

Cisco IOx & Covacsis Design Guide



This guide explains how to configure, manage, and troubleshoot Cisco IOx & Covacsis Validate Solution guide. This chapter contains the following sections:

- [Overview of Cisco IOx & Covacsis](#)
- [Supported Platforms](#)
- [Cisco IOx Local Manager](#)
- [Configuring Cisco IOx](#)
- [Local Manager Web Interface](#)
- [Configuring Guest Serial](#)
- [Troubleshooting](#)

Overview of Cisco IOx & Covacsis Solution

Cisco IOx is an end-to-end application enablement platform that provides application hosting capabilities for different application types in a consistent and uniform manner across various Cisco network platforms. The IOx platform allows you to manage the whole life cycle of applications including development, distribution, deployment, hosting, monitoring, and management.

Cisco IOx comes with the following features:

- A common software infrastructure to host applications in network devices such that they are independent of heterogeneous hosting hardware.
- User friendly interface that makes it easy for developers and administrators to build and deploy IOx applications.
- Provides a set of services that accelerate the development of IOx applications.
- Provides complete life cycle management capabilities for applications hosted on network devices.
- The Covacsis Intelligent Plant Framework™ (IPF) provides a unique platform that let you have a complete 360° visibility of the manufacturing floor in real time. It is a one of a kind product which provides vivid diagnostics input to production department to improve upon. It helps to redefine the sustainable economics of the shop floor, by collaborating across different departments of the organization in real time. With Covacsis IPF you will always be connected with your manufacturing floor, from any part of the world.

Covacsis Solution provides with the following Benefits:

Real Time Floor Economics:

Covacsis' IPF provides indepth understand of the financial impact of the manufacturing operations undertaken at the most granular level.

Improve Control and Communication:

Real-time visibility of the manufacturing floor through Covacsis' IPF ensures improved control over production.

Move from 'Reactive' to 'Predictive' Decision Making:

Big Data Analytics identifies non-linear relationships and renders actionable insights and reports.

Monitor your plant from 'anywhere':

Ubiquitous visibility and last mile connectivity gives an in-depth visibility of the shop floor in real time

Ensure Sustainable Improvement:

Instill a culture of improvement using data science within the organization

Covacsis powered by Cisco Network provides the benefits of

- Network processes Sensor data, aggregated and transported over Cisco's Intelligent WAN
- Drive Business Growth with real time Analytics
- Reduced WAN Bandwidth Utilization
- Data protection against loss of connectivity to Data Centre and Cloud services
- Monitor your business from- "anywhere"

Supported Platforms

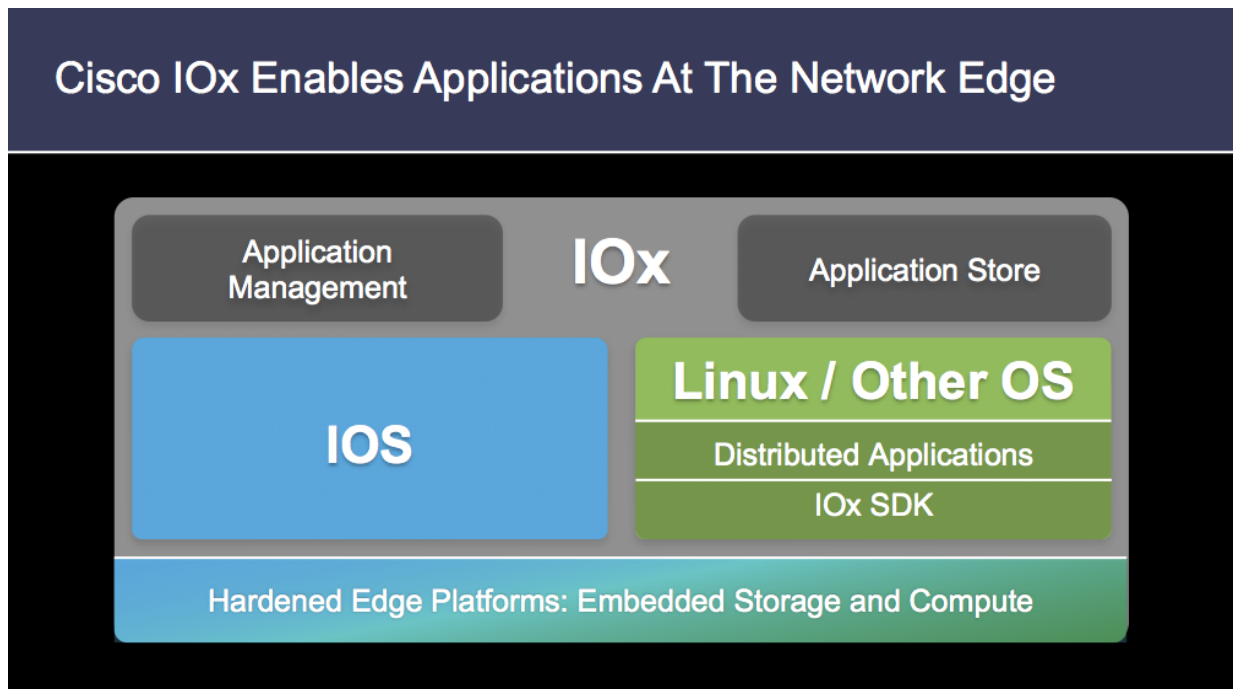
The table below outlines the minimum system requirements for Covacsis IPF implementation at a typical manufacturing plant having up to 10 machines with 50 parameters each. For larger number of machines, an appropriate sizing exercise needs to be conducted.

Components	Description
Cisco 819 Hardened Router. PID: C819G-4G-G-K9	<ul style="list-style-type: none">• In case of a serial port connection 1 unit per machine controller• In case of an Ethernet connection 1 unit per 2 machine controllers
IPF Blog Server	<ul style="list-style-type: none">• 2 X Intel Xenon X5650 @ 2.67GHz (6 Core, 2.67Ghz, 6.4GT/s QPI, 12MB Cache, 95W)• 32GB RAM• 2 X 1000GB 7.2K RPM SAS HDD/ Raid-1
Data Consolidation Server	<ul style="list-style-type: none">• 2 X Intel Xeon X5650 @ 2.67GHz (6 Core, 2.67Ghz, 6.4GT/s QPI, 12MB Cache, 95W)• 16GB RAM,• 2 X 500GB 7.2K RPM SAS HDD/ Raid-1
Panel PC	<ul style="list-style-type: none">• A standard Panel PC with Linux support is used with LIU application for operator entries. Typically 1 unit per machine.
Adapters	<ul style="list-style-type: none">• Various standard adapters may be required for establishing physical connection with the machine controllers.
Serial Cables PID: CAB-449MT=	<ul style="list-style-type: none">• Installed IOx applications to access the router's serial interface.

Software Version Details & Supported Protocols

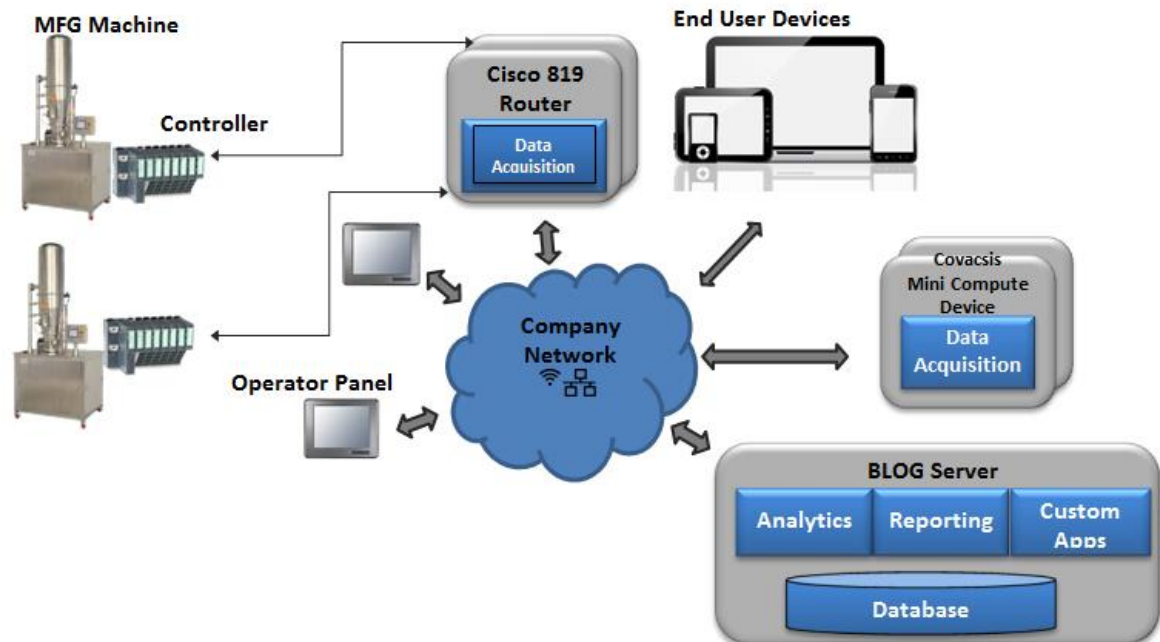
C819HG-4G-G-K9	<ul style="list-style-type: none"> • c800-universalk9_iox-mz.SPA. 15.5(1)T or later
Covacsis Software Details	<ul style="list-style-type: none"> • IPF – 2.1 or later
Protocols	<ul style="list-style-type: none"> • Modbus TCP • Modbus RTU • MPI/Ethernet • DF1 • MELSEC Serial • HostLINK Serial

CISCO IOS & IOx Architecture



Cisco IOS & Cisco IOx brings on 819 Router. Because Linux-based applications can run on Cisco devices in the Cisco IOx framework, businesses can bring their own applications and interfaces to the network. Cisco IOx offers developers a way to create IoT applications such as data aggregation, control systems, and access control and have them run on edge network devices.

Cisco IOx & Covacsis Solution Topology



The way CISCO 819 Router communicates with Machinery Controller (PLC) typically varies three different models.

Scenarios 1: (Where both Machinery Controller (PLC) IP network and Company LAN are in same network).

Scenarios 2: (Where both Machinery Controller (PLC) IP network and Company LAN are in Different network).

Scenarios 3: (In case of serial communication, Here Machinery Controller (PLC) IP network doesn't have any IP).

In all the three Scenarios bring up Cisco IOx Local manager is common. Following step by step will be the reference further, how to communicate from IOS to IOx.

Cisco IOx Local Manager

Cisco IOx Local Manager allows you to manage the complete life cycle of IOx applications hosted on your device. You can deploy, run, delete, stop, and troubleshoot IOx applications. You can use either a Web interface or CLI to manage IOx applications.

Configuring Cisco IOx

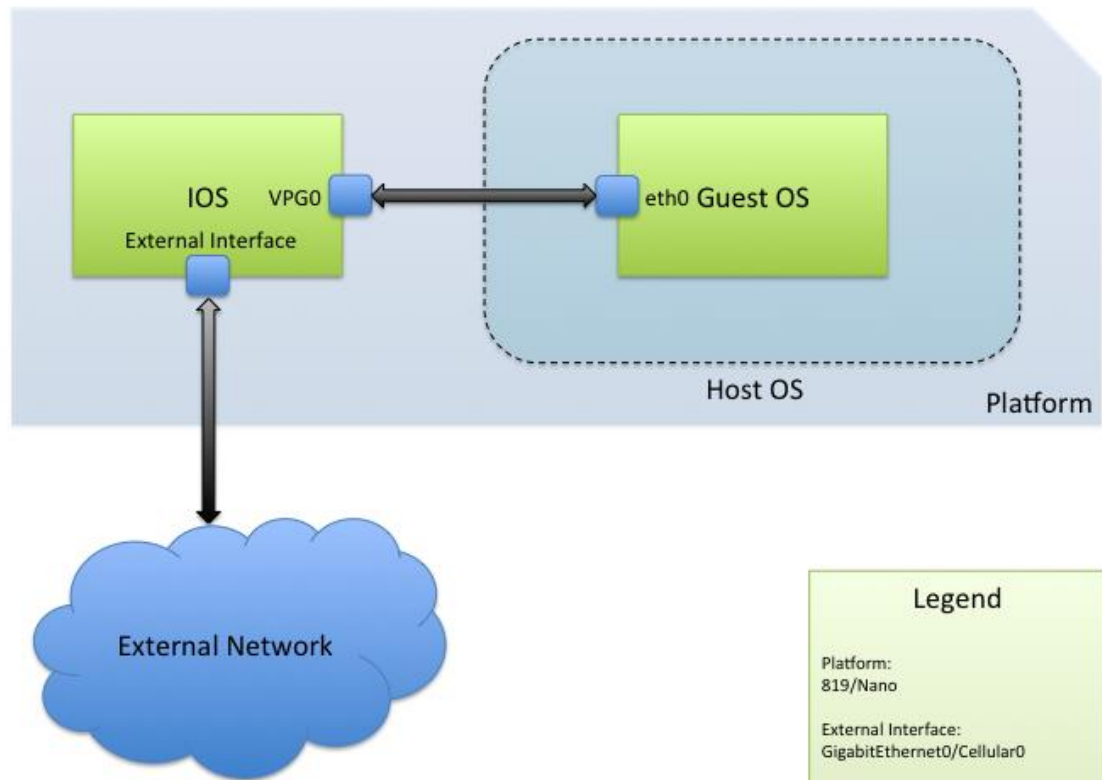
Before you deploy applications on your device, you have to configure IOx. On Cisco 800 series routers, IOS image runs on Core 1 and IOx runs on Core 2. Configuring IOx involves enabling IOx framework on Core 2 of your device.

Note: The prerequisite for configuring IOx on a device is that you should have an IOS image that supports IOx. The IOS image should be 15.5(1)T or later.

Setup 819 interfaces

In the example configuration used in the installation steps, it is assumed that 819 device Gigabit Ethernet 0 port is connected to network 172.x.x.x/24 and the IOx Host, Guest Operating will be behind internal NAT network

192.168.x.x. Eth0 of Guest OS will be connected to IOS VirtualPortGroup 0 interface



Configure the router ip address by executing the following example commands:

Step 1.

Enter the configure terminal command at the privileged EXEC prompt to enter global configuration mode:

Router# configure terminal Enter configuration commands, one per line. End with CNTL/Z.

Router(config)#

Step 2.

Enter the interface command. Specify the interface type, along with the slot number/port number to identify the interface to configure. The interface that you configure here is a physical interface like Gigabit Ethernet. You are configuring this interface for outside connectivity of Core 1 and Core 2.

Router(config)#interface GigabitEthernet0

Router(config-if)#

Step 3.

Assign an ip address and subnet mask to the interface. Enter no shutdown to enable the interface. Enter ip nat outside to specify that the interface is connected to the outside network. Enter Exit to exit the interface mode.

```
Router(config-if)#ip address 172.x.x.x 255.255.255.0 Router(config-if)#no shutdown
Router(config-if)#ip nat outside
Router(config-if)#exit
```

Step 4.

Enter the ip route command to establish static routes between the interfaces. Enter the ip default-gateway command to specify the default gateway.

```
Router(config)#ip route 0.0.0.0 0.0.0.0 172.x.x.x Router(config)#ip default-gateway 172.x.x.x
```

Step 5.

Enter interface command to specify the internal interface that helps the router's Core 1 and Core 2 to communicate. The interface name should be ethernet1 for Cisco 819 Series Routers, and ethernet0/1 for Cisco 800 M Series Routers. You should not use any other name for this interface. Assign an ip address and subnet mask to this interface. Enter ip nat inside to specify that the interface is connected to the inside network (the network subject to NAT translation). Enter Exit to exit the interface mode.

```
Router(config)#interface ethernet1
Router(config-if)#ip address 192.168.3.1 255.255.255.0 Router(config-if)#ip nat inside
Router(config-if)#exit
```

Step 6.

Enter interface command to specify the virtual interface of the single IOx application running on the second Core. This interface routes the application traffic. The interface name should be virtualportgroup 0. You should not use any other name for this interface. Assign an ip address and subnet mask to the virtual interface. Enter ip nat inside to specify that the interface is connected to the inside network (the network subject to NAT translation). Enter Exit to exit the interface mode.

```
Router(config)#interface virtualportgroup 0
Router(config-if)#ip address 192.168.1.1 255.255.255.0 Router(config-if)#ip nat inside
Router(config-if)#exit
```

Step 7.

Enter iox command to enter the iox configuration mode. Router(config)#iox
Router(config-iox)#

Step 8.

Configure the ip address of the host and default gateway. The IP address of the default gateway and the IP address of ethernet1 you configured above should be the same. Enter Exit to exit the IOx mode.

```
Router(config-iox)host ip address 192.168.3.2 255.255.255.0 Router(config-iox)host ip default-gateway 192.168.3.1 Router(config-iox)exit
```

Step 9.

Next you have to configure the NAT rules for application traffic. Enter the ip nat inside source list overload command. This command enables the router to use one global address for many local addresses. When overloading is configured, the TCP or UDP port number of each inside host distinguishes between the multiple conversations using the same local IP address. Packets with source addresses that pass the access list are dynamically translated using global addresses from the named pool. Enter ip access-list standard command to specify the standard IP access list. Enter permit command to permit the packets from the named pool.

```
Router(config)#ip nat inside source list NAT_ACL interface GigabitEthernet0 overload
Router(config)#ip access-list standard NAT_ACL
Router(config)#permit 192.168.0.0 0.0.1.255
```

Step 10.

Specify the PAT entry to direct the browser traffic via IOS to second Core web server using 8443 port.

```
Router(config)#ip nat inside source static tcp 192.168.3.2 8443 interface gigabitEthernet0 8443
```

Step 11.

Finally, configure DHCP pool for guest OS applications.

```
Router(config)# ip dhcp excluded-address 192.168.1.0 192.168.1.1
Router(config)# ip dhcp pool iox-apps
Router(config)# network 192.168.1.0 255.255.255.0
Router(config)# default-router 192.168.1.1
Router(config)# domain-name sample.com
Router(config)# dns-server 171.x.x.x
```

Local Manager Web Interface

After you configure IOx on your router, you can access the web interface to manage the IOx applications. The IP address of the Gigabit Ethernet interface of the router is used to generate the web URL. Foreexample, if the IP address of the GE interface is 172.x.x.x, then the web URL of the Local Manager is <https://172.x.x.x:8443>.

Log in to the Local Manager using your Username and Password. The User name and the Password is authenticated against the Username and Password used for router login. You should have **privilege 15** to access the Web interface. The following example shows how to enable **privilege 15** in your router:

```
username username privilege 15 password 0 password
```

Example:

```
username cisco privilege 15 password 0 cisco
```


Deploying Applications

The applications that you deploy on your device should be in the form of an OVA file. You can deploy a maximum of five applications on your device. At a time, you can run only one application.

Note: The OVA file size should not exceed 100 MB. The deployment may take up to 35 minutes to finish. It also depends on the size of the OVA file and the availability of the Linux host CPU. Do not reload the browser page. Reloading the browser page will terminate the deployment process. Usually, for 50MB OVA with lightly loaded Linux host, it takes about 17 minutes to finish the installation

To deploy an application, follow these steps:

- Step 1. Log in to Local Manager.
- Step 2. Choose Applications tab, then click Add/Deploy.
- Step 3. Click Choose File to select the application you want to deploy.

- 
- Step 4. Enter an application ID for the selected application. Valid characters for the application ID are 0-9, a-z, A-Z, and _. The application ID length must be less than 64 characters.

Starting Applications

After you successfully deploy an IOx application, you can see that application in the application list.

Note: At a time, you can start only one application.

To start an application, follow these steps:

- Step 1. Choose Applications tab.
- Step 2. From the listed applications, identify the application you want to start.
- Step 3. Click Start. The status of the application changes to RUNNING.

Stopping Applications

You can stop a running application. After you stop the applications, its status changes to STOPPED.

To stop an application, follow these steps:

- Step 1. Choose Applications tab.
- Step 2. From the listed applications, identify the application you want to stop.
- Step 3. Click Stop. The status of the application changes to STOPPED.

Deleting Applications

To delete an application, follow these steps:

- Step 1. Choose Applications tab.
- Step 2. From the listed applications, identify the application you want to delete.
- Step 3. Click Delete. The application will be deleted from your device.

Managing Deployed Applications

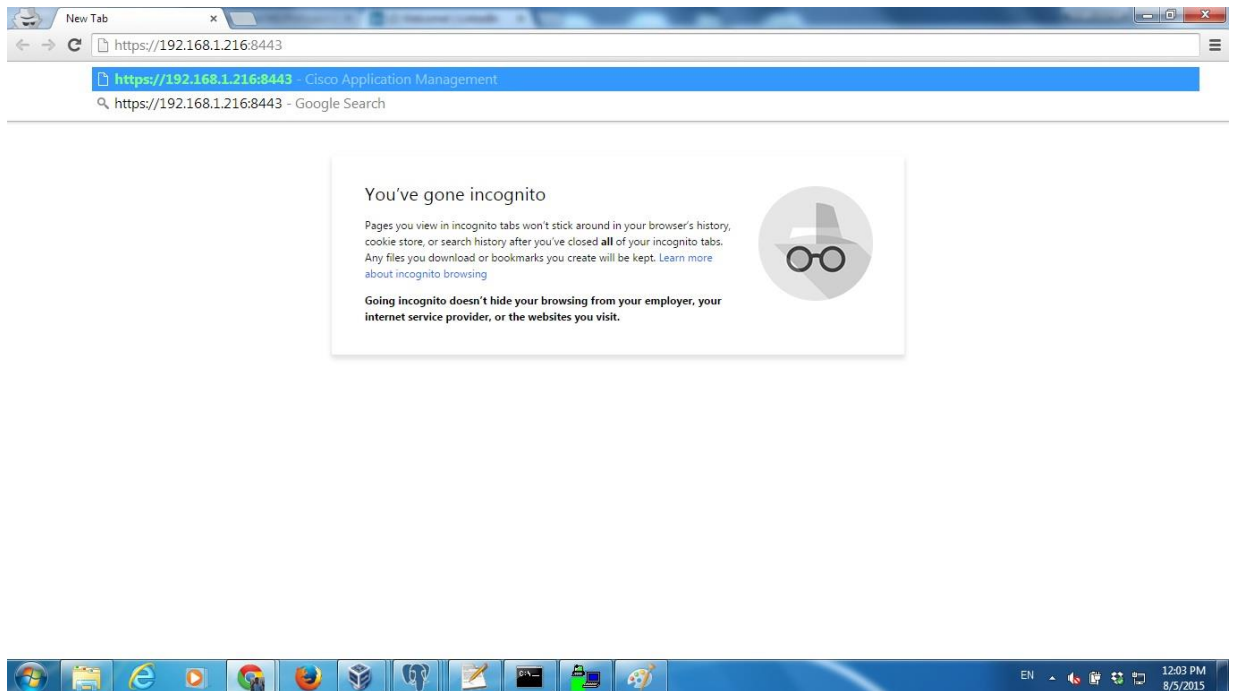
Local Manager allows you to configure a deployed application. The application parameters can be modified using the manage applications option. You can also view the log files of the application.

To manage an application, follow these steps:

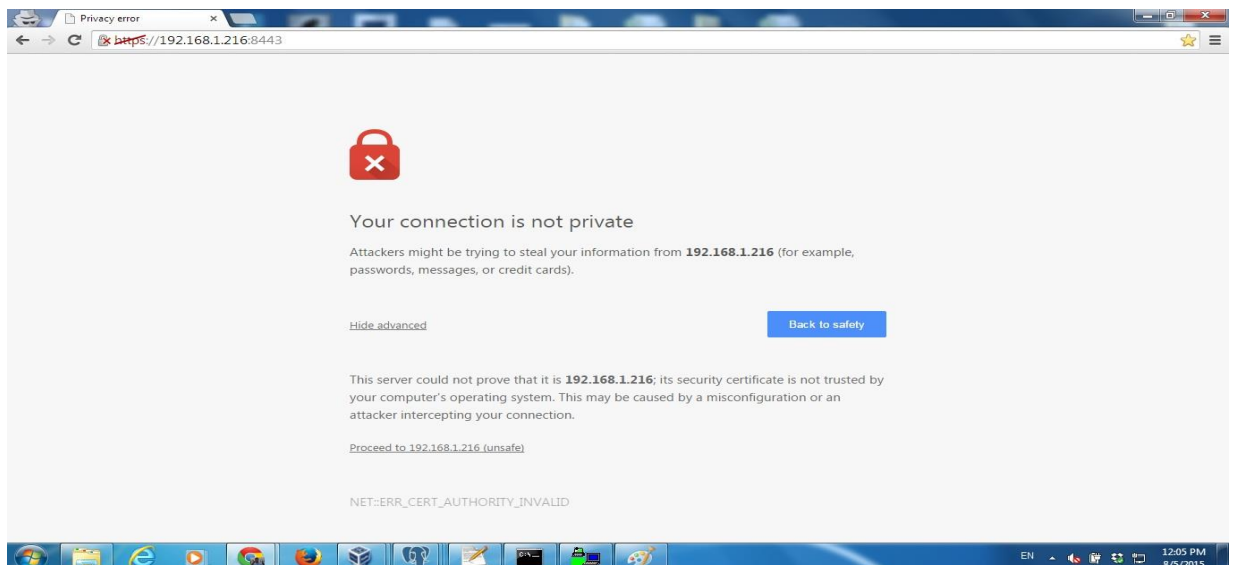
- Step 1. Choose Applications tab.
- Step 2. From the listed applications, identify the application you want to manage
- Step 3. Click Manage. A new tab appears for the application.
- Step 4. Choose Configuration tab.
- Step 5. Modify the application parameters.
- Step 6. Click Save.
- Step 7. Click Show Logs to view the application logs. You can download these logs.

Deploying Covacsis IPF Framework using Cisco Local Manger.

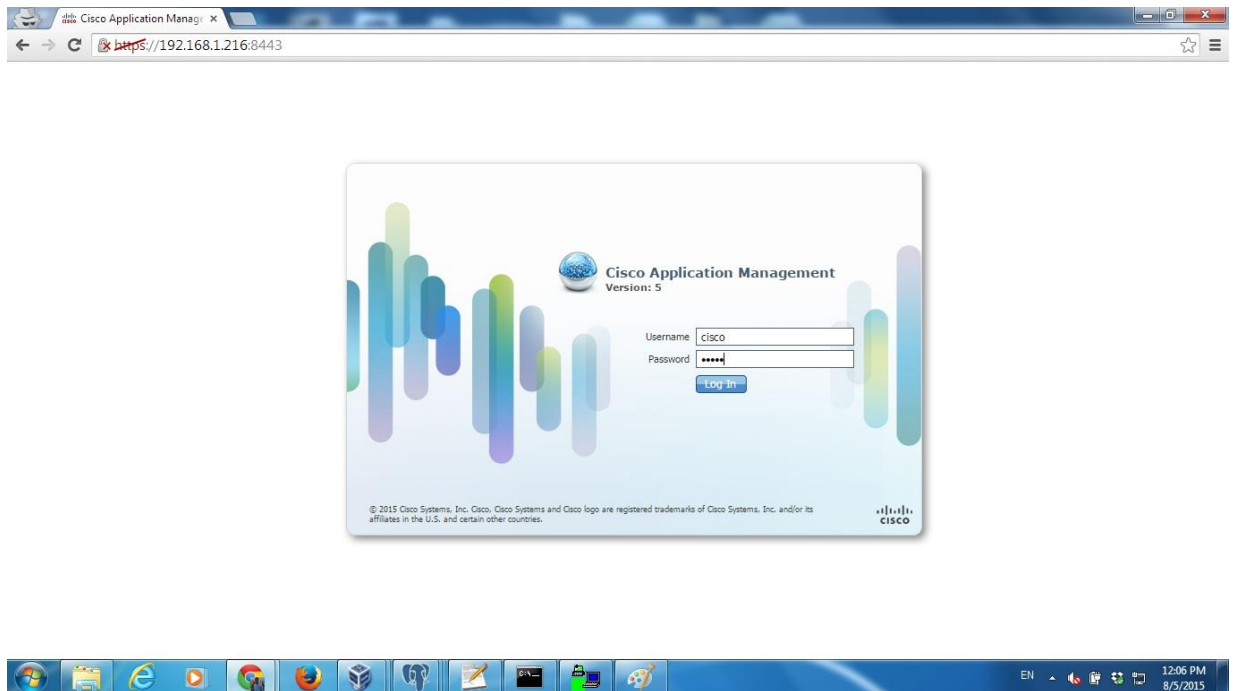
Step 1. Open the Browser to use Cisco Local Application Management page. Enter WAN Ethernet ip address in the browser on specified port of 8443. (Recommend Google chrome Browser).



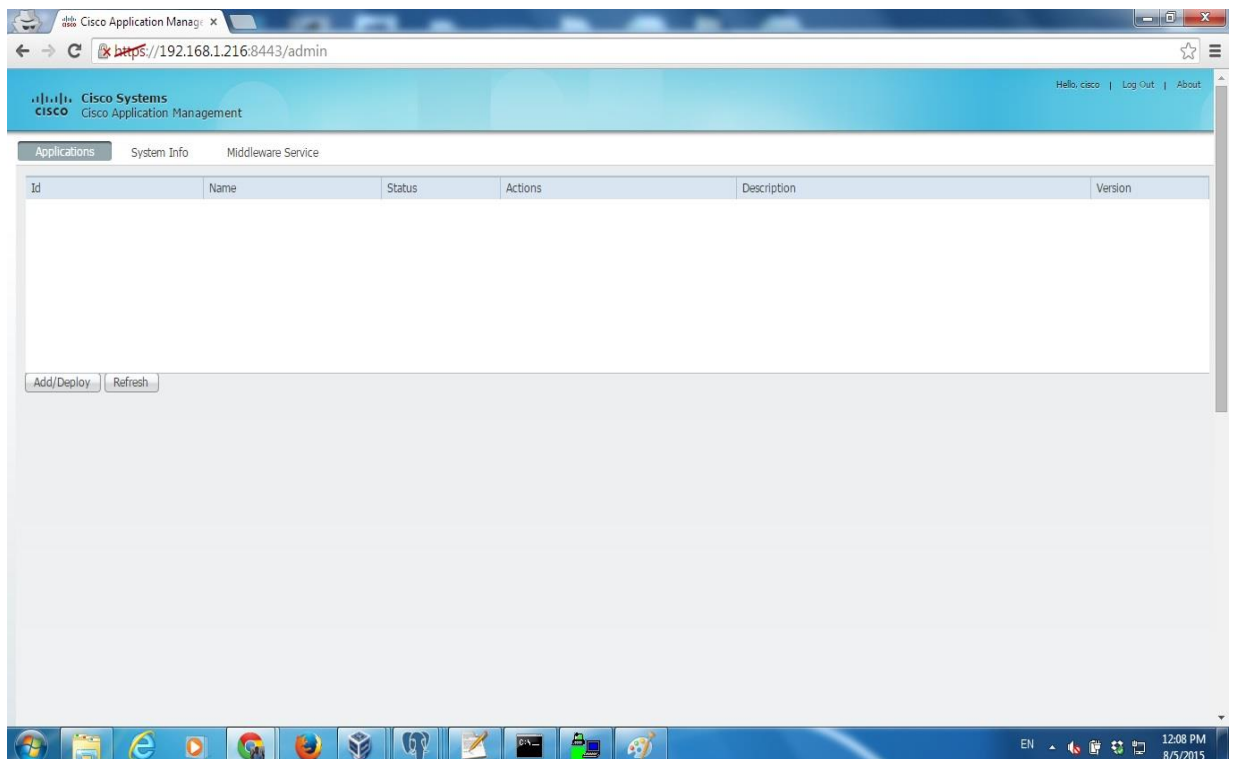
Step 2. Caution message will appear. Click on Proceed to 192.168.1.216(unsafe)



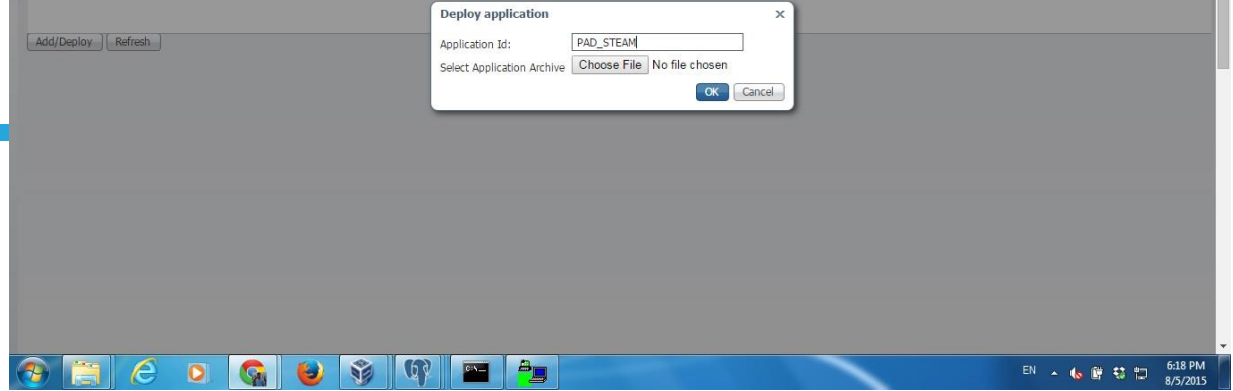
Step 3. Cisco Local Application Manager page open up. Enter cisco/cisco as username and password.



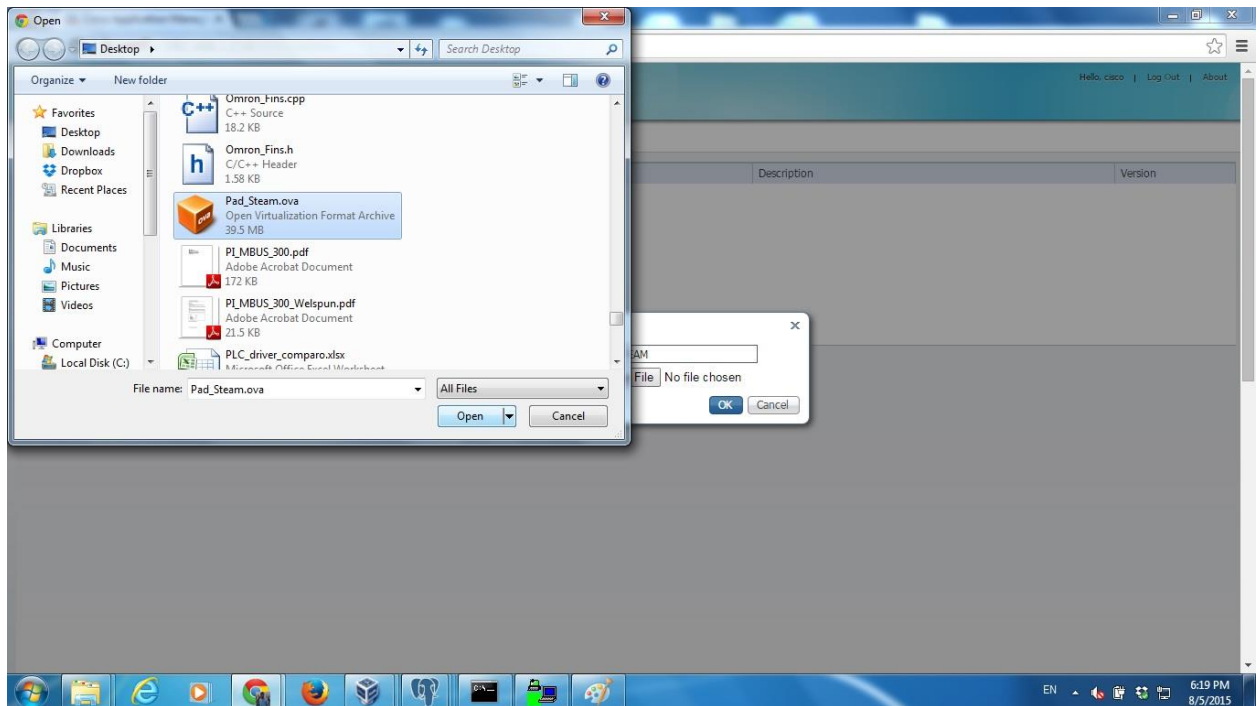
Step 4. Select the Application tab and click to Add/deploy Covacsis IPF Application.



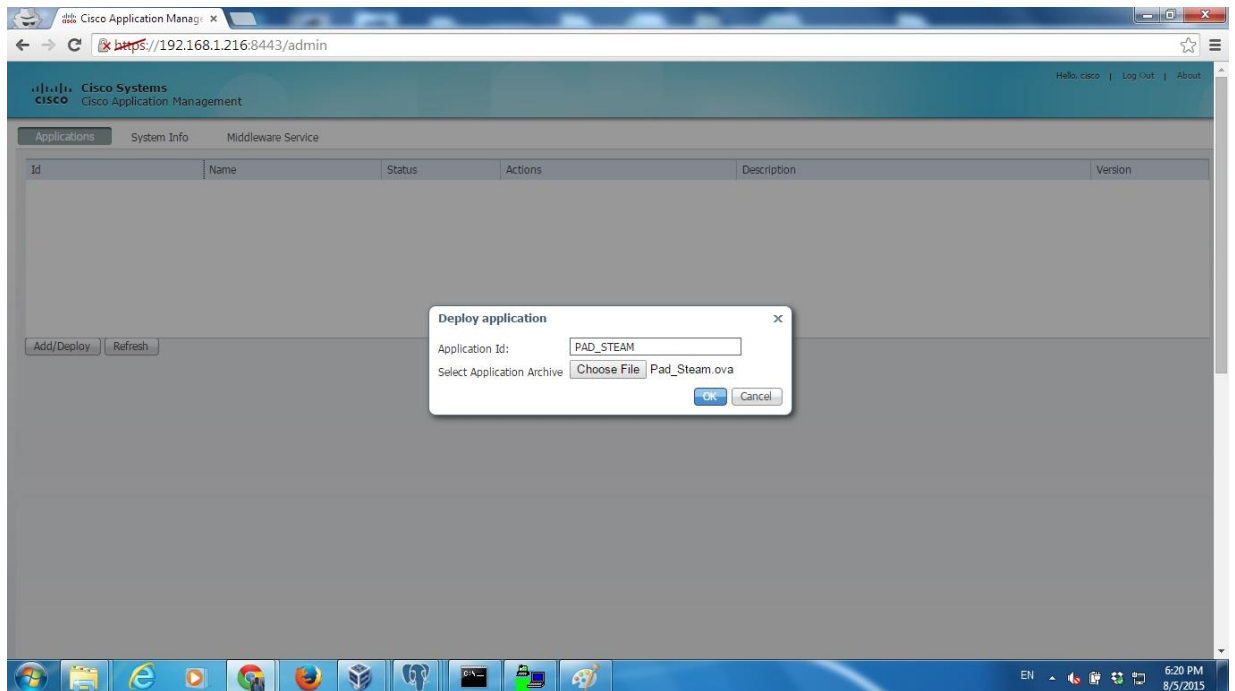
Step 5. Enter a name to your Application ID. Here we are addressing it as PAD_STEAM, since the application will run with PAD_STEAM Machine. Then Select application archive by clicking on



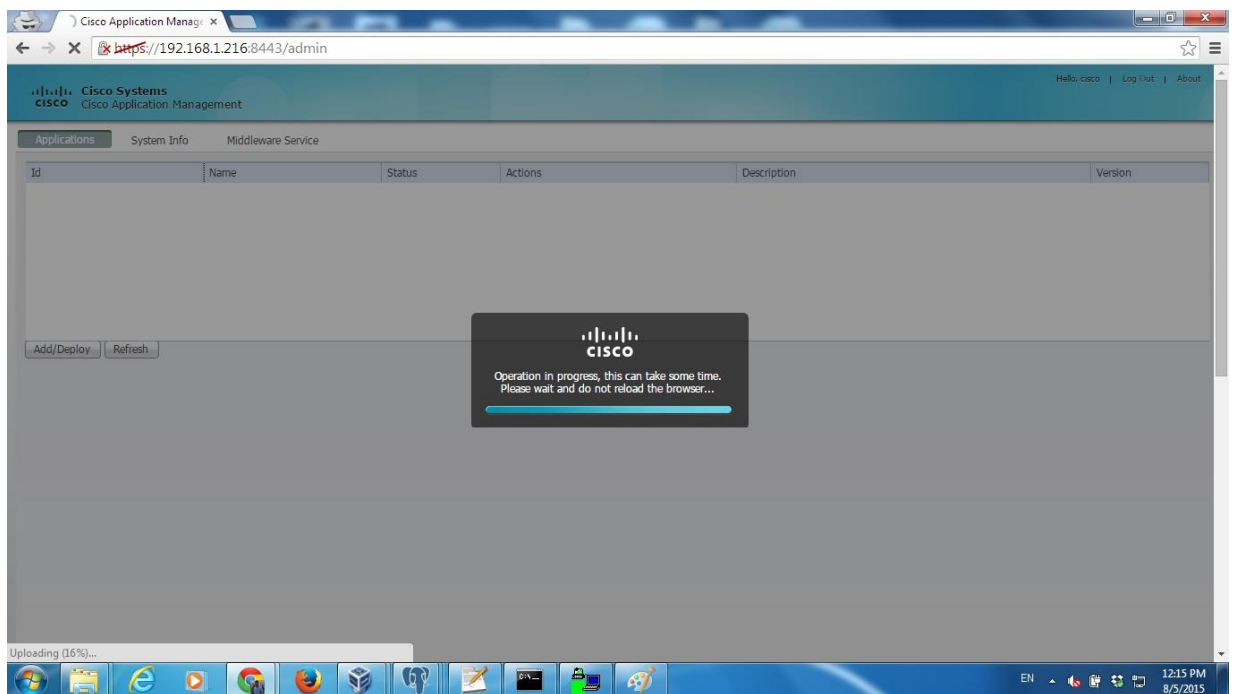
Step 6. Browse for a proper path of your application and Click on Open. Here we have selected Pad_Steam.ova .



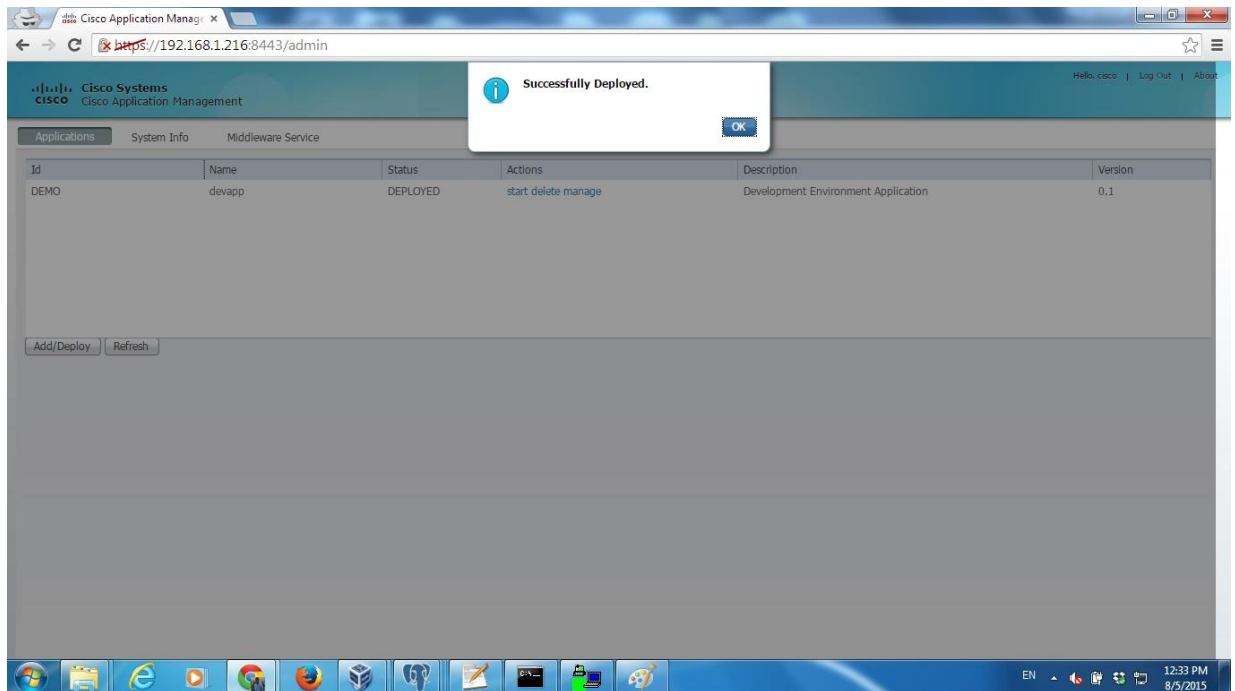
Step 7. After selection of proper Application file(Pad_Steam.ova) click on “OK” button to upload the application into IoX.



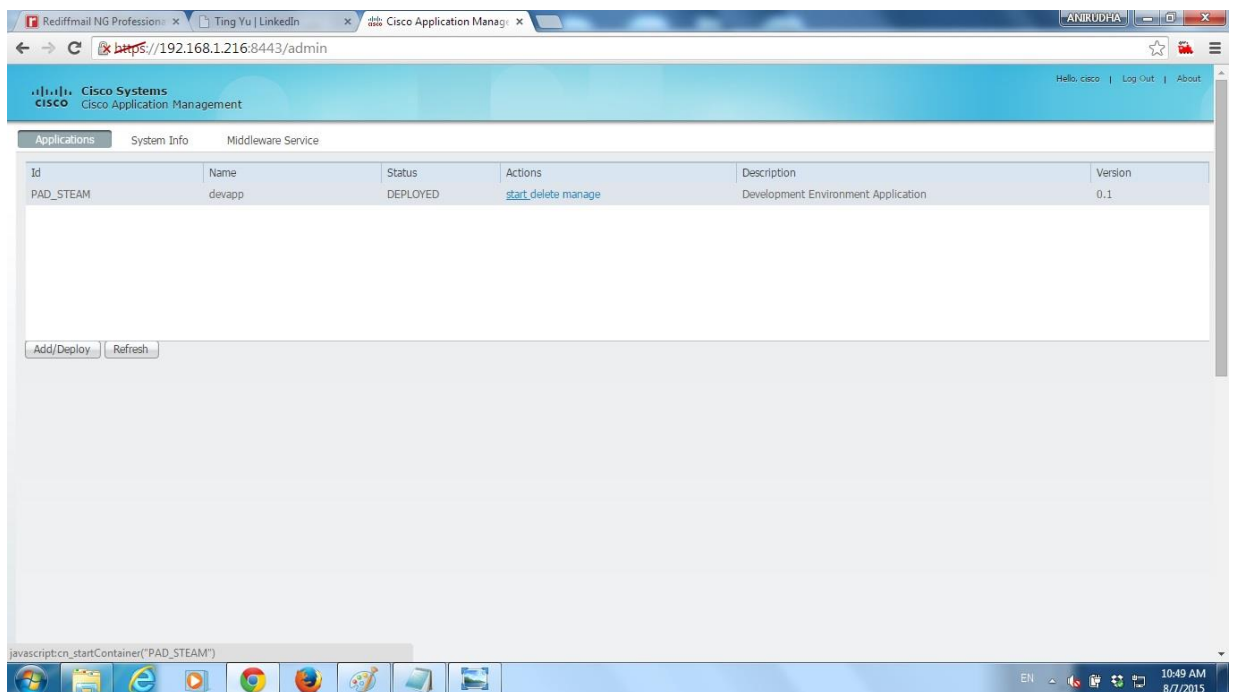
Step 8. Application deployment operation is in progress , it will take some to deploy.



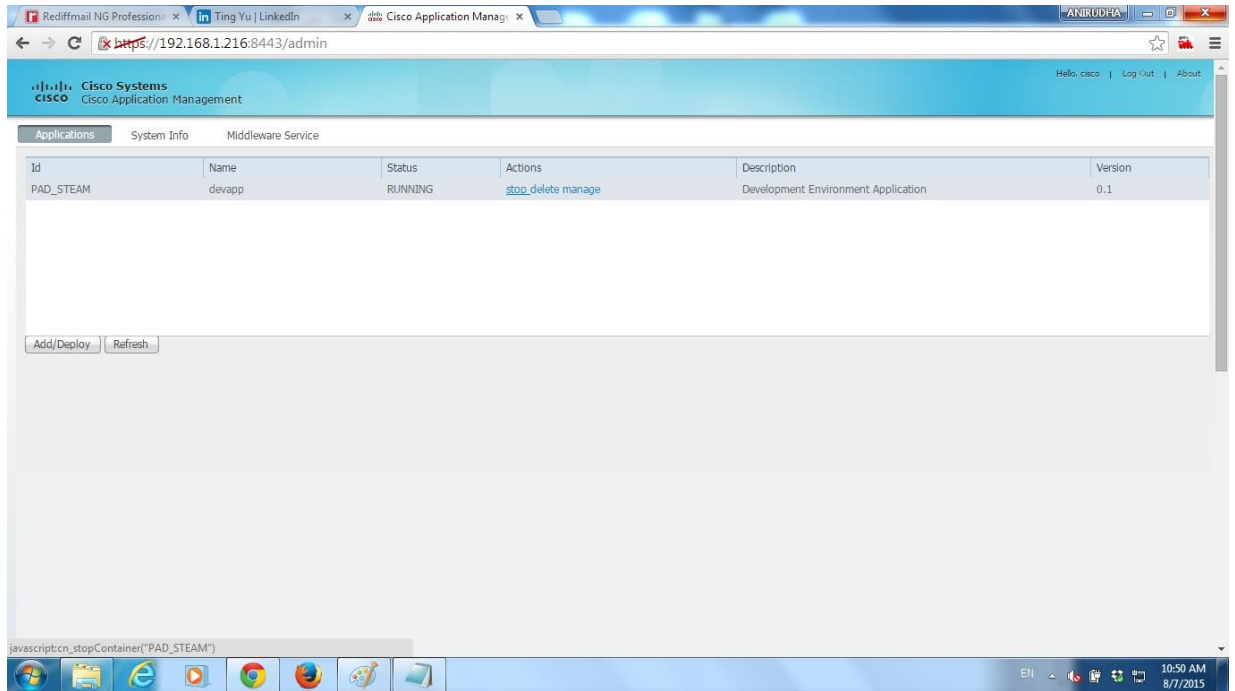
Step 9. Once the upload is completed, a Successfully Deployed message will appear on the screen. If there is some problem in your application ova file or some other error has occurred, the same will be displayed on the screen.



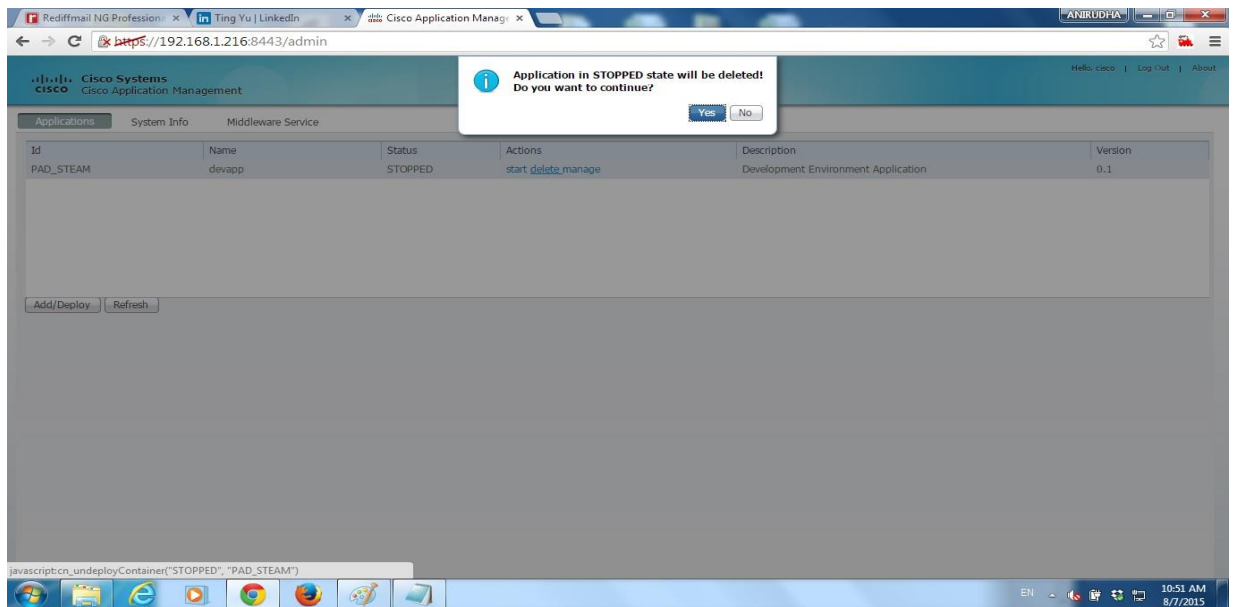
Step 10. Upon Successfully deployment of Application, Client on start in Action Column to start the application. Verify application status changed from DEPLOYED to RUNNING STATE.



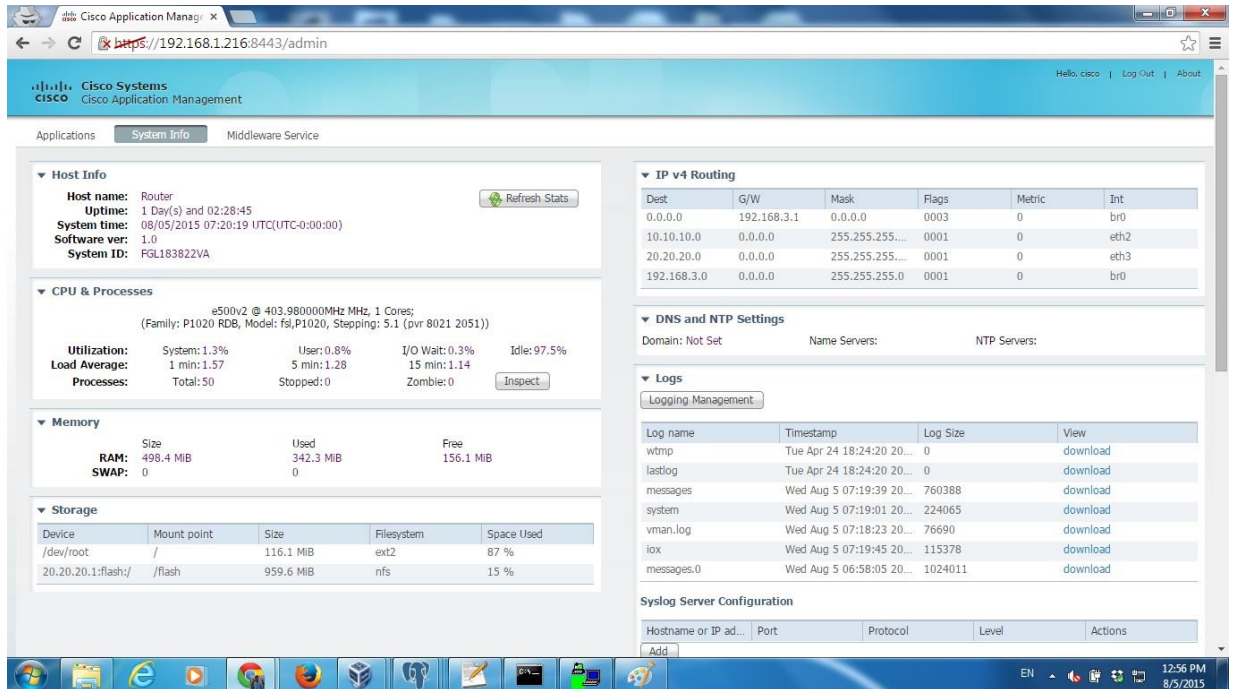
Step 11. If wish to stop the application for maintains of the application, Click on "STOP" in the Action column.



Step 12. Confirms "Yes" to stop the application.



Step 13. Confirms "Yes" to stop the application.



Configuring Guest Serial

This feature allows the installed IOx applications to access the router's serial interface. This configuration is optional because not all applications need this configuration.

After you install and activate a virtual serial enabled OVA package, the third party applications can see device nodes like `/dev/vport0p2`. The virtual port can be connected to a physical port on your router or on a serial extension module. This allows the third party applications to access and control the external device like a local device.

To configure a virtual serial port in guest, follow these steps:

- Step 1. Install and activate an OVA package with virtual serial port enabled.
- Step 2. Configure raw-socket TCP under serial interface and the corresponding async line.

The following example shows how to connect the serial port `s0` of a Cisco 819 router to guest:

```
interface serial0
physical-layer async
vrf forwarding internal-score-vrf
no ip address
encapsulation raw-tcp
end
line 7

raw-socket tcp client 192.168.3.2 32000
```

In the `raw-socket tcp client` command, 10.10.10.2 is the IP address for host Linux, and 32000 is the serial TCP port described in the VM description file (xml from OVA).

For C841M Series Routers, the interface name should be either `serial0/0/0` or `serial0/1/0` depending on the module slot.

The following example shows a sample configuration for a WIM-1T serial module installed on the C800M extension module slot 0:

```
interface Serial0/0/0
physical-layer async
no ip address
encapsulation raw-tcp
!
line 3
raw-socket tcp client 192.168.3.2 32000
```

The following example shows how to connect WIM-1T serial module installed on the C800M extension module slot1 to guest:

```
interface Serial0/1/0
physical-layer async
no ip address
encapsulation raw-tcp
end
line 19
raw-socket tcp client 192.168.3.2 32000
```

Note that the async line associated with the serial interface s0 on C819 is 7. The async lines associated with serial interface s0/0/0 and s0/1/0 on C841M platform are 3 and 19, respectively.

An async line is set to 9600 baud, no parity, and 1 stop bits by default. All TTY terminal settings need to be configured under the async line in IOS. The following example shows how to change the baud rate to 115Kbps:

```
line 7
raw-socket tcp client 192.168.3.2 32000
stopbits 1
speed 115200
```

Third party applications can access the external device by reading and writing to the device node using Linux system calls open(), close(), read(), and write(). The following example shows how to do it:

```
int main (int argc, char **argv)
{
    int fd;
    fd = open("/dev/vport0p2", O_RDWR);
```

You can also run shell commands from guest console to send the data to or read data from the external device. The following example shows how to do it:

```
[root@P1020RDB /root]# echo "hello" > /dev/vport0p2
[root@P1020RDB /root]# cat /dev/vport0p2
```

Complete loX Configuration Example:

```
PAD_Steam_PLC#sh run
Building Configuration...

Current configuration : 3544 bytes
!
! Last configuration change at 12:53:23 UTC Tue Aug 4 2015
!
version 15.5
service timestamps debug datetime msec
```



```

service timestamps log datetime msec
no service password-encryption
!
hostname PAD_Steam_PLC
!
boot-start-marker
boot system flash:c800-universalk9_iox-mz.SPA.155-0.25.T0.7
boot-end-marker
!
!
!
no aaa new-model
bsd-client server url https://cloudsso.cisco.com/as/token.oauth2
!
!
!
!
!
!
!
!
!

!
ip dhcp excluded-address 10.10.3.1 10.10.3.2
!
ip dhcp pool iox-apps
network 10.10.3.0 255.255.255.0
default-router 10.10.3.1
domain-name fake.com
lease infinite
!
!
!
ip cef
no ipv6 cef
!
!
!
!
!bundle-name authenticated
!
!
chat-script lte "" "AT!CALL" TIMEOUT 20 "OK"
!
!
!
!
!
cts logging verbose
license udi pid C819HG-4G-G-K9 sn FGL183822VD
!
!
username cisco privilege 15 password 0 cisco
!
!
!
!
!
controller Cellular 0
lte modem link-recovery rssi onset-threshold -110
lte modem link-recovery monitor-timer 20
lte modem link-recovery wait-timer 10
lte modem link-recovery debounce-count 6
!
!
!

```



```

!
!
ip nat inside source static tcp 10.10.2.2 8443 interface GigabitEthernet0 8443
ip nat inside source static tcp 10.10.2.2 22 interface GigabitEthernet0 22
ip nat inside source static tcp 10.10.2.2 65535 interface GigabitEthernet0 65535
ip nat inside source list 7 interface GigabitEthernet0 overload
ip nat inside source static tcp 10.10.3.3 6666 interface GigabitEthernet0 6666
ip route 0.0.0.0 0.0.0.0 192.168.2.1
!
no service-routing capabilities-manager
!
snmp-server community public RO
access-list 7 permit 10.10.0.0 0.0.255.255
access-list 7 permit 173.0.0.0 0.255.255.255
access-list 7 permit 10.61.0.0 0.0.255.255
!
control-plane
!
!
!
mgcp behavior rsip-range tgcp-only
mgcp behavior comedia-role none
mgcp behavior comedia-check-media-src disable
mgcp behavior comedia-sdp-force disable
!
mgcp profile default
!
!
!
!
!
!
!
line con 0
  no modem enable
line aux 0
line 2
-character
  no exec
  transport preferred none
  transport input all
  stopbits 1
line 3
  script dialer lte
  no exec
line 8
  no exec
line vty 0
  login
  transport input none
line vty 1
  login
  length 0
  transport input none
line vty 2 4
  login
  transport input none
!

```

Troubleshooting

Table 3 lists the commands available for troubleshooting.

Table 3. Commands for Troubleshooting IOx

Command Function	Command Syntax	Mode	Example
Debugs IOx configuration errors.	debug iox config level error	Privileged EXEC (#)	<pre> Router#debug iox config level error *Oct 7 08:30:27.951 PDT: if c800 iox infra cli handler.c :: debug_iox_configuration_command_handler() : 242 - Changed configuration debug level to 3 iox 819 2# iox_819_2# iox_819_2#conf t Enter configuration commands, one per line. End with CNTL/Z. iox_819_2(config)#iox iox 819 2(config-iox)#host ip add 192.168.100.2 255.255.255.0 iox 819 2(config-iox)# *Oct 7 08:30:44.043 PDT: if_c800_iox_trans_mgr.c :: iox_create_transaction() : 50 - Created transaction: tid=14, pid=155 *Oct 7 08:30:44.043 PDT: if c800 iox cli handler.c :: cfg iox host ip address cmd handler () : 387 - host ip address entered address: 192.168.100.2 mask: 255.255.255.0 *Oct 7 08:30:44.043 PDT: if c800 iox cli handler.c :: iox_validate_host_ip_address() : 309 - All checks passed </pre>
Debugs IOx configuration.	debug iox config level debug	Privileged EXEC (#)	<pre> Router#debug iox config level debug iox 819 2#conf t Enter configuration commands, one per line. End with CNTL/Z. iox 819 2(config)#iox iox 819 2(config-iox)#host ip default-gateway 192.168.100.1 % configuration failure: host ip default-gateway iox 819 2(config-iox)# *Oct 7 08:35:10.231 PDT: SCORE ERR: score ipc send msg socket 394 Send failed, socket down *Oct 7 08:35:10.231 PDT: if c800 iox cli handler.c :: cfg iox host default gateway cmd ha ndler() : 645 - Sending host ip message unsuccessful </pre>
Debugs IOx configuration warnings.	debug iox config level warning	Privileged EXEC (#)	<pre> Router#debug iox config level warning iox 819 2#conf t Enter configuration commands, one per line. End with CNTL/Z. iox_819_2(config)#iox iox_819_2(config-iox)#host ip default-gateway 192.168.100.1 </pre>

			<pre>% configuration failure: host ip default-gateway iox 819 2(config-iox)# *Oct 7 08:37:06.067 PDT: SCORE_ERR: score ipc send msg socket 394 Send failed, socket down *Oct 7 08:37:06.067 PDT: if_c800_iox_cli_handler.c :: cfg_iox_host_default_gateway_cmd_ha ndler() : 645 - Sending host ip message unsuccessful Router#debug iox host-agent level error Oct 23 22:37:40.598: if_c800_iox_trans_mgr.c :: iox_create_transaction() : 50 - Created transaction: tid=2, pid=103 *Oct 23 22:37:40.598: if_c800_iox_infra_cli_handler.c :: set_debug_level() : 151 - *****[IOS-DUMP]***** **** *Oct 23 22:37:40.598: if_c800_iox_infra_cli_handler.c :: set_debug_level() : 151 - 00 0F 00 06 00 00 00 02 07 01 00 08 01 03 *Oct 23 22:37:40.598: if_c800_iox_infra_cli_handler.c :: set debug level() : 151 - ***** **** *Oct 23 22:37:40.598: if_c800_iox_ipc_utils.c :: iox_msg_send() : 137 - Pid: 103 Sending iox message to Score *Oct 23 22:37:40.598: if_c800_iox_cli_handler.c :: iox_cli_wait_for_response() : 207 - CLI is waiting for response - pid: 103 *Oct 23 22:37:40.810: if_c800_iox_ipc_main.c :: iox_rcv_msg_from_ioxhad() : 35 - *****[IOS-DUMP]***** **** *Oct 23 22:37:40.810: if_c800_iox_ipc_main.c :: iox_rcv_msg_from_ioxhad() : 35 - 00 10 00 03 00 00 00 02 00 01 00 *Oct 23 22:37:40.810: if_c800_iox_ipc_main.c :: iox_rcv_msg_from_ioxhad() : 35 - ***** ****</pre>
Use this command if you face any IOx configuration issue from the IOS side. This allows you to monitor messaging between IOS and IOX framework. Note that this is for debugging IOS config/messaging and does not alter the debugging levels of IOX platform in general.	debug iox host-agent level error	Privileged EXEC (#)	
Debugs IOx host agent	debug iox host-agent level debug	Privileged EXEC (#)	<pre>Router#debug iox host-agent level debug *Oct 7 08:43:04.727 PDT: if_c800_iox_infra_cli_handler.c :: set_debug_level() : 151 - *****[IOS-DUMP]***** **** *Oct 7 08:43:04.727 PDT: if_c800_iox_infra_cli_handler.c :: set_debug_level() : 151 - 00 0F 00 06 00 00 00 14 07 01 00 08 01 03 *Oct 7 08:43:04.727 PDT: if_c800_iox_infra_cli_handler.c ::</pre>

			<pre> set debug level() : 151 - ***** ***** Router#debug iox host-agent level warning </pre>
Debugs IOx host agent warnings.	debug iox host-agent level warning	Privileged EXEC (#)	
Resets the IOx framework.	reset iox	Privileged EXEC (#)	<pre> Router#reset iox *Oct 23 22:41:05.406: if_c800_iox_trans_mgr.c :: iox_create_transaction() : 50 - Created transaction: tid=4, pid=103 *Oct 23 22:41:05.406: if c800 iox infra cli handler.c :: exec iox infra command handler() : 298 - *****[IOS-DUMP]***** ***** *Oct 23 22:41:05.406: if_c800_iox_infra_cli_handler.c :: exec iox infra command handler() : 298 - 00 11 00 03 00 00 00 04 07 01 01 *Oct 23 22:41:05.406: if_c800_iox_infra_cli_handler.c :: exec_iox_infra_command_handler() : 298 - ***** ***** *Oct 23 22:41:05.406: if c800 iox ipc utils.c :: iox_msg_send() : 137 - Pid: 103 Sending iox message to Score % Couldn't process IOx Infrastructure response Router#show raw-socket tcp sessions ----- ----- TCP Sessions ----- ----- interface tty socket mode local_ip_addr local_port dest ip addr dest port up time idle_time/timeout vrf_name Se0 7 0 client 10.10.10.1 34383 10.10.10.2 32000 00:00:10 00:00:10 /5 min internal-score-vrf Router# show raw-socket tcp statistic ----- ----- Network-Serial Statistics ----- ----- Interface tty sessions network_in_bytes network out bytes network_to_tty_frames tty_to_network_frames vrf_name Se0 7 1 6 6 1 1 internal-score-vrf ----- CEF Connections Statistics ----- </pre>
Displays the status of the raw socket session.	show raw-socket tcp sessions	Privileged EXEC (#)	
Displays the statistics of the raw sockets.	show raw-socket tcp statistic	Privileged EXEC	

			<pre> tty_id network_in_frames network_in_bytes network_out_frames network_out_bytes 0 0 0 0 0 0 0 0 0 0 </pre>
Displays application specific information.	show virtual-service detail	Privileged EXEC	<pre> Router# show virtual-service detail name APP Virtual service APP detail State : Activated Package information Name : APP Path : flash:/iox/tmp/APP.ova Application Name : KVM1_Application Installed version : 2.0 Description : KVM1 Linux Test Distro Signing Key type : Unsigned Method : SHA-1 Licensing Name : kvm1_license Version: 3.3 Activated profile name: Resource reservation Disk : 16 MB Memory : 256 MB CPU : 55% system CPU VCPUs : 1 (sockets:1 cores:1 threads:1) Attached devices Type Name Alias ----- ----- NIC dp_1_0 net1 Serial/shell serial0 Disk shared_moun Network interfaces MAC address Attached to interface ----- ----- 52:54:11:11:00:FE VirtualPortGroup0 Resource admission (without profile) : passed Disk space : 16MB Memory : 256MB CPU : 55% system CPU VCPUs : 1 (sockets:1 cores:1 threads:1) </pre>
Displays virtual service global information.	show virtual-service global	Privileged EXEC	<pre> Router# show virtual-service global Virtual Service Global State and Virtualization Limits: Infrastructure version : 1.7 Total virtual service installed : 1 Total virtual service activated : 1 Maximum VCPUs per virtual service : 1 Machine types supported : KVM </pre>

			Machine types disabled: LXC Resource virtualization limits: Name Quota Committed Available ----- ----- system CPU (%) 80 55 25 memory (MB) 256 256 0 flash (MB) 1024 11 625
Lists the applications.	show virtual-service list	Privileged EXEC	Router# show virtual-service list Virtual Service List: Name Status Package Name ----- ----- APP Activated APP.ova
Displays information about the appliance profile.	show virtual-service profile	Privileged EXEC	Router# show virtual-service profile
Displays information about the utilization of appliances.	show virtual-service utilization	Privileged EXEC	Router# show virtual-service utilization name APP Virtual-Service Utilization: CPU Utilization: Requested Application Utilization: 55 % Actual Application Utilization: 1 % (30 second average) CPU State: R : Running Memory Utilization: Memory Allocation: 262144 Kb Memory Used: 262144 Kb Network Utilization: Name: dp 1 0, Alias: net1 RX Packets: 16 TX Packets: 24 RX Bytes: 2416 TX Bytes: 6624 RX Errors: 0 TX Errors: 0 Storage Utilization: Name: shared mount, Alias: Capacity(1K blocks): 16384 Used(1K blocks): 20 Available(1K blocks): 16364 Usage: 1 %
Connects to the console of the VM environment that the IOx application is running.	virtual-service connect name appname console	Privileged EXEC	Router# virtual-service connect name sensorbot console
Displays the serial interface configuration and statistics	show interface serial	Privileged EXEC	Router# show interface serial erial0 is up, line protocol is up Hardware is Serial in async mode MTU 1500 bytes, BW 9 Kbit/sec, DLY 1000000 usec, reliability 255/255, txload 1/255, rxload 1/255 Encapsulation RAW-TCP, loopback not set

			Keepalive not set DTR is pulsed for 5 seconds on reset Last input never, output never, output hang never Last clearing of "show interface" counters 5d21h Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0 Queueing strategy: fifo Output queue: 0/10 (size/max) 30 second input rate 0 bits/sec, 0 packets/sec 30 second output rate 0 bits/sec, 0 packets/sec 391 packets input, 3247 bytes, 0 no buffer Received 0 broadcasts (0 IP multicasts) 0 runts, 0 giants, 0 throttles 1 input errors, 0 CRC, 1 frame, 0 overrun, 0 ignored, 0 abort 395 packets output, 3160 bytes, 0 underruns 0 output errors, 0 collisions, 0 interface resets 0 unknown protocol drops 0 output buffer failures, 0 output buffers swapped out 0 carrier transitions DCD=up DSR=up DTR=up RTS=up CTS=up Router# debug raw-socket tcp packet *Oct 23 18:52:25.912: [From Network]<-- received 8 bytes on socket 0 from 192.168.3.2 port 32000 *Oct 23 18:52:25.912: 010300000002C40B *Oct 23 18:52:25.912: [To Serial]<-- sending 8 bytes from socket 0 to interface 7 *Oct 23 18:52:25.912: rawsocket_async_output[tty 7]: Received 8 byte from socket... *Oct 23 18:52:25.912: [Socket --> Async] 01 03 00 00 00 02 C4 0B *Oct 23 18:52:25.948: [Async --> Socket] tty(7) Received 9 byte from serial... *Oct 23 18:52:25.948: [Async --> Socket] 01 03 04 89 2F 80 4F C1 92 *Oct 23 18:52:25.948: [From Serial]--> received 9 bytes from interface 7 tty 7 *Oct 23 18:52:25.948: 010304892F804FC192 *Oct 23 18:52:25.948: [To Network]--> dispatched 9 bytes on socket 0 to ip 192.168.3.2 port 32000
Monitors the serial data flow between IOS and host Linux. Note: This will dump the entire packet contents on the console. You might want to turn off logging console to prevent the debug messages from flooding your console.	debug raw-socket tcp packet debug raw-socket driver packet	Privileged EXEC	

System Information

You can view the health of the system by accessing the Systems Info tab from the Local Manager. The System Info tab allows you to access the following system information:

- Host Name
- CPU and Process Usage
- Memory Utilization
- Storage Utilization
- Routing Details
- DNS and NTP settings
- Syslog Server Configuration
- Interfaces

To see the System Info tab, follow these steps:

- Step 1. Log in to Local Manager.
- Step 2. Choose System Info tab.

The information presented in this tab is not specific to any running application. It is related to the IOx application hosting framework as a whole.

The Logs section in the System Info tab allows you to download the system log files. The most important log files are VMAN and IOX.

To download the log files, follow these steps:

- Step 1. Log in to Local Manager.
- Step 2. Choose System Info tab.
- Step 3. From the Logs section, identify the log you want to download.
- Step 4. Click download link. The log file will be saved in your system.

You can change the log level. The default log level is Info. To change the log level, follow these steps:

- Step 1. Log in to Local Manager.
- Step 2. Choose System Info tab.
- Step 3. From the Logs section, click Logging Management.
- Step 4. Choose the Log Level from the drop down.
- Step 5. Select the category. You have to select at least one category.
- Step 6. Click Save.

For More Information

Read more about the [Cisco Covacsis Solution](#), or contact your local account representative.