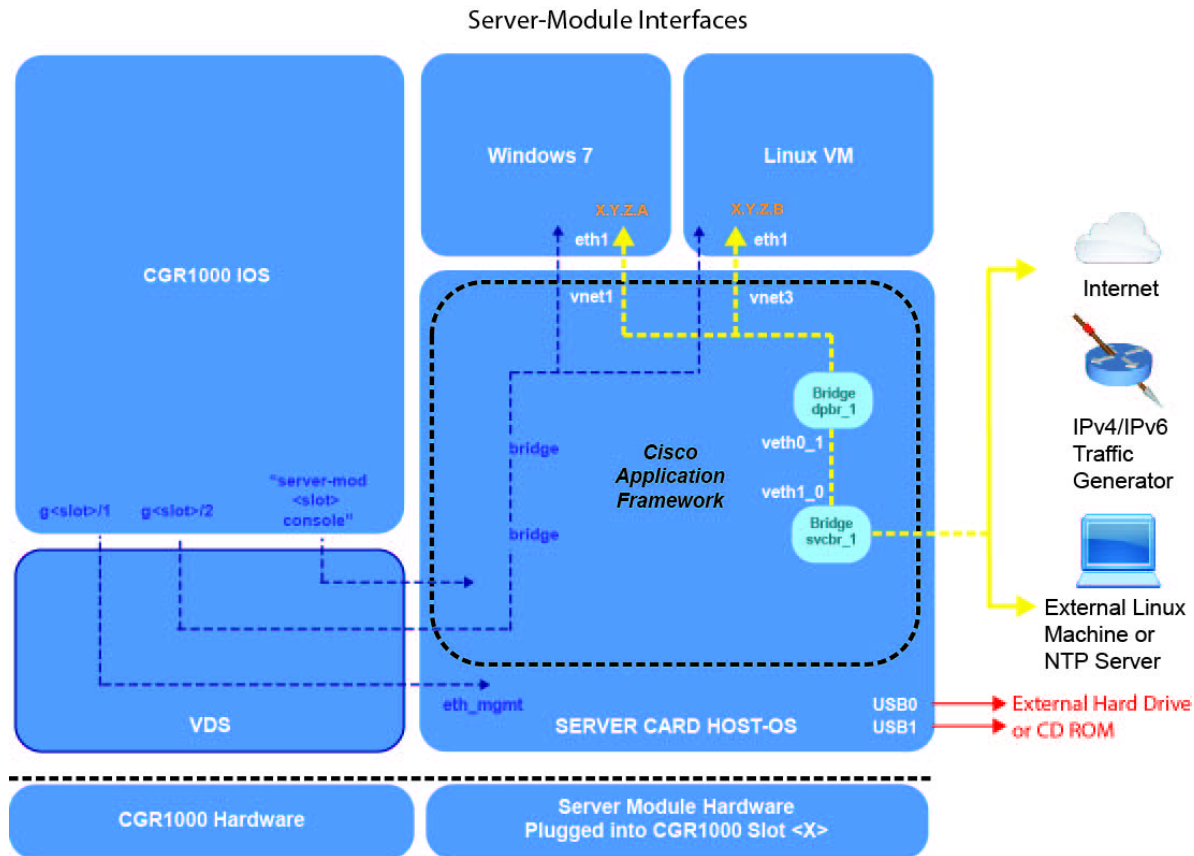


Figure 4 Server Module



## 3.0 Configuring the Compute Module Using CGR IOS

After you physically install and power up the module in the CGR and install the Host-OS image, you can now configure the compute module from the CGR.

**PLEASE READ THE FOLLOWING NOTES BEFORE CONFIGURING YOUR COMPUTE MODULE.**

**Note:** When sending traffic over the internal gig {slot}/2 interface, from the application to IOS or the external machine, we are rate-limited to 4Mbps. For higher traffic, the external interface on the server module is the recommended option.

In addition, the MTU Setting **must be** 1492 on **both** sender and receiver interfaces. The MTU setting on the Server module interface can scale only to 1492 and does not support path discovery. If the external machine supports MTU path discovery, then you can configure the CGR1000 gig {slot}/2 interface MTU to 1492, since IOS supports the feature. Using the above settings ensures fragmentation for any packet size will work.

**Note:** Throughout this section and others, the IP addresses we use are for illustration purposes only. Please update the IP addresses as needed to match your configuration. In many instances we have marked network-specific commands with: *Update to match your configuration.*

**Note:** You can employ Network Address Translation (NAT) in your configuration as seen in [Figure 3 Internal Interfaces - Config 02 with NAT, page 7](#). However, we **do not** provide specific NAT configuration details in this document.

For details on NAT configuration, refer to the following documentation:  
<http://www.cisco.com/c/en/us/tech/ip/network-address-translation-nat/index.html>

- [3.1 Define User Credentials, page 9](#)



---

### 3.0 Configuring the Compute Module Using CGR IOS

- [3.2 Enable Licensing, page 9](#)
- [3.3 Configure SSHv2 to Enable Hypervisor/IOS Bundle installation, page 9](#)
- [3.4 Create a DHCP Pool for IOx and CAF on the Module Host-OS, page 10](#)
- [3.5 DHCP Pool for Application Hosted on Module, page 10](#)
- [3.6 IOS Physical Interface Configuration, page 10](#)
- [3.7 Server-Module Internal Management Interface, page 10](#)
- [3.8 Server-Module Internal Data Interface, page 10](#)
- [3.9 Routing, page 10](#)
- [3.10 Enable HTTP and SCP, page 10](#)
- [3.11 CPU Allocation, page 11](#)
- [3.12 Line Setting, page 11](#)
- [3.13 Console Login, page 11](#)
- [3.14 Version Check of Host-OS Version on Module, page 11](#)
- [3.15 Server-Module Reload, page 12](#)
- [3.16 BIOS Validation, page 12](#)
- [3.17 Temperature Monitoring, page 12](#)
- [3.18 LED Monitoring, page 12](#)
- [3.19 Internal Management Interface Monitoring, page 12](#)

## 3.1 Define User Credentials

**Note:** You must have privilege 15 to deploy applications [CAF/IOx limitation]. Otherwise, you will encounter an IOx authentication failure.

```
configuration terminal
hostname CGR1120 <---Change to User Credentials
enable password cisco <---Change to User Credentials
username cisco privilege 15 password 0 cisco <---Change to User Credentials
```

## 3.2 Enable Licensing

Ensure licensing configuration in place:

```
license boot module cgr1000 technology-package securityk9
license boot module cgr1000 technology-package datak9
```

## 3.3 Configure SSHv2 to Enable Hypervisor/IOS Bundle installation

```
aaa new-model
ip ssh rsa keypair-name sshkeys
crypto key generate rsa label sshkeys key-pair modulus 2048
ip ssh version 2
```

### 3.4 Create a DHCP Pool for IOx and CAF on the Module Host-OS

```
ip dhcp pool smpool
network 9.1.2.3 255.255.255.252 <---Change to User IP and Subnet
default-router 9.1.2.3 <---Change according to user configuration
dns-server 192.168.1.1 192.168.2.1 <---Change according to user configuration
lease infinite
```

### 3.5 DHCP Pool for Application Hosted on Module

```
ip dhcp pool vmpool
network 9.1.2.5 255.255.255.252 <---Change to User IP and Subnet
default-router 9.1.2.5 <---Change according to user configuration
dns-server 192.168.1.1 192.168.2.1 <---Change according to user configuration
lease infinite
```

### 3.6 IOS Physical Interface Configuration

Interface ensures network access to the Host-OS and Application on the compute module via the IOS internal interface.

```
interface GigabitEthernet2/2
no switchport
ip address 9.1.2.2 255.255.255.0 <---Change to User Static IP and Subnet
ip virtual-reassembly in
duplex auto
speed auto
no shut
```

### 3.7 Server-Module Internal Management Interface

```
interface GigabitEthernet3/1 <---Change slot number as applicable
ip address 9.1.2.3 255.255.255.0 <---Change according to user configuration
ip virtual-reassembly in
no shut
duplex auto
speed auto
ipv6 enable
```

### 3.8 Server-Module Internal Data Interface

```
interface GigabitEthernet3/2 <---Change slot number as applicable
ip address 9.1.2.5 255.255.255.0 <---Change according to user configuration
ip virtual-reassembly in
no shut
duplex auto
speed auto
ipv6 enable
```

### 3.9 Routing

```
ip route 0.0.0.0.0.0.0.0 GigabitEthernet2/2 9.1.2.2 <---Change according to user configuration.
```

**Note:** In this case, the gateway to CGR 1000 is the switch (see [Figure 2 on page 6](#))

### 3.10 Enable HTTP and SCP

This is an example template to connect with the tools that manage application hosting (ioxclient, local manager, fog director). Enable SCP server in case required.

## 3.0 Configuring the Compute Module Using CGR IOS

```
ip http server
ip http secure-server
ip scp server enable
```

## 3.11 CPU Allocation

To optimize VDS/Hypervisor performance, enter the following configuration:

```
iox hypervisor sched-policy 100
```

## 3.12 Line Setting

Example template to allow all protocol traffic (User-configurable)

```
line con 0 <---Update template to match your needs
line 1/1 1/6
  transport input all
  transport output all
line vty 0 15
  transport input all
  transport output all
```

## 3.13 Console Login

You can log into the console of the compute module from CGR by entering:

```
cgr1120# server-module 3 console
```

Escape sequence: Ctrl-Shift x, then disconnect command

```
MontaVista Carrier Grade Express Linux 2.0.0 CGM-SRV-64-3 /dev/console
```

```
CGM-SRV-64-3 login: root
Last login: Tue Jun 21 00:42:35 UTC 2016 from 10.41.58.8 on pts/0
Stopping syslogd/klogd: stopped syslogd (pid 1283)
stopped klogd (pid 1286)
done
Starting syslogd/klogd: done
root@CGM-SRV-64-3:~#
root@CGM-SRV-64-3:~# uname -a
Linux CGM-SRV-64-3 4.1.17-rt17-yocto-standard #1 SMP PREEMPT Thu Jun 16 10:16:27 PDT 2016 x86_64
GNU/Linux
root@CGM-SRV-64-3:~#
root@CGM-SRV-64-3:~#
```

Escape sequence: Ctrl-Shift 6 + x

**Note:** To log out of the module session, hit simultaneously 'Ctrl+Shift+6' followed immediately by 'x'. Immediately enter 'disconnect' in IOS mode to prevent reconnection attempts to Host-OS

```
CGR1120# disconnect
Closing connection to 127.1.3.1 [confirm]confirm
```

## 3.14 Version Check of Host-OS Version on Module

To check Host-OS version installed on the compute module from CGR:

```
cgr1120# show iox host list detail | include Host OS Version
```

## 3.0 Configuring the Compute Module Using CGR IOS

Host OS Version 0.3

## 3.15 Server-Module Reload

To reload the module you can power on/power off the compute module from the CGR using one of the two command options below. You must include slot number in which the module resides: (In the examples below, 3 is the slot number).

```
cgr1120# hw-module reload 3
```

-or-

```
configuration terminal
hw-module poweroff 3
no hw-module poweroff 3
```

## 3.16 BIOS Validation

To verify that a BIOS validation has successfully occurred after a module or CGR reload (requires up to 20 minutes to complete):

```
show log | i BIOS
```

```
Jun 20 17:36:42.547: %CGR1K_SM-6-CGR1K_PLUGIN_MODULE_SM_BIOS_VALID_INFO: Server module BIOS validation
succeeded for module 3
```

## 3.17 Temperature Monitoring

**Note:** Server-module draws approximately 6 to 8W, which can cause it to heat up more than other modules.

The **show env temp** command displays the ambient temperature (CurTemp) of the module, not the CPU, which tends to be a few degrees higher.

The module supports a maximum temperature of to 50 C.

When higher temperatures exist, reduce frequent reloads of the module.

To check module temperature:

```
show env temp
```

Temperature:

```
-----
Module   Sensor      MajorThresh  MinorThres  CurTemp     Status
          (Celsius)   (Celsius)   (Celsius)
-----
1        Sensor0     075,-25     50,-15     43          OK
3        Sensor0     075,-25     50,-15     52          OK
```

## 3.18 LED Monitoring

To check LED status of the module from CGR:

```
show platform led | include Server Module
Server Module 3: green <--- indicates module is ON
```

**Note:** OFF status indicates the module is powered down.

## 3.19 Internal Management Interface Monitoring

In the configuration below, IOS acts as DHCP server and Host-OS acts as DHCP client, to verify address assignment to server-module Host-OS.

## 4.0 Monitoring and Verifying Server-Module Interfaces

```

cgr1120#show run int g3/1
Building configuration...

Current configuration : 149 bytes
!
interface GigabitEthernet3/1
 ip address 9.1.2.3 255.255.255.0
 ip virtual-reassembly in
 duplex auto
 speed auto
 ipv6 enable
end

cgr1120#show run | section dhcp
ip dhcp pool smpool
 network 9.1.2.3 255.255.255.252
 default-router 9.1.2.3
 dns-server 171.70.168.183 173.36.131.10
 lease infinite

cgr1120#show ip dhcp binding
Bindings from all pools not associated with VRF:
IP address      Client-ID/      Lease expiration      Type
                Hardware address/
                User name
9.1.2.3         0100.800f.1170.00    Infinite              Automatic

```

**Verify on Host-OS:**

```

CGR1120#server-module 3 console
Escape sequence: ctrl-shift-^ x, then disconnect command
root@CGM-SRV-64-3:~# ifconfig
eth-mgmt Link encap:Ethernet HWaddr 00:80:0F:11:70:00
inet addr:9.1.2.4 Bcast:9.1.2.5 Mask:255.255.255.252
inet6 addr: fe80::280:fff:fe11:7000/64 Scope:Link
UP BROADCAST RUNNING PROMISC MULTICAST MTU:1492 Metric:1
RX packets:7313 errors:0 dropped:0 overruns:0 frame:0
TX packets:11412 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:616458 (602.0 KiB) TX bytes:3550994 (3.3 MiB)

```

## 4.0 Monitoring and Verifying Server-Module Interfaces

The Compute module has two USB ports and one RJ45 Gigabit Ethernet Interface on its front panel. Details on access and information accessible through those ports is noted in the sections below.

For details on LEDs and the front panel, refer to following chapter and page: [Product Overview, page 1](#).

- [4.1 USB0 and USB1 Ports, page 13](#)
- [4.2 External RJ45 Gigabit Ethernet Interface and ETH LEDs, page 14](#)

### 4.1 USB0 and USB1 Ports

The two USB ports on the module support storage and file transfer. Use of the module's external Gigabit Ethernet interface is recommended for faster transfer rates.

USB ports cannot be used as a power source.

## 5.0 Application Hosting on Server Module

You can hot-swap USBs on the module.

You can view details on the USB ports through the Host-OS and VM (Windows or Linux) on the module.

**Note:** The Host-OS for the USB ports display will vary based on the VM in use.

- To check USB0 and USB1 port status from Host-OS:

```
root@CGM-SRV-64-3:~# lsusb -t
/: Bus 04.Port 1: Dev 1, Class=root_hub, Driver=ehci-pci/2p, 480M
  |__ Port 1: Dev 2, If 0, Class=Hub, Driver=hub/4p, 480M
/: Bus 03.Port 1: Dev 1, Class=root_hub, Driver=ehci-pci/2p, 480M
  |__ Port 1: Dev 2, If 0, Class=Hub, Driver=hub/4p, 480M
     |__ Port 3: Dev 3, If 0, Class=Mass Storage, Driver=usb-storage, 480M
/: Bus 02.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/2p, 5000M
/: Bus 01.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/2p, 480M
  |__ Port 1: Dev 2, If 0, Class=Vendor Specific Class, Driver=smsc9500, 480M
```

```
root@CGM-SRV-64-3:~# fdisk -l
:
  Device Boot      Start         End      Blocks   Id  System
/dev/sdb1   *            3         33033     61453632   c   Win95 FAT32 (LBA)
```

- To manually copy files from USB to Host-OS use the following example configuration:

```
mkdir /mnt/card
mount/dev/sdb1 /mnt/card
cp /mnt/card/vm.img /mnt/data/vm <---Default IOx path to store the Application
```

**Note:** The Windows 7 and Linux images are labeled as *vm.img* throughout this guide.

## 4.2 External RJ45 Gigabit Ethernet Interface and ETH LEDs

The Gigabit Ethernet Interface is IOS-independent.

The Gigabit Ethernet interface has two LEDs (Speed, Link Status) to monitor auto-negotiation. You can monitor the interface by physical inspection of the front panel. See [Product Overview, page 1](#).

**Note:** The Gigabit Ethernet interface cannot be used to SCP files (such as *vm.img*) to Host-OS/IOx since it is only accessible to the application only--not the Host-OS.

END INSTALLATION: You have now completed installation of the module and configured the basic steps to provide access to the Host-OS on the Compute module.

## 5.0 Application Hosting on Server Module

To install the Application Installation, follow these steps:

- **Packaging an application** [tool used: IOxclient] - Ensures application is compatible with our platform resources.
- **Deploying an application** [tool used: IOxclient or Local Manager] - Copy application to the server-module and deploy it based on user configuration.
- **Monitoring an application** [tool used: IOxclient, Local Manager] - Log in to a running application and monitor the same.
- **Removing an application** [tool used: IOxclient, Local Manager] - Stop, deactivate or uninstall an application.

## Using IOxclient to Bring Up Application

### 5.1 Application Readiness

#### BEFORE YOU BEGIN

VM application should be in .img format. Image size to conform with PID disk space limitations.

**Note:** This release does not support Container-based and PaaS-styled applications.

- For a 64 GB Compute module, ensure at least 50 GB of usable space is available.
- For a 128 GB Compute module, ensure at least 106 GB of usable space is available.

### 5.2 Proxy Settings

- Ensure Proxy settings are configured appropriately (example Ubuntu VM template below, refine to suit your environment).
- Example configuration only. User needs to change according to user environment.

```
username@username: ~/Desktop$ export http_proxy=http://proxy-wsa.esl.xyz.com:80/
username@username: ~/Desktop$ export ftp_proxy=http://proxy-wsa.esl.xyz.com:80/
username@username: ~/Desktop$ export https_proxy=http://proxy-wsa.esl.xyz.com:80/
```

### 5.3 Prerequisites

Download IOxclient onto a standalone Ubuntu PC/VM (2-core PC with 8GB RAM). (For example, we employed an Ubuntu 14.04LTS on a standalone 2-core PC with 8GB RAM in our testing). Ensure PC meets conditions noted in [5.1 Application Readiness, page 15](#).

IOxclient is a command-line interface tool, which can communicate with the Host-OS IOx on the Compute module (server-module).

We support two user-interfaces:

- Local Manager, which manages an individual device
- Fog Director, which manages multiple devices (Cisco-developed interface)

### 5.4 Setup IOx Client

Execute IOx client. For the first time, register with an example device to create the default profile.

```
~$ ./ioxclient
Config file not found : /home/username/.ioxclientcfg.yaml
Creating one time configuration..
Your / your organization's name : cisco <--User configurable
Your / your organization's URL : www.cisco.com ---User configurable
Your IOx platform's IP address [127.0.0.1] : 9.1.2.4 <--Eth_Mgmt of Host-OS IP Address
Your IOx platform's port number [8443] : 8443 <--Host-OS port. Use 9443,if 8443 in use for other tasks
Authorized user name [root] : cisco <--IOS username
Password for username: <Enter_user_password> <--IOS password
Local repository path on IOx platform [/software/downloads]: <--Can leave as default
URL Scheme (http/https) [https]: https
API Prefix [/iox/api/v2/hosting/]: <--Can leave as default. Not used in current implementation.
Your IOx platform's SSH Port [2222]: 22 ><--Can leave as default unless we have rules blocking specific
ports in the configuration.
```

Activating Profile default

## 5.5 Create IOxclient Profile

You must create a profile for each device. You can create any number of profiles for a device or many devices. However, you can **only** have one active profile.

Commands listed below are interactive.

```
username@username:~/Desktop/sdkdir13$ ./ioxclient profiles create
Active Profile: default <--No user entry required. It displays previously active profile.
Enter a name for this profile: cgr1120 <--Any name of user preference can be entered.
Your IOx platform IP address[127.0.0.1] 9.1.2.4 <---Eth_management of Host-OS IP address
Your IOx platform port number[8443] 8443 <-- Other values supported other than default
Authorized user name[root]cisco<-- Enter IOS username (user privilege 15)
Password for cisco: <-- Enter IOS password
Local repository path on IOx platform[software/downloads] <-- Leave as default
URL Scheme (http/https) [https]: <-- Can leave default value. HTTPS chosen in this example.
API Prefix[/iox/api/v2/hosting/]: <-- Can leave default value.
Your IOx platform's SSH Port[222]:22 <-- Can leave default value unless we have rules blocking specific
ports in our configuration
```

```
Activating Profile CGR1120
```

## 5.6 Create a New Project Folder

**Note: The following example shows steps required to package and deploy a Windows VM. The same steps must be followed for an Ubuntu Linux-based VM.**

From the ioxclient path, create a new project folder.

```
username@username~/Desktop/sdkdir13$ mkdir winapp
```

## 5.7 Add Package Descriptor File

In the winapp folder, manually add [vi edit] the package yaml file as shown below (copy and paste with exact syntax).

```
app:
  cpuarch: x86_64
  resources:
    cpu: "6000"
    devices:
      -
        label: HOST_DEV331
        type: usbport
        usage: "This is for hokkaido usb331"
    disk: "10"
    graphics:
      vnc: true
    memory: "3072"
    network:
      -
        interface-name: eth0
      -
        interface-name: eth1
  profile: custom
startup:
  disks:
    -
      file: "file://vm.img"
      target-dev: hda
  ostype: windows
```



```

    gemu-guest-agent: false
  type: vm
  descriptor-schema-version: "2.1"
  info:
    author-link: "http://www.cisco.com/"
    author-name: "Cisco Systems"
    description: "Sample Windows 7 VM for Hokkaido use case"
    name: win7
    version: "0.1"

```

The example above is a Windows Application with 2 USB ports (configuration is valid for any number of USB ports) and 2 Ethernet interfaces (1 internal data interface, 1 front panel external data interface).

If you need to modify the yaml file for the Ubuntu Application, comment out OS type (`#ostype: windows`) and the rest of the script remains as is.

Brief Purpose of descriptor file: This is the file usually provided by an application developer to the application user/deployer . It comprises of cpu, memory, app type, path of image file and such user-application-specific details, based on user's platform constraints.

CPU Units: Especially useful when we run multiple applications. Since engineering supports only one application for the current release, we can be generous in using the allowed ~7300 units. Example uses 6000 units. Disk: Random allocation of large numbers can result in corruption of image. For windows7 VM app, sample configuration recommendation of 10MB works fine. Memory: max. 3500MB.

If the image is located in the same 'winapp' folder, "file path" field need not be specified. If image however is too large to be hosted on the machine where IOxclient is installed (Specially true of 25GB+ Windows images), add 'file path'. If default path used for image is /mnt/data/vm, no additional path info need to be specified except image name/label

**Note:** Syntax spacing of package.yaml file has to be exact. Use <http://www.yamllint.com> to validate the same.

Given that we support only one application, we recommend that you retain the image name as *vm.img* and mirror the working example provided.

## 5.8 Image SCP to Host-OS

```

username@username:~/Desktop/sdkdir13$ ioxclient platform scp ./vm.img
Currently active profile : CGR1120
Command Name: plt-scp
2016/05/25 13:59:37 POST /iox/api/v2/hosting/platform/scpservice HTTP/1.1
Host: 9.1.2.4:8443
X-Token-Id: 2d09a545-c9da-4b55-8eb3-fdc077db0425
Downloaded scp keys to pscp.pem
Running command : [scp -P 22 -r -i pscp.pem ./
vm.img.tar scpuser@9.1.2.4:/]
The authenticity of host '[9.1.2.4]:2222 ([9.1.2.4]:22)' can't be established.
ECDSA key fingerprint is e8:a0:78:3e:ba:0c:3b:92:aa:e7:9f:61:65:a8:56:2c.
Are you sure you want to continue connecting (yes/no)? yes

```

Image can be copied using USB as well, if there is physical access to the device or a USB is already plugged in. Else, alternatively, IOxclient can be used to SCP image.

If the steps fail as provided in the example, check the ssh port setup while creating device profile (Step[3]). If not done, no need to recreate profile, alternatively, manually modify following file in the path where ioxclient tool is installed:

```
vi ~/.ioxclientcfg.yaml > goto our respective profile > edit ssh_port manually
```

Retry the ioxclient command line

SCP rate via Internal Interface ~3Mbps.

**Note:** We do not chmod 777 \* the created .pem file. Else the SCP will fail. .pem file permissions should be "chmod 0600 \*.pem"

## 5.9 Package Application

```
username@username:~/Desktop/sdkdir13/winapp$ ioxclient pkg .
Currently active profile : CGR1120
Command Name: package
Checking if package descriptor file is present..
Created Staging directory at : /tmp/533536799
Copying contents to staging directory
Checking for application runtime type
Couldn't detect application runtime type
Creating an inner envelope for application artifacts
Excluding package.tar
Generated /tmp/533536799/artifacts.tar.gz
Calculating SHA1 checksum for package contents..
Root Directory : /tmp/533536799
Output file: /tmp/081563890
Path: artifacts.tar.gz
SHA1 : 96e379fb98bd1b401c6ee52230c11d38251572cf
Path: package.yaml
SHA1 : 1695a21094f0331e711b6374114c14f5c4362327
Generated package manifest at package.mf
Generating IOx Package..
Created IOx package at : /home/example/Desktop/sdkdir13/winapp/package.tar
```

Packaging an application can be done using the IOxclient tool or SDK. For this Documentation Guide, we reference IOxclient as it can be used end-to-end from packaging to application deployment.

Once an application is packaged, any image name change, path change, descriptor file content change requires that the application be repackaged.

Once packaged, a package.tar will comprise of a manifest file, the yam descriptor file and the artifacts.gz which has all the actual application images/files.

## 5.10 Install Application

```
username@username~/Desktop/sdkdir13/winapp$ ioxclient app in win ./package.tar <--Any name can be user
in place of 'win'. Any name can be used for app-id
Currently active profile : CGR1120
Command Name: application-install
Installation Successful. %s is available at : App https://9.1.2.4:8443/iox/api/v2/hosting/apps/vm
Successfully deployed
```

In the steps above, we are creating an **appid** (example label for an instance of application) to be installed on the device. By using different application IDs, we can reuse the application and install it again on the same device with a different appid). For our module, since we support only one application for this release, this feature may not be relevant.

## 5.11 Payload Activation JSON File

```
username@username:~/Desktop/sdkdir13/winapp$ ioxclient application activate --payload payload_act win
Currently active profile : CGR1120
Command Name: application-activate
Payload file : payload_act. Will pass it as application/json in request body..
App win is Activated
Sample payload_act file:username@username:~/Desktop/sdkdir13/winapp$ vi payload_act
{
  "resources": {
    "profile": "custom",
    "graphics": {
      "vnc-password": "password"
    },
    "network": [{
      "interface-name": "eth0",
```

```

"network-name": "iox-bridge0"
}, {
"interface-name": "eth1",
"network-name": "iox-bridge1"
}],
"devices": [{
"device-id": "usb3/3-1",
"port-num": "undefined",
"bus-num": "undefined",
"type": "usbport",
"label": "HOST_DEV331"
}],
"cpu": "6000",
"memory": "3072"

```

End User needs to manually add a payload activation file in son format. File needs to be in 'winapp' location.

Purpose of Payload Activation file: For end users to invoke the interfaces they need for their application (2 Ethernet interfaces and 1 USB interface, in this example). It is also used to specify a mandatory VNC password (valid for Windows and Linux Applications, it is especially needed for Windows App login as ssh is unsupported).

Due to security constraints, VNC password setting is mandatory.

## 5.12 Start Application

```

username@username:~/Desktop/sdkdir13/winapp$ ioxclient app start win
Currently active profile : CGR1120
Command Name: application-start
App win is Started

```

Start Application. To verify if running, login to Host-OS, and check top -b for running qemu-kvm command with specified resources and Appid name.

## 5.13 Monitor Application

We can use the following ioxclient option to verify a running application parameters.

To VNC Login to Windows application, open up port ip\_addr:5900, and provide the specified vnc password in payload\_act file (eg: 'password', in this case)

Recommended VNCClient: TightVNC latest version

For ssh login to an Ubuntu application we can alternatively use : ssh -p <port> -i <app\_id.pem> appconsole@ip\_addr. However, because of the limitation is that we can open only one console port, for multiple ssh sessions, we need to login via console, configure ssh server on our VM and use external/internal data interface for access.

For Ubuntu VM, the recommendation is to install ssh server on the Ubuntu application on first time bring-up so as to be enable multiple ssh sessions.

```

username@username:~/Desktop/sdkdir13/winapp$ ioxclient app info win
Currently active profile : CGR1120
Command Name: application-info
Details of App : win
-----
{
"appCustomOptions": "",
"appType": "vm",
"author": "Cisco Systems",
"authorLink": "http://www.cisco.com/",
"dependsOn": {},

```

## Using IOxclient to Bring Up Application

```

"description": "Sample Windows 7 VM for Hokkaido use case",
"ostype": "Windows",

"env": {
  "CAF_APP_APPDATA_DIR": "/data/appdata",
  "CAF_APP_CONFIG_DIR": "/data",
  "CAF_APP_CONFIG_FILE": "/data/",
  "CAF_APP_LOG_DIR": "/data/logs",
  "CAF_APP_PERSISTENT_DIR": "/data",
  "CAF_APP_USERNAME": "root",
  "CAF_SYSTEM_UUID": "de277f88-9874-4175-a2b5-2aecd55d5121"
},
"id": "win",
"is_autoinstalled": false,
"is_service": false,
"name": "win7",
"networkInfo": {
  "eth0": {
    "libvirt_network": "dpbr_0",
    "mac_address": "52:54:dd:de:95:27",
    "network_name": "iox-bridge0",
    "port_mappings": {}
  },
  "eth1": {
    "libvirt_network": "dpbr_1",
    "mac_address": "52:54:dd:4c:dc:85",
    "network_name": "iox-bridge1",
    "port_mappings": {}
  }
},
"resources": {
  "cpu": 6000,
  "devices": [
    {
      "device-id": "usb3/3-1",
      "label": "HOST_DEV331",
      "type": "usbport",
      "usage": "This is for hokkaido usb331"
    }
  ],
  "disk": 10,
  "graphics": {
    "vnc": {
      "autoport": "yes",
      "listen": "0.0.0.0",
      "password": "password",
      "port": "5900"
    },

    "vnc-password": "password"
  },
  "memory": 3072,
  "network": [
    {
      "interface-name": "eth0",
      "network-name": "iox-bridge0",
      "port_map": null
    },
    {
      "interface-name": "eth1",
      "network-name": "iox-bridge0",
      "port_map": null
    }
  ],
  "profile": "custom",

```

```

    "vcpu": 1
  },
  "state": "RUNNING",
  "toolkitServicesUsed": null,
  "version": "0.1"
}

```

## 5.14 Stop, Deactivate or Uninstall an Application

Steps to execute:

- Stop an app (qemu-kvm process stops running).
- Deactivate an app (all the resources assigned to the app are released)
- Uninstall an application (remove all history of the AppID from CAF/IOx)

```

username@username:~/Desktop/sdkdir13/winapp$ ioxclient application stop win
Currently active profile : CGR1120
Command Name: application-stop
App win is Stopped

```

```

username@username:~/Desktop/sdkdir13/winapp$ ioxclient deactivate win
Currently active profile : CGR1120
Command Name: application-deactivate
App win is Deactivated

```

```

username@username:~/Desktop/sdkdir13/winapp$ ioxclient application stop win
Currently active profile : CGR1120
Command Name: application-uninstall
Successfully uninstalled app win

```

## 5.15 Troubleshooting

For troubleshooting, the following commands can be very useful. When providing logs to engineering for troubleshooting, please provide the information below.

```

username@username~/Desktop/sdkdir13/winapp$ioxclient debug info
username@username~/Desktop/sdkdir13/winapp$ioxclient platform techsupport create
username@username:~/Desktop/sdkdir13/winapp$ioxclient platform techsupport download
username@username:~/Desktop/sdkdir13/winapp$ioxclient platform techsupport delete

```

**Note:** The following VNC viewer (TightVNC version 2.8.5) configuration must be employed:

- ColourLevel: Replace *pal8* with **rgb332** or **rgb565**
- FullColour: **True**

For additional details:

<https://support.realvnc.com/knowledgebase/article/View/422/12/problems-connecting-to-vmware-qemu-xens-built-in-or-other-third-party-vnc-compatible-server-software>

