Zero-Touch Provisioning

To address network provisioning challenges, Cisco introduces a zero-touch provisioning model.

The zero-touch model provides the following functionalities:

- Guest Shell—Is a secure container that is an embedded Linux environment that allows customers to develop and run 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.
- Application Hosting—Is an end-to-end application framework that provides application hosting capabilities for different application types on Cisco networking devices by using IOx.

Note In Cisco IOS XE Everest 16.5.1, Cisco Catalyst 3650 Series Switches and Cisco Catalyst 3850 Series Switches do not support application hosting, and uses the Guest Shell to develop and run custom Python applications.

This module describes the Zero-Touch Provisioning features.

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 Zero-Touch Provisioning

Guest Shell Overview

Guest Shell is a 64-bit application environment, and allows customers to develop and run custom Python applications for automated control and management of Cisco switches. It is automatically enabled in the system and can be accessed using the **guestshell enable** IOS command. Guest Shell is bundled with Python 2.7.11 or later. Using Guest Shell customers can install, update, and operate third party Linux applications (such as Puppet agent, Chef agent, Splunk, etc) in a secure environment on a network device.

The Guest Shell environment is intended for App or script hosting only, and network device resources are not available from this environment

To provide an open Linux shell environment, an additional Linux rootfs is packaged with the Cisco IOS XE image that can be installed on a flash file system or any persistent storage device as a Linux container. The container shares the Montavista kernel with the host (Cisco switches and routers) system. Users can access the Linux shell of this container and update scripts and software packages in the container rootfs. However, users within the container cannot modify the host file system and processes. The Guest Shell container shares the host's network namespaces.

Guest Shell container is managed using IOx components. IOx is Cisco's implementation of Fog Computing. 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. IOx is the framework that manages containers on Cisco IOS XE-supported platforms.

Because of limited resources and CPU difference on Catalyst 3650 and 3850 Series Switches, a variant of Guest Shell, called Guest Shell Lite is available for these platforms.

The following table provides the Guest Shell versions and the supported platforms:

Table 1: Guest Shell Versions and Supported Platforms

Guest Shell Version	Platform
Guest Shell Lite (limited LXC container)	Cisco Catalyst 3650 Series Switches
	Cisco Catalyst 3850 Series Switches
Guest Shell (LXC container)	Cisco ISR 4000 Series Integrated Services Routers

Guest Shell Container

The Guest Shell container allows users to run their scripts and apps on the system. The Guest Shell container on Intel platforms will be a Linux container (LXC) with a Centos 7.0 minimal rootfs. Python Version 2.7.11 is pre-installed in this container. Users can install Python Version 3.0 during runtime using the Yum utility in Centos 7.0.

The Guest Shell container on Advanced RISC Machines (ARM) and MIPS platforms will be Carrier Grade Edition (CGE) 7.0 container that has Python Version 2.7.11 pre-installed.

Applications running in third-party containers and the Guest Shell must comply with the UAPI provided by the host Linux kernel.

Container Rootfs Overview

The rootfs can be installed on a flash file sytem or any other persistent storage device as a Linux container.

The choice of rootfs for a platform depends on the CPU architecture and storage resources available on the platform. On X86 and ARM CPU architectures, preference is given to CentOS/Debian Linux rootfs. For other CPU architectures and platforms with limited resources, a Montavista-based rootfs with software built from Yocto system is available. Cisco provides RPM packages for software that can be dynamically installed in the Guest Shell. Cisco provided RPM is only for CentOS distribution.

Rootfs is packaged with Cisco IOS XE Everest 16.5.x image as a subpackage and the subpackage can be upgraded through **install** commands. For more information, see the *In Service Model Update* feature.

Table 2: Rootfs Supported by Platforms

Platform (CPU)	Rootfs/ Linux /distro	Main Contents
Catalyst 3850 Series Switches	Montavista CGE7	Busybox, SSH, and Python: pip install
ASR 1000 Series Aggregation Services Routers	CentOs	yum (yum install for third party apps), SSH, and Python: pip install

Container Security

Cisco provides security to ensure that users or apps in the Guest Shell do not compromise the host system. To isolate the container or Guest Shell from the host, usernamespaces are used on the host kernel. With usernamespace support, the container or Guest Shell runs as an unprivileged container.

Cisco also uses SELinux to contain and isolate attacks from arbitrary third-party applications that users may run in the container or the Guest Shell. SELinux is a third-party kernel Linux Security Module (LSM) that allows the isolation of applications/containers at a granular level.

Hardware Requirements for Guestshell

In Cisco Catalyst 3000 Series Switches, a single software bundle is supported for multiple stock keeping units (SKUs). The following table is to highlight that the image size is constrained to the bootflash of the smallest SKU. Therefore accounting for the supported deployment models, we capture the storage requirements below:

Table 3: Guest Shell Requirement

Deployment Scenario	Guest Shell Requirement
Switch compressed image size increase	18 MB

Deployment Scenario	Guest Shell Requirement
IOx component size in image	14 MB
Flash Required	(53 + 2 * 32) = 117 MB
Expanded CAF required content to Flash	53 MB

Table 4: Hardware Requirement

Platforms	Catalyst 3850 Series Switches	Catalyst 3000 Series Switches (12/24 SFP + Models)	Catalyst 3000 Series Switches (48 SFP + Models)
Switch image upgrade flash requirements	2x Image Size – baseline size - 593MB		
RAM sizing	4 G	4 G	8 G
Flash sizing	2G (1.5G available for use)	4G (3.5G available for use)	8G (7.5G available for use)
Remaining Flash post Guest Shell	138 MB	220 MB	579 MB

Accessing Guest Shell from a Device

Network administrators can use IOS commands to manage files and utilities in the Guest Shell. Guest Shell access is provided using the SSH service running in the Linux container on the loopback interface on the management virtual routing and forwarding (VRF) instance.

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 in IOS. This user is granted root access into the Linux container. When the user accesses the Guest Shell, an IOS authentication token is provided in the environment variable of the Guest Shell session. This token is used to access the Plug N Play (PnP) agent without using an username or a password. Commands sent in PnP requests are executed with same privilege that the user had when logged into the IOS terminal. To access Device Management Instrumentation (DMI) services from the Linux container, username and password are required.

Management Networking

Note Management networking is supported only on Cisco Catalyst 3650 Series Switches and Cisco Catalyst 3850 Series Switches.

Guest Shell allows applications to access the management network by default. To support zero-touch configuration through the management network, and to save the management IP address, the host management

network namespace, the management virtual routing and forwarding (VRF) instance, and VRF 2 namespace of the host Linux container is mounted onto the Guest Shell. Network administrators/users can switch to the Guest Shell inside the container to access the management network.

All management networking services must start in this VRF 2 namespace from the Guest Shell. However, port conflicts between services started inside the container and services started from the host device can occur. In case of port conflicts, a different port number must be used when the service is started from the container.

To enable management networking, the VRF 2 namespace must be bind mounted onto the corresponding Guest Shell rootfs, before the Guest Shell is activated. The mount point must be unmounted after the Guest Shell is deactivated.

Users cannot change the management VRF networking configurations from inside the container.

Note For platforms without a management port, a virtual port group can be associated with Guest Shell in the IOS configuration.

Guest Shell Packaging

Cisco application hosting framework (CAF) is an IOx Python process that manages virtualized and container applications running on devices. Guest Shell is a specific application for CAF that is installed and managed from IOS commands. The Guest Shell container is started by configuring the **iox** command.

The lifecycle of Guest Shell is managed by CAF, and for CAF to install the Guest Shell, the contents must be an IOx application package. The IOx application package for Guest Shell is in the TAR file format. The CAF-formatted tar file (guest_shell.tar) content for Guest Shell is packaged into the Cisco IOS XE Everest 16.5.x image.

During the initialization of Cisco devices, the Guest Shell package content is mounted onto the root file system of the host system (under the */tmp/sw/isos* or an equivalent directory), and the Guest Shell package is made available for CAF to install rootfs on the non-volatile bootflash. After the Guest Shell package is installed, if a newer version of the Guest Shell rootfs is available, CAF will install it from the image during a reload.

The following are the contents of the application TAR file:

- A package descriptor file, named package.yaml that is present in the root directory.
- Zero or one application configuration file, named package_config.ini. This file, if present, must be in the root directory.
- An application manifest, named package.mf that is present in the root directory.
- Zero or one tar.gz envelop that contains application artifacts, named artifacts.tar.gz. These artifacts can be binaries, application code, application libraries, virtual disks, rootfs and so on.

Guest Shell Logging

Console and Systemd Logs

Console logs are written to the */dev/console* folder inside the Guest Shell, and console logs are saved into tracelog memory buffers. When the buffer is full, it is written to file.

Syslog

Syslogs generated in the Guest Shell is written to */var/log* directories. These directories are mounted from the RAM file system like tmpfs. These log will be lost on system reload.

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. It is a Python-based framework for hosting user-deployed Linux applications on designated Cisco networking systems. 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. App is an application that is deployed in the IOx framework. 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.

Cisco Application Hosting Framework Overview

Note Cisco Catalyst 3650 Series Switches and Cisco Catalyst 3850 Series Switches do not support application hosting.

Cisco application hosting framework (CAF) is an IOx Python process that manages virtualized and container applications running on devices. CAF provides the following services:

- Launches designated applications (like a Cisco Guest Shell) in containers.
- Checks available resources (memory, CPU, and storage), and allocate and manage them.
- Provides support for console logging.
- Provides access to relevant services via REST APIs.
- Provides a CLI endpoint.
- Provides a TDL-enabled message path for CAF.
- Helps in the setup of platform-specific networking (packet-path) via virtual port group (VPG) interfaces and Linux Shared Memory Punt Interface (LSMPI) or LFTS-based interception.

To ensure security and stability, Linux containers are used to deploy signed applications, and Kernel-based Virtual Machine (KVM) containers are used to deploy applications trusted by users.

Applications to be deployed in the containers are packaged as TAR files. The configuration that is specific to the application to be deployed in the container is also packaged as a TAR file. Applications, such as the Guest Shell, can be deployed to a system using a local store (for the TAR file), or via the Internet through Cisco Fog Services in conjunction with the Cisco Developer Network.

To manage IOx, a local Web Management Interface (Local Manager) and a multi-device remote application manager are available. The Local Manager management interface enables application life cycle management on the local IOx device. The remote manager can manage multiple IOx devices, and can perform bulk operations that manages the life cycle of an IOx application.

How to Configure Zero-Touch Provisioning

Managing IOx

Use the following commands to manage the IOx service:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **iox**
4. **exit**
5. **show app-hosting list**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device>`	Enables privileged EXEC mode. 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.

	Command or Action	Purpose
Step 5	**show app-hosting list** **Example:** `Device# show app-hosting list`	Displays a list of app-hosting services enabled on the device.

Managing the Guest Shell

You can start the Guest Shell container in IOS either through Guest Shell commands or from any IOS subsystem (such as, Plug N Play) that uses the Guest Shell.

SUMMARY STEPS

1. **enable**
2. **guestshell enable**
3. **guestshell enable** [**VirtualPortGroup** *port-number* **guest-ip** *ip-address* **gateway** *gateway-ip* **netmask** *netmask* [**name-server** *ip-address*]]
4. **guestshell run** *linux-executable*
5. **guestshell disable**
6. **guestshell destroy**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**guestshell enable** **Example:** `Device# guestshell enable`	Enables the Guest Shell service.
Step 3	**guestshell enable** [**VirtualPortGroup** *port-number* **guest-ip** *ip-address* **gateway** *gateway-ip* **netmask** *netmask* [**name-server** *ip-address*]] **Example:** `Device# guestshell enable VirtualPortGroup 0 guest-ip 198.51.100.1 gateway 192.168.2.1 netmask 255.255.255.0 name-server 10.1.1.1`	(Optional) Enables connectivity to the front panel ports. • In IOS XE Everest 16.5.1, this comamnd is supported only on Cisco ASR 1000 Series Aggregation Services Routers and Cisco ISR 4000 Series Integrated Services Routers.
Step 4	**guestshell run** *linux-executable*	Executes or runs a Linux program in the Guest Shell.

	Command or Action	Purpose
	Example: `Device# guestshell run python`	• The supported Python version is 2.7.11.
Step 5	**guestshell disable** **Example:** `Device# guestshell disable`	Disables the Guest Shell service.
Step 6	**guestshell destroy** **Example:** `Device# guestshell destroy`	Deactivates and uninstalls the Guest Shell service.

Enabling and Running the Guest Shell

The **guestshell enable** command installs Guest Shell from a Guest Shell software package. By default, the package embedded in the system image is used for the installation. If Guest Shell is not installed, this command extracts the Guest Shell image from the system image. 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 should be configured before the **guestshell enable** command is used.

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

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.

Starting Guest Shell Using Python

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

The following example shows how to enable Python:

```
Switch# python
Python 2.7.11 (default, March 16 2017, 16:50:55)
[GCC 4.7.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>>>
```

Additional References for Zero-Touch Provisioning

Related Documents

Related Topic	Document Title
Python Scripting	
REST API on Cisco CSR 1000V Series Routers	Enabling Management by REST API

Standards and RFCs

Standard/RFC	Title
RFC 7950	*The YANG 1.1 Data Modeling Language*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. 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. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for Zero-Touch Provisioning

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 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 5: Feature Information for Zero-Touch Provisioning

Feature Name	Release	Feature Information
Application Hosting	Cisco IOS XE Everest 16.5.1	Application Hosting is an end-to-end application framework that provides application hosting capabilities for different application types on Cisco network platforms by using IOx In Cisco IOS XE Everest 16.5.1, this feature is implemented on Cisco ISR 4000 Series Integrated Services Routers. The following command was introduced: **guestshell**.
Zero-Touch Provisioning	Cisco IOS XE Everest 16.5.1	To address network provisioning challenges, Cisco introduces a zero-touch provisioning model. The zero-touch model provides a secure container called Guestshell and Application Hosting. In Cisco IOS XE Everest 16.5.1, this feature is implemented on the following platforms: • Cisco ISR 4000 Series Integrated Services Routers The following command was introduced: **iox**.