Properly Managing Configuration Files:
From Standalone to Multi-Cluster

Laurie Hofer, Sr. Professional Services Consultant

Sanford Owings, Principal Consultant

splunk>

Disclaimer

During the course of this presentation, we may make forward looking statements regarding future events or the expected performance of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results could differ materially. For important factors that may cause actual results to differ from those contained in our forward-looking statements, please review our filings with the SEC. The forward-looking statements made in the this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, this presentation may not contain current or accurate information. We do not assume any obligation to update any forward looking statements we may make.

In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only and shall not, be incorporated into any contract or other commitment. Splunk undertakes no obligation either to develop the features or functionality described or to include any such feature or functionality in a future release.

Abstract

splunk>

Abstract

With a varied landscape of Splunk Enterprise configuration possibilities and configuration distribution managers now available, join us for a discussion on best practices for managing your Splunk configuration files in today's distributed environments.

Learn how and when to distribute configurations using your deployment server, your indexer cluster's master node, or your search head cluster's deployer – and how to get the right configurations to the right places in the most efficient and confusion-free methods possible.

Agenda

- Configuration Files – What and where are they?
- Modifying Configuration Files
- Configuration and Distribution Managers
- Distributed Splunk and Forwarder Management
- Search Head Cluster Management
- Indexer Cluster Management
- Best Practices and Version Control

Configuration Files:
What are they, and where are they?

splunk>

Configuration Files

- Files ending in .conf located in:
  - $SPLUNK_HOME/etc/system/default/ or *apps/*appname*/default/
    $SPLUNK_HOME/etc/system/local/ or *apps/*appname*/local/
- Parameters set within the files configure Splunk software to operate as you want it to.
- $SPLUNK_HOME/etc/system/default/*.conf or *apps/*appname*/default/*.conf
  - Files that ship with the software or app.
  - These files should not be modified.
- $SPLUNK_HOME/etc/system/local/*.conf or *apps/*appname*/local/*.conf
  - Your changes to the default configurations for Splunk software or app.
  - Apps can be custom or downloaded from Splunkbase.
  - *This behavior changes slightly for Search Head Clustering*

Example File: props.conf

```
#  /opt/splunk/etc/system/default/props.conf
# Version 6.2.5
# DO NOT EDIT THIS FILE!
# Changes to default files will be lost on update and are difficult to
# manage and support.
#
# To override a specific setting, copy the name of the stanza and
# setting to the file where you wish to override it.
#
[default]
CHARSET = UTF-8
LINE_BREAKER_LOOKBEHIND = 100
TRUNCATE = 10000
DATETIME_CONFIG = /etc/datetime.xml
ANNOTATE_PUNCT = True
```

Modifying Configurations

splunk>

Updating Configuration Files

- Two methods to edit configs
  - Using the GUI
    - Updates are placed automatically into local versions of the files based on login
  - Within the .conf files directly
    - More granular functionality
    - Spec files for each .conf file list all the possible values for each parameter

- When should I not use the GUI to make changes?
  - In any clustered environment on cluster members

How Splunk Prioritizes Configuration Files

- Which of the following locations has the highest file precedence?
  - $SPLUNK_HOME/etc/system/local/
  - $SPLUNK_HOME/etc/apps/
  - $SPLUNK_HOME/etc/deployment-apps/
  - $SPLUNK_HOME/etc/slave-apps/

Your poll will show here

1

Install the app from pollev.com/app

2

Make sure you are in Slide Show mode

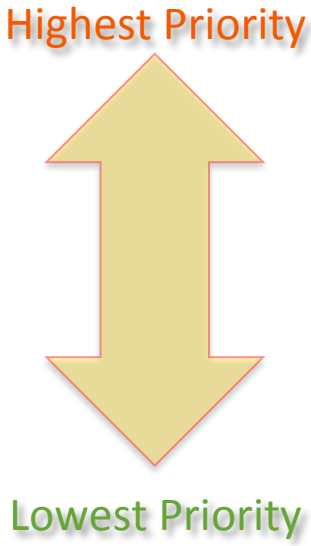
Still not working? Get help at pollev.com/app/help

or

Open poll in your web browser

Configuration File Precedence

- $SPLUNK_HOME/etc/slave-apps/*/local/
  - (indexer cluster peers only)
- $SPLUNK_HOME/etc/system/local/
- $SPLUNK_HOME/etc/apps/*/local/
- $SPLUNK_HOME/etc/slave-apps/*/default/
  - (indexer cluster peers only)
- $SPLUNK_HOME/etc/apps/*/default/
- $SPLUNK_HOME/etc/system/default/



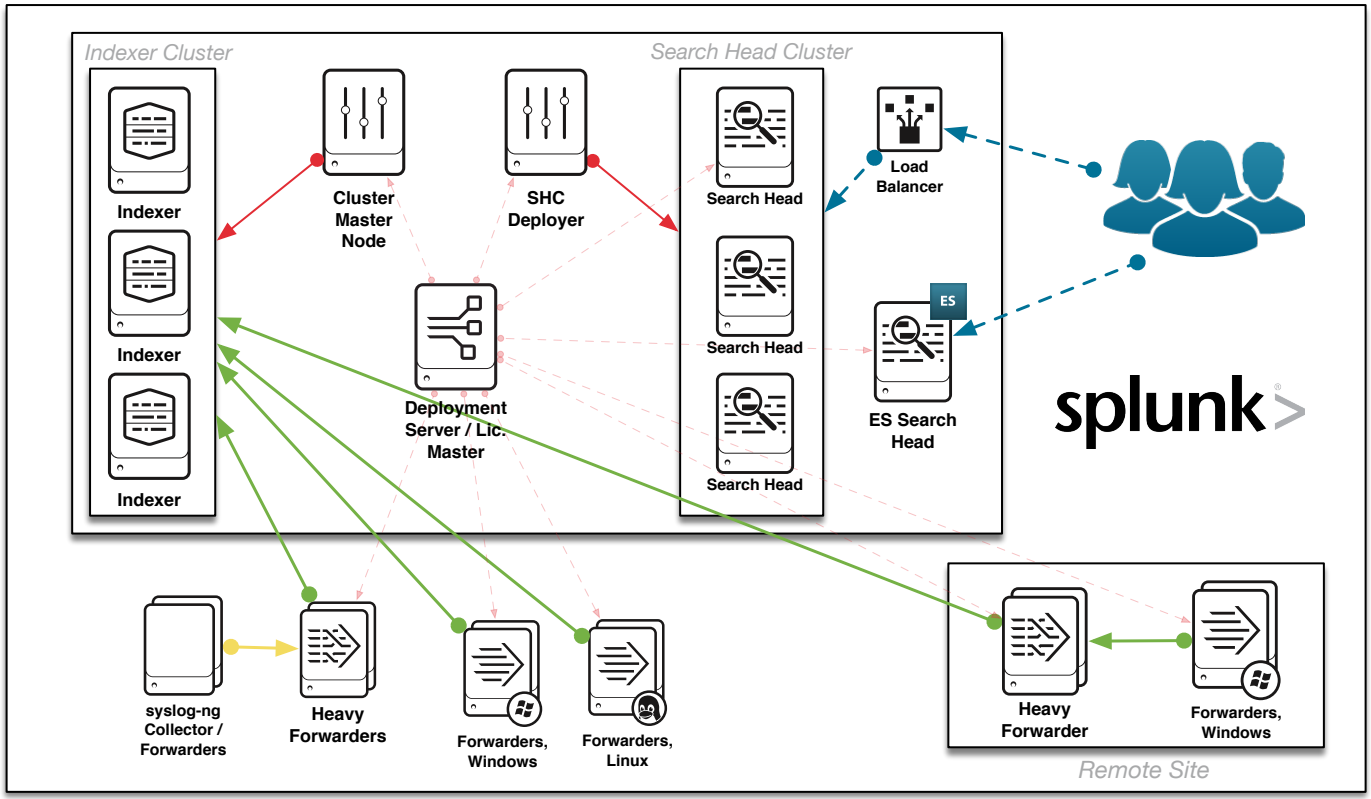
Multiple Configurations – Which One Wins?

```
bash-3.2# /opt/splunk/bin/splunk btool conf list --debug | grep conf.conf  | grep -v confdb
/opt/splunk/etc/system/default/conf.conf [bootstrap]
/opt/splunk/etc/system/default/conf.conf $apps-base$/* = default
/opt/splunk/etc/system/default/conf.conf $apps-base$/*/bin = searchscripts
/opt/splunk/etc/system/default/conf.conf $apps-base$/*/default = conf
/opt/splunk/etc/system/default/conf.conf $apps-base$/*/default/data/models = models
/opt/splunk/etc/system/default/conf.conf $apps-base$/*/default/data/ui = xml
/opt/splunk/etc/system/default/conf.conf $apps-base$/*/local = conf
/opt/splunk/etc/system/default/conf.conf $apps-base$/*/local/data/models = models
/opt/splunk/etc/system/default/conf.conf $apps-base$/*/local/data/ui = xml
/opt/splunk/etc/system/default/conf.conf $apps-base$/*/lookups = lookups
/opt/splunk/etc/system/default/conf.conf $apps-base$/learned = default
```

/opt/splunk/bin/splunk btool conf list --debug | grep conf.conf | grep -v confdb

- Configurations are combined based on precedence
  - Use btool via the CLI to see which are being applied

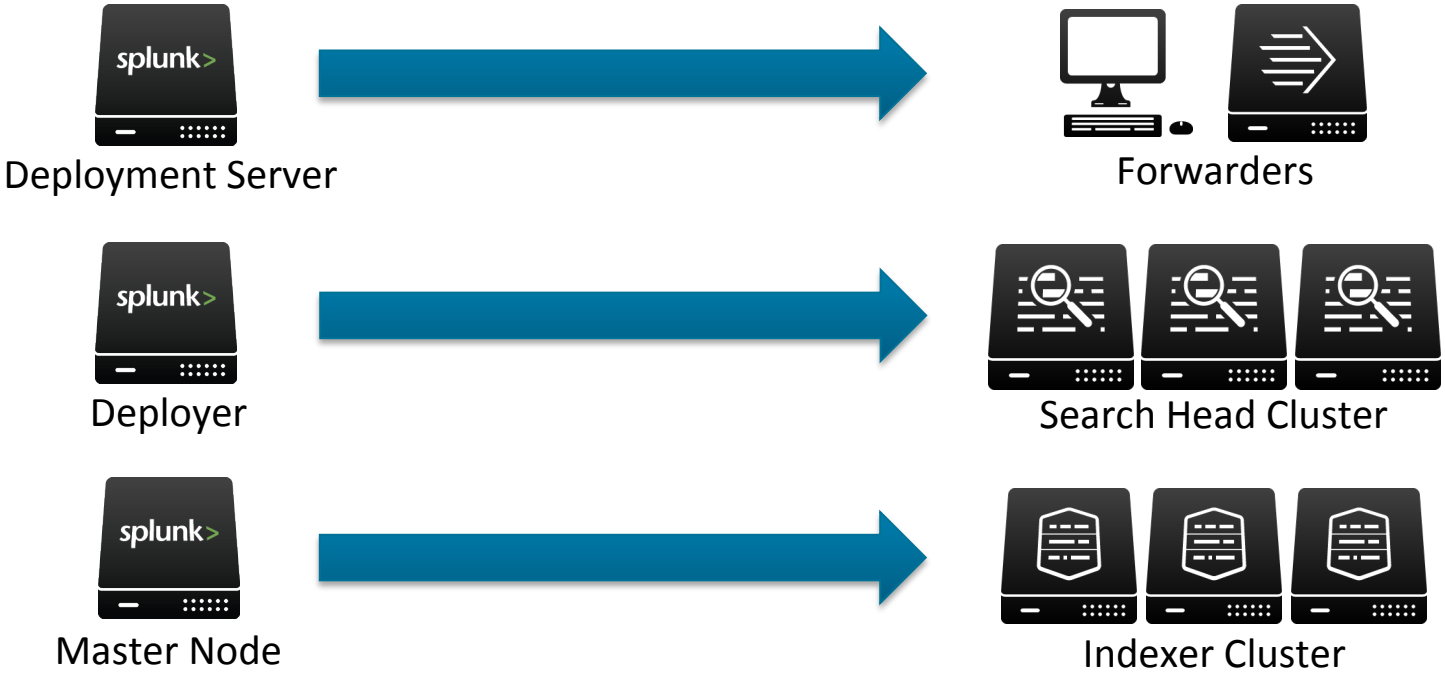
Example Splunk Architecture



Configuration and Distribution Managers

splunk>

Configuration Distribution Management

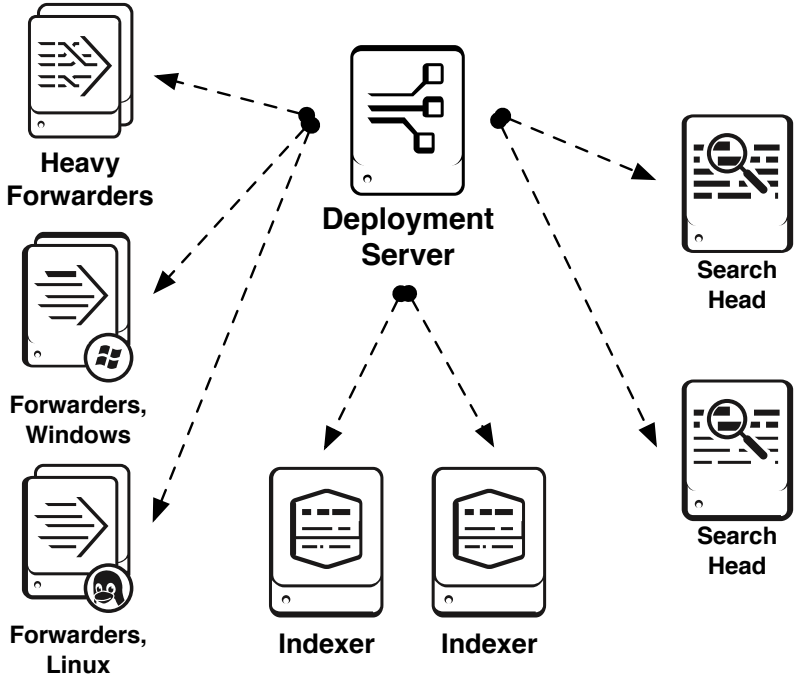


Distributed Splunk and Forwarder Management

splunk>

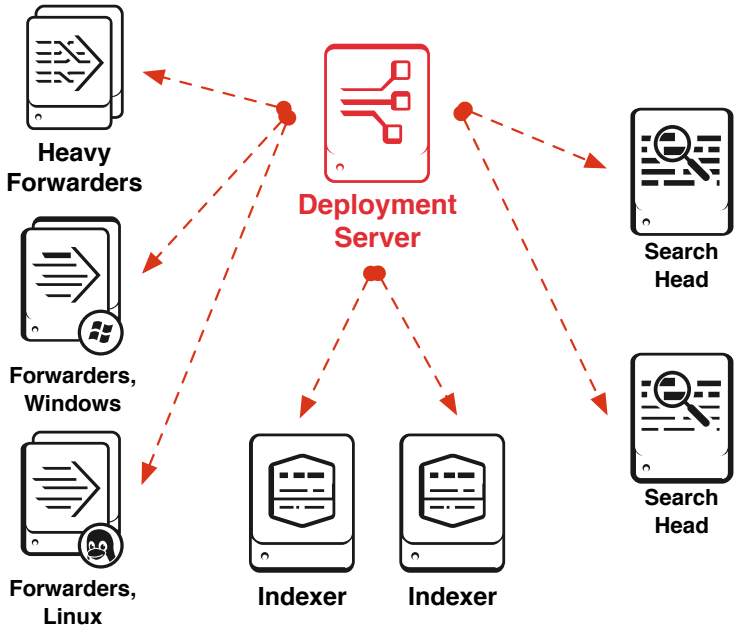
Standard Distributed Environment

- Deployment server manages all or most configurations
- No search head or indexer clustering
- Configurations are kept consistent across nodes of the same role/class



The Deployment Server

- Deployment Server distributes configurations to clients
- Apps that contain configurations are stored in $SPLUNK_HOME/etc/deployment-apps/
- Clients poll Deployment Server on interval set in deploymentclient.conf (defaults to every 60 seconds)
- serverclass.conf, located at $SPLUNK_HOME/etc/system/local/ indicates which clients are members of which client grouping, and indicates which apps those clients receives
- Client management is also modifiable from the GUI under Settings | Forwarder management



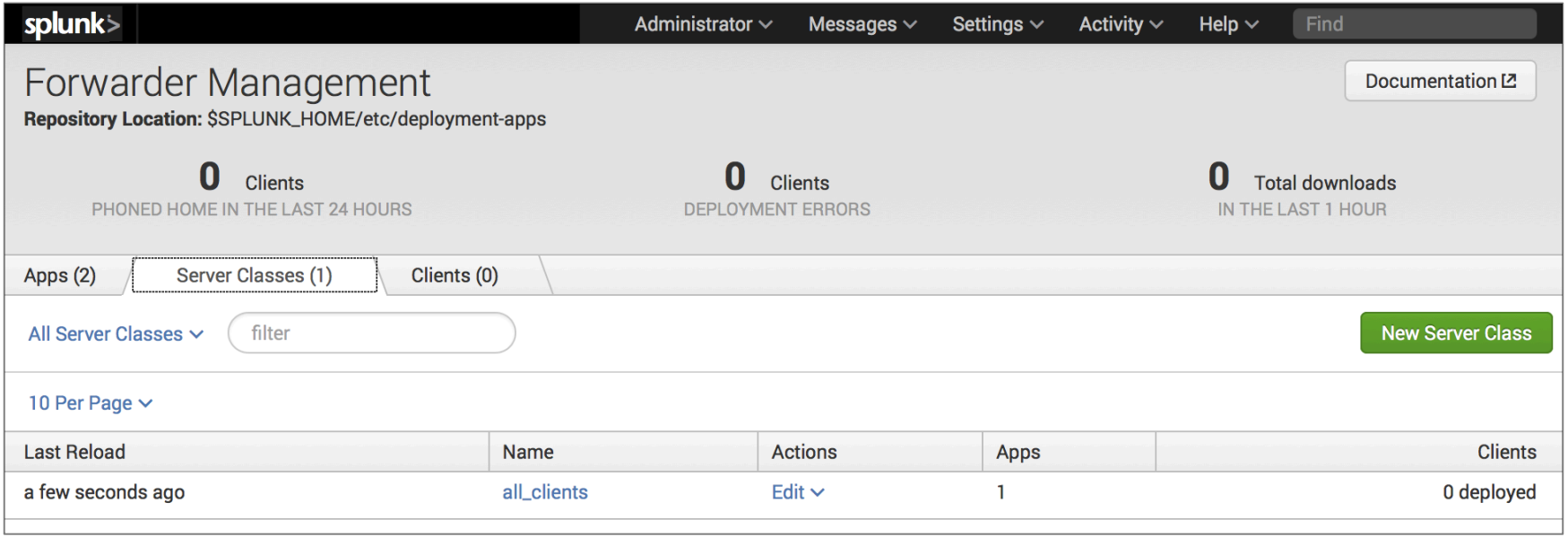
Example File: serverclass.conf

```
# ALL CLIENTS - KEEP DEPLOYMENT CLIENT UP TO DATE
[serverClass:all_clients]
whitelist.0 = *
[serverClass:all_clients:app:org_all_deploymentclient]

# FULL INSTANCES - SEARCH HEADS, HEAVY FORWARDERS, INDEXERS
[serverClass:full_instances]
whitelist.0 = search_head_one.mycompany.com
whitelist.1 = indexer-[1-3].example.com
[serverClass:full_instances:app:org_all_indexes]

# ALL SEARCH HEADS
[serverClass:all_search]
whitelist.0 = search_head_one.mycompany.com
whitelist.1 = search_head_two.mycompany.com
# This is a search head.
[serverClass:all_search:app:org_all_search_base]
```

GUI Client Management



Deployment Management

- Calculate deployment server performance*
  - http://docs.splunk.com/Documentation/Splunk/latest/Updating/Calculatedeploymentserverperformance
  - Linux: Up to about 10,000 clients*
  - Windows: Up to about 7,000 clients*
- Multiple Deployment Servers
  - 6.3 allows deployment servers sharing the same content behind a load balancer, due to a fix to the way the checksum of an app is done
    - New settings to control this behavior (details available in upcoming release documentation)
      - crossServerChecksum in serverclass.conf
      - reloadDSOnAppInstall in deploymentclient.conf
  - Not available in 6.2 and below
    - Current behavior: DS on host-ds1.company.com computes checksum 12345 for app A, but host-ds2.company.com computes checksum 12543 (modtime) so clients would perpetually keep downloading depending upon who they ask

**Using reference hardware*

Best Practices

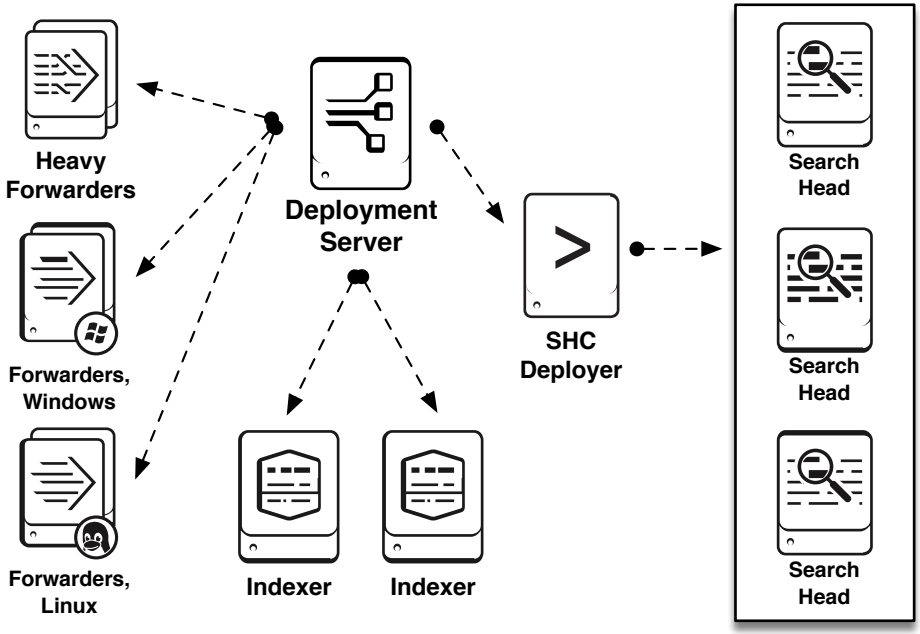
- Use naming conventions and be consistent
- Names should be descriptive
  - yourcompany_all_remoteheavyforwarders
- Differentiate between TAs from Splunkbase and those that are custom
- Use descriptive/dated comments in both serverclass.conf and your custom/ local configuration files
- Use version control: Git or SVN
- Carefully utilize any configurations outside of $SPLUNK_HOME/etc/apps/ on clients, as the deployment server can only overwrite files in this location

Search Head Cluster Management

splunk>

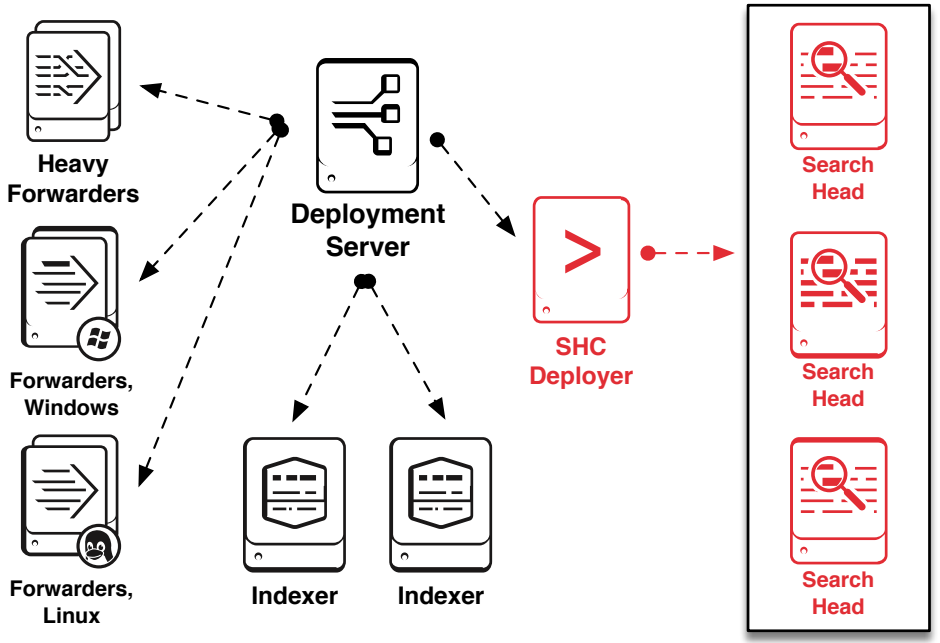
Search Head Clustering

- ALL apps from /etc/shcluster/ get pushed to /etc/apps/
- App Configs and User Content "flatten" to default instead of local
- Cluster replication is managed within the search head cluster by the elected captain
- Captain replicates search artifacts and run-time changes between members to keep them in sync
- GUI management are not available at this time



The Deployer

- Deployer distributes configurations to clients
- Apps that contain configurations are stored in $SPLUNK_HOME/etc/shcluster/ on the deployer
- Search head clustering is enabled and managed via the CLI
- No GUI management at this time



splunk show shcluster-status

```
Captain:
                    elected_captain : Thu Sep 8 14:11:21 2015
                                 id : C0C81FAC-0192-424A-A073-C2FFB2192FBF
                     initialized_flag : 1
                                 label : app011
                    maintenance_mode : 0
                               mgmt_uri : https://app011:8089
                    min_peers_joined_flag : 1
                    rolling_restart_flag : 0
                    service_ready_flag : 1
Members:
        app011
                                 label : app011
                                 mgmt_uri : https://192.168.1.100:8089
                                 status : Up
        app012
                                 label : app012
                                 mgmt_uri : https://192.168.1.101:8089
                                 status : Up
        app013
                                 label : app013
                                 mgmt_uri : https://192.168.1.102:8089
                                 status : Up
```

Differences of Search Head Cluster Management

- Search head cluster management differs in several ways
  - Configurations (apps) are only distributed to the cluster members when "splunk apply shcluster-bundle" is run from the CLI, or when a cluster member joins or rejoins the cluster and it polls the deployer for any updates
  - Apps' configurations are "flattened" into the /appname/default/ directory on the deployer, so that run-time changes which are applied to the /local/ directories of the apps can be replicated between members by the captain
  - Search artifacts and changes made to whitelisted .conf files via Splunk Web, Splunk CLI or the REST API are replicated between members by the captain
    - *See Splunk docs for configuration files on the replication whitelist defaults*

Search Head Clustering Replication Overview

- **Changes that ARE propagated by the captain**
  - Changes to configuration files made at runtime via Splunk Web, the CLI or the REST API
    - Synced and replicated approximately every five seconds
    - Controlled via a whitelist in server.conf in $SPLUNK_HOME/etc/system/local/
      Changes to this must be made on* **each *cluster member*
- **Changes that are NOT propagated**
  - MANUAL EDITS to configuration files on the cluster members
  - Index-time settings such as data inputs or indexes
  - Configuration changes made to configuration files not on the whitelist

Best Practices

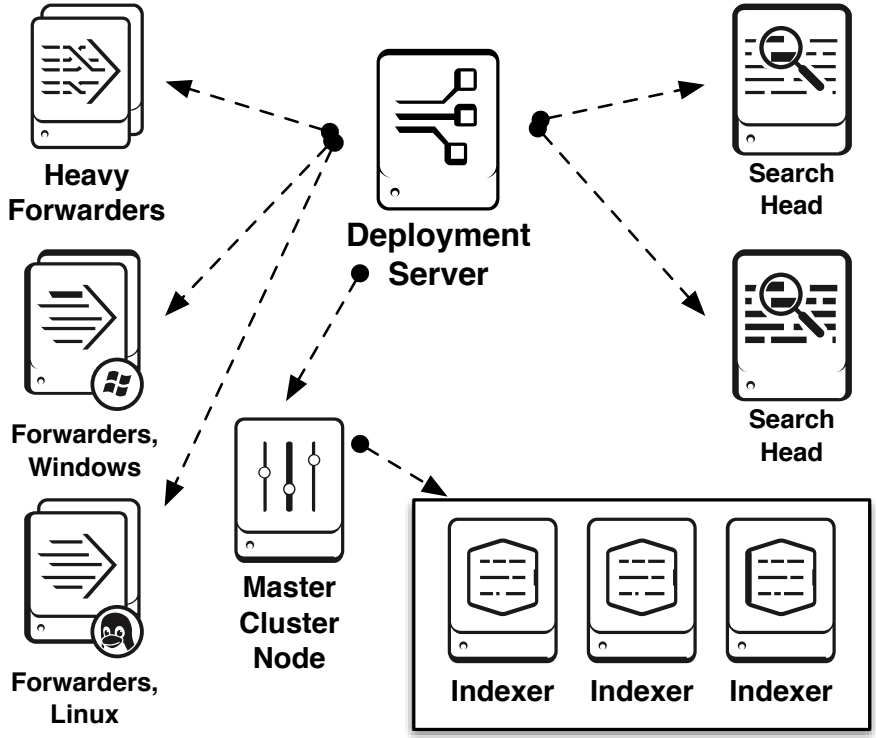
- Use naming conventions and be consistent
- Utilize a staging search head node to test changes to apps before moving them to the deployer
- When moving from a distributed architecture to a clustered architecture always ensure that only one distribution manager is managing a Splunk node
- Use descriptive/dated comments in your custom configuration files
- Use version control

Indexer Cluster Management

splunk>

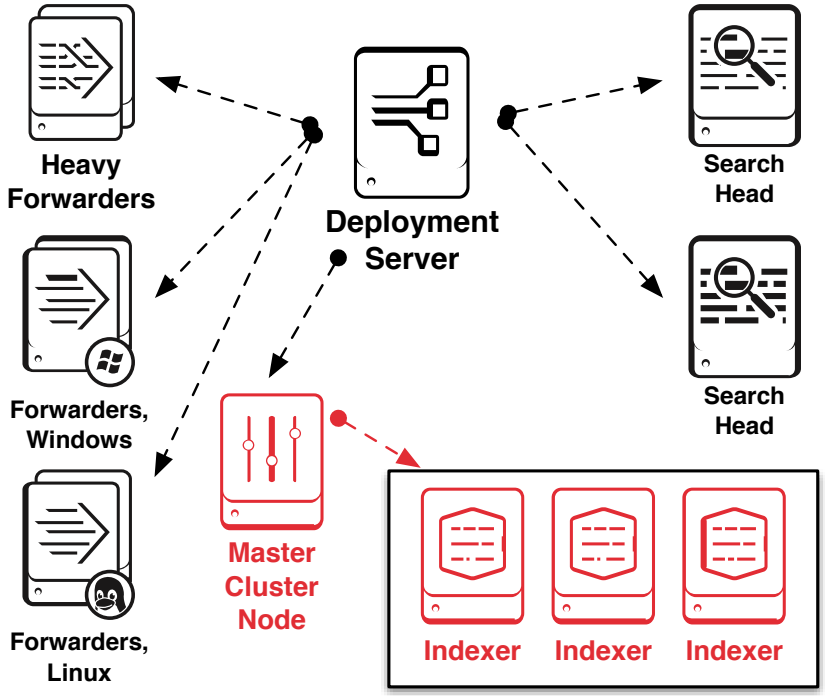
Indexer Clustering

- ALL apps from /etc/master-apps/ get pushed to /etc/slave-apps/
- Cluster replication is managed within the indexer cluster by master cluster node
- The master node replicates copies of data, not configurations, based on replication and search factor settings
- Multi-site clustering and GUI management available



The Master Cluster Node

- Master cluster node distributes configurations to cluster members
- Apps that contain configurations are stored in $SPLUNK_HOME/etc/master-apps/ on the master node
- Cluster bundles can be applied via Splunk Web or the CLI
- Bucket replication status and search factor status available in Splunk Web via Settings | Indexer clustering



Differences of Indexer Cluster Management

- Indexer cluster configuration management differs in several ways
  - Configurations (apps) are only distributed to the cluster members when:
    - "splunk apply cluster-bundle" is run from the CLI
    - the Distribute Configuration Bundle button is clicked in Splunk Web
    - a cluster member joins or rejoins the cluster and it polls the master node for any updates
  - Configuration changes are not replicated between members; changes are managed only from the master node
  - Copies of the data and searchable copies of the data are replicated between members

Best Practices

- Use naming conventions and be consistent
- Be aware that certain configuration changes within a distribution bundle cause the master node in a cluster to initiate a rolling restart of the peers
- When moving from a distributed architecture to a clustered architecture always ensure that only one distribution manager is managing a Splunk node
- Use descriptive/dated comments in your configuration files
- Use version control

Clustering: Distributed Search and Bundles

- Distributed search: all props (search and index-time) are distributed to indexers and search heads
  - Each side knows which to use based upon its role (what it's doing) and it doesn't care if the others are there
    - Search heads don't index, so they don't touch those props
    - Indexer uses index-time props, but is provided search-time rules from the search head

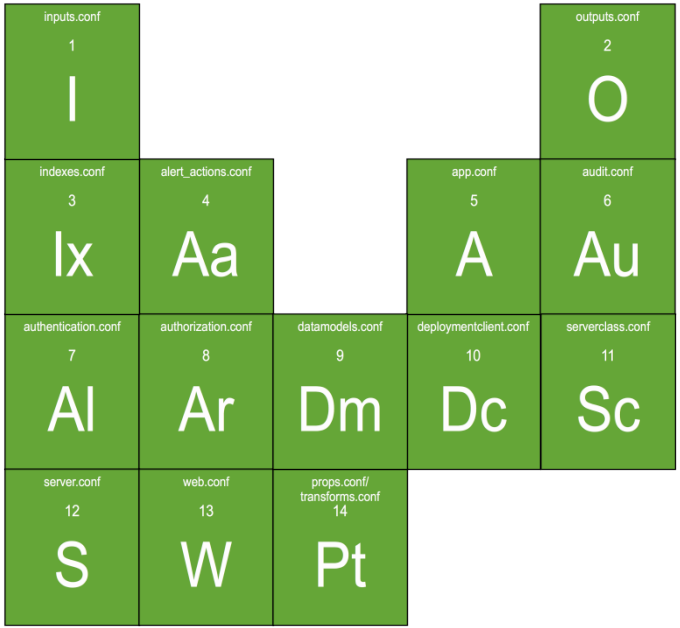
Best Practices and Version Control

splunk>

Create smaller, more discrete apps

- Keep the number of configuration files per app low
  - This creates smaller, reusable modules and lets you take advantage of Splunk's configuration layering
  - Easier to debug
- Use a naming convention for the apps
  - Example: DS_<org group>_<class>_<config_file>
    DS_dmz_forwarder_outputs
- Create classes of apps
  - Input apps
  - Index apps
  - Web control apps
- The more places configuration files need to be managed, the more likelihood for error and/or confusion

Atomic Apps Combine To Make Larger Config Molecules

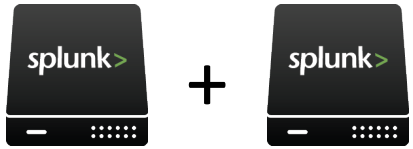


Why not larger apps?

- Very hard to reuse
- Configurations quickly become clumsy
- Makes debugging problems more difficult
- Not as flexible

Combining Instance Roles

- Splunk full instances can serve more than one function
  - Deployment server & license master
    - Depends on number of clients/load on the deployment server
  - Deployment server and master cluster node
    - /etc/master-apps/ management becomes a manual process
    - The deployment server cannot be a client of itself
  - Search head cluster deployer and master cluster node
    - Deployer and master cluster node are explicitly YES in current versions and beyond



Source/Version Control

- Source control is a good thing, use it!
  - It gives you a place to revert to if there are problems. Many customers already have process around change control, and this can reinforce those practices.
  - GIT, SVN, etc.
- Promoting from Dev or QA into Production
  - For customers who have different “lanes” or “stages”, they can have a dev repository, test it, deploy by whatever mechanism, validate it and then check the diff in for the next lane (i.e., promote it to production).

Alternate Configuration Management Tools

- If you already have a configuration management system in place you can use that system
  - Build with our best practices in mind
  - Easier to understand / maintain the aggregate state
- Rather than having to figure out “which pieces went into server.conf”, you can look at the app manifest and see from their clear(!) naming convention what’s going to be there
- Also helpful for managing large numbers of endpoint forwarder changes

What Now?

Related breakout sessions and activities…

- **Onboarding Data Into Splunk** : Tuesday Sep. 22nd
  - **Andrew Duca**, Principal Consultant, Splunk
- **Search Head Clustering**: Tuesday Sep. 22nd
  - **Manu Jose**, Senior Engineer, Splunk
  - **Eric Woo**, Senior Engineer, Splunk
- **Simplified Forwarder Deployment and Deployment Server Techniques** : Tuesday Sep. 22nd
  - **Cary Petterborg**, Senior Monitoring Engineer, LDS
- **Indexer Clustering Best Practices, Tips and Tricks** : Wednesday Sep. 23rd
  - **Da Xu**, Senior Software Engineer, Splunk
- **Splunk App Certification Criteria** : Wednesday Sep. 23rd
  - **Damien Dallimore**, Developer Evangelist, Splunk
- **Splunk Configuration Management and Deployment with Ansible** : Wednesday Sep. 23rd
  - **Sean Delaney**, Client Architect, Splunk
    Jose Hernandez, Director of Big Data Security Solutions, Zenedge
- **Add-On Best Practices Check Tool** : Thursday Sep. 24th
  - **Brian Wooden**, Senior Manager of Security Add-Ons, Splunk
  - **Jack Coates**, Director, Product Management, Splunk

THANK YOU

splunk>