Linux Device Management

VMware Workspace ONE UEM

You can find the most up-to-date technical documentation on the VMware website at:

https://docs.vmware.com/

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Contents

Workspace ONE UEM on Linux

1

Use Workspace ONE UEM to manage and secure your enterprise Linux devices. The Workspace ONE UEM console gives you tools and features to manage the entire lifecycle of Linux devices.

The flexibility of the Linux operating system makes it a preferred platform for a wide range of uses, including developer workstations, Raspberry Pi devices, and many IoT devices. With Workspace ONE UEM, you can build on the flexibility and ubiquity of Linux devices and manage them along-side your other enterprise devices in one central location.

This chapter includes the following topics:

Requirements for Workspace ONE UEM on Linux

Workspace ONE UEM is compatible with all distributions of Linux running on x86_64, ARM5, or ARM7 architectures, although not all features might be available on every distribution. Make sure that your system meets the Workspace ONE UEM version and network requirements before you deploy your Linux devices.

Linux Device Requirements

You can enroll devices running any distribution of Linux running on x86_64, ARM5, or ARM7 architectures into Workspace ONE UEM.

- Installers are created for specific distributions and architectures. Ensure that you are using the correct installer for your use case.
- To run Workspace ONE Intelligent Hub as a system service, the device must be running System D or System V.
- Configurations using Workspace ONE profiles requires a Puppet agent (open source). When running a Debian-based (deb) or Red Hat-based (rpm) system, the puppet agent installs automatically with Hub. For other systems, or when using the Tarball method of installation, install the Puppet agent manually prior to Workspace ONE enrollment. For more information, see the Installation section.

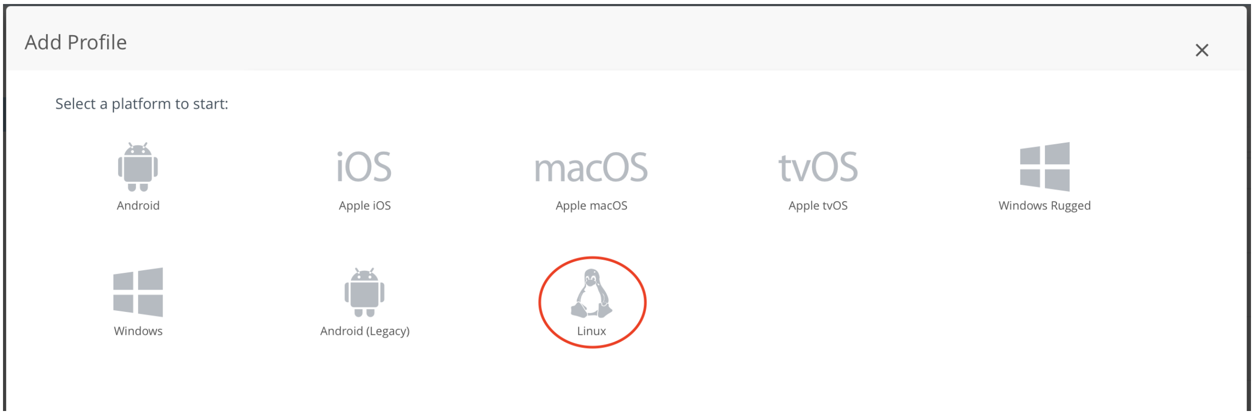
Workspace ONE UEM Requirements

Workspace ONE UEM for Linux is generally available. However, it might not be available in your environment. Workspace ONE UEM for Linux is available in all shared SaaS environments and deploys to dedicated SaaS environments. Workspace ONE UEM for Linux is not available for on-premises installations of Workspace ONE UEM.

- You must deploy the Workspace ONE Intelligent Hub for Linux v22.06 or later. Previous versions including v21.10, v21.01, and v1 of the Workspace ONE Intelligent Hub for Linux do not support the features described in this guide.
- To use Linux Management, Workspace ONE UEM 2206 or later with Control Plane implemented are both required.

Before attempting to enroll a Linux device, determine if Linux management is enabled in your Workspace ONE UEM environment. The easiest way to validate this is to create a new profile. Navigate to **Resources > Profiles & Baselines> Profiles> Add> Add Profile** to see if Linux is an available option.

If the Linux icon shows on the Add Profile page, then Linux management is enabled. You can enroll a Linux device. If there is no Linux icon, then your environment does not allow Linux device enrollment. To read more about environments upgrade scheduling, see the following KB article.



Enroll Your Linux Devices 2

Enroll your Linux devices to establish a secure connection between the devices and Workspace ONE UEM. This topic describes web-based enrollment and command line enrollment for Linux devices.

Unlike other platforms, the Workspace ONE Intelligent Hub for Linux does not have a graphical user interface and requires the command line to install, enroll, and interact with it on the device itself. For hub-based enrollment, you need a working knowledge of the Linux command line.

Enrollment Methods

There are two ways to initiate enrollment for Linux devices:

- Hub-based Enrollment - Enroll a Linux device using the command line in `ws1HubUtil`, that is included as part of the Hub installation.

  Note Command line enrollment does not support SAML authentication, or directory lookup. The enrollment user must exist in the Workspace ONE UEM console (basic or pre-synced Directory user accounts).

- Web-based Enrollment - Enroll a Linux device with a user-initiated, web-based enrollment wizard.

  Note Web enrollment is the only enrollment method that supports integrated user authentication, such as Workspace ONE Access or SAML authentication.

  Neither enrollment method supports advanced and single-user staging enrollments. Single-user staging enrollment is where an admin enrolls on behalf of a user or enrolls and waits for a user to enter credentials.

Prerequisites

Before enrolling a Linux endpoint, you need the Workspace ONE UEM Device Services URL, organization Group ID, and the enrollment user's user name and password.

Ensure that you are using the correct installer for your device and distribution. Installers are created for specific distributions and architectures.

For more information before enrollment, see Command-line Utilities for Workspace ONE Intelligent Hub on Linux.

Enroll Linux Devices using Web Enrollment

Web enrollment makes enrolling devices easier by not requiring manual Workspace ONE Intelligent Hub downloads. Because credentials are entered in the browser, web enrollment supports integrated and modern user authentication (such as Workspace ONE Access, SAML, and others). To use web enrollment, send users a URL.

Currently, Chrome and Firefox are the only supported browsers for web enrollment. Web enrollment does not support the Tarball installer.

Procedure

1. In Chrome or Firefox, go to the enrollment URL. This URL is built into your environment and is accessible by adding "/enroll/welcome" to your active environment.

   For example, a Web Enrollment URL looks like https://cn135.awmdm.com/enroll/welcome.

2. Enter the Group ID, Username, and Password.

3. To download the Workspace ONE Intelligent Hub, accept the Terms of Use.

   Configure the Terms of Use in Workspace ONE UEM under **Settings > System > Terms of Use**.

4. Select the appropriate installer for your device.

5. To download the installer and the enrollment package, select **Next**.

   - Use the Deb Installer for all Debian based distributions including Ubuntu, Debian, Raspbian, and others.
   - Use the RPM Installer for any Red Hat based distributions including RHEL, CentOS, Fedora, SUSE, and others.

6. After downloading the package to your device, close your browser.

   Important: When the browser is closed, a cookie is written with the enrollment credentials for the device. The Hub fails to complete enrollment if the browser is not closed.

7. Run the command for installing the package using the command line.

   For the Deb package on Ubuntu:

   ```
   $ sudo apt install "./Downloads/com.airwatch.linux.agent.amd64.deb"
   ```

   Note: Do not use the dpkg to install the Hub. Prerequisites are not installed and Workspace ONE UEM does not work properly if dpkg is used.

   For RPM package on Fedora:

   ```
   $ sudo dnf install ./com.airwatch.linux.agent.amd64.rpm
   ```

   For RPM package on OpenSUSE:

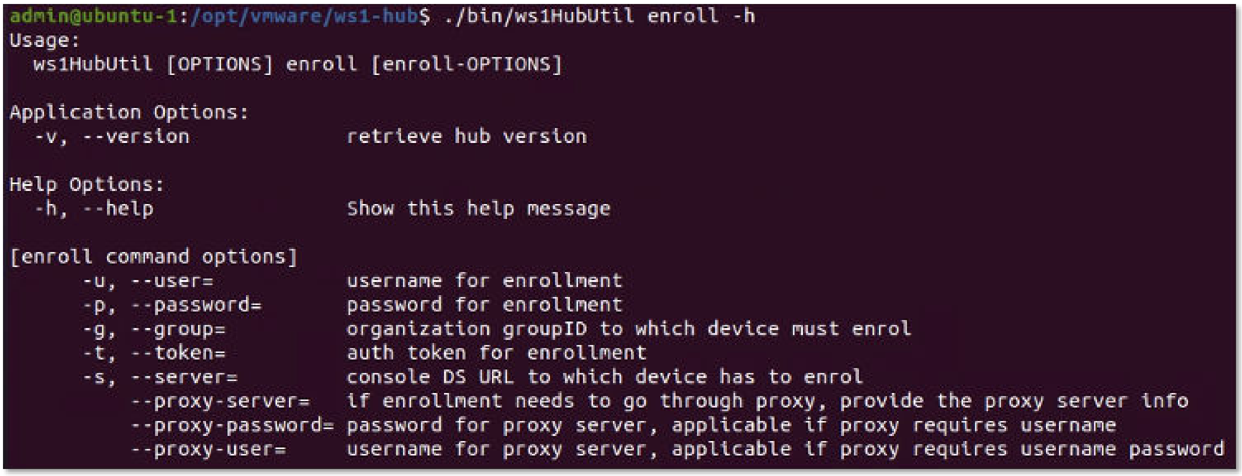
   ```
   $ sudo zypper install ./com.airwatch.linux.agent.amd64.rpm
   ```

This installed package takes care of prerequisites such as Ruby and Puppet, installing the Workspace ONE Intelligent Hub and ws1HubUtil, and enrolling the device into Workspace ONE UEM using the credentials entered in the wizard.

Enroll Linux Devices using the Command Line

Command line enrollment is a two-step process. First, you install the Intelligent Hub. Then, you enroll devices with the `ws1HubUtil` command. You can fully script the enrollment in a single command or prompt the user to enter enrollment information.

You can also enroll with a token using the `--token` argument and use it in the `--group` argument. Enter the username and the password. You can leave those text boxes blank when enrolling with a token. For more details on the enrollment arguments and options available, or run the command `./ws1HubUtil enroll -h`, see the ws1HubUtil Enroll Command section of Command-line Utilities for Workspace ONE Intelligent Hub on Linux.



```
admin@ubuntu-1:/opt/vmware/ws1-hub$ ./bin/ws1HubUtil enroll -h
Usage:
  ws1HubUtil [OPTIONS] enroll [enroll-OPTIONS]

Application Options:
  -v, --version             retrieve hub version

Help Options:
  -h, --help                Show this help message

[enroll command options]
      -u, --user=           username for enrollment
      -p, --password=       password for enrollment
      -g, --group=          organization groupID to which device must enrol
      -t, --token=          auth token for enrollment
      -s, --server=         console DS URL to which device has to enrol
          --proxy-server=   if enrollment needs to go through proxy, provide the proxy server info
          --proxy-password= password for proxy server, applicable if proxy requires username
          --proxy-user=     username for proxy server, applicable if proxy requires username password
```

To run as a system service, the device must be running SystemD or System V init for Hub. Custom configurations require a Puppet agent. The agent automatically installs with the Workspace ONE Intelligent Hub if you are running a Debian-based (deb) or Red Hat-based (rpm) system. For other systems, manually install Ruby and the Puppet agent prior to Workspace ONE enrollment.

Procedure

1 Download the Workspace ONE Intelligent Hub for Linux to your intended device. The downloaded file must correspond to the targeted processor architecture and distribution. The agent is available as deb, rpm, or tgz packages. Download the agent directly or transfer it to your Linux device using USB or SSH.

Retrieve installers from the following locations:

Architecture	Debian Based	Red Hat Based	Other (Tarball)
x86_64	amd64-22.06.0.7.deb	amd64-22.06.0.7.rpm	amd64-22.06.0.7.tgz
ARM5	arm5-22.06.0.7.deb	arm5-22.06.0.7.rpm	arm5-22.06.0.7.tgz
ARM7	arm7-22.06.0.7.deb	arm7-22.06.0.7.rpm	arm7-22.06.0.07.tgz

To access current and previous installers, see the Workspace ONE install packages.

2. Run the Workspace ONE Intelligent Hub client installer with root privileges.

   Example:

   Deb package on Ubuntu:

   ```
   $ sudo apt install "/tmp/workspaceone-intelligent-hub-amd64-21.10.0.1.deb"
   ```

   Note: Do not use the dpkg to install the Hub. Prerequisites are not installed and Workspace ONE UEM does not work properly if dpkg is used.

   RPM package on Fedora:

   ```
   $ sudo dnf install workspaceone-intelligent-hub-amd64-22.6.0.7.rpm
   ```

   RPM package on openSUSE:

   ```
   $ sudo zypper install workspaceone-intelligent-hub-amd64-22.06.0.7.rpm
   ```

   Tarball (any other Linux distribution):

   ```
   Extract the Package using: $ tar xvf workspaceone-intelligent-hub-<arch>.22.06.0.7.tgz
   ```

   ```
   Install the Package using: $ sudo ./install.sh
   ```

   Before installing the Intelligent Hub when using Tarball, install Ruby and the Puppet agent manually.

3. Enroll your device in Workspace ONE UEM after the installation by using the `ws1HubUtil`.

4. Change the directory to the Hub binary directory under the installation directory by using `$ cd /opt/vmware/ws1-hub/bin`.

5. Send a user's enrollment details using a single command or follow enrollment prompts.

   - Single command (include enrollment arguments in order):

     ```
     $ sudo ./ws1HubUtil enroll --server https://host.com --user <username> --password
     <password> --group <organization group id>
     ```

   - Enrollment Prompts (without more arguments):

     ```
     $ sudo ./ws1HubUtil enroll
     ```

6 After a successful enrollment, the Linux device is listed in the Workspace ONE UEM console.

This chapter includes the following topics:

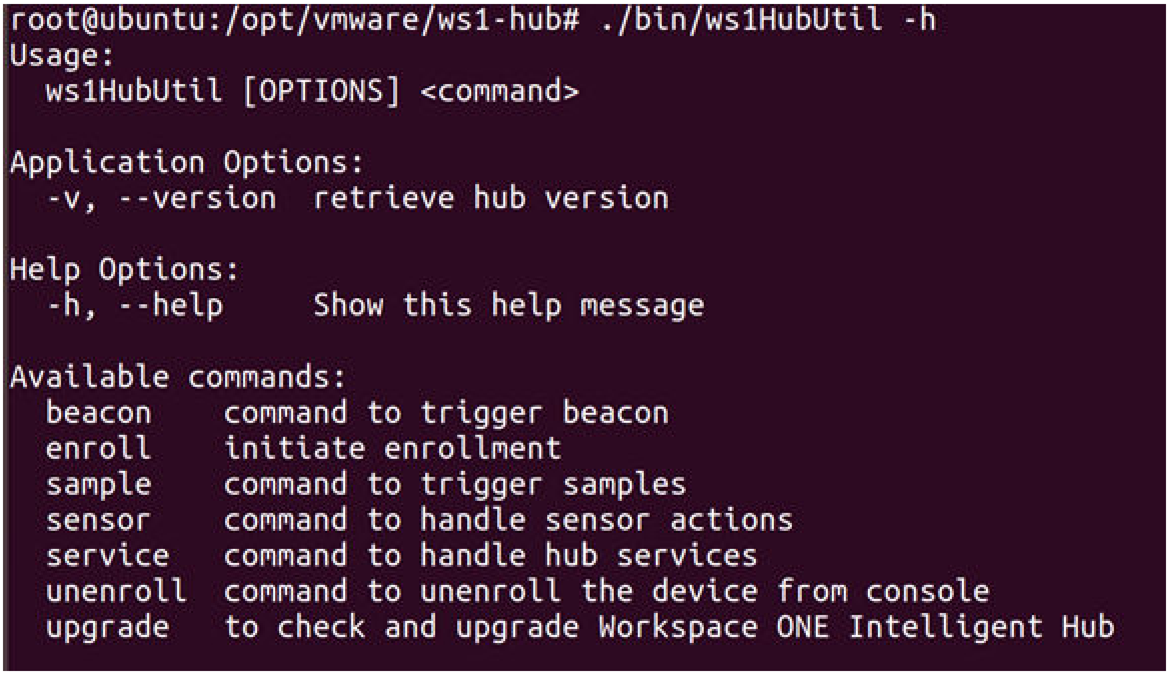
- Command-line Utilities for Workspace ONE Intelligent Hub on Linux

Command-line Utilities for Workspace ONE Intelligent Hub on Linux

Use these command-line utilities to expedite the deployment of Workspace ONE Intelligence Hub on your Linux devices. The `ws1HubUtil` application is located at the agent binary directory under the installation directory: /opt/vmware/ws1-hub/bin.

The `ws1HubUtil` installed on the Linux device includes the following commands:

- Version
- Enroll
- Beacon
- Sample
- Sensor
- Service
- Upgrade
- Unenroll



```
root@ubuntu:/opt/vmware/ws1-hub# ./bin/ws1HubUtil -h
Usage:
  ws1HubUtil [OPTIONS] <command>

Application Options:
  -v, --version  retrieve hub version

Help Options:
  -h, --help     Show this help message

Available commands:
  beacon    command to trigger beacon
  enroll    initiate enrollment
  sample    command to trigger samples
  sensor    command to handle sensor actions
  service   command to handle hub services
  unenroll  command to unenroll the device from console
  upgrade   to check and upgrade Workspace ONE Intelligent Hub
```

Version: The version argument prints the hub installed version.

`./ws1HubUtil –version` or `./ws1HubUtil -v`

Enroll: The enroll command handles the hub native enrollment process.

```
./ws1HubUtil enroll --user xyz --password xyz --group xyz --server https://<host>.com
```

Table 2-1. Supported Command-line Arguments for the ws1HubUtil Enroll Command

Command-line argument	Short Name	Value	Description	Comments
--user	-u	Enrollment user string	User credentials generated from console.	Prompts you to enter the details if the command line argument is not entered.
--password	-p	Password String	Credentials generated from console.	Prompts you to enter the details if the command line argument is not entered.
--group	-g	Organization group String	Organization groupID to which device must enroll.	Prompts you to enter the details if the command line argument is not entered.
--token	-t	Enrollment Token	Used for token based enrollment	Used if OG enrollment type is set for token. Token can also be passed as --group field.
--server	-s	Server String	**Fully qualified** Workspace ONE UEM console URL to which the device has to enroll. The URL is typically the device services URL not necessarily the console URL. For example, https://ds135.awmdm.com.	Prompts you to enter the details if the command line argument is not entered.
--proxy-server	N/A	Proxy server info	Use during enrollment if hub needs to use proxy info (optional)	Provide the proxy server info using this argument. If enrollment goes through a proxy,

Table 2-1. Supported Command-line Arguments for the ws1HubUtil Enroll Command (continued)

Command-line argument	Short Name	Value	Description	Comments
--proxy-user	N/A	Proxy username	Username for the proxy (optional)	Applicable if --proxyserver is provided and if proxy requires username and password.
--proxy-password	N/A	Proxy password	Password for the proxy (optional)	Applicable if --proxyserver is provided and if proxy requires username and password.

Beacon: The beacon command notifies the hub scheduler to trigger the beacon (heartbeat) immediately.

`./ws1HubUtil` beacon

Sample: The sample command notifies the hub scheduler to trigger a sample immediately. By default, the hub collects and sends all the samples. Also, a customized sample can be triggered. The allowed sample types are - system, network, certificate, profile, or all.

`./ws1HubUtil sample` [will trigger all sample].

Or

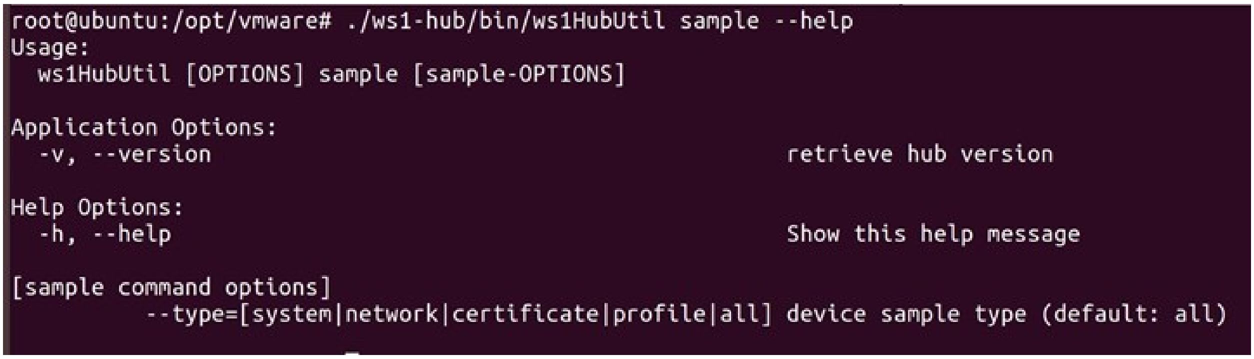
`./ws1HubUtil sample --type` [system], [network], [certificate], [profile], or [all].

Table 2-2. Supported Command-line Arguments for the Agent Utility

Command-line argument	Value	Description	Comments
--type	system | network | certificate | profile | all	Notify scheduler to trigger sample immediately	Queues a sample job to the hub scheduler with the specified sample type.

Sample Type	Samples Collected
system	System | memory | device capability
network	Network adapter | WLAN
certificate	certificate
profile	profile
all	All of the sample types

Figure 2-1.

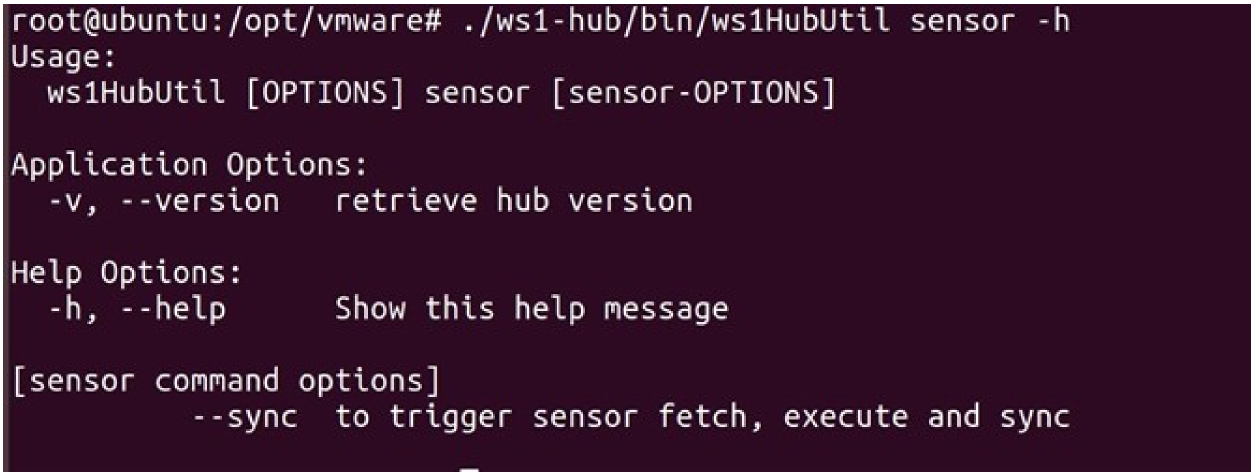


```
root@ubuntu:/opt/vmware# ./ws1-hub/bin/ws1HubUtil sample --help
Usage:
  ws1HubUtil [OPTIONS] sample [sample-OPTIONS]

Application Options:
  -v, --version                                         retrieve hub version

Help Options:
  -h, --help                                            Show this help message

[sample command options]
          --type=[system|network|certificate|profile|all] device sample type (default: all)
```

Sensor: The Sensor command notifies hub scheduler to trigger a Workspace ONE sensor sync immediately. The sync fetches the latest Workspace ONE sensors, runs periodic Workspace ONE sensors, and transmits the latest values from the device to the Workspace ONE UEM console.

```
./ws1HubUtil sensor --sync
```

Figure 2-2.



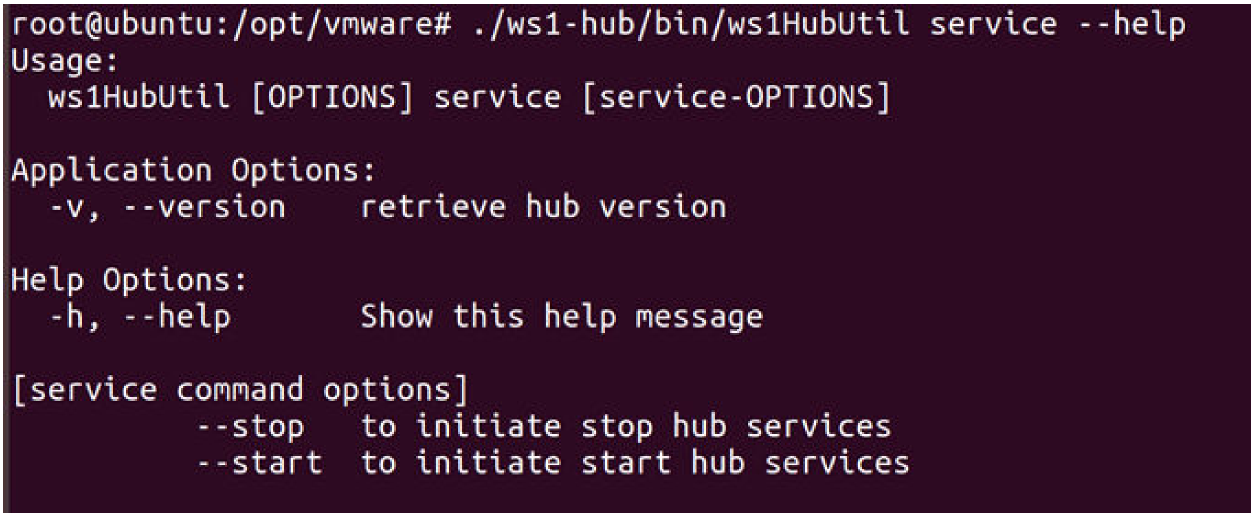
```
root@ubuntu:/opt/vmware# ./ws1-hub/bin/ws1HubUtil sensor -h
Usage:
  ws1HubUtil [OPTIONS] sensor [sensor-OPTIONS]

Application Options:
  -v, --version   retrieve hub version

Help Options:
  -h, --help      Show this help message

[sensor command options]
          --sync  to trigger sensor fetch, execute and sync
```

Service: The Service command provides the option to either start or stop hub services running on the device.

```
./ws1HubUtil service [--start] or [--stop]
```

Command options long name	value	description	comments
--start	N/A	Starts hub services	
--stop	N/A	Stops hub services	

Figure 2-3.



```
root@ubuntu:/opt/vmware# ./ws1-hub/bin/ws1HubUtil service --help
Usage:
  ws1HubUtil [OPTIONS] service [service-OPTIONS]

Application Options:
  -v, --version    retrieve hub version

Help Options:
  -h, --help       Show this help message

[service command options]
          --stop   to initiate stop hub services
          --start  to initiate start hub services
```

Unenroll: The Unenroll command sends a request to the Workspace ONE UEM console to unenroll the device. This command requires connectivity to Workspace ONE UEM to execute. If the command is successful, the Workspace ONE UEM console marks the device as unenrolled and sends a successful response back to the device. After the response, the Hub is uninstalled on the device. If not successful, then the device logs captures the failure and the device remains enrolled.

```
./ws1HubUtil unenroll
```

If preferred, you can manually uninstall the device side. Unenroll the device before the uninstall command is used device side.

Examples:

Debian:

```
$ sudo apt remove workspaceone-intelligent-hub
```

RPM for Fedora:

```
$ sudo zypper remove workspaceone-intelligent-hub
```

Tarball:

```
$ sudo /opt/Workspace-ONE-Intelligent-Hub/uninstall.sh
```

Upgrade: The upgrade command queues an upgrade hub check job with the hub scheduler. The job checks if a newer version of Workspace ONE Intelligent Hubis available. If a newer version is available, the latest package is fetched and the Hub is upgraded.

```
./ws1HubUtil upgrade
```

Linux Device Management

3

After your devices are enrolled and configured, manage these devices using the Workspace ONE UEM console. The management tools and functions enable you to keep an eye on your devices and remotely perform administrative functions.

While you can manage all your devices from the UEM console, the reporting details and available actions for enrolled devices may vary based on your deployment type and device platform.

The Device Dashboard is a searchable, customizable view where you use to the filter and find specific devices. This feature makes it easier to perform administrative functions on a particular set of devices.

The Device List View displays all devices currently enrolled in your Workspace ONE UEM environment and their status. You can filter the list view specific to Linux and see how devices are being managed at a glance.

The Device Details page provides device-specific information such as hardware details, profiles, and network details. You can also perform remote actions on the device from the Device Details page that are platform-specific.

This guide takes you through some specifics related to the management of Linux devices. For further information on any of these functions, see Managing Devices in the Workspace ONE UEM Documentation.

This chapter includes the following topics:

- Device Dashboard
- Device List View
- Linux Device Details Page

Device Dashboard

As devices are enrolled, you can manage them from the **Device Dashboard** in Workspace ONE UEM powered by AirWatch.

The **Device Dashboard** provides a high-level view of your entire fleet and allows you to act on individual devices quickly.

You can view graphical representations of relevant device information for your fleet, such as device ownership type, compliance statistics, and platform breakdowns. The Device Dashboard includes the following sections:

- **Security** – View the top causes of security issues in your device fleet. Selecting any of the doughnut charts displays a filtered **Device List** view comprised of devices affected by the selected security issue. If supported by the platform, you can configure a compliance policy to act on these devices.
  - **Compromised** – The number and percentage of compromised devices (jailbroken or rooted) in your deployment (Linux currently does not support this)
  - **No Passcode** – The number and percentage of devices without a passcode configured for security. (Linux currently does not support this)
  - **Not Encrypted** – The number and percentage of devices that are not encrypted for security. (Linux currently does not support this)
- **Ownership** – View the total number of devices in each ownership category. Selecting any of the bar graph segments displays a filtered **Device List** view comprised of devices affected by the selected ownership type.
- **Last Seen Overview/Breakdown** – View the number and percentage of devices that have recently communicated with the Workspace ONE UEM MDM server. For example, if several devices have not been seen in over 30 days, select the corresponding bar graph to display only those devices. You can then select all these filtered devices and send them a message requesting that they check in.
- **Platforms** – View the total number of devices in each device platform category. Selecting any of the graphs displays a filtered **Device List** view comprised of devices under the selected platform.
- **Enrollment** – View the total number of devices in each enrollment category. Selecting any of the graphs displays a filtered **Device List** view comprised of devices with the selected enrollment status.
- **Operating System Breakdown** – View devices in your fleet based on operating system. There are separate charts for each supported OS. Selecting any of the graphs displays a filtered **Device List** view comprised of devices running the selected OS version. (Linux currently does not support this)

Device List View

Use the Device List View in Workspace ONE UEM to see a full listing of devices in the currently selected organization group.

The Last Seen column displays an indicator showing the number of minutes elapsed since the device has checked-in. The indicator is red or green, depending on how long the device is inactive. The default value is 480 minutes (8 hours). You can customize this value by navigating to **Groups & Settings** > **All Settings** > **Devices & Users** > **General** > **Advanced** and change the Device Inactivity Timeout (min) value.

Note For Linux devices, the version number in the Device List View is the kernel version for the enrolled device, not the version of the distribution.

Select a device-friendly name in the General Info column at any time to open the details page for that device. A Friendly Name is the label you assign to a device to help you differentiate devices of the same make and model.

To review activity based on specific information, sort by columns and configure information filters. For example, sort by the Enrollment column to see any devices that are currently unenrolled from Workspace ONE UEM. Search across all devices for a friendly name or the user name to isolate one device or user.

Device List View Action Button Cluster

With one or more devices selected in the Device List View, you can perform common actions with the action button cluster including Query, and other actions accessed through the More Actions button.

Available Device Actions vary by platform, device manufacturer, model, enrollment status, and the specific configuration of your Workspace ONE UEM console. The following options are available:

- **Query** - Submit an on-demand request for a device to send updated sample data back to the Console.
- **Enterprise Wipe** - Unenroll and remove any Wi-Fi and credentials that Workspace ONE UEM pushed down to the device and then ultimately removes the Workspace ONE Intelligent Hub from the device. This option does not remove any custom configuration profiles that were sent to the device, but it runs any defined removal manifests. See Custom Configuration profiles for more information.
- **Manage Tags** - View the currently assigned device tags and see a list of tags available to be assigned with the Manage Tags screen.
- **Assign Tags** - Assign a customizable tag to a device, which can be used to identify a special device in your fleet.
- **Change Organizational Group** - Change the device's home organization group to another existing OG. Includes an option to select a static or dynamic OG.
- **Change Ownership** - Change the Ownership setting for a device, where applicable. Choices include Corporate-Dedicated, Corporate-Shared, Employee Owned and Undefined.
- **Delete Device** - If a device is still enrolled, this option issues an unenroll command before removing the entry from Workspace ONE UEM.

Customize Device List View Layout

Display the full listing of visible columns in the Device List view by selecting the Layout button and select the Custom option. This view enables you to display or hide Device List columns per your preferences.

There is also an option to apply your customized column view to all administrators at or below the current organization group (OG). For example, hide Asset Number from the Device List views of the current OG and of all the OGs underneath.

Once all your customizations are complete, select the Accept button to save your column preferences and apply this new column view. To update your column display preferences, select the Layout button settings at any time.

Some notable device list view custom layout columns include the following:

- SSID (Service Set Identifier or Wi-Fi network name)
- Wi-Fi MAC Address
- Wi-Fi IP Address
- Public IP Address

Exporting List View

To save an XLSX or CSV (comma-separated values) file of the entire Device List View for viewing and analyzing with Microsoft Excel, select the Export button. If you have a filter applied to the Device List View, then the exported listing reflects the filtered results.

Search in Device List View

Search for a single device for quick access to its information and take remote action on the device.

Run a search:

1. Navigate to **Devices > List View**.
2. Select the Search List bar.
3. Enter a user name, device-friendly name, or other device-identifying element. This action initiates a search across all devices, using your search parameter, within the current organization group and all child groups.

Linux Device Details Page

The Linux Device Details page in the Workspace ONE UEM console shows options available for customizing your enrolled Linux devices. Use the Device Details page to review and modify user and device actions.

Device Details

You can access Device Details by selecting a device's Friendly Name from the Device List View, using one of the Dashboards, or with any of the search tools.

From the Device Details page, you can access device information broken into different tabs. Each tab contains related device information, which can vary depending on your Workspace ONE UEM deployment.

- **Summary** - View general statistics such as enrollment status, compliance, last seen, platform/model/OS, organization group, serial number, storage capacity, and physical memory. The Security Panel on the Summary page reports if a Linux device is encrypted and if a firewall is enabled.
  - **Encryption Detection** - Workspace ONE UEM only detects devices encrypted with Linux Unified Key Setup (LUKS).
  - **Firewall Detection** - Workspace ONE UEM shows enabled when firewall or UFW services are running on the device.
- **Profiles** - Displays all profiles assigned to the selected device both installed (active) and assigned (inactive).
- **Apps** - Displays a list of installed desktop apps along with the app status, installation status, and assignment status.
- **Sensors** – View all sensors assigned to the selected device. This data includes the name, value, and last executed date.
- **User** - Access details about the user of a device and the status of the other devices enrolled to this user.
- **Network** – View current network (Wi-Fi) status of a device.
- **Security** - View the last received security information statuses from the device. The Security tab shows if a device is enrolled, if profiles are installed, and the status of certificates. The Security tab also shows the Disk Encryption Status for a device. Passcode and Applications, while shown on the security tab, are not supported for Linux.
- **Notes** - View and add notes regarding the device. For example, note the shipping status or if the device is in repair and out of commission.
- **Certificates** – Identify device certificates by name and issuer. This tab also provides information about certificate expiration.
- **Terms of Use** - View a list of End User License Agreements (EULAs) which have been accepted during the device enrollment.
- **Troubleshooting** - View Event Log and Commands logging information.
- **Status History** - View history of device in relation to enrollment status.
- **Attachments** - Use this storage space on the server for screenshots, documents, and links for troubleshooting and other purposes without taking up space on the device itself.

The More Actions drop-down on the Device Details page is similar to same function on the list view, but with extra features. Perform remote actions over the air to the selected device using these actions. The actions available vary depending on factors such as the device platform, Workspace ONE UEM console settings, and enrollment status.

- **Query** - This option submits an on-demand request for a device to send an updated sample data back to the Console. Selecting the Query option submits a request for all of the following. To request one of the items, access the More Actions menu.
  - **Device Information** - Send an MDM query command to the device to return information on the device such as friendly name, platform, model, organization group, operating system version, and ownership status.
  - **Profiles** – Send an MDM query command to the device to return a list of the installed device profiles.
  - **Certificates** – Send an MDM query command to the device to return a list of the installed certificates.
  - **Sensors** – Send an MDM query command to the device to return updated Sensor values.
- **Enterprise Wipe** – This option unenrolls and removes any Wi-Fi and credentials that Workspace ONE UEM pushed down to the device and removes the Workspace ONE Intelligent Hub from the device. This option does not remove custom configuration profiles that are sent to the device, but it runs any defined removal manifests. See Custom Configuration profiles for more information.
- **Change Organization Group** – Change the home organization group for the device to another pre-existing OG. Includes an option to select a static or dynamic OG.
- **Manage Tags** – Assign a customizable tag to a device, which can be used to identify a special device in your fleet.
- **Edit Device** – Edit device information such as Friendly Name, Asset Number, Device Ownership and Device Category.
- **Delete Device** – Delete and unenroll a device from the Workspace ONE UEM console. This action performs an Enterprise Wipe and removes the device from the Workspace ONE UEM console.

Linux Profiles

4

Profiles are the primary means to manage devices. Configure profiles so that your Linux devices remain secure and configured to your preferred settings.

Think of profiles as the settings and rules that, when combined with compliance policies, help you enforce corporate rules and procedures. Profiles contain the settings, configurations, and restrictions that you want to enforce on devices.

A profile consists of the general profile settings and a specific payload. Profiles work best when they contain only a single payload.

Wi-Fi Profile for Linux

Configuring a Wi-Fi profile lets devices connect to corporate networks, even if they are hidden, encrypted or password protected.

Procedure

1. Navigate to **Resources** > **Profiles & Baselines** > **Profiles** > **Add** > **Add Profile** > **Linux**.
2. Configure the General Settings for the profile as appropriate.
3. Select the Wi-Fi payload.
4. Configure Wi-Fi settings, including:

Setting	Description
Service Set Identifier	Provide the name of the network.
Hidden Network	Indicate if the Wi-Fi network is hidden.
Set as Active Network	Indicate if the device connects to the network with no end-user interaction.

Setting	Description
Security Type	Specify the access protocol used and certificate are requirements. Required fields depend on the selected security type. If None or WPA/WPA 2 are selected, then the Password field displays. If WPA/WPA 2 Enterprise is selected, then the Protocols and Authentication fields display. Protocols - Use Two Factor Authentication SFA Type Authentication - Identity Anonymous Identity Username Password Identity Certificate Root Certificate
Password	To connect to the network, provide the required credentials for the device. The password field displays when WPA/WPA 2 is selected from the Security Type field.
Proxy Type	To configure the Wi-Fi proxy settings, enable Proxy Type.
Proxy Server	Enter the hostname or the IP address for the proxy server.
Proxy Server Port	Enter the port for the proxy server.
Exclusion List	To exclude from the proxy, enter the hostnames. Hostnames entered here are not routed through the proxy. Use the * as a wildcard for the domain. For example: *.vmware.com or *vmware.com.

5 Select Save and Publish.

Credential Profile for Linux

To protect corporate assets and for greater security, implement digital certificates. To implement digital certificates you must define a certificate authority, then configure a Credentials payload alongside your Wi-Fi payload. Each payload has settings for associating the certificate authority defined in the Credentials payload.

Procedure

1 Navigate to **Resources** > **Profiles & Baselines** > **Profiles** > **Add** > **Add Profile** > **Linux**.

2 Configure the profile's General settings as appropriate.

3 Select the Credentials profile.

4 Select Configure.

5 To select either Upload or Defined Certificate Authority for the Credential Source, use the drop-down menu. The remaining profile options are source-dependent. If you select Upload, then you must enter a Credential Name and upload a new certificate. If you select Defined Certificate Authority, then you must choose a predefined Certificate Authority and a template.

6 Select Save and Publish.

Custom Configuration Profile

The Custom Configuration payload can be used to configure your Linux devices with features that the Workspace ONE UEM console does not currently support through its native payloads. This payload currently uses open source Puppet for this configuration, so nothing other than the free Puppet agent installed on the device to support this functionality.

When a custom configuration profile is assigned to a Linux device, Workspace ONE UEM passes the manifest to the Puppet running on the device. Currently, when a device is enterprise wiped or unenrolled, these configuration changes are not removed from the device unless a removal manifest is defined in the profile.

For more information on Puppet, including sample manifests, see: http://forge.puppet.com.

To validate the syntax of your puppet code, see: https://validate.puppet.com.

Procedure

1 Navigate to **Resources > Profiles & Baselines > Profiles > Add > Add Profile > Linux**.

2 Configure the General settings for the profile as appropriate.

3 Select the Custom Configuration profile.

4 Select Configure.

5 Configure the payload including:

Setting	Description
Name	Populate a name that distinguishes this payload from others.
Enforce Manifest	If selected, then reapply the manifest at the data transmit interval configured in **Settings > Device & Users > Linux > Intelligent Hub Settings**. If deselected, then the manifest executes once when the profile initially pushes to the device.
Check for Dependency	If the puppet manifest has a required dependency, then it can be included here. For example, "puppetlabs/stdlib".
Install Manifest	Copy and paste the content of your Puppet Manifest here. This manifest implements on the device assigned in the general tab.
Remove Manifest	This manifest executes on the device when this profile is unassigned from a device. If this manifest is left blank, when a custom configuration profile is removed from a device, the action dictated by the Install Manifest remains on the device.

6 Select Save and Publish.

Custom Configuration Examples

Puppet Manifest Examples

Learn and explore Puppet if you are interested in creating custom configuration profiles. To get started, the following are examples of Puppet code that can be used on standard Ubuntu. They do not work on other distributions of Linux.

Install Chrome Browser on Ubuntu:

- Dependency: None
- Installation Manifest:

```
file { 'google-chrome-stable_current_amd64.deb': source => 'https://dl.google.com/
linux/direct/google-chrome-stable_current_amd64.deb', path => '/tmp/google-chrome-
stable_current_amd64.deb', ensure => present, } exec { 'install-chrome': command =>
'/usr/bin/dpkg -i /tmp/google-chrome-stable_current_amd64.deb', logoutput => true, }
```

- Removal Manifest:

```
package { 'google-chrome-stable': ensure => 'absent', }
```

Deactivating SSH Server on Ubuntu:

- Dependency: `puppetlabs-stdlib`
- Installation Manifest:

```
service { 'ssh': name => 'sshd', ensure => false, enable => false,}
```

Removal Manifest:

```
service { 'ssh': name => 'sshd', ensure => true, enable => true,}
```

Sensors for Linux Based Devices

5

Linux based devices contain multiple attributes such as hardware, certificates, patches, apps, and more. With Sensors, you can collect data for these attributes using the Workspace ONE UEM console and display the data returned by Sensors in Workspace ONE.

Linux devices have a huge number of attributes associated with them. This number increases when you track the different apps, distro versions, patches, and other continually changing variables. It can be difficult to track all of these attributes.

Workspace ONE UEM tracks a limited number of device attributes by default. However, sensors enable you to track specific device attributes. For example, you can create a sensor that tracks the number of battery charge cycles, last updated date of a virus definition file, or the build version of a specific security agent. Sensors allow you to track various attributes across your devices using Bash. For your Linux devices, sensor scripts can be configured to run periodically.

Find sensors in the main Workspace ONE UEM console navigation under Resources.

Workspace ONE UEM Options

- **Bash Scripts** - The Bash script determines the value of each sensor. For examples of what scripts you can create, see Bash Examples.
- **Support for Variables** - If your sensor script requires that dynamic or sensitive information be defined outside of the script, then use variables to securely store this information. Variable data is encrypted at-rest and in-transit. The variables can be referenced in the code directly by name $myvariable.
- **Sensors Triggers** - When configuring Sensors, control when the device reports the sensor data back to the Workspace ONE UEM console with triggers. Trigger choices for Linux devices include:
  - Periodically (based on a sample schedule)
  - When a user logs in or out
  - When a device starts up
  - When the device's network changes
- **Sensors** - See data for single devices on the Sensors tab in a device's Device Details page.

Workspace ONE Intelligence Options

To view and interact with data from your sensors when you use the VMware Workspace ONE Intelligence service, run a report or create a dashboard. When you run reports, use the Workspace ONE UEM category, Device Sensors. Find your sensors and select them for queries in reports and dashboards.

For details on how to work in Workspace ONE Intelligence, see VMware Workspace ONE Intelligence Products.

Creating Sensors for Linux Devices

To track specific device attributes such as remaining battery, specific version, build information, or average CPU usage, create sensors in the Workspace ONE console. Each sensor includes a script of code to collect the desired data. You can upload these scripts or enter them directly into the console.

To gather attribute values, sensors use Bash scripts. You must create these scripts before creating a sensor or during configuration in the scripting window.

Note To view the sensors for multiple devices and interact with the data in reports and dashboards, you must opt into VMware Workspace ONE Intelligence. If you want to view sensor data for a single device, you do not need VMware Workspace ONE Intelligence. To view the data, go to the Device Details page for the device and select the Sensors tab.

1 In the Workspace ONE console, navigate to **Resources > Sensors**.

2 On the Sensors page, click **Add** and select **Linux**.

3 On the New Sensor page, navigate to **General > Name** and enter the following:

Setting	Description
Name	Enter the name of the sensor. The name must start with a lowercase letter followed by alpha-numeric characters and underscores. The name must be from 2 through 64 characters.
Description	Enter the description of the sensor.

4 Click **Next**.

5 Configure the sensor settings in the Details tab.

Setting	Description
Language	Select the language. Currently, only Bash is supported for Linux devices.
Execution Context	This setting controls the context with which the script runs. Currently, only System is support for Linux devices.

Setting	Description
Response Data Type	Select the type of response the script will return. You can choose between: ■ String ■ Integer ■ Boolean ■ Date Time
Code	Upload a script for the sensor or write your own in the text box provided.

6 Click **Next**.

7 Optional: On the Variables tab, define variable names and values to use in your sensor script. These variables are securely stored, encrypted at-rest, and only used temporarily during script execution in the scripting environment.

Variables support static text or the Workspace ONE UEM lookup values. The lookup values are resolved before being delivered to the device for execution.

Bash scripts can reference the variables directly by name from the environment like $myvariable.

8 Click **Save** or **Save** and **Assign**.

To add sensors to a smart group, save the sensors information and go back to menu or move to the Assignment page.

What to do next

To add a sensor to a smart group, perform the following steps:

1 On the New Assignment page, enter the Assignment Name and Select Smart Group.

2 Click **Next**.

3 On the Deployment page, select when you want to trigger the sensor to be captured on the devices. For the periodically trigger, the script runs periodically based on the Intelligent Hub Data Transmit Interval configured in **All Settings > Device & Users > Linux > Intelligent Hub Settings**.

4 Click **Save**.

After the assignment group is saved, prioritize the assignments if multiple smart groups are configured with potentially overlapping sets of devices. Once this step is done, devices with Workspace ONE Intelligent Hub installed receive the Sensor configurations on the next check-in. Workspace ONE Intelligent Hub then runs the sensors and reports the data back to Workspace ONE.

View Sensors in Linux Device Details

View sensor data in the Workspace ONE UEM console on the Sensors tab in Device Details.

1 In the Workspace ONE UEM console, navigate to **Device > Details View** and select the Sensors tab. The following details are displayed in the Sensors tab:

   a **Name** - Name of the sensor.

   b **Value** - Value reported by the device.

   c **Last executed date** - The timestamp for the collection of the sensor value.

2 To request the device to on-demand, run the Sensor. Report the value. Select a Sensor name and click **Run**.

   Note The Run button displays in Device Details only for supported Hub versions. The minimum supported Linux Hub version is 21.05.

3 To view information about Sensors execution, navigate to **Details View > Troubleshooting**. In the event log filters, select **Sensors**.

   Note This is seen only if the event log level is set to capture information or debug messages.

Linux Sensor Examples

When you create sensors for Linux devices, you must upload a bash script or enter the bash script in the text box provided during the configuration in the Workspace ONE UEM console. These commands return the values for the sensor attributes.

Bash Examples: The following examples contain the settings and the code needed for standard Ubuntu and may not work on other Linux distributions.

Get the current Host Name:

Language: Bash

Execution Context: System

Response Data Type: String

```
cat /proc/sys/kernel/hostname |
```

Get a list of all users currently logged in:

Language: Bash

Execution Context: System

Response Data Type: String

```
who | cut -d' ' -f1 | sort | uniq
```

Get current distribution version:

Language: Bash

Execution Context: System

Response Data Type: String

```
( lsb_release -ds || cat /etc/*release || uname -om ) 2>/dev/null | head -n1 |
```

Determine if an SSH Server is running:

Language: Bash

Execution Context: System

Response Data Type: Integer

```
ps -ef | grep sshd | grep -v "grep" | wc -l
```

Note The returned value equates to the number of SSH daemons running on the endpoint.