Forwarder management overview

The forwarder management interface is an interactive, visual tool for creating **server classes**, which map **deployment clients** to **deployment apps**. You can also use forwarder management to manage and monitor your deployment.

The interface saves server class configurations to a serverclass.conf file, located under `$SPLUNK_HOME/etc/system/local` on the deployment server.

Capabilities of the interface

The main purpose of the forwarder management interface is to create and edit server classes. You can also use it for a number of other purposes:

- To track the status of the system
- To monitor deployment activity
- To view the associations between apps, clients, and server classes
- To configure app behavior
- To uninstall apps from clients

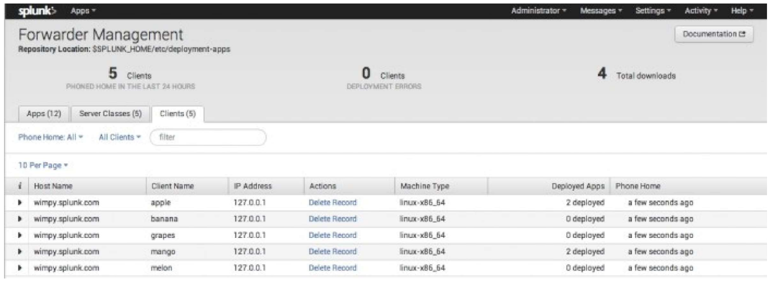
Access the forwarder management interface

You access the forwarder management interface through Splunk Web on the deployment server. To open the interface:

1. Click the **Settings** link at the top of Splunk Web. A window pops up with links to the set of system interfaces.

2. Select "Forwarder Management" in the **Distributed Environment** section. This takes you to the main page of the interface.

The following example shows the forwarder management home page as it might appear when you already have some apps, clients, and server classes:



Forwarder Management

Repository Location: $SPLUNK_HOME/etc/deployment-apps

5 Clients PHONED HOME IN THE LAST 24 HOURS

0 Clients DEPLOYMENT ERRORS

4 Total downloads

i	Host Name	Client Name	IP Address	Actions	Machine Type	Deployed Apps	Phone Home
▸	wimpy.splunk.com	apple	127.0.0.1	Delete Record	linux-x86_64	2 deployed	a few seconds ago
▸	wimpy.splunk.com	banana	127.0.0.1	Delete Record	linux-x86_64	0 deployed	a few seconds ago
▸	wimpy.splunk.com	grapes	127.0.0.1	Delete Record	linux-x86_64	0 deployed	a few seconds ago
▸	wimpy.splunk.com	mango	127.0.0.1	Delete Record	linux-x86_64	2 deployed	a few seconds ago
▸	wimpy.splunk.com	melon	127.0.0.1	Delete Record	linux-x86_64	0 deployed	a few seconds ago

The page contains these features (going from top to bottom):

- The **Repository Location**. The repository location is where deployment apps reside on the deployment server.
- A status section, with information on deployment clients and recent downloads.
- Three tabs:
    - **Apps**. This tab lists the set of deployment apps in the repository location, along with their status. You can edit some app properties from here.
    - **Server Classes.** This tab lists the set of server classes, along with their status. From here, you can create new server classes and edit existing ones. You can also drill down on existing server classes for information on their associated apps and clients. For a new, unconfigured deployment server, the list will be empty.
    - **Clients.** This tab lists all of the deployment server's clients, along with status information. You can filter this list in various ways to limit the current view.

For more information on this page, read "Use forwarder management to define server classes".

Limitations of the interface

The forwarder management interface supports the vast majority of deployment server use cases. For some complex configuration requirements, however, you might need to edit `serverclass.conf` directly.

Important: If you switch from forwarder management to direct editing of `serverclass.conf`, you might not be able to use forwarder management for any subsequent configuration. This is because the forwarder management interface can handle only a subset of the configurations possible through `serverclass.conf`.

Here are some limitations of the interface, compared to direct editing of the configuration file:

- Deployment behavior of any app must be the same across server classes. For example, you cannot specify that an app initiate a restart on a client when it's downloaded via one server class but not when it's downloaded via another server class.
- You cannot modify the default `repositoryLocation` on the deployment server.
- The `filterType`, which governs the combined behavior of allow lists and deny lists, must use the default value of `whitelist`.
- Client filters are only supported at the server class level.

For details on forwarder management limitations and compatibility between forwarder management and `serverclass.conf`, see the topic "Compatibility and forwarder management".