Splunk Cluster Administration - Class Lab Exercises

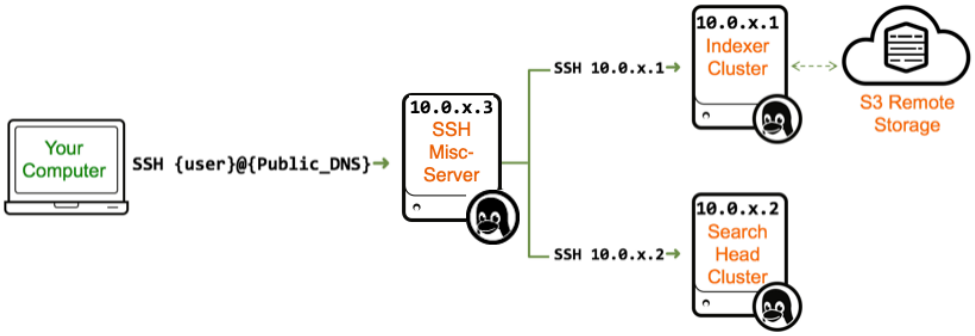
Training Lab Environment

Throughout the course, you will be working in a private network environment. Your instructor will provide the following information to configure and test your Splunk Cluster environment:

- Student ID {**x**}: .....
- Public address {**Public_DNS**}: ..... **C<session>-<x>.class.splunk.com**
- SSH user name {**user**}: .....
- SSH password {**password**}: .....
- Splunk user name: ..... **admin**
- Splunk user password: ..... **same as SSH user password**

NOTE: Your SSH password and Splunk **admin** password are the same on all instances.

You will use the Splunk CLI for configuration tasks and use Splunk Web for monitoring and verification tasks. To access your Splunk CLI consoles, you will first SSH into your designated Misc-Server address. From there, you will remote-SSH into other nodes in the network using their private IP addresses.



Your Misc-Server is also configured to be the reverse-proxy web server for your Splunk Web instances. To access a particular instance, go to **https://{Public_DNS}/{splunk server name}**. For example, to access Splunk Web for **cmanager**, direct your web browser to **https://{Public_DNS}/cmanager**.

Module 1 Lab Exercise – Configure Splunk License Manager

Description

In this exercise, you will perform basic discovery tasks to learn about your lab environment and start the Splunk License Manager.

Steps

Task 1: Access your designated Splunk environment.

1. SSH to the Misc-Server with the credentials provided.

```
ssh os_user@{Public_DNS of your Misc-Server}
os_user@Cnnnnn-x.class.splunk.com's password: {SSH user password}

[os_user@ip-10-0-x-3 ~]$
```

NOTE: Use the **SSH password** given to you at the beginning of the class.

2. Check the prompt and verify your student ID and the host you are on:

   The third number of the prompt represents your student ID. (**x** in this example)

   The last number of the prompt indicates the current node you are on:
   - **-1** is your indexer cluster and referenced as **IDX-Cluster** in this lab exercise.
   - **-2** is your search head cluster and referenced as **SH-Cluster** in this lab exercise.
   - **-3** hosts miscellaneous servers and referenced as **Misc-Server** in this lab exercise.

3. Type **pwd** to display your current directory and type **ls** to list the instances on this server.

```
[os_user@ip-10-0-x-3 ~]$
pwd
/opt/home/os_user

ls
cmanager  dserver  fwdr
```

Task 2: Set up password-less SSH connection to IDX-Cluster and SH-Cluster.

4. For convenience, set up password-less SSH connections using ssh keys from your **Misc-Server**.

NOTE: This allows remote-ssh to your IDX-Cluster and SH-Cluster without entering a password in this course. Your production environment might have a more secure implementation.

5. Generate and share the public key with the IDX-Cluster and SH-Cluster nodes using **scp**.

```
[os_user@ip-10-0-x-3 ~]$
ssh-keygen -t rsa -P ""
 Generating public/private rsa key pair.
 Enter file in which to save the key (/opt/home/os_user/.ssh/id_rsa):
   Press Enter to accept the default value

Created directory '/opt/home/os_user/.ssh'.
 Your identification has been saved in /opt/home/os_user/.ssh/id_rsa.
 Your public key has been saved in /opt/home/os_user/.ssh/id_rsa.pub.
 The key fingerprint is: ...
 The key's randomart image is:
 +--[ RSA 2048]----+
 |                 |
 |               ..|
 |          + .  .+|
 |         o = + =o|
 |        S o = =.o|
 |           o oooE|
 |            ..oo.|
 |             .o o|
 |            .. o |
 +-----------------+

cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys

chmod 600 ~/.ssh/authorized_keys

scp -r ~/.ssh 10.0.X.1:~/
 The authenticity of host '10.0.x.1 (10.0.x.1)' can't be established.
 ECDSA key fingerprint is fa:57:03:83:e0:58:ba:29:d5:68:e3:75:2e:8c:9f:1d.
 Are you sure you want to continue connecting (yes/no)? yes
 Warning: Permanently added '10.0.x.1' (ECDSA) to the list of known hosts.
 os_user@10.0.x.1's password: {use your SSH password}
 known_hosts                                 100%  444     0.4KB/s   00:00
 id_rsa.pub                                  100%  398     0.4KB/s   00:00
 authorized_keys                             100%  398     0.4KB/s   00:00
 id_rsa                                      100% 1675     1.6KB/s   00:00

scp -r ~/.ssh 10.0.x.2:~/
 The authenticity of host '10.0.x.2 (10.0.x.2)' can't be established.
 ECDSA key fingerprint is fa:57:03:83:e0:58:ba:29:d5:68:e3:75:2e:8c:9f:1d.
 Are you sure you want to continue connecting (yes/no)? yes
 Warning: Permanently added '10.0.x.2' (ECDSA) to the list of known hosts.
 os_user@10.0.x.2's password: {use your SSH password}
 ...
```

From this point onwards, you should be able to **ssh** to each node without entering the password.

6. Remote-ssh to your **IDX-Cluster**, list the instances on the node, and exit back to **Misc-Server**.

```
[os_user@ip-10-0-x-3 ~]$
ssh 10.0.x.1

[os_user@ip-10-0-x-1 ~]$
ls
idx1  idx2  idx3  idx4

exit
logout
Connection to 10.0.x.1 closed.
```

7. Remote-ssh to your **SH-Cluster**, list the instances on the node, and exit back to **Misc-Server**.

```
[os_user@ip-10-0-x-3 ~]$
ssh 10.0.x.2

[os_user@ip-10-0-x-2 ~]$
ls
sh1  sh2  sh3  sh4  sh5

exit
logout
Connection to 10.0.x.2 closed.
```

Task 3: Configure your License Manager instance.

8. In the **Misc-Server** session, start the **`dserver`** Splunk instance and check its **`servername`**, **`splunkd-port`**, and **`web-port`**.

```
[os_user@ip-10-0-x-3 ~]$
~/dserver/bin/splunk start --accept-license
This appears to be your first time running this version of Splunk.
...
The Splunk web interface is at http://ip-10-0-x-3:8100

~/dserver/bin/splunk show servername
 Splunk username: admin
Password: {Splunk user password}
Server name: dserver

~/dserver/bin/splunk show splunkd-port
Splunkd port: 8189

~/dserver/bin/splunk show web-port
Web port: 8100
```

© 2021 Splunk Inc. All rights reserved.

9. Configure the **dserver** instance as the License Manager for your deployment by adding the license file **splunk.license.big.license** from the **/opt/license** directory.

```
~/dserver/bin/splunk add licenses /opt/license/splunk.license.big.license
The licenses object has been added. You need to restart the Splunk Server
(splunkd) for your changes to take effect.

~/dserver/bin/splunk restart
```

Check Your Work

Task 4: Confirm the license information.

10. Use your web browser to access the Splunk Web interface of your License Manager (**dserver**).

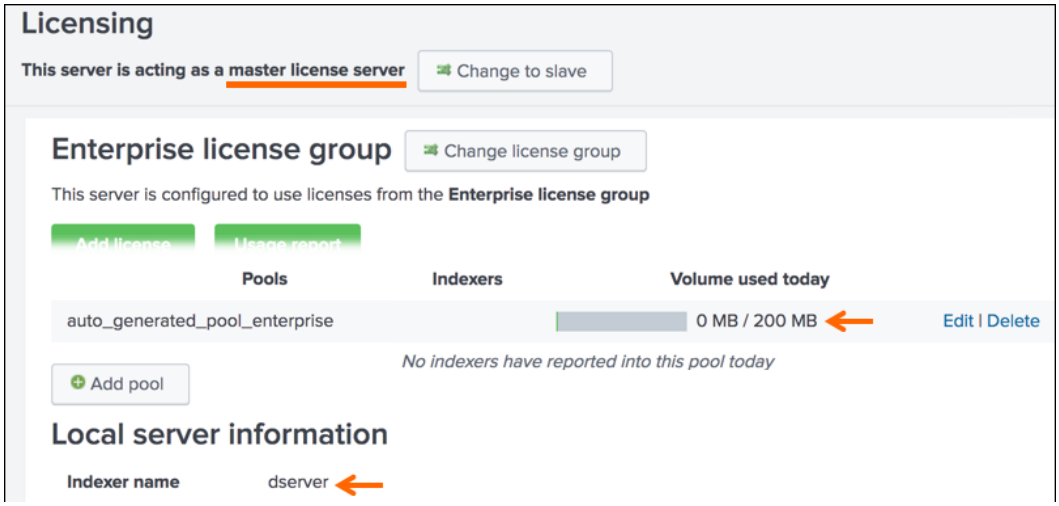
    https://{your assigned Public_DNS}/dserver

NOTE:	In this lab environment, a proxy web server in the Misc-Server has mapped all your Splunk Web instances to **https://{Public_DNS}/{servername}**.

11. Log in as **admin** using your assigned Splunk password.

NOTE:	When you see the prompt *Help us improve Splunk software*, click **OK**. When you see the *Important changes coming*, click **Don't show me this again**.

12. Click **Settings** > **Licensing** and verify the information on the **Licensing** page.



Troubleshooting Suggestions

If your configuration is not returning the expected results, troubleshoot by isolating the issue.

1. Verify the command syntax and spelling.

   ```
   [os_user@ip-10-0-x-3 ~]$ ~/dserver/bin/splunk btool check --debug
   ```

2. Check **splunkd.log** for any errors:

   ```
   tail -50 ~/dserver/var/log/splunk/splunkd.log
   ```
   Or, **egrep 'ERROR|WARN' ~/dserver/var/log/splunk/splunkd.log**

3. Compare the output of **~/dserver/etc/system/local/server.conf** with **lab_conf_outputs.txt**.

© 2021 Splunk Inc. All rights reserved.

Module 2 Lab Exercise – Enable Single-site Indexer Cluster

Description

In this exercise, you will configure a Splunk single-site indexer cluster with three peer nodes and one search head. You will then simulate a peer node failover scenario.

Steps

Task 1: Configure the manager node for a single-site indexer cluster.

1. In the **Misc-Server** `ssh` session, navigate to the `cmanager` Splunk instance and start Splunk.

```
[os_user@ip-10-0-x-3 ~]$
~/cmanager/bin/splunk start --accept-license
This appears to be your first time running this version of Splunk...
The Splunk web interface is at http://ip-10-0-x-3:8000
```

2. Check its `servername`, `splunkd-port`, and `web-port`.

```
~/cmanager/bin/splunk show servername
Splunk username: admin
Password:
Server name: cmanager

~/cmanager/bin/splunk show splunkd-port
Splunkd port: 8089

~/cmanager/bin/splunk show web-port
Web port: 8000
```

3. Configure `cmanager` to be a license client to `dserver` (`10.0.x.3:8189`):

```
~/cmanager/bin/splunk edit licenser-localslave -master_uri https://10.0.x.3:8189
The licenser-localslave object has been edited.
You need to restart the Splunk Server (splunkd)...
```

4. Configure `cmanager` to be the manager node with the following indexer cluster options:
   - `replication_factor` `2`
   - `search_factor` `2`
   - `secret` `idxcluster`

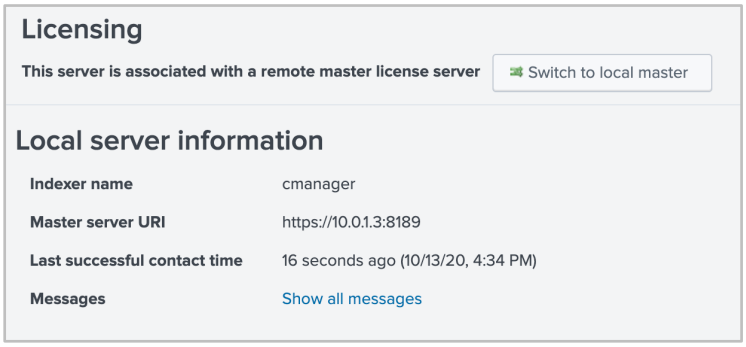
```
~/cmanager/bin/splunk edit cluster-config -mode manager -replication_factor 2
-search_factor 2 -secret idxcluster
... The cluster-config property has been edited...
You need to restart...

~/cmanager/bin/splunk restart
```

Task 2: Monitor the cluster status from cmanager's Splunk Web.

5. Access the Splunk Web interface for **`cmanager`**: **`https://{Public_DNS}/cmanager`**

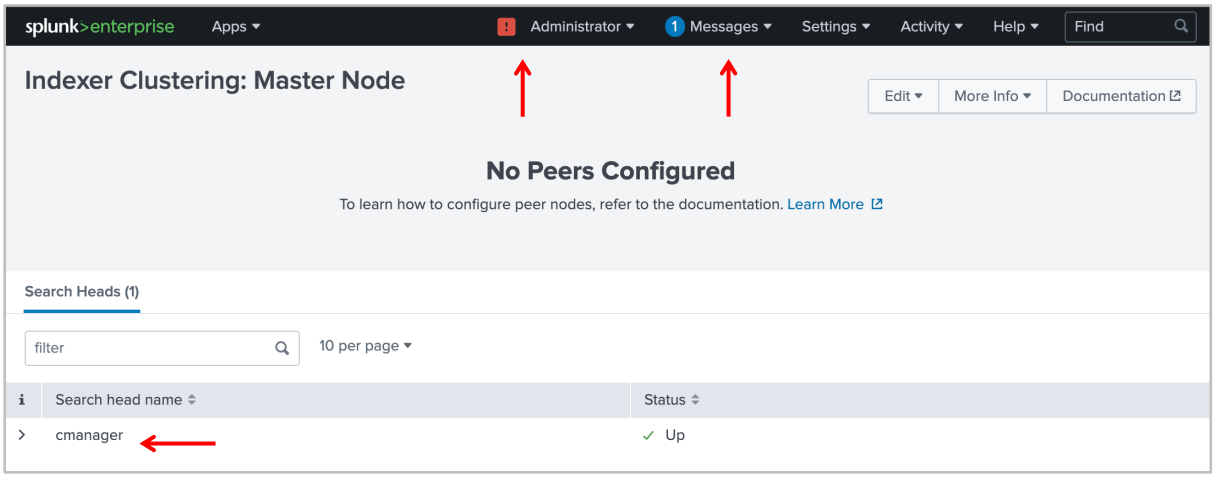
6. Log into Splunk Web as **`admin`** using your assigned password.

7. Navigate to **Settings** > **Licensing** and review the information on the **Licensing** page.

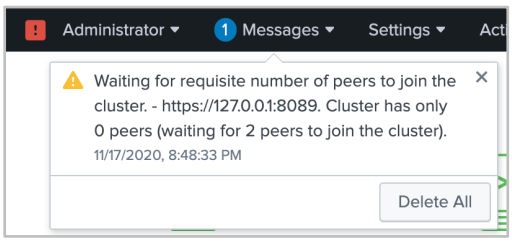


8. Navigate to **Settings** > **Indexer clustering** and monitor the cluster status.

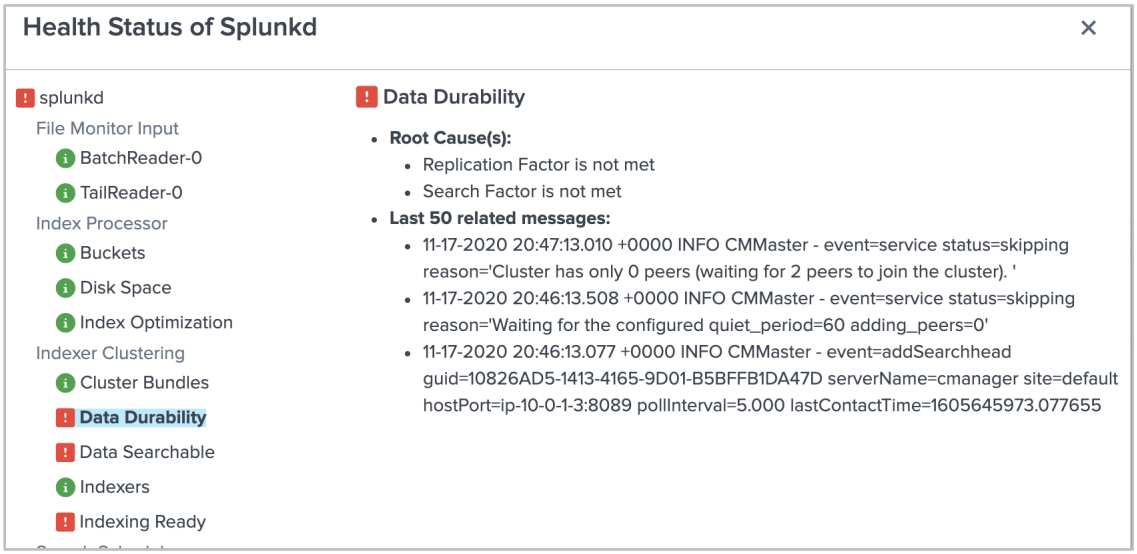
   `cmanager` is listed as the only search head. The health status indicator is red and there is a message notification.



9. Click the **Messages** drop-down menu and read the message about the requisite number of peers required to join cluster. Acknowledge the message by deleting it.



10. Click the red **Health Status** icon and drill down each red indicator to learn more about the status. Note that they are all related to health and status of the cluster.



11. Close the **Health Status** page but keep the **Indexer Clustering: Master Node** page in view.

Task 3: Configure three indexers to form the replication peers for the single-site indexer cluster.

Each indexing node in a production environment must run on a dedicated host. However, to simulate a working cluster in this lab environment, each Linux host has been configured to run multiple Splunk instances. To accommodate this, each instance has been carefully assigned unique port numbers.

Reference the following port matrix when you configure each indexer instance:

Server Name	Splunkd-port	Web-port	Listening-port	Replication-port
idx1	8189	8100	9197	9100
idx2	8289	8200	9297	9200
idx3	8389	8300	9397	9300

12. To form a cluster of indexing peers, remote-ssh to **IDX-Cluster**.
13. Bring up **`idx1`**, **`idx2`**, and **`idx3`** and perform the following:
    a. Configure each indexer peer as a License Manager to **`dserver`**.
    b. To receive forwarder data, configure a listening port on each peer.
    c. To save CPU resource, disable Splunk Web on each peer.
    d. Configure each instance to join the indexer cluster.
    e. Monitor the **Indexer Clustering: Master Node** page as you bring up each node.

NOTE: Be sure to use the same secret that the manager node has used.
Defer restarts until you configure the cluster setting.

```
[os_user@ip-10-0-x-3 ~]$
ssh 10.0.x.1

[os_user@ip-10-0-x-1 ~]$
~/idx1/bin/splunk start --accept-license

~/idx1/bin/splunk edit licenser-localslave -master_uri https://10.0.x.3:8189
Splunk username: admin
Password:
The licenser-localslave object has been edited. You need to restart the
Splunk Server (splunkd) for your changes to take effect.

~/idx1/bin/splunk enable listen 9197
Listening for Splunk data on TCP port 9197.

~/idx1/bin/splunk disable webserver
You need to restart the Splunk Server (splunkd) for your changes to take
effect.

~/idx1/bin/splunk edit cluster-config -mode peer -master_uri https://10.0.x.3:8089
-secret idxcluster -replication_port 9100
The cluster-config property has been edited.
You need to restart the Splunk Server (splunkd) for your changes to take
effect.

~/idx1/bin/splunk restart
```

14. Confirm the **`idx1`** peer node has appeared on the **`cmanager`**'s indexer clustering view.

15. Repeat the steps to configure **`idx2`**:

```
[os_user@ip-10-0-x-1 ~]$
~/idx2/bin/splunk start --accept-license

~/idx2/bin/splunk edit licenser-localslave -master_uri https://10.0.x.3:8189

~/idx2/bin/splunk enable listen 9297

~/idx2/bin/splunk disable webserver

~/idx2/bin/splunk edit cluster-config -mode peer -master_uri https://10.0.x.3:8089
-secret idxcluster -replication_port 9200

~/idx2/bin/splunk restart
```

16. Repeat the steps to configure **idx3**:

```
~/idx3/bin/splunk start --accept-license

~/idx3/bin/splunk edit licenser-localslave -master_uri https://10.0.x.3:8189

~/idx3/bin/splunk enable listen 9397

~/idx3/bin/splunk disable webserver

~/idx3/bin/splunk edit cluster-config -mode peer -master_uri https://10.0.x.3:8089
-secret idxcluster -replication_port 9300

~/idx3/bin/splunk restart
```

Task 4: Configure a search head to join the cluster.

17. Connect to the **SH-Cluster** session and identify what instances are on the host.

```
[os_user@ip-10-0-x-1 bin]$
ssh 10.0.x.2
```

18. To add a search head to the indexer cluster, bring up **sh1**:
    a. Configure it as a License Peer to **dserver**.
    b. Configure it to be the search head of the indexer cluster.

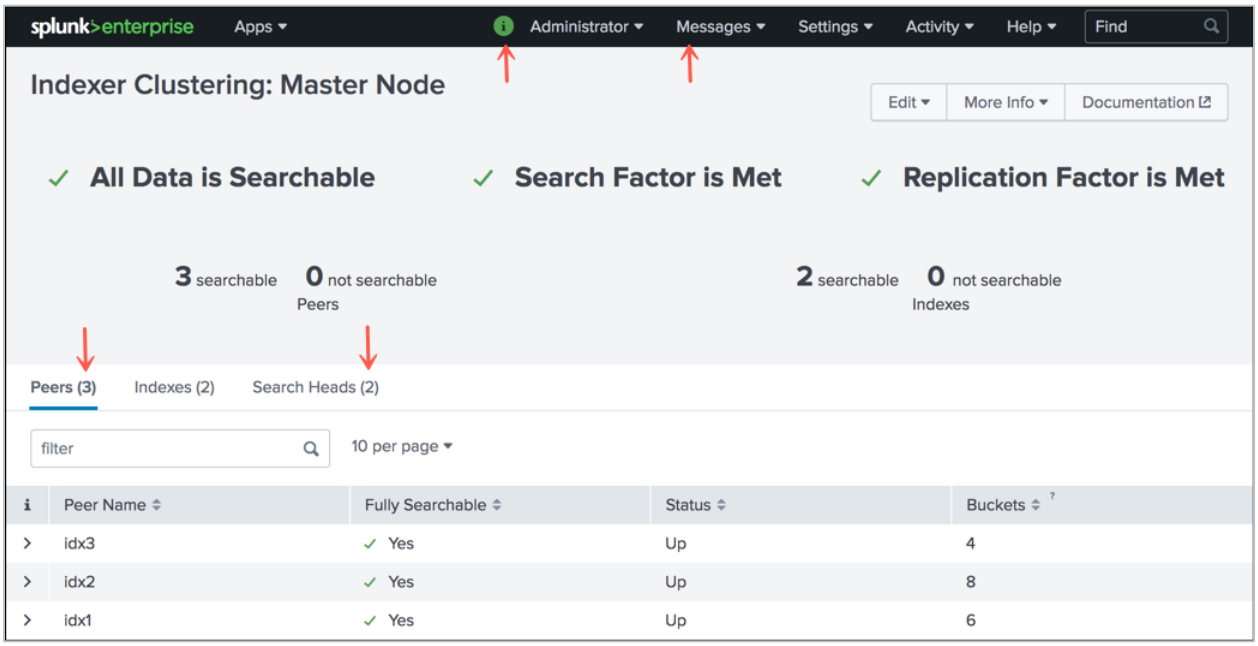
NOTE: Don't forget to use the same secret that the Manager Node has used.

```
[os_user@ip-10-0-x-2 ~]$
~/sh1/bin/splunk start --accept-license
~/sh1/bin/splunk edit licenser-localslave -master_uri https://10.0.x.3:8189
~/sh1/bin/splunk edit cluster-config -master_uri https://10.0.x.3:8089 -mode
searchhead -secret idxcluster
~/sh1/bin/splunk restart
exit

[os_user@ip-10-0-x-1 bin]$
exit

[os_user@ip-10-0-x-3 bin]$
```

Check Your Work

Task 5: Verify that your indexer cluster is functioning properly.

19. Access **cmanager**'s **Indexer Clustering: Master Node** page and confirm the cluster status.
    - The Health Status indicator is green.
    - The message about the cluster has cleared.
    - The **Indexer Clustering: Master Node** page lists 3 peers and 2 search heads.



NOTE:	If any status indicator is NOT green, then check the configuration of the missing node(s) before you proceed to the next task.

20. Open a new browser tab and go to: **`https://{Public_DNS}/sh1`**

    Log into Splunk Web as **`admin`** using your assigned password.

21. Navigate to **Settings** > **Distributed search** > **Search peers.**

    Notice the search peers are automatically added.

22. In the **Search & Reporting** app, search **`index=_audit`** over **Last 24 hours**.

    a. Click **`host`** and see that the results contain events from **`idx1`**, **`idx2`**, **`idx3`**, and **`sh1`**.

    b. Click **`splunk_server`** and see that the results are from **`idx1`**, **`idx2`**, **`idx3`**, and **`sh1`**.

© 2021 Splunk Inc. All rights reserved.

Configuration Troubleshooting Suggestions

1. Verify the command syntax and spelling on each instance with: **`splunk btool check --debug`**
2. In the manager node's Splunk Web, search for any cluster errors (adjust the time range):

   ```
   index=_internal sourcetype=splunkd (warn OR error) component=CM*
   ```

3. Check **`splunkd.log`** of each instance for any errors:

   ```
   egrep 'ERROR|WARN' ~/<instance_name>/var/log/splunk/splunkd.log
   ```

4. Compare the output of **`.conf`** files with **`lab_conf_outputs.txt`**.

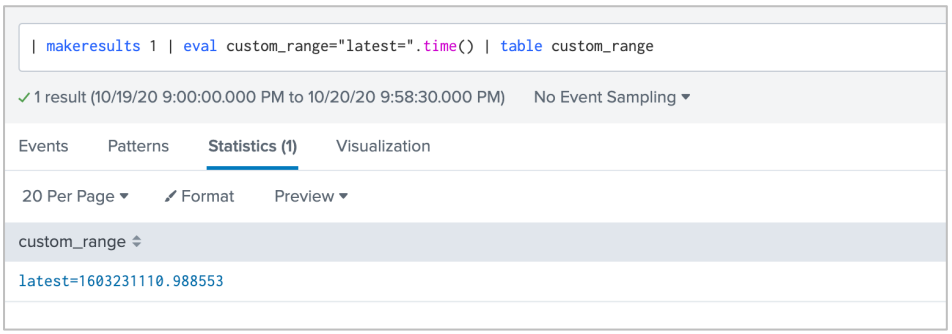
> If you are done with the configuration and have about 15 minutes to spare, try this optional failover test.

Optional Peer Node Failover Test Steps

Task 6: Test a peer node failover scenario.

23. From **`sh1`**, run the following search to establish a baseline search window (Last 24 hours):

    ```
    | makeresults 1 | eval custom_range="latest=".time() | table custom_range
    ```



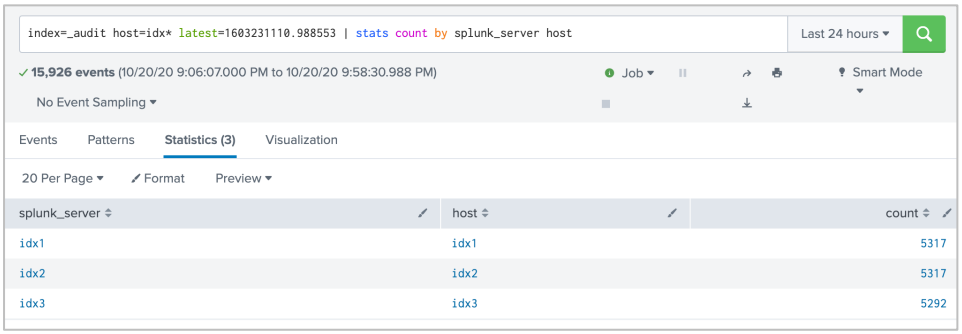
> **NOTE:** You cannot start a search with the **`eval`** command. To work around this limitation, use the **`makeresults`** command and create a dummy event. The above search generates a fixed search window that you use to validate the subsequent test.

24. Copy the calculated search window value (**`custom_range`**) from the result.

    Example above: **`latest=1603231110.988553`**

25. Run a baseline search that displays the event count per host and its data provider:

```
index=_audit host=idx* {copied value} | stats count by splunk_server host
```



NOTE: Note the count values. Your result will vary.

26. On **cmanager**'s browser window, navigate to **Settings** > **Indexer clustering** and monitor the page.

    https://{Public_DNS}/cmanager/en-US/manager/system/clustering

27. To simulate a peer failure, connect to the **IDX-Cluster** ssh session.

28. Identify the parent **splunkd** process of **idx1** and stop the process.

NOTE: This is only to simulate a peer failure in this lab environment. Your cluster peer nodes are running on a single host.

```
[os_user@ip-10-0-x-3 ~]$
ssh 10.0.x.1

[os_user@ip-10-0-x-1 ~]$
ps -ef | grep "splunkd -p 8189"
user 25586 1 0 ? 00:12:47 splunkd -p 8189 restart
user 25587 25586 0 ? 00:00:15 [splunkd pid=25586] splunkd -p 8189 restart
[process-runner]

kill 25586
```

29. In the cluster status page, wait until the status of **idx1** becomes **Down** but all data is searchable.

NOTE: The **idx1** status should transition from **Up** > **Pending** > **Shutting down** while the cluster tries to recover. The search capability is ready when you see the green checkmark next to **All Data is Searchable** and the status of **idx1** is **Stopped**. The fixup process begins and eventually the cluster will reach the complete state.

30. Re-run the exact search (on **sh1**) from Step 25.

```
index=_audit host=idx* {copied value} | stats count by splunk_server host
```

NOTE: The result still contains events from all peer nodes but the data providers are different now.

✓ 11,431 events (9/25/18 9:58:37.000 PM to 9/25/18 10:54:46.947 PM) No Event Sampling Job Smart Mode

Events Patterns **Statistics (3)** Visualization

20 Per Page Format Preview

splunk_server	host	count
idx2	idx1	3815
idx2	idx2	3816
idx3	idx3	3800

The **idx1** count doesn't exactly match the original count. Why?

31. To restore the full cluster, start the Splunk instance for **idx1** in the **IDX-Cluster** ssh session.

```
[os_user@ip-10-0-x-1 ~]$
~/idx1/bin/splunk start
exit
```

32. Check the cluster status and verify that all peers are **Up**.

33. Re-run the exact search again and verify the result.

```
index=_audit host=idx* {copied value} | stats count by splunk_server host
```

NOTE: The total counts for each host should now match your initial results. The search you ran while **idx1** was down only includes replicated data from **idx1**, not the events that were indexed before you implemented indexer clustering. This search once again includes the non-replicated events from **idx1**.

✓ 11,457 events (9/25/18 9:55:22.000 PM to 9/25/18 10:54:46.947 PM) No Event Sampling Job Smart Mode

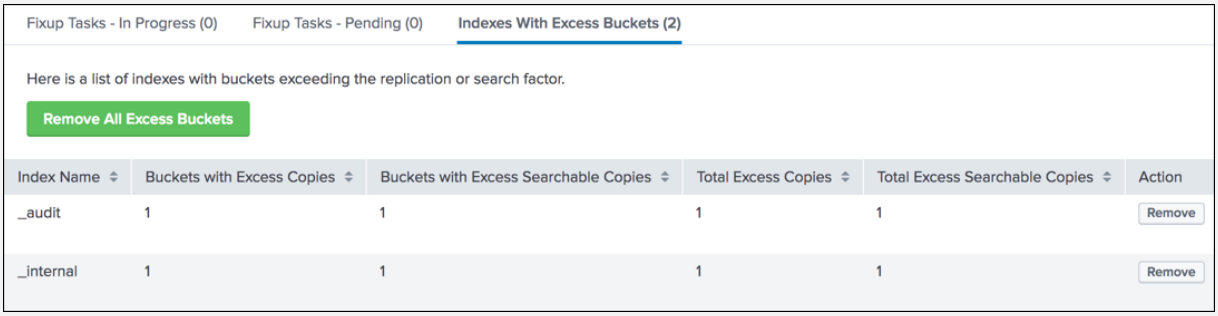
Events Patterns **Statistics (4)** Visualization

20 Per Page Format Preview

splunk_server	host	count
idx1	idx1	3841
idx2	idx2	31
idx3	idx2	3785
idx3	idx3	3800

34. On the **cmanager**'s **Indexer clustering** page, click the **Indexes** tab > the **Bucket Status** button > the **Indexes With Excess Buckets** tab.

NOTE: The page lists the number of buckets exceeding the replication or search factor per index.



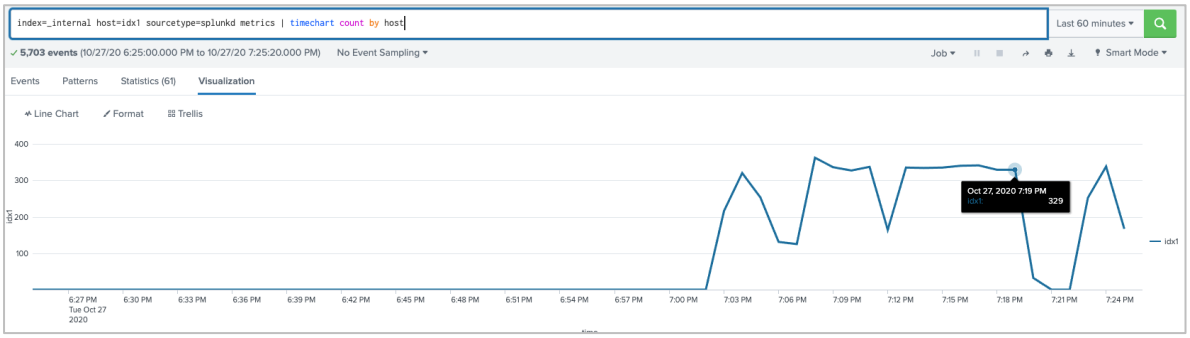
35. To reclaim the storage space, click **Remove All Excess Buckets** > **Confirm**.

36. Reload the browser page and verify that the excess copies are removed.

Task 7: Investigate the peer outage with Splunk internal logs.

From **cmanager**, search the **metrics.log** of **idx1** to determine the time of outage (Last 60 minutes).

```
index=_internal host=idx1 sourcetype=splunkd metrics | timechart count by host
```



NOTE: Look for the drop-off in indexed events for **idx1**.

37. Mouse over **idx1**'s trendline at the point immediately before the dropoff, then click the square in the pop-up that displays the timestamp and count.

    A drilldown search executes and displays the events during that time period.

38. Click the time range picker and select the **Date & Time Range** tab.

39. Change the scope from **Between** to **Since** and then click **Apply**.

40. To display the fixup summary, search using your selected time range (Since date time):

```
index=_internal sourcetype=splunkd_access uri_path="/services/cluster*"
| timechart values(uri_path) by host
```

NOTE: Depending on the time range of your search, a default span value is used to group the results. You can include the **span** option to adjust as necessary (**span=1m** for example). The results of this search cannot be charted on the Visualizations tab because you are creating multi-valued fields using the **values** function. To see the fixup tasks, look in the **idx2** and **idx3** columns.

41. To identify the outage and recovery from the manager node, search using your selected time range:

```
index=_internal sourcetype=splunkd component=CM*
```

NOTE: To determine the time of outage, you can search for the event where the manager first detected the peer outage: (first entry)

```
index=_internal sourcetype=splunkd component=CMMaster streaming error
```

You can further investigate the cause by searching **component=CMPeer**:

```
index=_internal sourcetype=splunkd component=CMPeer
| timechart span=1m values(event_message)
```

```
 ...transitioning from=Up to=Pending reason="non-streaming failure"
 ...transitioning from=Pending to=Down reason="heartbeat or restart
    timeout=60"
```

Other notable event messages:

When did the cluster recover to the complete state again? (last entry)

```
CMReplicationRegistry Finished replication...
CMBucket event=replicationDone...
CMMaster replication success...
CMBucket event=searchableDone...
CMMaster change bucket success...
```

When did the downed peer rejoin the cluster?

```
CMMaster scheduled rebalance primaries
```

Module 3 Lab Exercise – Migrate Single-site Cluster to Multisite Cluster

Description

In this exercise, you will migrate the single-site indexer cluster you configured in the previous lab exercise to a multisite cluster with two sites. The first two indexers, **idx1** and **idx2**, will be assigned to **site1**. The third indexer, **idx3**, will be assigned to **site2**. The original search head, **sh1**, will be assigned to **site1**.

In order to meet the minimum number of nodes to form a multisite cluster with search affinity, you will add an additional indexer **idx4** and a search head **sh2** for **site2**. Once the cluster reaches the complete state, you will simulate a site failure scenario.

Steps

Task 1: Migrate the single-site manager node to the multisite mode.

1. Open the **Misc-Server** session and migrate the **cmanager** instance to support a multisite cluster.

   Use the following options:

   - Manager node site association **site1**
   - available_sites **site1 & site2**
   - site_replication_factor **origin:1,total:2**
   - site_search_factor **origin:1,total:2**
   - replication_factor **1** (it was set to **2** in module 2)
   - search_factor **1** (it was set to **2** in module 2)
   - secret **idxcluster**

```
[os_user@ip-10-0-x-3 ~]$
~/cmanager/bin/splunk edit cluster-config -mode manager -multisite true -site site1
-available_sites site1,site2 -site_replication_factor origin:1,total:2
-site_search_factor origin:1,total:2 -replication_factor 1 -search_factor 1 -secret
idxcluster

~/cmanager/bin/splunk restart
```

2. To perform the migration of peers, enable maintenance mode on the manager node.

```
~/cmanager/bin/splunk enable maintenance-mode
Warning: In maintenance mode, the cluster manager will not attempt to
replace any missing replicated or searchable bucket copies. This mode should
be enabled only while performing maintenance on peers. Do you want to
continue? [y/n]: y
... Maintenance mode set
```

NOTE: If you are creating a script, append **--answer-yes** to bypass the prompt:

splunk enable maintenance-mode --answer-yes

Task 2: Migrate the existing three peer nodes to form the multisite indexer cluster.

NOTE: Each indexing peer in a production environment must run on a dedicated host. However, to simulate a working cluster in this lab environment, each host has been configured to run multiple Splunk instances. To accommodate this simulation, each instance has been carefully assigned unique port numbers.

Reference the port matrix below to configure each indexer instance.

Server Name	Splunkd-port	Web-port	Listening-port	Replication-port
idx1	8189	8100	9197	9100
idx2	8289	8200	9297	9200
idx3	8389	8300	9397	9300
idx4	**8489**	**8400**	**9497**	**9400**

3. Connect to the **IDX-Cluster** session and convert **`idx1`** and **`idx2`** to be **`site1`** peer nodes.

```
[os_user@ip-10-0-x-3 ~]$
ssh 10.0.x.1

[os_user@ip-10-0-x-1 ~]$
~/idx1/bin/splunk edit cluster-config -site site1

~/idx1/bin/splunk restart

~/idx2/bin/splunk edit cluster-config -site site1

~/idx2/bin/splunk restart
```

4. On the **`cmanager`**'s **Indexer clustering** page, confirm that both **`idx1`** and **`idx2`** are members of **`site1`**. **`idx3`** is not listed yet as it has not been assigned.

5. Convert **`idx3`** to be a **`site2`** peer node.

```
~/idx3/bin/splunk edit cluster-config -site site2

~/idx3/bin/splunk restart
```

© 2021 Splunk Inc. All rights reserved.

6. Launch **idx4** and configure it as a **site2** peer node.

```
~/idx4/bin/splunk start --accept-license

~/idx4/bin/splunk edit licenser-localslave -master_uri https://10.0.x.3:8189

~/idx4/bin/splunk enable listen 9497

~/idx4/bin/splunk disable webserver

~/idx4/bin/splunk edit cluster-config -master_uri https://10.0.x.3:8089 -mode peer
-site site2 -replication_port 9400 -secret idxcluster

~/idx4/bin/splunk restart

exit
```

7. To perform the deferred fixups and rebalancing, disable maintenance mode on the master node.

```
[os_user@ip-10-0-x-3 ~]$
~/cmanager/bin/splunk disable maintenance-mode
```

8. Confirm that all 4 peer nodes are now listed on the **cmanager**'s **Indexer clustering** page.

Task 3: Convert sh1 to be a multisite search head for site1.

9. Connect to the **SH-Cluster** session.
10. Migrate the existing search head **sh1** to **multisite** mode and set it as the search head for **site1**.

```
[os_user@ip-10-0-x-3 ~]$
ssh 10.0.x.2

[os_user@ip-10-0-x-2 ~]$
~/sh1/bin/splunk edit cluster-master https://10.0.x.3:8089 -multisite true
-site site1
```

© 2021 Splunk Inc. All rights reserved.

Task 4: Configure `sh2` as the search head for `site2`.

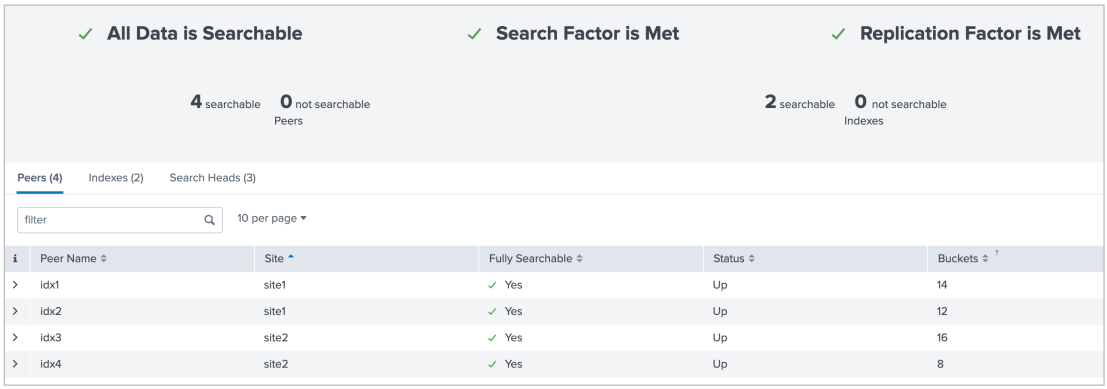
11. Start search head **`sh2`** in **`multisite`** mode and configure it as the search head for **`site2`**.

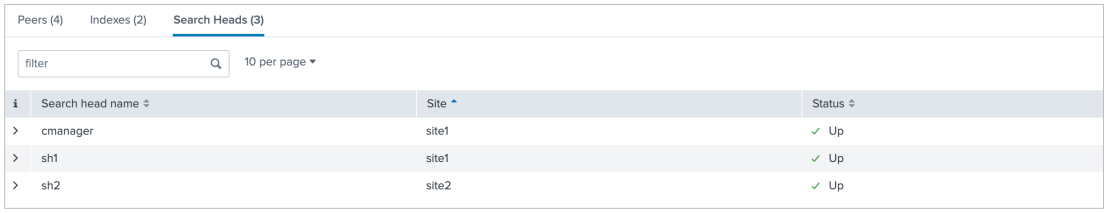
NOTE:	Don't forget to use the same indexer cluster secret that the manager node has used.

```
[os_user@ip-10-0-x-2 ~]$
~/sh2/bin/splunk start --accept-license

~/sh2/bin/splunk edit licenser-localslave -master_uri https://10.0.x.3:8189

~/sh2/bin/splunk edit cluster-config -mode searchhead -master_uri
https://10.0.x.3:8089 -site site2 -secret idxcluster

~/sh2/bin/splunk restart

exit
```

Task 5: Check the cluster status with Splunk Web.

12. Go to **`cmanager`**'s Splunk Web, **`https://{Public_DNS}/cmanager`** and check the cluster status.
    a. Navigate to **Settings** > **Indexer clustering**.
    b. Confirm that 4 peers and 3 search heads are shown, and all status indicators are green.





Check Your Work

Task 6: Run a search from sh1 to establish a search result baseline.

13. Access **sh1**'s Splunk Web by pointing the browser to **https://{Public_DNS}/sh1**.
14. Search the following in the Search app **(All time):**

```
index=_internal host=idx* component=CMSlave Maintenance mode finished | stats
latest(_time) as t1 | eval t2 = time() | eval custom_range = "starttimeu=" + t1 |
eval custom_range = custom_range + " endtimeu=" + t2 | fields custom_range
```

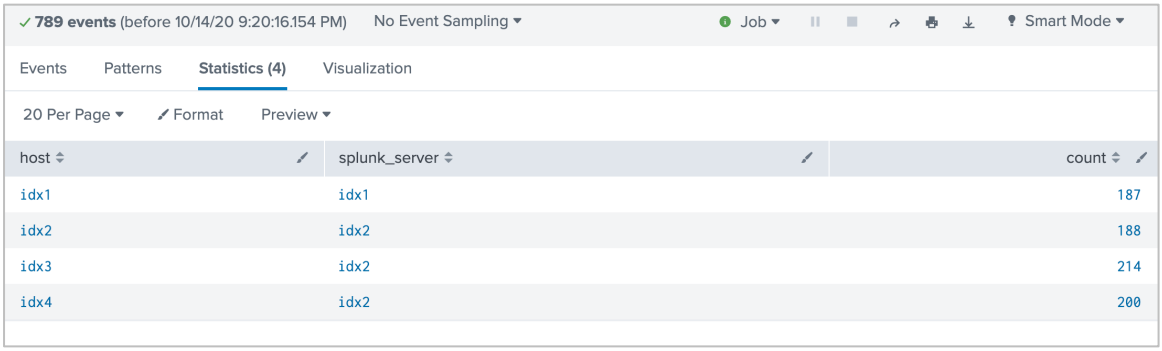
NOTE: This search example provides a fixed search window that you can use to validate a Splunk cluster site recovery test.

15. Copy the calculated search window value (**custom_range**) from the result.

    Example above: **starttimeu=1602710385.311 endtimeu=1602710416.154722**

16. Run a baseline search that displays the event count per host and its data provider:

```
index=_internal host=idx* {copied value of custom_range}
| stats count by host splunk_server | sort host
```



NOTE: Your result may vary depending on the state of replication. Run the search again until total event count no longer changes.

Configuration Troubleshooting Suggestions

If your configuration is not returning the expected results, troubleshoot by isolating the issue.

1. Verify the command syntax and spelling on each instance with: **`splunk btool check --debug`**
2. In Cluster Manager's Splunk Web, search for any cluster errors:

   `index=_internal sourcetype=splunkd (warn OR error) component=CM*`

3. Check **`splunkd.log`** of each instance for any errors:

   `tail -40 ~/<instance_name>/var/log/splunk/splunkd.log`

4. Compare the output of **`.conf`** files with **`lab_conf_outputs.txt`**.

> If you are done with the configuration and have about 15 minutes to spare, try this optional failover test.

Optional: Simulate Site1 Failover Test Steps

Task 7: Test the indexer site1 failover scenario.

> **NOTE:** The indexer cluster peers are running on a single Linux host and the following step is performed only to simulate a site failure.

17. In the **IDX-Cluster** session, identify the parent **`splunkd`** process of **`idx1`** and stop the process.

```
ssh 10.0.x.1

[os_user@ip-10-0-x-1 ~]$
ps -ef | grep "splunkd -p 8189"
 jane 25586     1 0 ?  00:12:47 splunkd -p 8189 restart
 jane 25587 25586 0 ?  00:00:15 [splunkd pid=25586] splunkd -p 8189 restart
[process-runner]

kill 25586
```

18. Identify the parent **`splunkd`** process of **`idx2`** and stop the process.

```
ps -ef | grep "splunkd -p 8289"
 jane 25875     1 0 ?  00:22:36 splunkd -p 8289 restart
 jane 25876 25875 0 ?  00:00:15 [splunkd pid=25875] splunkd -p 8289 restart

kill 25875
```

19. Check the **Indexer Clustering: Master Node** status page and confirm that the search functionality has been restored.

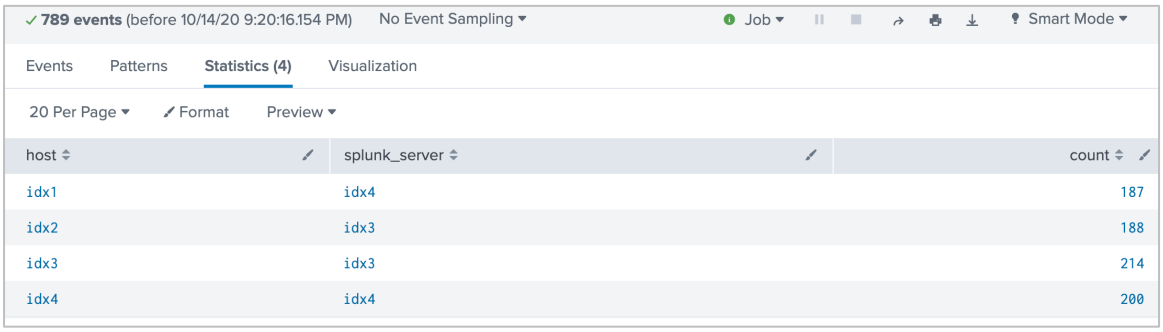
    `idx1` and **`idx2`** are marked briefly as *Shutting Down* and the cluster is not searchable. They should transition to *Stopped* and the search capability will recover quickly.

    You should see a green checkmark next to **Fully Searchable** for **`idx3`** and **`idx4`**.

    Due to the replication factors used in this cluster, the site replication factors cannot be met.

20. Go back to **`sh1`**'s Splunk Web: **`https://{Public_DNS}/sh1`**.
21. Re-run the exact search in the Search app with the same time range captured in Step 16.

```
index=_internal host=idx* {copied value of custom_range}
| stats count by host splunk_server | sort host
```



NOTE:	The results still contain events from all indexer peers and the counts should be exactly the same as before. However, now the data providers are from only **`site2`** peers. This can take some time to complete, if you only see one or two indexers, refresh your screen.

Task 8: Restore the peer processes for `site1`.

22. In the **IDX-Cluster** ssh session, start the Splunk instances for **`idx1`** and **`idx2`**.

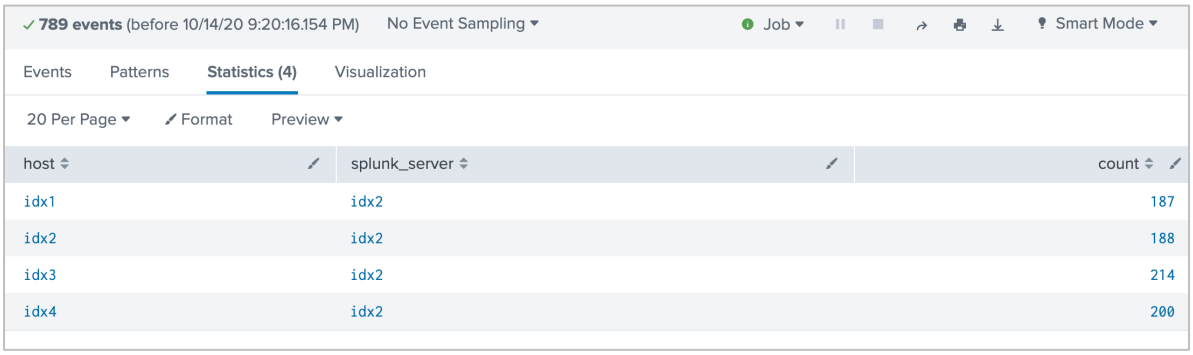
```
[os_user@ip-10-0-x-1 ~]$
~/idx1/bin/splunk start
~/idx2/bin/splunk start

exit
```

23. Check the cluster status and verify that all peers are up.

© 2021 Splunk Inc. All rights reserved.

24. Re-run the exact search again and verify the result:

```
index=_internal host=idx* {copied value of custom_range}
| stats count by host splunk_server | sort host
```



NOTE: The event count for each host should be unchanged; however, the data providers are again from **`site1`** peers (they may or may not be the same original providers).

Module 4 Lab Exercise – Configure and Monitor a Cluster Environment

Description

In this exercise, you will complete your indexer clustering environment by deploying an add-on to the indexing layer and enabling the Monitoring Console (MC) in distributed mode. MC will be enabled on the **`dserver`** instance.

The **bcg_web_idx** add-on is one of the three **Buttercup Games** app packages used throughout this course. They have been placed in the **`/opt/apps/LSD_apps`** directory on your **Misc-Server**.

	Search Head	Indexer	Forwarder
`bcg_web_idx` (indexer add-on)		x	
`bcg_web` (app)	x		
`bcg_web_TA` (add-on)			x

Steps

Task 1: Stage the `bcg_web_idx` app and deploy to the indexer cluster.

1. Confirm the apps exist in the **`/opt/apps/LSD_apps`** directory.

2. Copy the **`bcg_web_idx`** app from the **`/opt/apps/LSD_apps`** directory to the **`master-apps`** directory of the manager node.

NOTE: In a production environment, you will probably use the **`scp`** command.

```
os_user@ip-10-0-x-3 ~]$
ls /opt/apps/LSD_apps
bcg_web  bcg_web_idx  bcg_web_TA

cp -r /opt/apps/LSD_apps/bcg_web_idx ~/cmanager/etc/master-apps
```

3. Run the **`find`** command to check which custom indexes the app uses.

   Examine the content of its **`indexes.conf`** file.

```
cd ~/cmanager/etc/master-apps/bcg_web_idx
find . -name "indexes.conf"
./default/indexes.conf

cat ./default/indexes.conf
[web]
coldPath = $SPLUNK_DB/web/colddb
homePath = $SPLUNK_DB/web/db
maxTotalDataSizeMB = 50000
thawedPath = $SPLUNK_DB/web/thaweddb
```

© 2021 Splunk Inc. All rights reserved.

4. Configure the **web** index to automatically replicate across the peer nodes.

   Create an **indexes.conf** file in the app's **local** directory with the **repFactor** attribute set to **auto**.

```
vi ~/cmanager/etc/master-apps/bcg_web_idx/local/indexes.conf
[web]
repFactor=auto
```

5. From the manager node, run **splunk validate cluster-bundle --check-restart** to check for any errors and determine whether it requires a rolling-restart.

   To check the status, run **splunk show cluster-bundle-status**

```
os_user@ip-10-0-x-3 ~]$
~/cmanager/bin/splunk validate cluster-bundle --check-restart
Validating new bundle and checking if its application results in a restart.
Please run 'splunk show cluster-bundle-status' to check the status of the
bundle validation.

Created new bundle with checksum=019BB4FA06321CFDE415316AC448A222

~/cmanager/bin/splunk show cluster-bundle-status
...
idx1   CD3C8A8A-AFAA-45FF-B179-2ECF8B0F68F2      site1
       active_bundle=73A84BA3D1639A72B84A659060742671
       latest_bundle=73A84BA3D1639A72B84A659060742671
       last_validated_bundle=019BB4FA06321CFDE415316AC448A222
       last_bundle_validation_status=success
       last_bundle_checked_for_restart=019BB4FA06321CFDE415316AC448A222
       last_check_restart_result=restart not required
       restart_required_apply_bundle=0
       status=Up
...
```

6. If no errors are reported, deploy the bundle to the peer nodes: **splunk apply cluster-bundle**

```
~/cmanager/bin/splunk apply cluster-bundle
 Warning: Under some circumstances, this command will initiate a rolling
restart of all peers. This depends on the contents of the configuration
bundle. For details, refer to the documentation. Do you wish to continue?
[y/n]: y
Created new bundle with checksum=019BB4FA06321CFDE415316AC448A222
...
```

© 2021 Splunk Inc. All rights reserved.

7. Check the status of the applied bundle.

NOTE: Run **`splunk show cluster-bundle-status`** until all peer nodes report a status of **`Up`** and the *active_bundle* checksum value matches the checksum value from Step 5.

```
~/cmanager/bin/splunk show cluster-bundle-status
master cluster_status=None
       active_bundle
            checksum=019BB4FA06321CFDE415316AC448A222
            timestamp=1538090505 (in localtime=Thu Sep 27 23:21:45 2020)
       latest_bundle
            checksum=019BB4FA06321CFDE415316AC448A222
            timestamp=1538090505 (in localtime=Thu Sep 27 23:21:45 2020)
       last_validated_bundle
            checksum=019BB4FA06321CFDE415316AC448A222
            last_validation_succeeded=1
            timestamp=1538090505 (in localtime=Thu Sep 27 23:21:45 2020)
       last_check_restart_bundle
            last_check_restart_result=restart not required
            checksum=019BB4FA06321CFDE415316AC448A222
            timestamp=1538090321 (in localtime=Thu Sep 27 23:18:41 2020)

idx4   16F7CABD-FB80-4801-A3FB-DC07585D90DA      site2
       active_bundle=019BB4FA06321CFDE415316AC448A222
       latest_bundle=019BB4FA06321CFDE415316AC448A222
       last_validated_bundle=019BB4FA06321CFDE415316AC448A222
       last_bundle_validation_status=success
       restart_required_apply_bundle=0
       status=Up
...
```

Task 2: Disable indexing on the manager node and the monitoring console instance (`dserver`).

8. To forward all index data from **`cmanager`** and **`dserver`**, create an **`outputs.conf`** for **`cmanager`**.

```
os_user@ip-10-0-x-3 ~]$
vi ~/cmanager/etc/system/local/outputs.conf
[indexAndForward]
index = false

[tcpout]
defaultGroup = default-autolb-group
forwardedindex.filter.disable = true
indexAndForward = false

[tcpout:default-autolb-group]
server=10.0.x.1:9197,10.0.x.1:9297,10.0.x.1:9397,10.0.x.1:9497
```

9. Copy **`outputs.conf`** to **`dserver`**.

```
cp ~/cmanager/etc/system/local/outputs.conf ~/dserver/etc/system/local/outputs.conf
```

NOTE: In a production environment, you will probably use the **scp** command.

You will defer the steps to disable the indexing on search heads until you have search head clustering configured in Module 7.

10. Restart only **cmanager** for now (you will restart **dserver** in the next task.)

```
~/cmanager/bin/splunk restart
```

Task 3: Enable the Monitoring Console to run in distributed mode on dserver.

11. To conveniently group and identify the indexer cluster nodes, add a label for the cluster.

```
~/cmanager/bin/splunk edit cluster-config -cluster_label idxc-<user>
```

12. To enable the distributed search capabilities from the Monitoring Console, configure the master node as a search peer of **dserver** and configure the **dserver** instance as a cluster search head with its search affinity disabled.

```
os_user@ip-10-0-x-3 ~]$
~/dserver/bin/splunk add search-server 10.0.x.3:8089 -remoteUsername admin
-remotePassword <pw>

~/dserver/bin/splunk edit cluster-config -mode searchhead -master_uri
https://10.0.x.3:8089 -site site0 -secret idxcluster
```

13. Restart the instance.

```
~/dserver/bin/splunk restart
```

14. Log into **https://{Public_DNS}/dserver**.

15. Navigate to **Settings** > **Monitoring Console**.

16. Click **Settings** > **General Setup** on the Monitoring Console menu bar.

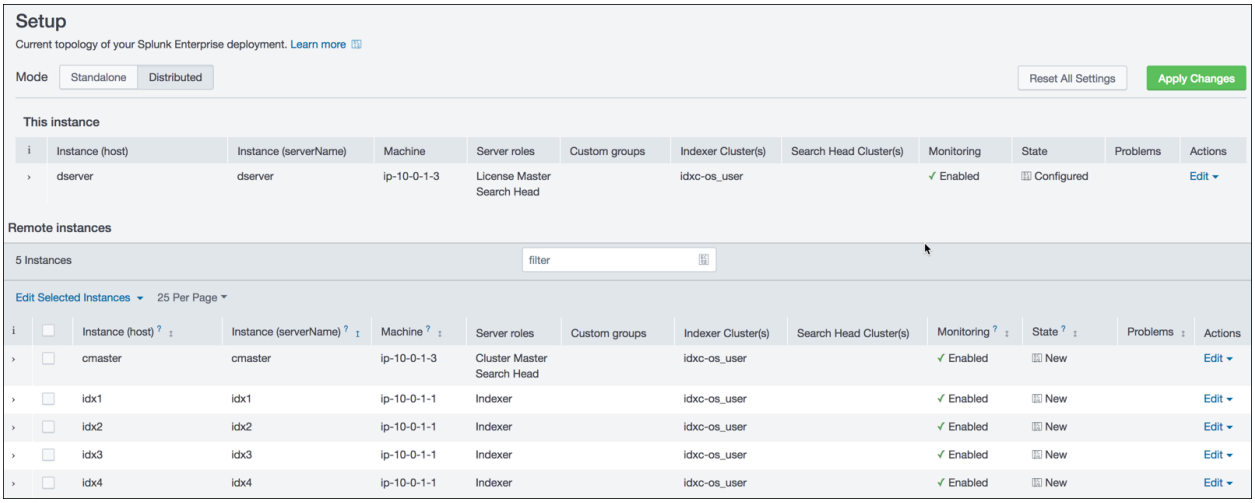
17. Select **Distributed** under the **Mode** option to run MC in distributed mode.

18. Click **Continue** and you should get a list of remote instances.

19. Examine the auto-selected **Server roles** of each instance and adjust the auto-identified roles as needed.

    a. Click **Edit** > **Edit Server Roles** on **dserver**.

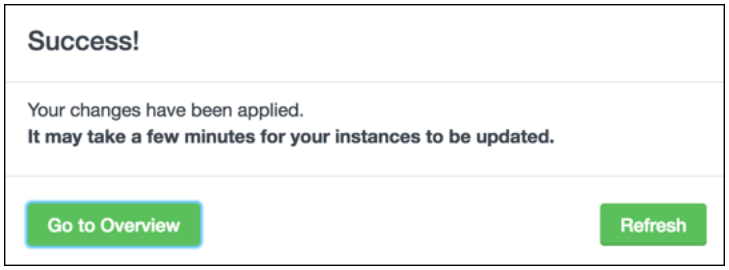
b. Select only the **License Master** and **Search Head** roles.
c. Verify `cmanager` has only **Cluster Master** and **Search Head**.
d. Verify that `idx1 – idx4` has only **Indexer**.



20. Click **Apply Changes** when you are ready to save the setup.

    If you get an informational message about sharing roles, ignore and click **Save**.

21. Continue on to the next step only when you get the **Success!** dialog box.



    If you encounter an error or no such prompt, repeat steps starting on step 14 one more time.

22. Click **Go to Overview**.

    The overview page should display the status of 4 indexers, 2 search heads, 1 cluster manager, and 1 license manager.

Check Your Work

Task 4: Monitor the indexer clustering service activities from Monitoring Console.

23. From the **Overview** page, click on the number (1) next to the **Cluster Master** panel header.

    The **Instance** page opens with the manager node information. You can further drill down to the resource usage page by selecting an option under **Views**.

24. To view the clustering service activities, click **Indexing** > **Indexer Clustering** > **Indexer Clustering: Service Activity**.

    For a healthy indexer cluster, the Warning and Error Patterns panel should be empty.

25. To confirm that the `web` index from `bcg_web_idx` app has been deployed, go to **Indexing** > **Indexes and Volumes** > **Indexes and Volumes: Deployment**.

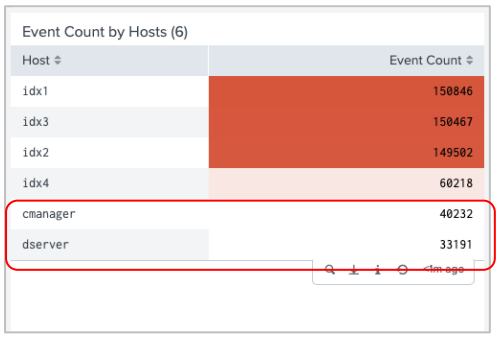
26. From the **Indexes** panel, click `web` to drill down to the **Index Detail: Deployment** dashboard.

    While there are no events in the `web` index yet, it should be present on all 4 indexers (see the **Index Structure Overview** and the **Instances** panels).

27. Confirm that forwarding from `cmanager` and `dserver` to the indexing layer is working:

    a. Change the selected **Index** in the **Index Detail: Deployment** dashboard from `web` to `_internal`.

    b. Scroll down to the **Event Count by Hosts** panel.



    c. You can also verify the forwarding by searching from `sh1`:

    ```
    index=_internal sourcetype=splunkd tcpoutputproc host=* | stats count by host
    ```

    Both `cmanager` and `dserver` should be listed. If not, verify the **outputs.conf** settings from Step 8.

Configuration Troubleshooting Suggestion

1. Compare the output of `.conf` files with `lab_conf_outputs.txt`.

Module 5 Lab Exercise – Configure a Forwarder

Description

In the lab environment, your **dserver** instance (**10.0.x.3:8189**) serves multiple server roles. Configure **dserver** to function as your deployment server.

When you launched your peer nodes in previous lab exercises, you enabled them to receive data from forwarders. In this lab exercise, you will configure a forwarder to use the indexer discovery option using the deployment server and install the forwarder add-on portion of the three-part **Buttercup Games** app.

	Search Head	Indexer	Forwarder
bcg_web_idx (indexer add-on)		x	
bcg_web (app)	x		
bcg_web_TA (add-on)			x

Steps

Task 1: On the manager node, enable the indexer discovery option with forwarder site failover.

1. Open the **Misc-Server** session and configure the manager to enable indexer discovery.
   - To enable, add the **indexer_discovery** stanza in **server.conf**
   - Set **pass4SymmKey = idxforwarders**

```
os_user@ip-10-0-x-3 ~]$
vi ~/cmanager/etc/system/local/server.conf
...
[indexer_discovery]
pass4SymmKey = idxforwarders
```

2. Configure forwarder site failover from **site1** to **site2**.

```
os_user@ip-10-0-x-3 ~]$
~/cmanager/bin/splunk edit cluster-config -forwarder_site_failover site1:site2
The cluster-config property has been edited.
You need to restart the Splunk Server (splunkd) for your changes to take
effect.

~/cmanager/bin/splunk restart
```

Task 2: Configure the deployment server.

3. Copy the forwarder configuration app **uf_base** from **/opt/apps** to the staging directory in **dserver**.

```
[os_user@ip-10-0-x-3 ~]$
cp -r /opt/apps/uf_base ~/dserver/etc/deployment-apps
```

© 2021 Splunk Inc. All rights reserved.

4. Edit the empty **~/dserver/etc/deployment-apps/uf_base/local/outputs.conf** file to enable indexer discovery, indexer acknowledgment, and volume-based forwarding.
   - The **master_uri** is the address to your cluster manager node
   - Set the **autoLBVolume** size to 256KB (**262144**)

```
os_user@ip-10-0-x-3 ~]$
vi ~/dserver/etc/deployment-apps/uf_base/local/outputs.conf
[tcpout]
defaultGroup = default-autolb-group

[tcpout:default-autolb-group]
indexerDiscovery = idxc1
useACK = true
autoLBVolume = 262144

[indexer_discovery:idxc1]
master_uri = https://10.0.x.3:8089
pass4SymmKey = idxforwarders
```

5. Create the **~/dserver/etc/deployment-apps/uf_base/local/props.conf** file to enable the forwarder event breaking option for all single-line events.

```
vi ~/dserver/etc/deployment-apps/uf_base/local/props.conf
[default]
EVENT_BREAKER_ENABLE = true
```

6. Create the **~/dserver/etc/deployment-apps/uf_base/local/server.conf** file to configure the forwarder to send data to all peers in **site1**.

```
vi ~/dserver/etc/deployment-apps/uf_base/local/server.conf
[general]
site = site1
```

7. Copy the **bcg_web_TA** app from the **/opt/apps/LSD_apps** directory to the deployment server (**dserver**).

```
cp -r /opt/apps/LSD_apps/bcg_web_TA ~/dserver/etc/deployment-apps
```

NOTE: In a production environment, you will probably use the **scp** command.

8. Configure the **eng-uf** server class so it deploys to both the **uf_base** and **bcg_web_TA** add-ons to forwarders. Create the server class in **~/dserver/etc/system/local**.

```
vi ~/dserver/etc/system/local/serverclass.conf
[serverClass:eng_uf]
whitelist.0 = 10.0.x.3

[serverClass:eng_uf:app:uf_base]
restartSplunkWeb = 0
restartSplunkd = 1
stateOnClient = enabled

[serverClass:eng_uf:app:bcg_web_TA]
restartSplunkWeb = 0
restartSplunkd = 1
stateOnClient = enabled
```

9. Reload the server class.

```
~/dserver/bin/splunk reload deploy-server
Reloading serverclass(es).
```

Task 3: Enable the deployment client setting on the forwarder.

10. Start the forwarder with the **auto-ports** option and configure it as a deployment client.

    In your lab environment, the forwarder instance is installed in the **fwdr** directory of **Misc-server**.

```
[os_user@ip-10-0-x-3 ~]$
~/fwdr/bin/splunk start --accept-license --answer-yes --auto-ports --no-prompt

~/fwdr/bin/splunk set deploy-poll 10.0.x.3:8189
Splunk username: admin
Password:
Configuration updated.
```

Task 4: Update the instance server role in Monitoring Console.

11. Log into **https://{Public_DNS}/dserver**.
12. Navigate to **Settings** > **Monitoring Console**.
13. Click **Settings** > **General Setup** on the Monitoring Console menu bar.

14. -Examine the auto-selected **Server roles** for `dserver` and adjust the auto-identified roles as needed.
    a. Click **Edit** > **Edit Server Roles** on `dserver`.
    b. Add **Deployment Server** to the selected server roles.
    c. Click **Save** > **Done**.
    d. Click **Apply Changes** > **Save**.

15. When you get the **Success!** dialog box, click **Go to Overview**.

    The overview page should now include 1 deployment server. Refresh the page until the client count in the deployment server panel changes to `1`.

Check Your Work

Task 5: Verify the forwarder app deployment.

16. Log into `https://{Public_DNS}/sh1`.

17. Search the last 15 minutes of internal logs to confirm the indexer discovery activities on the forwarder:

```
index=_internal host=uforwarder TcpOutputProc | timechart values(event_message)
as messages | search messages=*
```

NOTE:	Be patient. Take a quick break. It will take about 3 - 5 minutes to complete the deployment. The stats should list only the peers from `site1`.

```
index=_internal host=uforwarder TcpOutputProc | stats count by splunk_server
```

18. Search the last 15 minutes of internal metrics to verify input data is flowing to the `web` index:

```
index=_internal sourcetype=splunkd component=Metrics series=web | timechart
span=1m max(kb) by series
```

19. To confirm, search the `web` index:

    `index=web` (Last 24 hours)

After you confirm that data is flowing to the `web` index, test the forwarder site failover.

© 2021 Splunk Inc. All rights reserved.

Task 6: Test the forwarder site failover scenario.

20. In the **IDX-Cluster** session, identify the parent **`splunkd`** process of **`idx1`** and stop the process.

```
[os_user@ip-10-0-x-3 ~]$
ssh 10.0.x.1

[os_user@ip-10-0-x-1 ~]$
ps -ef | grep "splunkd -p 8189"
 jane 25586     1 0 ?  00:12:47 splunkd -p 8189 restart
 jane 25587 25586 0 ?  00:00:15 [splunkd pid=25586] splunkd -p 8189 restart
[process-runner]

kill 25586
```

21. Identify the parent **`splunkd`** process of **`idx2`** and stop the process.

```
ps -ef | grep "splunkd -p 8289"
 jane 25875     1 0 ?  00:22:36 splunkd -p 8289 restart
 jane 25876 25875 0 ?  00:00:15 [splunkd pid=25875] splunkd -p 8289 restart
[process-runner]

kill 25875
```

22. Go to the Monitoring Console **Overview** page on **`https://{Public_DNS}/dserver`**.

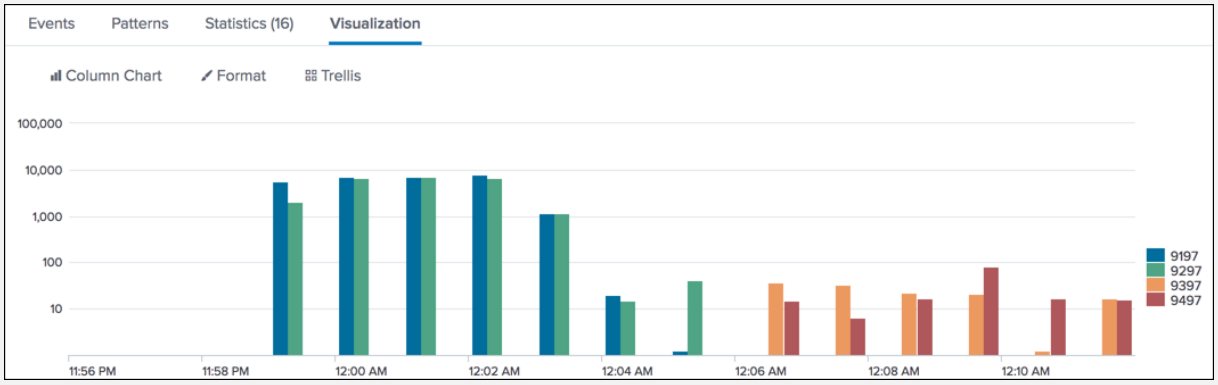
NOTE: The indexers panel should report that two instances are not reachable.

23. Search the last 15 minutes of internal metrics to verify that the forwarder failover is working:

```
index=_internal host=uf* component=Metrics group=tcpout_connections | timechart
span=1m sum(kb) by destPort
```

NOTE: The time chart should display the destination switchover. The failover can take up to 5 minutes. You may need to adjust the search window and **`timechart span`**.

For the visualization, change the **Scale** option in **Format** to **Log** on the **Y-Axis** tab.



24. To restore the service, start Splunk for **`idx1`** and **`idx2`** in the **IDX-Cluster** ssh session.

```
[os_user@ip-10-0-x-1 ~]$
~/idx1/bin/splunk start
~/idx2/bin/splunk start

exit

[os_user@ip-10-0-x-3 bin]$
```

25. Confirm that the warning message is cleared from the Monitoring Console page.

Configuration Troubleshooting Suggestions

If your configuration is not producing the expected results, troubleshoot by isolating the issue.

1. Verify that the forwarder has downloaded the **uf_base** app and restarted.

```
[os_user@ip-10-0-x-3 ~]$
ls ~/fwdr/etc/apps
bcg_web_TA introspection_generator_addon  learned  search  splunk_httpinput
SplunkUniversalForwarder  uf_base
```

2. On the **cmanager**, search the internal index to check if the forwarder has contacted the manager node.

   index="_internal" component=CMIndexerDiscovery host=cmanager

   You should have an event indicating a new forwarder has contacted the cluster manager:
   CMIndexerDiscovery - Registering new forwarder <some_GUID> (total: 1). Heartbeat assigned for next check: 30 seconds

   If you see such an event, then go to Troubleshooting Step 3.

   If you get no result, then stop and check the **master_uri** value specified in **~/dserver/etc/deployment-apps/uf_base/local/outputs.conf**.

3. Check the forwarder **splunkd.log** for any failed heartbeat messages or any other clues:

   tail -100 ~/fwdr/var/log/splunk/splunkd.log

   tail ~/fwdr/var/log/splunk/splunkd.log | egrep 'HttpPubSubConnection'

   tail -f ~/fwdr/var/log/splunk/splunkd.log | egrep 'TcpOutProc'

   A *failed heartbeat* message indicates an issue with the value of **pass4SymmKey**. Make sure the same value is used for the cluster manager and the forwarder. If you are not sure:

   a. Set it again in clear-text on both **cmanager** (Configuration Step 1) and **fwdr** (Configuration Step 4)
   b. Restart **cmanager** and then the **fwdr**.

   A *failed to extract FwdTarget* message indicates misconfigured listening ports on the indexer peers.

4. Compare the output of **.conf** files with **lab_conf_outputs.txt**.

© 2021 Splunk Inc. All rights reserved.

Module 6 Lab Exercise – Enable Search Head Cluster

Description

Currently, you have two dedicated search heads – one for each site -- independently searching the indexer cluster sites.

In this lab exercise, you will add the existing **`site2`** search head to a new 3-member search head cluster. Keep the **`site1`** search head as stand-alone for comparison. To complete the search head cluster, you will launch two more search head instances and associate them with **`site2`**. You will also integrate the new search head cluster to your existing indexer cluster.

After the search head cluster is up, you will run a quick test to validate its functionality.

Steps

Task 1: Add two more search heads to `site2`.

NOTE: Each member in a search head cluster in a production environment must run on a dedicated host. However, to simulate a working search head cluster in this lab environment, a single host is configured to run multiple Splunk instances.

To accommodate this simulation, each instance has been carefully assigned unique port numbers. Reference this port matrix to configure each search head instance

Server Name	Splunkd-port	Web-port	Replication-port
sh1	8189	8100	9100
sh2	8289	8200	9200
sh3	8389	8300	9300
sh4	8489	8400	9400

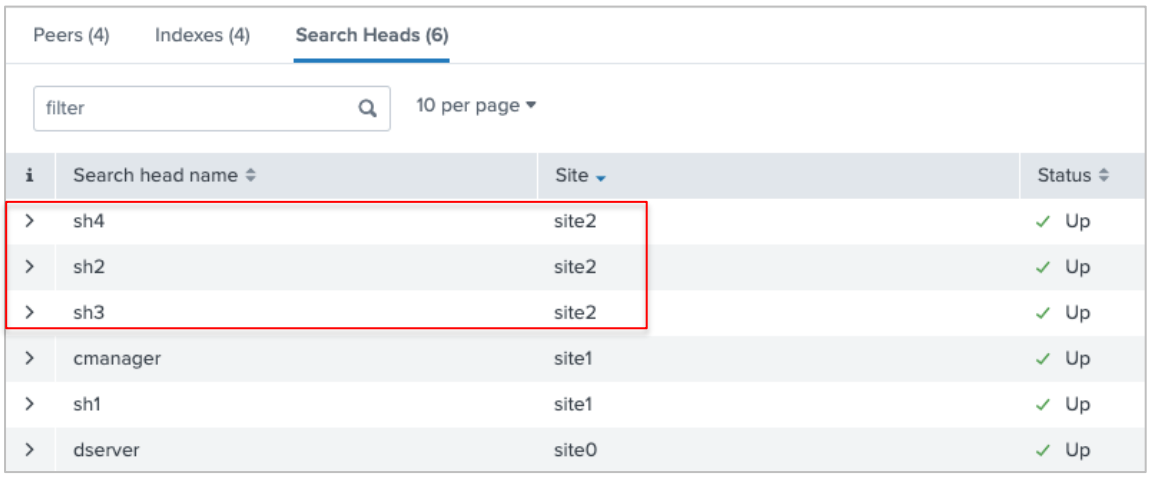
1. In the **SH-Cluster** session, bring up **`sh3`** and **`sh4`** as search heads for **`site2`**.
   a. Configure search heads as license peers to **`dserver`**.
   b. Integrate search heads with the **`site2`** indexer cluster.
   Hint: Don't forget to use the same secret that the manager node has used.

```
[os_user@ip-10-0-x-3 ~]$
ssh 10.0.x.2

[os_user@ip-10-0-x-2 ~]$
~/sh3/bin/splunk start --accept-license

~/sh3/bin/splunk edit licenser-localslave -master_uri https://10.0.x.3:8189

~/sh3/bin/splunk edit cluster-config -mode searchhead -master_uri
https://10.0.x.3:8089 -site site2 -secret idxcluster

~/sh3/bin/splunk restart
# Repeat to configure sh4
```

© 2021 Splunk Inc. All rights reserved.

```
~/sh4/bin/splunk start --accept-license

~/sh4/bin/splunk edit licenser-localslave -master_uri https://10.0.x.3:8189

~/sh4/bin/splunk edit cluster-config -mode searchhead -master_uri
https://10.0.x.3:8089 -site site2 -secret idxcluster

~/sh4/bin/splunk restart
```

2. On **cmanager**, check the Cluster Manager Node status page and confirm that **`site2`** now has three search heads.



Task 2: Enable a search head cluster with `site2` search heads.

3. In the **SH-Cluster** session, initialize **`sh2`** to be a member of the search head cluster.

   Use **`splunk init shcluster-config`** with:

   - **`mgmt_uri`** **`https://10.0.x.2:8289`**
   - **`replication_port`** **`9200`**
   - **`secret`** **`shcluster`**

   Repeat the process to initialize **`sh3`** and **`sh4`** to be members of the same search head cluster.

NOTE: To get more CLI help, run **`splunk help init shcluster-config`**

You are allowed to initialize **`sh2`** without re-installing Splunk because no knowledge objects and artifacts were created on this instance.

Reference the port matrix to configure each search head instance. If you get the "*This command needs splunkd to be up, and splunkd is down*" message, it probably means there is a typo or syntax error in the command.

© 2021 Splunk Inc. All rights reserved.

```
[os_user@ip-10-0-x-2 ~]$
~/sh2/bin/splunk init shcluster-config -mgmt_uri https://10.0.x.2:8289
-replication_port 9200 -secret shcluster

~/sh3/bin/splunk init shcluster-config -mgmt_uri https://10.0.x.2:8389
-replication_port 9300 -secret shcluster

~/sh4/bin/splunk init shcluster-config -mgmt_uri https://10.0.x.2:8489
-replication_port 9400 -secret shcluster

~/sh2/bin/splunk restart

~/sh3/bin/splunk restart

~/sh4/bin/splunk restart
```

4. Bootstrap **sh2** to be the initial captain with the **splunk bootstrap shcluster-captain** command and add members one by one with the **splunk add shcluster-member** command.

```
~/sh2/bin/splunk bootstrap shcluster-captain -servers_list https://10.0.x.2:8289

~/sh2/bin/splunk add shcluster-member -new_member_uri https://10.0.x.2:8389

~/sh2/bin/splunk add shcluster-member -new_member_uri https://10.0.x.2:8489
```

NOTE: You will learn more about adding SHC members in Module 7.

5. Check the running state of the search head cluster from any member.

```
~/sh2/bin/splunk show shcluster-status
 Captain:
                    dynamic_captain : 1
                    elected_captain : Wed Oct 15 21:33:43 2019
                                 id : CDBA2E24-E286-43D5-AD7C-D67625D2DF32
                  initialized_flag : 1
                              label : sh2
                           mgmt_uri : https://10.0.x.2:8289
              min_peers_joined_flag : 1
               rolling_restart_flag : 0
                 service_ready_flag : 1
 Members:
      sh4
                               label : sh4
                last_conf_replication : Wed Oct 15 21:37:03 2019
                            mgmt_uri : https://10.0.x.2:8489
                       mgmt_uri_alias : https://10.0.x.2:8489
                               status : Up
...
```

6. To easily group and identify the search head cluster members, add the label for the cluster.

```
~/sh2/bin/splunk edit shcluster-config -shcluster_label shc-<user>
```

When the search head cluster is up and running for the first time, it automatically disables Monitoring Console from all members. For this to fully take effect, you need to restart all members.

7. To restart the search head cluster, execute the **`rolling-restart`** from the captain's command line.

```
[os_user@ip-10-0-x-2 ~]$
~/sh2/bin/splunk rolling-restart shcluster-members

~/sh2/bin/splunk rolling-restart shcluster-members -status 1
Rolling restart Success : 1
  Message : Rolling Restart of all the search head cluster members has been
kicked off. It might take some time for completion. After restart the
information will be logged at audit log, Meanwhile you can check the
progress of this transaction using
       "splunk  rolling-restart shcluster-members -status 1"
exit
```

Check Your Work

Task 3: Verify that your search head cluster is functioning properly.

8. Log into your **sh3**: **https://{Public_DNS}/sh3**.
9. Navigate to **Settings** > **Search head clustering**.

   Identify the current captain and wait until all members are **Up**.

10. In the Search app, search over the **Last 15 minutes**:

    ```
    index=_internal sourcetype=splunkd error
    ```

11. Save the search as a scheduled report.
    - Title: **My Cluster Errors - last 15 minutes**
    - Time Range Picker: No
12. Click **Schedule** and set the following:
    - Schedule Report: ☑
    - Schedule: **Run on Cron Schedule**
    - Cron Expression: ***/10 * * * ***
    - Time Range: **Last 15 minutes**
    - Schedule Priority: **Default**
    - Schedule Window: **No window**

    Do not enable any actions and **Save**.

13. Open another browser tab and go to **https://{Public_DNS}/sh4**.
    a. Go to the **Search & Reporting** app and click **Reports**.
    b. Confirm that the **My Cluster Errors - last 15 minutes** scheduled report from **sh3** has been replicated on this member.
    c. Run the **My Cluster Errors - last 15 minutes** report.

NOTE:	Ignore the message, "*There are no results because the first scheduled run of the report has not completed*" for now.

    d. Navigate to **Settings > Users**.
    e. Click **New** and create an account for **emaxwell** with the following field values:
       - Username: **emaxwell**
       - Assign to roles: **power** and **user**
       - Password: (for simplicity, use the same password as **admin**)
       - Confirm password: (for simplicity, use the same password as **admin**)
       - Require password change on first login: Uncheck

14. Open another browser tab and log into **https://{Public_DNS}/sh2** as **emaxwell**.

    If you are able to log in as **emaxwell**, log out.

15. Log back into **https://{Public_DNS}/sh2** as **admin**.
16. Confirm that the **My Cluster Errors - last 15 minutes** scheduled report from **sh3** has been replicated on this member.

17. Click **Activity** > **Jobs** and confirm that the search job from the scheduled report is shown.
18. Click the **Job** > **Inspect Job** link associated with the matching search.

    The SID includes the GUID of the member who ran the search. Click **Search job properties** to expand the properties. Scroll to the bottom of the page to find the **searchProviders** info. From the **searchProviders** value, you can also deduce which member ran the job.

19. Go to **`https://{Public_DNS}/sh1`**
20. Verify that there is no such report, no job artifacts, nor the user **`emaxwell`** on this search head.

Troubleshooting Suggestions

If your configuration is not returning the expected results, troubleshoot by isolating the issue.

1. Verify the command syntax and spelling on each instance with: **`splunk btool check --debug`**
2. From the would-be captain, search (Last 60 minutes) for the captain election messages:

   ```
   index=_internal sourcetype=splunkd component= SHCRaftConsensus | reverse
   ```

3. Check **`splunkd.log`** of each instance for any errors:

   ```
   tail -40 ~/<instance_name>/var/log/splunk/splunkd.log
   ```

4. Compare the output of **`.conf`** files with **`lab_conf_outputs.txt`**.

© 2021 Splunk Inc. All rights reserved.

Module 7 Lab Exercise – Deploy an App to SHC

Description

In this exercise, you will complete your Splunk clustering environment and perform basic administration tasks such as deploying an app and monitoring the search head cluster activities.

You will deploy the final part of the three-part **Buttercup Games Web app** into the SHC with deployer. In this lab environment, you will configure **`dserver`** (**`10.0.x.3:8189`**) to function as your deployer instance.

	Search Head	Indexer	Forwarder
`bcg_web_idx` (indexer add-on)		x	
`bcg_web` (app)	x		
`bcg_web_TA` (add-on)			x

Steps

Task 1: Configure the SHC members and the deployer.

1. To enable the deployer component on **`dserver`**, add the search head cluster's **`pass4SymmKey`** in **`server.conf`**.

```
[os_user@ip-10-0-x-3 ~]$
vi ~/dserver/etc/system/local/server.conf
...
[shclustering]
pass4SymmKey = shcluster

~/dserver/bin/splunk restart
ssh 10.0.x.2
```

2. Add the deployer's address to the existing search head cluster members.

   Defer **`splunk restart`** for now. You will use the web UI after all members are edited.

```
[os_user@ip-10-0-x-2 ~]$
~/sh2/bin/splunk edit shcluster-config -conf_deploy_fetch_url https://10.0.x.3:8189

~/sh3/bin/splunk edit shcluster-config -conf_deploy_fetch_url https://10.0.x.3:8189

~/sh4/bin/splunk edit shcluster-config -conf_deploy_fetch_url https://10.0.x.3:8189

exit
```

© 2021 Splunk Inc. All rights reserved.

3. Log into any SHC member and navigate to **Settings** > **Search head clustering**.
4. Click **Begin Rolling Restart** > **Restart**.

   You may continue on to Task 2 while SHC processes the rolling-restart.

Task 2: Stage and distribute apps to search head cluster members.

5. Copy an existing app **`bcg_web`** in the **`/opt/apps/LSD_apps`** directory to the deployer's **`shcluster`** directory.

NOTE: In a production environment, you will probably use the **`scp`** command.

```
[os_user@ip-10-0-x-3 ~]$
cp -r /opt/apps/LSD_apps/bcg_web ~/dserver/etc/shcluster/apps
```

6. Create a new app called **`shc_base`** in the deployer's **`shcluster`** directory.

NOTE: This app disables indexing on the search head members. In **`outputs.conf`**, be sure to use the correct IP addresses and ports for your indexer peer nodes.

```
mkdir -p ~/dserver/etc/shcluster/apps/shc_base/{default,metadata}

touch ~/dserver/etc/shcluster/apps/shc_base/metadata/local.meta

vi ~/dserver/etc/shcluster/apps/shc_base/default/app.conf
[ui]
is_visible = 0

[package]
id = shc_base
check_for_updates = 0

vi ~/dserver/etc/shcluster/apps/shc_base/default/outputs.conf
[indexAndForward]
index = false

[tcpout]
defaultGroup = default-autolb-group
forwardedindex.filter.disable = true
indexAndForward = false

[tcpout:default-autolb-group]
server=10.0.x.1:9197,10.0.x.1:9297,10.0.x.1:9397,10.0.x.1:9497
```

© 2021 Splunk Inc. All rights reserved.

7. To manually control the deployment of your staged apps to the search head members, run **`splunk apply shcluster-bundle`** with the **`-action`** parameter from the deployer.

   You may use any member of the search head cluster as the target.

```
[os_user@ip-10-0-x-3 ~]$
~/dserver/bin/splunk apply shcluster-bundle -action stage --answer-yes
```

NOTE: Ignore the message *`Bundle has been pushed successfully to all the cluster members`* in this phase. No actual apps have been sent yet. This step only checks the validity of your app bundles.

```
[os_user@ip-10-0-x-3 ~]$
~/dserver/bin/splunk apply shcluster-bundle -action send -target
https://10.0.x.2:8289 --answer-yes
```

NOTE: The message *`Bundle has been pushed successfully to all the cluster members`* indicates a successful execution. To confirm, search the internal logs.

8. Log into **`https://{Public_DNS}/dserver`** and search the following for confirmations (last 15 minutes):

   `index=_internal confdeployment data.task=createDeployableApps | table host, data.task, log_level, data.source_area, data.staging_area` (This confirms the staging.)

   `index=_internal confdeployment data.task=sendDeployableApps | table host, data.target_label, data.target_uri, data.status` (This confirms the sending.)

Check Your Work

Task 3: Verify the app deployment.

9. Log into **`https://{Public_DNS}/sh2`**.
   The **BCG Web** app should appear in the list of apps.

10. To verify the successful app deployment from the deployer, repeat the steps on all remaining search head members.
    - **`https://{Public_DNS}/sh3`**
    - **`https://{Public_DNS}/sh4`**

11. On any search head member (e.g. **`sh4`**), test the following:
    a. Go to the **BCG Web** app.

b. Confirm the **Web Store Status** dashboard panels are populating with results.

Task 4: Complete the Monitoring Console setup on `dserver`.

12. Add **`sh2`**, **`sh3`**, and **`sh4`** as search peers of **`dserver`**.

```
[os_user@ip-10-0-x-3 ~]$
~/dserver/bin/splunk add search-server 10.0.x.2:8289 -remoteUsername admin
-remotePassword <pw>

~/dserver/bin/splunk add search-server 10.0.x.2:8389 -remoteUsername admin
-remotePassword <pw>

~/dserver/bin/splunk add search-server 10.0.x.2:8489 -remoteUsername admin
-remotePassword <pw>
```

13. Log into **`https://{Public_DNS}/dserver`**.

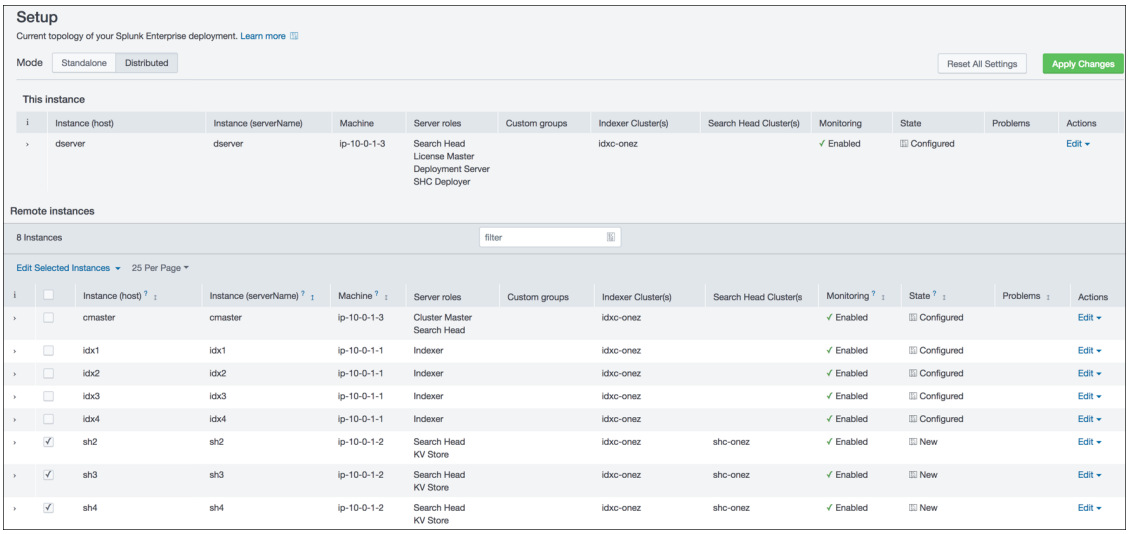
14. Navigate to **Settings** > **Monitoring Console**.

15. Click **Settings** > **General Setup** on the MC menu bar.

16. Examine the **Server roles** of each instance again and adjust the auto-identified roles if required.
    a. Click **Edit** > **Edit Server Roles** on **`dserver`**.
    b. Add the **SHC Deployer** role.
    c. Click **Save** > **Done**.

17. Examine the **Server roles** of each SHC member instance and adjust the auto-identified roles.
    a. Select the check box next to **`sh2`**, **`sh3`** and **`sh4`**.
    b. Click **Edit Selected Instances** > **Set Server Roles**.
    c. Select only the **Search Head** and **KV Store** roles.
    d. Click **Save** > **Done**.

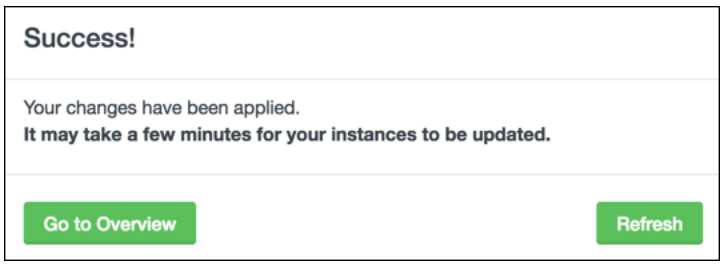
© 2021 Splunk Inc. All rights reserved.



18. Click **Apply Changes** when you are ready to save the setup.

    If you get an informational message about sharing roles, ignore and click **Save**.

19. Continue on to the next step only when you get the **Success!** dialog box.

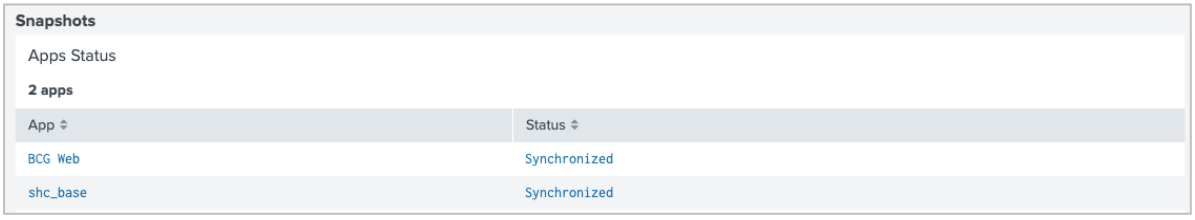


20. Click **Go to Overview**.

NOTE: The overview page should now display the status of 4 indexers, 5 search heads (because `sh1` is not configured), 1 cluster master, 1 license master, and 1 deployment server.

Task 5: Review the search head cluster dashboards.

21. Click **Search** > **Search Head Clustering** > **Search Head Clustering: Status and Configuration** on the MC menu bar.
    a. Confirm that the Health Check panel indicates no issue.
    b. Check the Status panel for the current captain and its term.
22. Navigate to the **Search Head Clustering: App Deployment** dashboard and confirm that the status of each app is *Synchronized*.



Snapshots

Apps Status

2 apps

App	Status
BCG Web	Synchronized
shc_base	Synchronized

Troubleshooting Suggestions

If your configuration is not returning the expected results, troubleshoot by isolating the issue.

1. Verify the command syntax and spelling on each instance with: **`splunk btool check --debug`**
2. To check the deployer deployment status, search the last 60-minutes of internal events in **`dserver`**.

   ```
   index=_internal component=ConfDeployment data.task=sendDeployableApps | table
   data.target_label data.status
   ```

3. Check **`splunkd.log`** of each instance for any errors:

   ```
   tail -40 ~/<instance_name>/var/log/splunk/splunkd.log
   ```

4. Compare the output of **`.conf`** files with **`lab_conf_outputs.txt`**.

© 2021 Splunk Inc. All rights reserved.

Module 8 Lab Exercise – Add a KV Store Collection

Description

The **bcg_web** app you have deployed to your search head cluster in the previous module is shipped with collection configurations and a CSV file you can use to populate the initial KV store collection. However, the collection will not work with the indexer cluster because the app is configured for a non-clustered deployment.

In this lab exercise, you will manage a KV store lookup to work within a clustered search head environment.

Steps

Task 1: Identify the current SHC captain and KV store captain.

1. In the **SH-Cluster** session, identify the current SHC captain.

```
[os_user@ip-10-0-x-3 ~]$
ssh 10.0.x.2

[os_user@ip-10-0-x-2 ~]$
~/sh2/bin/splunk show shcluster-status
Captain:
                    dynamic_captain : 1
                    elected_captain : Tue Oct 15 23:01:20 2020
                                 id : CDBA2E24-E286-43D5-AD7C-D67625D2DF32
                  initialized_flag : 1
                              label : sh2
                           mgmt_uri : https://10.0.x.2:8289
              min_peers_joined_flag : 1
               rolling_restart_flag : 0
                 service_ready_flag : 1
 ...
```

NOTE: In this example, **sh2** is identified as the SHC captain.

2. Run the **`show kvstore-status`** command to identify the current KV store captain (primary).

```
~/sh2/bin/splunk show kvstore-status
This member:
                         date : Tue Oct 15 23:39:19 2020
                      dateSec : 1555371559.164
                     disabled : 0
                         guid : 977E4E87-02A4-4AF0-BCBB-4BB088470525
            oplogEndTimestamp : Mon Oct 15 23:39:08 2020
          oplogEndTimestampSec : 1555371548
           oplogStartTimestamp : Tue Oct 15 21:34:18 2020
        oplogStartTimestampSec : 1555364058
                         port : 8291
                   replicaSet : splunkrs
            replicationStatus : Non-captain KV store member
                   standalone : 0
                       status : ready

Enabled KV store members:
...

KV store members:
10.0.x.2:8491
                 configVersion : 1
                  electionDate : Tue Oct 15 23:01:36 2020
              electionDateSec : 1555369296
                  hostAndPort : 10.0.x.2:8491
                   optimeDate : Tue Oct 15 23:39:08 2020
                optimeDateSec : 1555371548
            replicationStatus : KV store captain
                       uptime : 2285

exit
```

NOTE: In this example, **`sh4`** (**`10.0.x.2:8491`**) is identified as the KV store captain.

Task 2: Verify the state of the KV store service from Monitoring Console.

3. In **`dserver`**'s Monitoring Console, navigate to **Search** > **KV Store** > **KV Store: Deployment**.

4. In the **Warning and Error Patterns** panel, change the **Time Range** to **Last 15 minutes**.

NOTE: The panel should not have any reported issues.

5. Review the **KV Store Status** panel and identify the primary instance in the **Replication Role** column.

Snapshots

KV Store Status

Instance	Physical Memory Usage (MB)	Mapped Memory Usage (MB)	Page Faults per Operation	Total Queued	Active Connections	Lock (%)	Last Flush (ms)	Network Traffic (MB)	Uptime (hours)	Replication Role
sh2			0.00	0	17		5	10.34	0.64	Secondary
sh3			0.00	0	18		6	12.87	0.65	Secondary
sh4			0.00	0	22		2	14.10	0.64	Primary

Click instance name for more details. Total queued is operations (readers and writers) waiting for a read or write lock to be cleared.

6. Log into the primary instance's Splunk Web. (**sh4** in this example)
7. Navigate to **Settings** > **Lookups** > **Lookup definitions**.
8. Verify the **product_lookup** KV store is configured under the **bcg_web** app.

NOTE: The configured KV store supports the following lookup fields:

productId, product_name, categoryId, price, sale_price, Code

9. Navigate to the **BCG Web** app > **Search**, then run the following search:

```
| inputlookup product_lookup
```

NOTE: This search returns the *No results found* message because the KV store collection is currently empty.

10. Run a search to populate the KV store with the CSV file contents in the **lookups** directory:

```
| inputlookup products.csv | outputlookup product_lookup
```

NOTE: Note the processed count on the **Statistics** tab. It should be 15. Click the **Job** menu to display a message indicating that the records were written to **product_collection** and there were warnings.

11. In **dserver**'s Monitoring Console, navigate to **Search** > **KV Store** > **KV Store: Instance**.

NOTE: The number of objects in the **product_collection** shows **14**, regardless of which SHC member instance you select. Why?

12. To investigate, go to **Activity** > **Jobs** page on any SHC member (**sh4** for example) and select the **Inspect Job** option associated with the search from Step 10.

NOTE: You should have messages such as:

```
info : Results written to collection 'product_collection'.
info : Successfully read lookup file '.../bcg_web/lookups/products.csv'.
```

```
warn : There were warnings when executing this outputlookup. See
search.log for more information.
```

13. Click **search.log** and find a log entry **`WARN  KVStoreLookup`**.

NOTE: You should have messages such as:

```
WARN  KVStoreLookup - Skipping row due to bad value '{[0]='NA'}' for field
'price' (expected type: 1)
...
INFO  outputcsv - 14 events written to product_collection
```

These are the process artifacts from KV store data type enforcement.

14. Run the search from Step 10 again and confirm that you get **`14`** results.

15. Run a search to verify the lookup knowledge enhancement (All time):

```
index=web sourcetype=access_combined action=purchase | lookup product_lookup
productId | stats sum(price) by product_name
```

a. Open the job inspector and write down the search duration.

b. Scan the **Execution costs** histogram and record the values of the **`command.stats`** and **`dispatch.stream.remote`** components.

Example: *This search has completed and has returned 14 results by scanning 26,995 events in 0.782 seconds*

Duration	Component	Invocations	Input count	Output count
0.03	command.stats	28	26,996	14
0.57	dispatch.stream.remote	22	-	12,870,775

c. Expand the **Search job properties** and click the remote log from **`idx3`**.

d. Check if the remote search performed any lookups by searching for **`product_lookup`**.

NOTE: No **`product_lookup`** entry exists in any of the **`idx#.log`** files.

Task 3: Enable the KV store collection replication.

16. In the **Misc-Server** session, edit the **`bcg_web`** app's **`collections.conf`** file in the deployer's **`shcluster`** directory to enable the collection replication.

    Apply the bundle after the edit.

```
[os_user@ip-10-0-x-3 ~]$
vi ~/dserver/etc/shcluster/apps/bcg_web/default/collections.conf
[product_collection]
enforceTypes = true
field.price = number
field.sale_price = number
replicate = true

~/dserver/bin/splunk apply shcluster-bundle -target https://10.0.x.2:8289
```

NOTE: Updating **`collections.conf`** does not require a search head cluster rolling-restart.

17. Run the same search in **BCG Web** to verify the lookup replication (All time):

    `index=web sourcetype=access_combined action=purchase | lookup product_lookup productId | stats sum(price) by product_name`

    a. Open the job inspector and compare the execution costs to the values you recorded earlier.
    b. Scan the **Execution costs** histogram and compare the **`command.stats`** and **`dispatch.stream.remote`** values you recorded earlier.
    c. Expand the Search job properties and click the remote log from **`idx3`**.
    d. Check if the remote search performed any lookups by searching for **`product_lookup`**.

© 2021 Splunk Inc. All rights reserved.

Example: *This search has completed and has returned 14 results by scanning 27,012 events in 0.511 seconds*

```
Duration  Component        Invocations   Input count   Output count
0.01      command.stats             18            97             14
0.24      dispatch.stream.remote    11             -         97,155
```

NOTE: **product_lookup** log events can be found in all logs now.

SearchParser - PARSING: litsearch (action=purchase index=web sourcetype=access_combined) | lookup product_lookup productId | addinfo type=count label=prereport_events track_fieldmeta_events=true | fields keepcolorder=t "prestats_reserved_*" "price" "product_name" "psrsvd_*" | prestats sum(price) by product_name

The peer nodes were able to filter out the events based on the lookup and send much less data to the search head. With the larger dataset, you may see more search performance improvement.

Troubleshooting Suggestions

1. If you make a mistake and you want to clean the KV store collection and start over, you can run the **clean kvstore** command from the KV store captain's terminal.

   For example, if you have determined that the current KV store captain is on **sh4** by running the **show kvstore-status**, you can run the following:

```
[os_user@ip-10-0-x-3 ~]$
~/sh4/bin/splunk clean kvstore -app bcg_web
This action will permanently erase KVStore data.
Are you sure you want to continue [y/n]? y
```

2. To confirm the cleaning, search: **| inputlookup product_lookup**

NOTE: It should return *No results found.* And also, the number of objects in the **MC** > **Search** > **KV Store** > **KV Store: Instance** page should show **0** in all instances.

Module 9 Lab Exercise – Migrate the Indexer Cluster to use SmartStore

Description

In this scenario, your indexer cluster has been in production for some time and the storage requirement has outgrown the compute resources. Unfortunately, your environment doesn't have any room to scale out the peer nodes. However, you already have a S3-compliant storage with plenty of storage capacity.

In order to meet the growing storage requirements, you will migrate your multisite indexer cluster to utilize SmartStore to increase the storage capacity without adding more peer nodes. You will use the single volume storage option in this exercise.

Your instructor will provide the following remote storage service access information:

- `path = <s3_bucket_ns>`
- `remote.s3.access_key = <access_key>`
- `remote.s3.secret_key = <secret_key>`
- `remote.s3.endpoint = <aws_region_uri>`

NOTE:	Copy the above attributes to a text editor and replace/add your specific values.
WARNING:	Migrating indexes to SmartStore is a one-way option and cannot be reverted.

Steps

Task 1: Verify the SmartStore connectivity from a test instance.

1. In the **SH-Cluster** session, create a test file.

```
[os_user@ip-10-0-x-3 ~]$
ssh 10.0.x.2

[os_user@ip-10-0-x-2 ~]$
echo "Hello World" > test99.txt                Replace 99 with your student ID
```

NOTE:	To verify the S3 connectivity and SmartStore functionalities, you are going to start a standalone test instance.

2. Start a test instance **`sh5`**.

```
~/sh5/bin/splunk start --accept-license
```

3. From **`~/sh5/bin`**, verify the SmartStore connectivity using the provided remote storage credentials.

- To list the content of a s3 bucket, run:

  ```
  ./splunk cmd splunkd rfs -- --access-key <access_key> --secret-key
  <secret_key> --endpoint <aws_region_uri> ls --starts-with <s3_bucket_ns>
  ```

- To copy a file to a s3 bucket, run:

  ```
  ./splunk cmd splunkd rfs -- --access-key <access_key> --secret-key
  <secret_key> --endpoint <aws_region_uri> putF <local_file> <s3_directory>
  ```

NOTE: The **<s3_directory>** is defined with **<s3_bucket_ns>** and **<directory>**. To test the connectivity, this lab environment will share a test directory called **all**.

The **<s3_bucket_ns>** in the following example is **s3://lsd1234** and the resulting **<s3_directory>** is: **s3://lsd1234/all**.

```
./splunk cmd splunkd rfs -- --access-key ABCDEFGHIJKLMNOPQRST --secret-key
aBBc1dEEfGhi2jKK3LLmMM4no5pqRRs6TTuVwxYZ --endpoint https://s3.ap-northeast-
1.amazonaws.com ls --starts-with s3://lsd1234

./splunk cmd splunkd rfs -- --access-key ABCDEFGHIJKLMNOPQRST --secret-key
aBBc1dEEfGhi2jKK3LLmMM4no5pqRRs6TTuVwxYZ --endpoint https://s3.ap-northeast-
1.amazonaws.com putF ~/test99.txt s3://lsd1234/all/

./splunk cmd splunkd rfs -- --access-key ABCDEFGHIJKLMNOPQRST --secret-key
aBBc1dEEfGhi2jKK3LLmMM4no5pqRRs6TTuVwxYZ --endpoint https://s3.ap-northeast-
1.amazonaws.com ls --starts-with s3://lsd1234/all
size,name
14,all/test01.txt
12,all/test99.txt
```

NOTE: You will see files other students have uploaded in the **<s3_bucket_ns>/all** directory.

Look for your own test file. If you don't see your file, check **splunkd-utility.log** in **sh5/var/log/splunk** for any errors. Check your CLI attribute values carefully and try again.

Task 2: Test the SmartStore configuration on a test instance.

4. To specify the SmartStore settings, edit **~/sh5/etc/system/local/indexes.conf** and restart.

NOTE: In this lab environment, you will uniquely identify your remote storage namespace using your student ID. Re-compose your **path** by replacing **all** with **test-<x>** for the **indexes.conf** settings. For example: **path = s3://lsd1234/test-99**

```
vi ~/sh5/etc/system/local/indexes.conf
[default]
remotePath = volume:s3vol/$_index_name
maxGlobalDataSizeMB = 500000
maxDataSize = auto

[volume:s3vol]
storageType = remote
path = s3://lsd1234/test-99
remote.s3.access_key = ABCDEFGHIJKLMNOPQRST
remote.s3.secret_key = aBBc1dEEfGhi2jKK3LLmMM4no5pqRRs6TTuVwxYZ
remote.s3.endpoint = https://s3.ap-northeast-1.amazonaws.com

./splunk restart
```

NOTE: This configuration sets all indexes to use the SmartStore volume **s3vol.** It limits the size of each index shared across all peers to 500 GB (**maxGlobalDataSizeMB**). Upon restart, the cache manager uploads qualifying buckets and their metadata to the remote storage.

© 2021 Splunk Inc. All rights reserved.

5. To confirm SmartStore is working, run the **rfs** command to list the buckets from your test instance.

   Compare the GUID of the instance against the remote bucket namespace.

```
cat ~/sh5/etc/instance.cfg
[general]
guid = BF18D6D8-3193-49C9-82C0-E72B56A87548

./splunk cmd splunkd rfs -- ls --starts-with volume:s3vol

size,name
7,/_audit/db/5d/fc/0~BF18D6D8-3193-49C9-82C0-E72B56A87548/guidSplunk-
BF18D6D8-3193-49C9-82C0-E72B56A87548/.rawSize
538408,/_audit/db/5d/fc/0~BF18D6D8-3193-49C9-82C0-E72B56A87548/guidSplunk-
BF18D6D8-3193-49C9-82C0-E72B56A87548/1539203640-1539192620-
14335030555952092335.tsidx
98,/_audit/db/5d/fc/0~BF18D6D8-3193-49C9-82C0-E72B56A87548/guidSplunk-
BF18D6D8-3193-49C9-82C0-E72B56A87548/Hosts.data
111,/_audit/db/5d/fc/0~BF18D6D8-3193-49C9-82C0-E72B56A87548/guidSplunk-
BF18D6D8-3193-49C9-82C0-E72B56A87548/SourceTypes.data
107,/_audit/db/5d/fc/0~BF18D6D8-3193-49C9-82C0-E72B56A87548/guidSplunk-
BF18D6D8-3193-49C9-82C0-E72B56A87548/Sources.data
254,/_audit/db/5d/fc/0~BF18D6D8-3193-49C9-82C0-E72B56A87548/guidSplunk-
BF18D6D8-3193-49C9-82C0-E72B56A87548/Strings.data
...
```

NOTE: The **rfs** command is now using the values provided in the **indexes.conf** and recursively lists all Splunk buckets present in the remote storage. If you see the list of index files, your SmartStore configuration is set correctly.

6. Log into **https://{Public_DNS}/sh5**.

7. Run a search to confirm the **CacheManager** activities (Last 15 minutes):

   index=_internal sourcetype=splunkd component IN(CacheManager*) | table _time, action, status, cache_id

   To verify that the cache manager is functioning properly, select **Job** > **Inspect Job**.

NOTE: Scroll down the **Execution costs** list and locate:

command.search.index.bucketcache.error
command.search.index.bucketcache.hit
command.search.index.bucketcache.miss

The listing of these components and their invocation counts indicate functioning SmartStore configuration.

Task 3: Run the migration on the indexer cluster.

8. Access the Splunk Web interface for **cmanager**: **https://{Public_DNS}/cmanager**

   Confirm that the **Health Status** indicator is green.

9. To convert any preexisting, single-site buckets to follow the multisite replication and search policies, edit **~/cmanager/etc/system/local/server.conf**.

```
[os_user@ip-10-0-x-2 bin]$
exit

[os_user@ip-10-0-x-3 ~]$
vi ~/cmanager/etc/system/local/server.conf
...
[clustering]
mode = master
pass4SymmKey = $7$SbhODP0GfZ0wHE3/2KTBWNbiVQEACQ0hiVlPnBcP8F3TykW3d8znQ5pb
multisite = true
replication_factor = 1
search_factor = 1
available_sites = site1,site2
site_replication_factor = origin:1,total:2
site_search_factor = origin:1,total:2
maintenance_mode = false
cluster_label = idxc-os_user
forwarder_site_failover = site1:site2
constrain_singlesite_buckets = false
...

~/cmanager/bin/splunk restart
```

10. Log back into **cmanager** and verify again the status of the cluster.

    The manager node **Health Status** indicator may be in red initially. Be patient. Wait for it to turn green.

11. Access the **Indexer Clustering** page and confirm that the cluster is in the complete state.

12. Click the **Indexes** tab > the **Bucket Status** button, and confirm no buckets are stuck in fixup tasks.

NOTE: Buckets can be stuck in fixups for various reasons. If you have buckets that are stuck in this exercise, click the **Action** button and **delete** them.

13. Edit the **~/cmanager/etc/master-apps/_cluster/local/indexes.conf** file with the SmartStore settings provided.

    Once again, you will uniquely identify your remote storage namespace. Re-compose your **path** by replacing **test-<x>** with **idxc-<x>** in **indexes.conf**.

```
vi ~/cmanager/etc/master-apps/_cluster/local/indexes.conf
[default]
remotePath = volume:s3vol/$_index_name
maxGlobalDataSizeMB = 500000
maxDataSize = auto

[volume:s3vol]
storageType = remote
path = s3://lsd1234/idxc-99
remote.s3.access_key = ABCDEFGHIJKLMNOPQRST
remote.s3.secret_key = aBBc1dEEfGhi2jKK3LLmMM4no5pqRRs6TTuVwxYZ
remote.s3.endpoint = https://s3.ap-northeast-1.amazonaws.com
```

NOTE:	In production, the migration process will take a while to complete. If you have a large amount of data, expect some degradation of indexing and search performance during the migration. Schedule the migration for a time when your cluster activities will be idle.
WARNING	Don't forget to adjust the **`maxGlobalDataSizeMB`** and **`frozenTimePeriodInSecs`** settings to avoid unwanted bucket freezing and possible data loss. Remember SmartStore bucket-freezing behavior is different from the non-SmartStore behavior.

14. From the manager node, run **`splunk validate cluster-bundle`** to check for any errors.

15. If no errors are reported, deploy the bundle to the peer nodes: **`splunk apply cluster-bundle`**

```
~/cmanager/bin/splunk validate cluster-bundle
    Created new bundle with checksum=C26EAF9A245D6E8C649316082D0896D9

~/cmanager/bin/splunk show cluster-bundle-status

~/cmanager/bin/splunk apply cluster-bundle

~/cmanager/bin/splunk show cluster-bundle-status
master
    cluster_status=Rolling restart of the peers is in progress.
...
```

Wait for the rolling restart to complete.

Check Your Work

Task 4: Confirm remote storage access across the indexer cluster.

16. Verify the status of the cluster from **`cmanager`**'s Splunk Web.

The **Health Status** indicator is still green, and the cluster is still complete. This may take some time.

17. Run a search to monitor the migration process:

```
| rest /services/admin/cacheman/_metrics splunk_server=idx* | fields
splunk_server migration.*
```

18. Run a search to check if the migration was successful:

```
| rest /services/admin/cacheman splunk_server=idx* | search cm:bucket.stable=0 |
stats count by splunk_server
```

> **NOTE:** When **migration.status** in search from #17 returns `finished` on all peers and the result of search #18 is no results found, the migration is complete.
>
> At this point, you should be able to run normal searches without much delay because the data is already in the local cache.

19. In `dserver`'s Monitoring Console, navigate to **Indexing** > **SmartStore** > **SmartStore Activity: Deployment**.

20. Select the **Enable** check box under **Show Migration Progress** and verify that the remote storage is `ONLINE` and the migration progress is at `100`.

21. Run a search to confirm the **CacheManager** activities (Last 15 minutes):

```
index=_internal host=idx* sourcetype=splunkd component IN(CacheManager*) | table
_time, action, status, cache_id, splunk_server
```

To verify that the cache manager is functioning properly, select **Job** > **Inspect Job**.

> **NOTE:** Scroll down the **Execution costs** list and locate:
>
> ```
> command.search.index.bucketcache.error  -   -   -
> command.search.index.bucketcache.hit   12   -   -
> command.search.index.bucketcache.miss   -   -   -
> ```
>
> The listing of these components and their invocation counts indicate a functioning SmartStore configuration. Ideally, you want the results from only the `command.search.index.bucketcache.hit` invocation.
> If not, wait a little bit and re-run the search.

Troubleshooting Suggestions

If your configuration is not returning the expected results, troubleshoot by isolating the issue.

1. Verify the command syntax and spelling on each instance with: **`splunk btool check --debug`**
2. Carefully review the values specified in **`~/cmanager/etc/system/local/server.conf`**.
3. Change the log level for **`S3Client`** and **`StorageInterface`** to **`DEBUG`** and check **`splunkd.log`** and **`splunkd-utility.log`** for any error details.

© 2021 Splunk Inc. All rights reserved.