



Guest Shell

Guestshell is a virtualized Linux-based environment, designed to run custom Linux applications, including Python for automated control and management of Cisco devices. It also includes the automated provisioning (Day zero) of systems. This container shell provides a secure environment, decoupled from the host device, in which users can install scripts or software packages and run them.

This module describes Guest Shell and how to enable it.

- [Finding Feature Information, on page 1](#)
- [Information About Guest Shell, on page 1](#)
- [How to Enable Guest Shell, on page 5](#)
- [Configuration Examples for Guest Shell, on page 9](#)
- [Additional References for Guest Shell, on page 12](#)
- [Feature Information for Guest Shell, on page 12](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Information About Guest Shell

Guest Shell Overview

Guestshell is a virtualized Linux-based environment, designed to run custom Linux applications, including Python for automated control and management of Cisco devices. Using Guest Shell, customers can also install, update, and operate third-party Linux applications. It is bundled with the system image and can be installed using the **guestshell enable** IOS command.

The Guest Shell environment is intended for tools, Linux utilities, and manageability rather than networking.

Guest Shell shares the kernel with the host (Cisco switches and routers) system. Users can access the Linux shell of Guest Shell and update scripts and software packages in the container rootfs. However, users within the Guest Shell cannot modify the host file system and processes.

Guest Shell container is managed using IOx. IOx is Cisco's Application Hosting Infrastructure for Cisco IOS XE devices. IOx enables hosting of applications and services developed by Cisco, partners, and third-party developers in network edge devices, seamlessly across diverse and disparate hardware platforms.

This table provides information about the various Guest Shell capabilities and the supported platforms.

Table 1: Cisco Guest Shell Capabilities

	Guest Shell Lite (Limited LXC Container)	Guest Shell (LXC Container)
Operating System	Cisco IOS XE	Cisco IOS XE
Supported Platforms		<ul style="list-style-type: none"> • Cisco Catalyst 9500 Series Switches (all models)
Guest Shell Environment	Montavista CGE7	CentOS 7
Python 2.7	Supported (Python V2.7.11)	Supported (Python V2.7.5)
Custom Python Libraries	<ul style="list-style-type: none"> • Cisco Embedded Event Manager • Cisco IOS XE CLIs • Ncclient 	<ul style="list-style-type: none"> • Cisco Embedded Event Manager • Cisco IOS XE CLIs
Supported Rootfs	Busybox, SSH, and Python PIP install	SSH, Yum install, and Python PIP install
GNU C Compiler	Not supported	Not supported
RPM Install	Not supported	Supported
Architecture	MIPS	x86

Guest Shell Vs Guest Shell Lite

The Guest Shell container allows users to run their scripts and apps on the system. The Guest Shell container on Intel x86 platforms will be a Linux container (LXC) with a CentOS 7.0 minimal rootfs. You can install other Python libraries such as, Python Version 3.0 during runtime using the Yum utility in CentOS 7.0. You can also install or update python packages using PIP.

The Guest Shell Lite container on MIPS platforms such as, Catalyst 3650 and Catalyst 3850 Series Switches have the Montavista Carrier Grade Edition (CGE) 7.0 rootfs. You can only install or run scripts in Guest Shell Lite. Yum install is not supported on these devices.

Guest Shell Security

Cisco provides security to ensure that users or apps in the Guest Shell do not compromise the host system. Guest Shell is isolated from the host kernel, and it runs as an unprivileged container.

Hardware Requirements for Guestshell

This section provides information about the hardware requirements for supported platforms.

Table 2: Guest Shell Support on Catalyst Switches

Platforms	Default DRAM	Guest Shell Support
C9500-24Q-x (all)	16 GB	Supported



Note Virtual-service installed applications and Guest Shell container cannot co-exist.

Guest Shell Storage Requirements

On Catalyst 3650 and Catalyst 3850 Series Switches, Guest Shell can only be installed on the flash filesystem. Bootflash of Catalyst 3850 Series Switches require 75 MB free disk space for Guest Shell to install successfully.

Cisco Catalyst 9300 and Catalyst 9500 Series Switches require 1100 MB free hard disk space for Guest Shell to install successfully.

On Cisco 4000 Series Integrated Services Routers, Guest Shell is installed on the Network Interface Module (NIM)-Service Set Identifier (SSID) (hard disk), if available. If the hard disk drive is available, there is no option to select bootflash to install Guest Shell. Cisco 4000 Series Integrated Services Routers require 1100 MB free hard disk (NIM-SSID) space for Guest Shell to install successfully.

During Guest Shell installation, if enough hard disk space is not available, an error message is displayed.

Bootflash or hard disk space can be used to store additional data by Guest Shell. On Cisco Catalyst 3850 Series Switches, Guest Shell has 18 MB of storage space available and on Cisco 4000 Series Integrated Services Routers, Guest Shell has 800 MB of storage space available. Because Guest Shell accesses the bootflash, it can use the entire space available.

Table 3: Resources Available to Guest Shell and Guest Shell Lite

Resource	Default	Minimum/Maximum
CPU	1% Note 1% is not standard; 800 CPU units/ total system CPU units.	1/100%
Memory	256 MB	256/256 MB

Accessing Guest Shell on a Device

Network administrators can use IOS commands to manage files and utilities in the Guest Shell.

During the Guest Shell installation, SSH access is setup with a key-based authentication. The access to the Guest Shell is restricted to the user with the highest privilege (15) in IOS. This user is granted access into the Linux container as the *guestshell* Linux user, who is a sudoer, and can perform all root operations. Commands

executed through the Guest Shell are executed with the same privilege that a user has when logged into the IOS terminal.

At the Guest Shell prompt, you can execute standard Linux commands.

Accessing Guest Shell Through the Management Port

By default, Guest Shell allows applications to access the management network. Users cannot change the management VRF networking configurations from inside the Guest Shell.



Note For platforms without a management port, a `VirtualPortGroup` can be associated with Guest Shell in the IOS configuration. For more information, see the *Sample VirtualPortGroup Configuration* section.

Stacking with Guest Shell

When Guest Shell is installed, a `gs_script` directory is automatically created in the flash filesystem. This directory is synchronized across stack members. During a switchover, only contents of the `gs_script` directory are synchronized across all stack members. To preserve data during high availability switchover, place data in this directory.

During a high availability switchover, the new active device creates its own Guest Shell installation; the old filesystem is not maintained. The Guest Shell state is maintained during a switchover.

IOx Overview

IOx is a Cisco-developed end-to-end application framework that provides application hosting capabilities for different application types on Cisco network platforms. The Cisco Guest Shell, a special container deployment, is one such application, that is useful in system deployment/use.

IOx facilitates the life-cycle management of app and data exchange by providing a set of services that helps developers to package pre-built apps, and host them on a target device. IOx life-cycle management includes distribution, deployment, hosting, starting, stopping (management), and monitoring of apps and data. IOx services also include app distribution and management tools that help users discover and deploy apps to the IOx framework.

App hosting provides the following features:

- Hides network heterogeneity.
- IOx application programming interfaces (APIs), remotely manage the life cycle of applications hosted on a device.
- Centralized app life-cycle management.
- Cloud-based developer experience.

How to Enable Guest Shell

Managing IOx

Before you begin

IOx takes upto two minutes to start. CAF, IOXman, and Libirdtd services must be running to enable Guest Shell successfully.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	iox Example: Device(config)# iox	Configures IOx services.
Step 4	exit Example: Device(config)# exit	Exits global configuration mode and returns to privileged EXEC mode.
Step 5	show iox-service Example: Device# show iox-service	Displays the status of the IOx service
Step 6	show app-hosting list Example: Device# show app-hosting list	Displays the list of app-hosting services enabled on the device.

What to do next

The following is sample output from the **show iox-service** command on an ISR 4000 Series Router:

```
Device# show iox-service
```

```
Virtual Service Global State and Virtualization Limits:
```

```
Infrastructure version : 1.7
Total virtual services installed : 0
Total virtual services activated : 0
```

```

Machine types supported   : KVM, LXC
Machine types disabled   : none

Maximum VCPUs per virtual service : 6
Resource virtualization limits:
Name                      Quota      Committed   Available
-----
system CPU (%)            75         0           75
memory (MB)               10240     0           10240
bootflash (MB)           1000      0           1000
harddisk (MB)             20000     0           18109
volume-group (MB)        190768    0           170288

```

IOx Infrastructure Summary:

```

-----
IOx service (CAF)       : Running
IOx service (HA)       : Not Running
IOx service (IOxman)   : Running
Libvirtd               : Running

```

The following is truncated sample output from the **show iox-service** command on a Catalyst 3850 Series Switch:

```
Device# show iox-service
```

IOx Infrastructure Summary:

```

-----
IOx service (CAF)       : Running
IOx service (HA)       : Running
IOx service (IOxman)   : Running
Libvirtd               : Running

```

The following is sample output from the **show app-hosting list** command:

```
Device# show app-hosting list
```

```

App id                      State
-----
guestshell                   RUNNING

```

Managing the Guest Shell

You can start the Guest Shell container in IOS through Guest Shell commands.

Before you begin

IOx must be configured and running for Guest Shell access to work. If IOx is not configured, a message to configure IOx is displayed. Removing IOx removes access to the Guest Shell, but the rootfs remains unaffected.

Procedure

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: Device> enable	<ul style="list-style-type: none"> Enter your password if prompted.
Step 2	<ul style="list-style-type: none"> guestshell enable guestshell enable [VirtualPortGroup port-number guest-ip ip-address gateway gateway-ip netmask netmask [name-server ip-address]] Example: Device# guestshell enable Example: Device# guestshell enable VirtualPortGroup 0 guest-ip 192.168.35.2 gateway 192.168.35.1 netmask 255.255.255.0 name-server 10.1.1.1	Enables the Guest Shell service. or Enables connectivity to the front panel ports. Note <ul style="list-style-type: none"> The guestshell enable command without any arguments uses the management virtual routing and forwarding (VRF) instance for networking. When using VirtualPortGroups (VPGs) for front panel networking, the VPG must be configured first. The guest IP address and the gateway IP address must be in the same subnet. Front panel ports are not supported Cisco Catalyst 3650 Series Switches, Cisco Catalyst 3850 Series Switches, Cisco Catalyst 9300 Series Switches, and Cisco Catalyst 9500 Series Switches.
Step 3	guestshell run linux-executable Example: Device# guestshell run python	Executes or runs a Linux program in the Guest Shell. <ul style="list-style-type: none"> Python Version 2.7.11 is pre-installed on Catalyst 3650 and Catalyst 3850 Series Switches, and Python Version 2.7.5 is pre-installed on ISR 4000 Series Routers.
Step 4	guestshell run bash Example: Device# guestshell run bash	Starts a Bash shell to access the Guest Shell.
Step 5	guestshell disable Example: Device# guestshell disable	Disables the Guest Shell service.

	Command or Action	Purpose
Step 6	guestshell destroy Example: Device# <code>guestshell destroy</code>	Deactivates and uninstalls the Guest Shell service.

Enabling and Running the Guest Shell

The **guestshell enable** command installs Guest Shell. This command is also used to reactivate Guest Shell, if it is disabled.

When Guest Shell is enabled and the system is reloaded, Guest Shell remains enabled.



Note IOx must be configured before the **guestshell enable** command is used.

The **guestshell run bash** command opens the Guest Shell bash prompt. Guest Shell must already be enabled for this command to work.



Note If the following message is displayed on the console, it means that IOx is not enabled; check the output of the **show iox-service** command to view the status of IOx.

```
The process for the command is not responding or is otherwise unavailable
```

Disabling and Destroying the Guest Shell

The **guestshell disable** command shuts down and disables Guest Shell. When Guest Shell is disabled and the system is reloaded, Guest Shell remains disabled.

The **guestshell destroy** command removes the rootfs from the flash filesystem. All files, data, installed Linux applications and custom Python tools and utilities are deleted, and are not recoverable.

Accessing the Python Interpreter

Python can be used interactively or Python scripts can be run in the Guest Shell. Use the **guestshell run python** command to launch the Python interpreter in Guest Shell and open the Python terminal.



Note The **guestshell run** command is the IOS equivalent of running Linux executables, and when running a Python script from IOS, specify the absolute path. The following example shows how to specify the absolute path for the command:

```
Guestshell run python /flash/sample_script.py parameter1 parameter2
```


Configuration Examples for Guest Shell

Example: Managing the Guest Shell

The following example shows how to enable Guest Shell on a Catalyst 3850 Series Switch:

```
Device> enable
Device# guestshell enable

Management Interface will be selected if configured
Please wait for completion
Guestshell enabled successfully

Device# guestshell run python

Python 2.7.11 (default, Feb 21 2017, 03:39:40)
[GCC 5.3.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.

Device# guestshell run bash

[guestshell@guestshell ~]$

Device# guestshell disable

Guestshell disabled successfully

Device# guestshell destroy

Guestshell destroyed successfully
```

Sample VirtualPortGroup Configuration

When using the VirtualPortGroup interface for Guest Shell networking, the VirtualPortGroup interface must have a static IP address configured. The front port interface must be connected to the Internet and Network Address Translation (NAT) must be configured between the VirtualPortGroup and the front panel port.

The following is a sample VirtualPortGroup configuration:

```
Device> enable
Device# configure terminal
Device(config)# interface VirtualPortGroup 0
Device(config-if)# ip address 192.168.35.1 255.255.255.0
Device(config-if)# ip nat inside
Device(config-if)# no mop enabled
Device(config-if)# no mop sysid
Device(config-if)# exit
Device(config)# interface GigabitEthernet 0/0/3
Device(config-if)# ip address 10.0.12.19 255.255.0.0
```

```

Device(config-if)# ip nat outside
Device(config-if)# negotiation auto
Device(config-if)# exit
Device(config)# ip route 0.0.0.0 0.0.0.0 10.0.0.1
Device(config)# ip route 10.0.0.0 255.0.0.0 10.0.0.1
!Port forwarding to use ports for SSH and so on.
Device(config)# ip nat inside source static tcp 192.168.35.2 7023 10.0.12.19 7023 extendable
Device(config)# ip nat outside source list NAT_ACL interface GigabitEthernet 0/0/3 overload
Device(config)# ip access-list standard NAT_ACL
Device(config-std-nacl)# permit 192.168.0.0 0.0.255.255
Device(config-std-nacl)# exit
Device(config)# exit
Device#

```

Example: Guest Shell Usage

Example: Guest Shell Networking Configuration

For Guest Shell networking, the following configurations are required.

- Configure Domain Name System (DNS)
- Configure proxy settings
- Configure YUM or PIP to use proxy settings

Sample DNS Configuration for Guest Shell

The following is a sample DNS configuration for Guest Shell:

```

[guestshell@guestshell ~]$ cat/etc/resolv.conf
nameserver 192.0.2.1

```

Other Options:

```

[guestshell@guestshell ~]$ cat/etc/resolv.conf
domain cisco.com
search cisco.com
nameserver 192.0.2.1
search cisco.com
nameserver 198.51.100.1
nameserver 172.16.0.6
domain cisco.com
nameserver 192.0.2.1
nameserver 172.16.0.6
nameserver 192.168.255.254

```

Example: Configuring Proxy Environment Variables

If your network is behind a proxy, configure proxy variables in Linux. If required, add these variables to your environment.

The following example shows how to configure your proxy variables:

```
[guestshell@guestshell ~]$cat /bootflash/proxy_vars.sh
export http_proxy=http://proxy.example.com:80/
export https_proxy=http://proxy.example.com:80/
export ftp_proxy=http://proxy.example.com:80/
export no_proxy=example.com
export HTTP_PROXY=http://proxy.example.com:80/
export HTTPS_PROXY=http://proxy.example.com:80/
export FTP_PROXY=http://proxy.example.com:80/
guestshell ~] source /bootflash/proxy_vars.sh
```

Example: Configuring Yum and PIP for Proxy Settings

The following example shows how to use Yum for setting proxy environment variables:

```
cat /etc/yum.conf | grep proxy
[guestshell@guestshell~]$ cat /bootflash/yum.conf | grep proxy
proxy=http://proxy.example.com:80/
```

PIP install picks up environment variable used for proxy settings. Use sudo with -E option for PIP installation. If the environment variables are not set, define them explicitly in PIP commands as shown in following example:

```
sudo pip --proxy http://proxy.example.com:80/install requests
sudo pip install --trusted-host pypi.example.com --index-url
http://pypi.example.com/simple requests
```

The following example shows how to use PIP install for Python:

```
Sudo -E pip install requests
[guestshell@guestshell ~]$ python
Python 2.17.11 (default, Feb 3 2017, 19:43:44)
[GCC 4.7.0] on linux2
Type "help", "copyright", "credits" or "license" for more information
>>>import requests
```

Additional References for Guest Shell

Related Documents

Related Topic	Document Title
Python module	• <i>CLI Python Module</i>
Zero-Touch Provisioning	

MIBs

MB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	http://www.cisco.com/support

Feature Information for Guest Shell

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use the Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 4: Feature Information for Guest Shell

Feature Name	Release	Feature Information
Guest Shell	Cisco IOS XE Everest 16.5.1a	<p>Guest Shell is a secure container that is an embedded Linux environment that allows customers to develop and run Linux and custom Python applications for automated control and management of Cisco switches. It also includes the automated provisioning (Day zero) of systems. This container shell provides a secure environment, decoupled from the host device, in which users can install scripts or software packages and run them.</p> <p>In Cisco IOS XE Everest 16.5.1a, this feature was implemented on the following platforms:</p> <ul style="list-style-type: none">• Cisco Catalyst 9500 Series Switches

