



Guest Shell

LAB - 9

Disclaimer

This Lab Guide is designed to assist candidates to facilitate Technology learning. While every effort has been made to ensure that all material is as complete and accurate as possible, the enclosed material is presented on an “as is” basis. Neither the authors nor RSTForum assume any liability or responsibility to any person or entity with respect to loss or damages incurred from the information contained in this Lab guide. This workbook was developed by RSTForum. Any similarities between material presented in this Lab Guide and any other Lab Guide or any other material is completely coincidental.



Configuring a Guest Shell on Cisco Routers.

Step 1: Enable IOx

1. On cisco IOS XE CRS1000v router configure the following command.

```
config t
iox
end
```

iox will take some time to start. So wait about 5 minutes

2. Issue following command to check status of IOx service

```
show iox-service
```

Following output will be seen

IOx Infrastructure Summary:

IOx service (CAF) 1.8.0.3	: Running
IOx service (HA)	: Not Supported
IOx service (IOxman)	: Running
Libvirtd 1.3.4	: Running

IOxman and Libvirtd services must be running to enable GuestShell successfully.



Step 2: Enabling Guest Shell

Configuration must be done to enable communication of Guest Shell container with outside network and ISO-XE router.

To enable the communication between container and IOS XE router it is necessary to create a new VirtualPortGroup interface,

VirtualPortGroup interface should be in same broadcast domain as the Guest Shell container.

Network configuration on the host router.

1. Configure following commands:

```
conf t
interface VirtualPortGroup0
 ip address 172.16.1.1 255.255.255.0
end
```

Network configuration of the Container.

Guest Shell should be configured with IP address, default gateway and DNS server.

2. Configure following commands:

```
conf t
app-hosting appid guestshell
vnic gateway1 virtualportgroup 0 guest-interface 0 guest-ipaddress 172
.16.1.2 netmask 255.255.255.0 gateway 172.16.1.1 name-server 8.8.8.8
end
```



NAT configuration of the host.

3. Container should be connected to internet and to achieve this NAT should be configured:

```
conf t
interface VirtualPortGroup0
  ip nat inside
!
interface GigabitEthernet3
  ip nat outside
!
ip access-list extended NAT-ACL
  permit ip 172.16.1.0 0.0.0.255 any
!
ip nat inside source list NAT-ACL interface GigabitEthernet3 overload
end
```

All required configurations are done and router is ready to start Guest Shell:

Starting Guest Shell

4. To enable the Guest Shell, Type the following command:

```
guestshell enable
```

following output will be seen:

```
Interface will be selected if configured in app-hosting
Please wait for completion
guestshell activated successfully
Current state is: ACTIVATED
guestshell started successfully
Current state is: RUNNING
Guestshell enabled successfully
```



Verify your Guest Shell

5. Type the following command:

```
#show app-hosting detail

App id           : guestshell
Owner            : iox
State            : RUNNING
Application
  Type           : lxc
  Name           : GuestShell
  Version        : 2.5.1
  Description    : Cisco Systems Guest Shell XE for x86_64
  Path           : /guestshell:/guestshell.tar
Activated profile name : custom

Resource reservation
  Memory         : 512 MB
  Disk           : 1 MB
  CPU            : 800 units
  VCPU          : 1

Attached devices
  Type           Name           Alias
-----
serial/shell    iox_console_shell serial0
serial/aux      iox_console_aux   serial1
serial/syslog   iox_syslog        serial2
serial/trace    iox_trace         serial3

Network interfaces
-----
eth0:
  MAC address    : 52:54:dd:55:f2:70
  IPv4 address   : 192.168.1.2
  Network name   : VPG0

Port forwarding
Table-entry  Service  Source-port  Destination-port
-----
```



Step 3: Working with the Guest Shell

1. To enter Guest Shell, give following command:

```
guestshell
```

Guest Shell container provides CentOS flavor and shell.

sudo command are allowed without a password as the guestshell user is member of sudo group.

2. Let us play around with the shell:

```
[guestshell@guestshell ~]$ pwd
/home/guestshell

[guestshell@guestshell ~]$ whoami
guestshell

[guestshell@guestshell ~]$ hostnamectl
  Static hostname: guestshell
            Icon name: computer-container
            Chassis: container
            Machine ID: d1133315e44e4ef0b1baef5c0d0eecc9
            Boot ID: 36e84546a1b147ec928b2f9036d87ec3
    Virtualization: lxc-libvirt
 Operating System: CentOS Linux 7 (Core)
      CPE OS Name: cpe:/o:centos:centos:7
           Kernel: Linux 4.19.64
    Architecture: x86-64
[guestshell@guestshell ~]$
[guestshell@guestshell ~]$ cat /etc/centos-release
CentOS Linux release 7.5.1804 (Core)
```



Running CLI command from Guest Shell

3. Type CLI command from within the guest shell use dohost binary:

```
[guestshell@guestshell ~]$ dohost "sho ip int brief"
```

Interface	IP-Address	OK?	Method	Status	Protocol
GigabitEthernet1	10.0.0.1	YES	NVRAM	up	up
GigabitEthernet2	10.255.255.1	YES	NVRAM	down	down
GigabitEthernet3	192.168.0.169	YES	NVRAM	down	down
Loopback1	88.88.88.88	YES	NVRAM	up	up
VirtualPortGroup0	172.16.1.1	YES	NVRAM	up	up

```
[guestshell@guestshell ~]$
```

Installing git and nano on Guest Shell

4. Type following CLI command in guest shell:

```
[guestshell@guestshell ~]$ sudo yum install -y git nano
```

Cloning RSTForum git repository

5. Type following CLI command in guest shell:

```
[guestshell@guestshell ~]$ git clone https://github.com/RSTForum/knowledgebase
```

Build env: NETCONF / RESTCONF/ ANSIBLE

6. Type following CLI command in guest shell:

```
[guestshell@guestshell ~]$ cd knowledgebase/netconf/
```

```
[guestshell@guestshell netconf]$ sudo pip install -r requirements.txt
```

Setup SSH with Host Machine

7. Run following CLI command in guest shell:

```
[guestshell@guestshell netconf]$ ssh cisco@10.0.0.1
The authenticity of host '10.0.0.1 (10.0.0.1)' can't be established.
RSA key fingerprint is SHA256:nJ+pjK9MKzpHnHhZgBOEBS+XxXr5r85ZYJDfYDuLex0.
RSA key fingerprint is MD5:21:bf:73:78:a2:ef:77:c3:40:10:a8:4c:5c:92:88:4b.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.0.1' (RSA) to the list of known hosts.
cisco@10.0.0.1's password:
```

Check Python NETCONF Script

8. Type following CLI command in guest shell:

```
[guestshell@guestshell netconf]$ python get-int-details.py
-----
<?xml version="1.0" ?>
<rpc-reply message-id="urn:uuid:21ab2152-b9c9-4a02-b3d1-41247747c000" xmlns="urn:iETF:params:xml:ns:netconf:base:1.0" xmlns:nc="urn:iETF:params:xml:ns:netconf:base:1.0">
  <data>
    <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
      <interface>
        <GigabitEthernet>
          <name xmlns:nc="urn:iETF:params:xml:ns:netconf:base:1.0">2
          <description>RSTForum NETCONF Script</description>
          <ip>
            <address>
              <primary>
                <address>10.255.255.1</address>
                <mask>255.255.255.0</mask>
              </primary>
            </address>
          </ip>
          <mop>
            <enabled>false</enabled>
            <sysid>false</sysid>
          </mop>
          <negotiation xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-ethernet">
            <auto>true</auto>
          </negotiation>
        </GigabitEthernet>
      </interface>
    </native>
  </data>
</rpc-reply>
-----
```



Test cURL RESTCONF Script

9. Type following CLI command in guest shell:

```
[guestshell@guestshell netconf]$ curl -i -k -X "GET" "https://10.0.0.1:443/restconf/data/Cisco-IOS-XE-native:native/interface?fields=GigabitEthernet/ip/address/primary;name" -H 'Accept: application/yang-data+json' -u 'cisco:cisco'
```

```
-----
HTTP/1.1 200 OK
Server: nginx
Date: Fri, 12 Jun 2020 15:26:51 GMT
Content-Type: application/yang-data+json
Transfer-Encoding: chunked
Connection: keep-alive
Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
Pragma: no-cache
```

```
{
  "Cisco-IOS-XE-native:ip": {
    "address": {
      "primary": {
        "address": "10.255.255.1",
        "mask": "255.255.255.0"
      }
    }
  }
}
[guestshell@guestshell restconf]$
-----
```

