# Base Configurations

Whether you are into a multiple Splunk instance configuration, consisting of multiple indexers, search heads and/or forwarders, or just a single Splunk instance it makes sense to utilize a configuration management tool. Splunk has this tool built in, the [deployment server](http://docs.splunk.com/Documentation/Splunk/latest/Deploy/Aboutdeploymentserver).

The deployment server is utilized to make the management and configuration of Splunk instances easy and cohesive. This document will outline some best practices and configuration files used by Splunk Professional Services to baseline a new Splunk deployment.

### Deployment Server

The deployment server is a Splunk instance that acts as a centralized configuration manager, grouping together and collectively managing any number of Splunk instances. Any Splunk instance can act as a deployment server, even one that is indexing data locally. Splunk instance, e.g., forwarder, which is remotely configured by a deployment server, is called a deployment client.

Create the Deployment App Directories

Splunk's configuration information is stored in configuration files, identified by their ***.conf*** extension. The default location (repository) on the deployment server to store these configurations is $SPLUNK\_HOME/etc/deployment-apps. A number of deployment apps have accompanied this document. While all of them may or may not be used depending on what type of Splunk configuration you have, distributed or single instance, it’s a good idea to understand all of them.

At this point it makes sense to place all of the accompanying base configurations apps on the deployment server in the repository location mentioned above. These apps are simply file system directories, consisting of ***.conf*** files that Splunk understands how to process.

Next, let’s start working through the functionality of each by setting up the deployment server itself' configuration file, serverclass.conf, as well as the first app that allows communication from the deployment clients to the deployment server.

serverclass.conf and Deployment Client App

Splunk deployment clients use the **deploymentclient.conf** to receive configurations from a deployment server. The deployment app responsible for this is the **org\_all\_deploymentclient** app.

*$SPLUNK\_HOME/etc/deployment-apps/org\_all\_deploymentclient/local/deploymentclient.conf*

[deployment-client]

[target-broker:deploymentServer]

targetUri= deploymentserver.splunk.mycompany.com:8089

* Update the name of the configuration bundle, org\_all\_deploymentclient, to reflect the name of your organization, e.g., acme\_all\_deploymentclient
  + **Note:**  This process will be done for every single deployment app that accompanied this document
* Update target-broker parameter name (optional)
* Update targetUri to reflect the URL of your deployment server (preferably a DNS name)

Eventually we will use the app, org\_all\_deploymentclient, on all of the Splunk instances so that they know how to communicate with the deployment server.

Define Server Classes

A [server class](http://docs.splunk.com/Splexicon:Serverclass) defines a deployment configuration shared by a group of [deployment clients](http://docs.splunk.com/Splexicon:Deploymentclient). It defines both the criteria for being a member of the class and the set of content to deploy to members of the class. You can define different server classes to reflect the different requirements, OSes, machine types, or functions of your deployment clients.

You define server classes in [serverclass.conf](http://docs.splunk.com/Documentation/Splunk/latest/admin/Serverclassconf) on the deployment server. The server classes will logically group servers in the environment to send particular configuration files to. The whitelist and blacklist properties of each server class are used to identify deployment clients that phone home by IP address, hostname, client name or machine type. Once the deployment clients are configured to phone home they will always supply these four pieces of information.

A starter serverclass.conf file also accompanied this document. It needs to be copied to the $SPLUNK\_HOME/etc/system/local directory on the deployment server. It is written to reflect the original naming as well as to utilize the client name feature of the deployment server for use in a distributed environment to identify Splunk components as indexers (**splk-indexer**) or search heads (**splk-search**). It an environment with an increased number of indexers and/or search heads this makes for easier maintenance of the whitelist and blacklist parameters as we can simply use the client name instead of having a long list of all the indexers called out individually by IP address or hostname. The use of these client names will be covered in more detail below.

*$SPLUNK\_HOME/etc/system/local/serverclass.conf*

[global]

restartSplunkd = false

# ALL INSTANCES

[serverClass:all\_systems]

whitelist.0 = \*

restartSplunkd = true

[serverClass:all\_systems:app:org\_all\_deploymentclient]

# ALL SEARCH HEADS

[serverClass:all\_search]

whitelist.0 = splk-search

restartSplunkd = true

#[serverClass:all\_search:app:org\_all\_search\_base]

# Next two stanzas only used if pushing indexes from the search head to the indexers

#[serverClass:all\_search:app:org\_all\_search\_outputs]

#[serverClass:all\_search:app:org\_all\_forwarder\_outputs]

# ALL INDEXERS

[serverClass:all\_indexer]

whitelist.0 = splk-indexer

restartSplunkd = true

#[serverClass:all\_indexer:app:org\_all\_indexer\_base]

#[serverClass:all\_indexer:app:org\_all\_indexes]

# ALL FORWARDERS

[serverClass:all\_forwarder]

whitelist.0 = \*

blacklist.0 = splk-\*

restartSplunkd = true

#[serverClass:all\_forwarder:app:org\_all\_forwarder\_outputs]

* Update the whitelist and blacklist parameters
* Update the title of all the deployment apps, for example, org\_all\_deploymentclient, to reflect the name of your organization, e.g., acme\_all\_deploymentclient
* Restart the deployment server

As you can see the serverclass.conf file is configured with server classes for specific Splunk components: indexers, search heads and forwarders. However, all of the lines configuring the apps to be sent to the matching Splunk components have been commented out. These will be processed as we proceed through this document. Let’s start configuring the deployment clients to communicate with the deployment server.

### Indexers

After the Splunk installation process is complete, copy the deployment client configuration bundle (acme\_all\_deploymentclient) into the $SPLUNK\_HOME/etc/apps directory. The second step is to set up the client name, splk-indexer. A sample has been supplied with this document. Simply copy the deploymentclient.conf.indexer file to the following directory on the indexer and rename the file to deploymentclient.conf:

*$SPLUNK\_HOME/etc/system/local/deploymentclient.conf*

[deployment-client]

clientName = splk-indexer

Upon restart the indexer will attempt to communicate or check in with the deployment server. Let’s ensure that this is working before moving on.

After you have restarted the indexer open up a terminal window on the deployment server. Change the directory to $SPLUNK\_HOME/bin and run the following command:

./splunk list deploy-clients

It may take a minute or so for the indexer to check in but a listing should be displayed showing this successful connection from the indexer. If it does not then a configuration is not set properly either in the deployment app with the correct IP or hostname of the deployment server or the port could be blocked between the indexer and the deployment server.

* Ensure that TCP port 8089 is open between the deployment client and the deployment server
* Ensure that Splunk is running on both the deployment client and the deployment server
  + Run .**/splunk status** from **$SPLUNK\_HOME/bin/**
* Ensure on the deployment client that the deploymentclient.conf file contains the proper FQDN or IP address of the deployment server.

If a successful check in has occurred repeat the same process for each and every indexer in your environment. Next, let’s do the same process on the search heads.

### Search Heads

### The process is identical to the indexer steps except for the deploymentclient.conf file identifying the client name. After the Splunk installation process is complete, copy the deployment client configuration bundle (acme\_all\_deploymentclient) into the $SPLUNK\_HOME/etc/apps directory. The second step is to set up the client name, splk-search. A sample has been supplied with this document. Simply copy the deploymentclient.conf.search file to the $SPLUNK\_HOME/etc/system/local on the search head and rename the file to deploymentclient.conf:

### Again, confirm after the search head has been restarted that you can see the component checking in with the deployment server. If you are on linux you can run a command to search directly for the search heads by running:

./splunk list deploy-clients | grep splk-search

### Forwarder

In nearly all respects, the universal forwarder represents the best tool for forwarding data to indexers. The universal forwarder's sole purpose is to forward data. Unlike a full Splunk instance, you cannot use the universal forwarder to index or search data. To achieve higher performance and a lighter footprint, it has several limitations. If you want to see all the things you can and cannot do with the universal forwarder please review the documentation.

Install Forwarder

When installing the forwarder, do not select or add any configurations. After the installation process is complete, copy the deploymentclient app (org\_all\_deploymentclient) into the forwarder’s $SPLUNK\_HOME/etc/apps directory.

Start the forwarder and ensure that it is checking in to the deployment server. You can use a similar technique above and grep for the hostname or IP address of the forwarder to ensure successful communication:

./splunk list deploy-clients | grep <hostname>

### App Configuration

At this point all, or at least a handful of Splunk components are successfully checking in to the deployment server. The process now moves to configuring the other apps that accompanied this document that were placed on the deployment server in the $SPLUNK\_HOME/etc/deployment-apps directory. They consist of:

* org\_all\_forwarder\_outputs
* org\_all\_indexer\_base
* org\_all\_indexes
* org\_all\_search\_base
* org\_all\_search\_outputs
* org\_dept\_app\_inputs (not used for this document)
* org\_dept\_app\_props (not used for this document)

Each of these applications consist of Splunk configurations files to serve a specific purpose on their destined servers. Each app may consist of one or more configuration files and changes may be required to some or all of them depending upon each customer’s situation.

org\_all\_forwarder\_outputs

This app is destined for the forwarders in your environment. It is what will allow the forwarders to connect, load balance and send data to your Splunk indexers. The outputs.conf file located in $SPLUNK\_HOME/etc/deployment-apps/org\_all\_fowarder\_outputs/local/ is what enables this. Simply update the server attribute with your list of indexers. In our example, ***server\_one*** and ***server\_two*** need to be updated with the IP or hostname of your indexers. This is a comma-delimited list and can contain as little as one entry.

[tcpout]

defaultGroup = primary\_indexers

[tcpout:primary\_indexers]

server = server\_one:9997, server\_two:9997

autoLB = true

org\_all\_indexer\_base

*$SPLUNK\_HOME/etc/deployment-apps/org\_all\_indexer\_base*

* ***inputs.conf*** – this file sets up the listening port for splunk data, default is 9997. Update this if you would like to use a different port. If a new port is assigned ensure the outputs.conf file is updated from org\_all\_forwarder\_outputs documented above to reflect this port instead of 9997.
* ***web.conf*** – this file has a single configuration commented out by default to disable Splunk Web. If you are in a distributed environment with a dedicated search head you may uncomment this line out as to disable the web front end on the indexers.

org\_all\_indexes

*$SPLUNK\_HOME/etc/deployment-apps/org\_all\_indexes*

* ***indexes.conf*** – this file does a number of configurations for index storage. It utilizes the volume settings that are configured to the default index storage location, $SPLUNK\_HOME/var/lib/splunk. It applies arbitrary sizing for both the volume as a whole as well as the hot/warm and cold path sizing for the volume as well. It also updates the out of the box Splunk index definitions that ship with the product to utilize the new volume setting, **volume:primary**. This file could require the most amount of custom configuration depending on a number of factors at the customer site. Do they have different mount points for hot/warm and cold database locations? Do they want to age out data based upon overall volume or do they want to ensure storage for a certain amount of time? Further information can be found online in the [indexes.conf](http://docs.splunk.com/Documentation/Splunk/latest/admin/Indexesconf) spec.

org\_all\_search\_base

*$SPLUNK\_HOME/etc/deployment-apps/org\_all\_search\_base*

* Empty currently, depending upon environment set up could be used to hold indexes.conf file for volume settings or index definitions.

org\_all\_search\_outputs

*$SPLUNK\_HOME/etc/deployment-apps/org\_all\_search\_outputs*

* ***outputs.conf*** – only required in a distributed Splunk environment with separate search heads and indexers. The purpose is to push all internal Splunk indexes from the search head to the indexers. If this application is used then all of the search heads should also be sent the org\_all\_forwarder\_outputs application. This can be accomplished by using the all\_search serverclass in serverclass.conf.

Server Class Update

At this point the serverclass.conf file can be updated on the deployment server to reflect the apps that have been configured for your environment. Simply uncomment out any of the lines registering the app for the server class and issue a reload of the deployment server, ./splunk reload deploy-server.

*$SPLUNK\_HOME/etc/system/local/serverclass.conf*

[global]

restartSplunkd = false

# ALL INSTANCES

[serverClass:all\_systems]

whitelist.0 = \*

restartSplunkd = true

[serverClass:all\_systems:app:org\_all\_deploymentclient]

# ALL SEARCH HEADS

[serverClass:all\_search]

whitelist.0 = splk-search

restartSplunkd = true

#[serverClass:all\_search:app:org\_all\_search\_base]

# Next two stanzas only used if pushing indexes from the search head to the indexers

[serverClass:all\_search:app:org\_all\_search\_outputs]

[serverClass:all\_search:app:org\_all\_forwarder\_outputs]

# ALL INDEXERS

[serverClass:all\_indexer]

whitelist.0 = splk-indexer

restartSplunkd = true

[serverClass:all\_indexer:app:org\_all\_indexer\_base]

[serverClass:all\_indexer:app:org\_all\_indexes]

# ALL FORWARDERS

[serverClass:all\_forwarder]

whitelist.0 = \*

blacklist.0 = splk-\*

restartSplunkd = true

[serverClass:all\_forwarder:app:org\_all\_forwarder\_outputs]