Splunk4Admins - Indexer Performance

Lab Guide

Overview

This lab guide contains the hands-on exercises for the Splunk4Admins - Admin Config Service Workshop. Before proceeding with these exercises, please ensure that you have a copy of the workshop slide deck, which will help to put into context the tasks you are carrying out.

Download the workshop slide deck: Splunk4Admins - Admin Config Service - Presenter - March 2025 (…

Prerequisites

In order to complete these exercises, you will need your own Splunk instance. Splunk's hands-on workshops are delivered via the Splunk Show portal and you will need a splunk.com account in order to access this.

If you don't already have a Splunk.com account, please create one here before proceeding with the rest of the workshop.

⚠️ Troubleshooting Connectivity

If you experience connectivity issues with accessing either your workshop environment or the event page, please try the following troubleshooting steps. If you still experience issues please reach out to the team running your workshop.

- **Use Google Chrome** (if you're not already)
- If the event page (i.e. *https://show.splunk.com/event/<eventID>*) didn't load when you clicked on the link, try **refreshing the page**
- **Disconnect from VPN** (if you're using one)
- **Clear your browser cache and restart your browser** (if using Google Chrome, go to: Settings > Privacy and security > Clear browsing data)
- **Try using private browsing mode** (e.g. Incognito in Google Chrome) to rule out any cache issues
- **Try using another computer** such as your personal computer - all you need is a web browser! Cloud platforms like AWS can often be blocked on corporate laptops.

Table of Contents

Description

Steps

1. On the admin host, use the existing file ‘additional_syslog_tokens.json’

Exercise 6 – Manage Roles

Description

Steps

1. On the admin host, create a new role
2. Validate via the GUI
3. Experience an error:

Exercise 7 – Manage Users

Description

Steps

1. On the admin host, create a new user
2. Update the user to add the ‘index-syslog’ role
3. Validate with the CLI

Exercise 8 – Add App - private

Description

Steps

1. On the admin server CLI, install a private app
2. Validate the installation of the app via GUI
3. Experience an App Vetting failure

Exercise 9 – Add App - Splunkbase

Description

Steps

1. On the admin server CLI, install a Splunkbase app
2. Validate the install of Splunk Add-on for Unix and Linux with the GUI:

Links

Exercise 1 – Configure ACS

Description

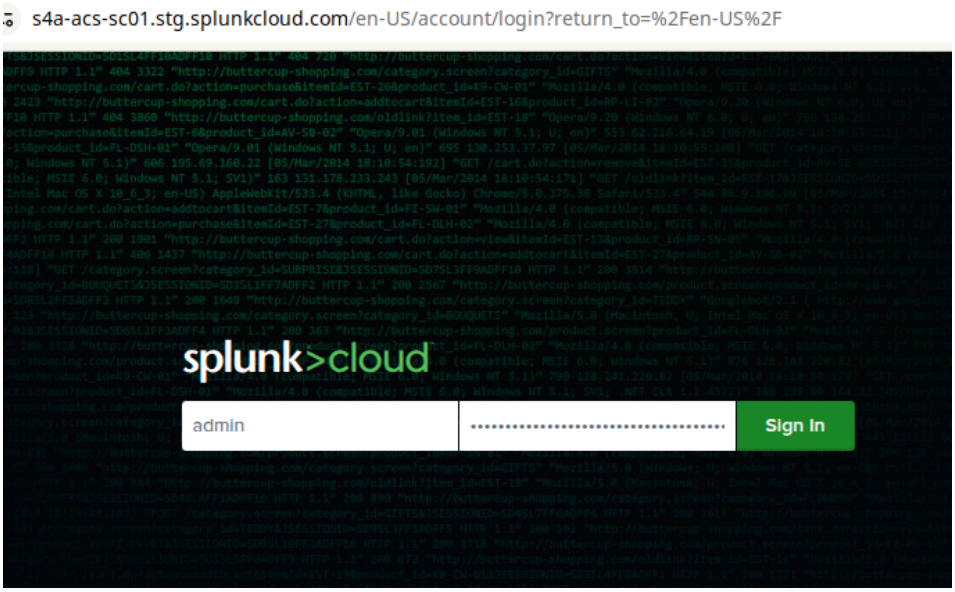
Access the Splunk Cloud stack to configure a user to provide access to ACS services. Create a token to provide authentication for the user.

Access an admin host (small AWS instance) to setup and configure to use the ACS REST interface. Install the Splunk-built ‘acs’ tool from homebrew, configure a stack and setup token access (from the previously set up Cloud user) for using the ACS REST system.

Steps

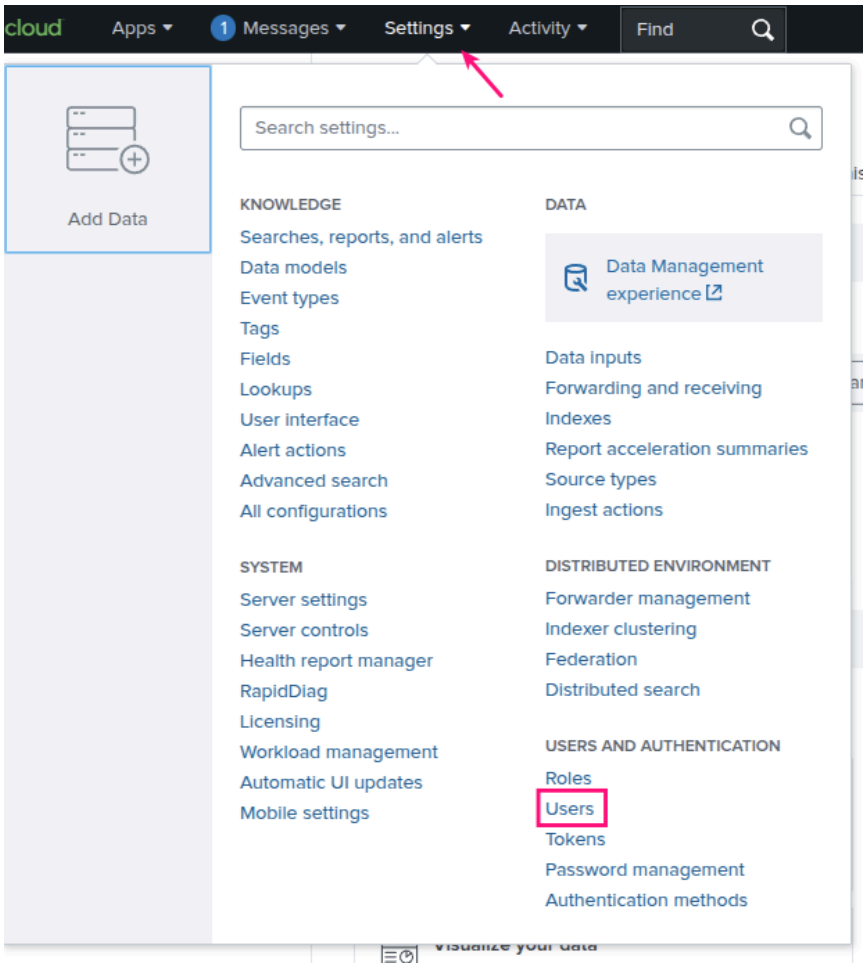
1. Configure ACS - Splunk Cloud

1. **Use the SHOW interface to determine the Cloud Stack and admin host**
   a. Access the environment by clicking on the Splunk Cloud link provided in the Splunk Show interface or in the address bar of a browser window and typing “***<your_assigned_stack>***”.
2. **Login to the GUI**
   a. Using the following credentials:
   Username: `admin`
   Password: `<provided_password>`

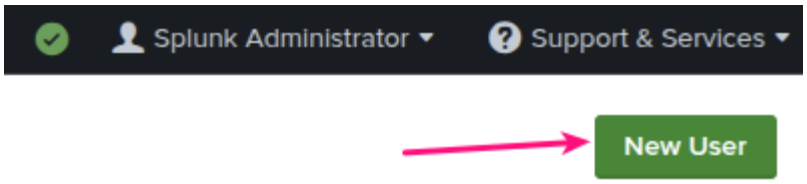


3. **Create the new user**

   a. Once logged in to the GUI, access the 'Users' menu by clicking on the 'Settings' menu item in the black-bar (from now on called 'Black-bar Settings') and then clicking on the blue 'Users' selection.



   b. Click the "New User" green button on the upper right.



   c. In the dialogue that follows, create a new user. the pink boxes provide the location (with blue for the checkbox)

      i. Name = `acs_admin`

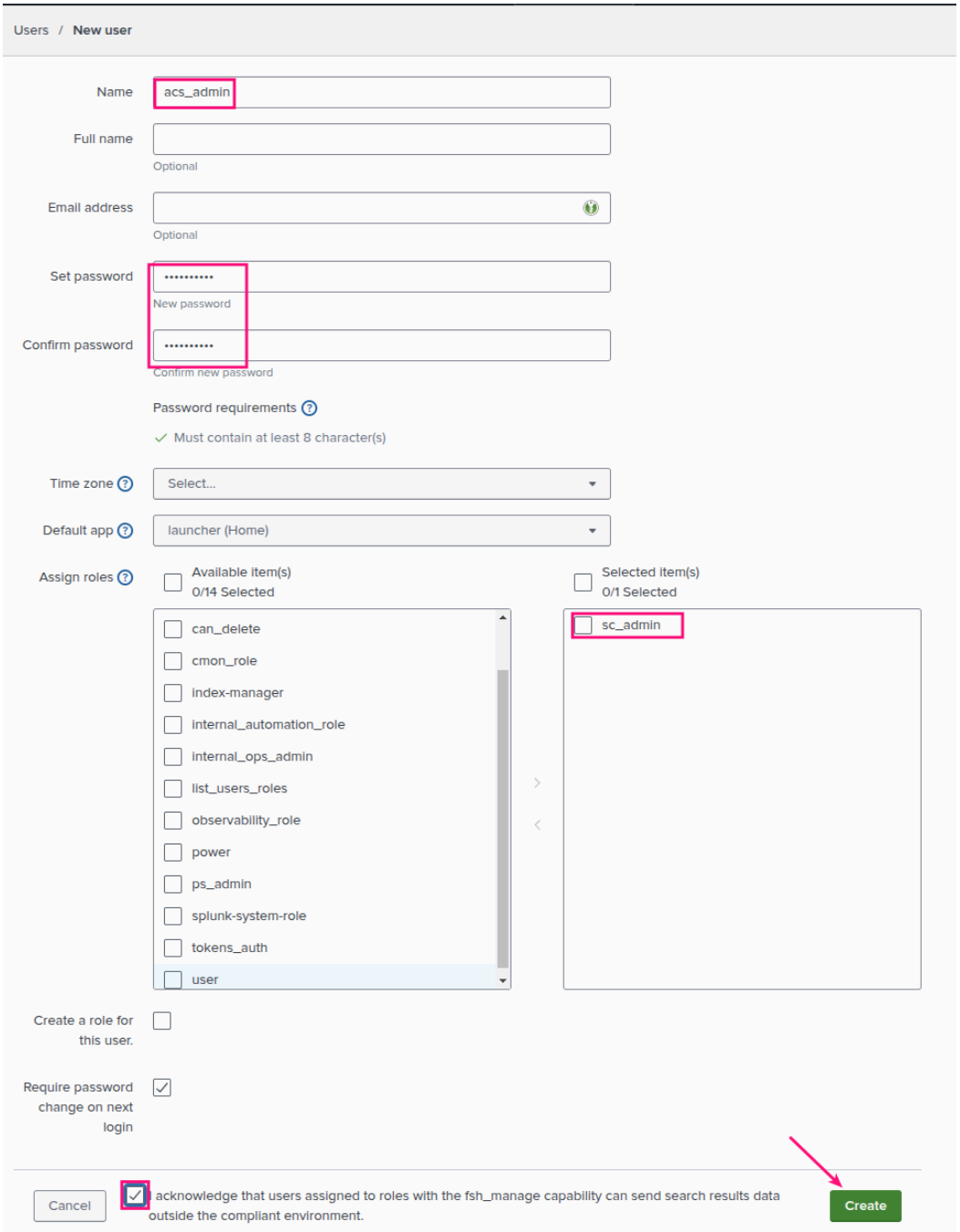
      ii. Password = `5p1unk.conf`

      iii. Remove the "`user`" role

iv. Add the “`sc_admin`” role

v. Uncheck the “`Require password change on next login`”

vi. Select “I acknowledge that users …” checkbox

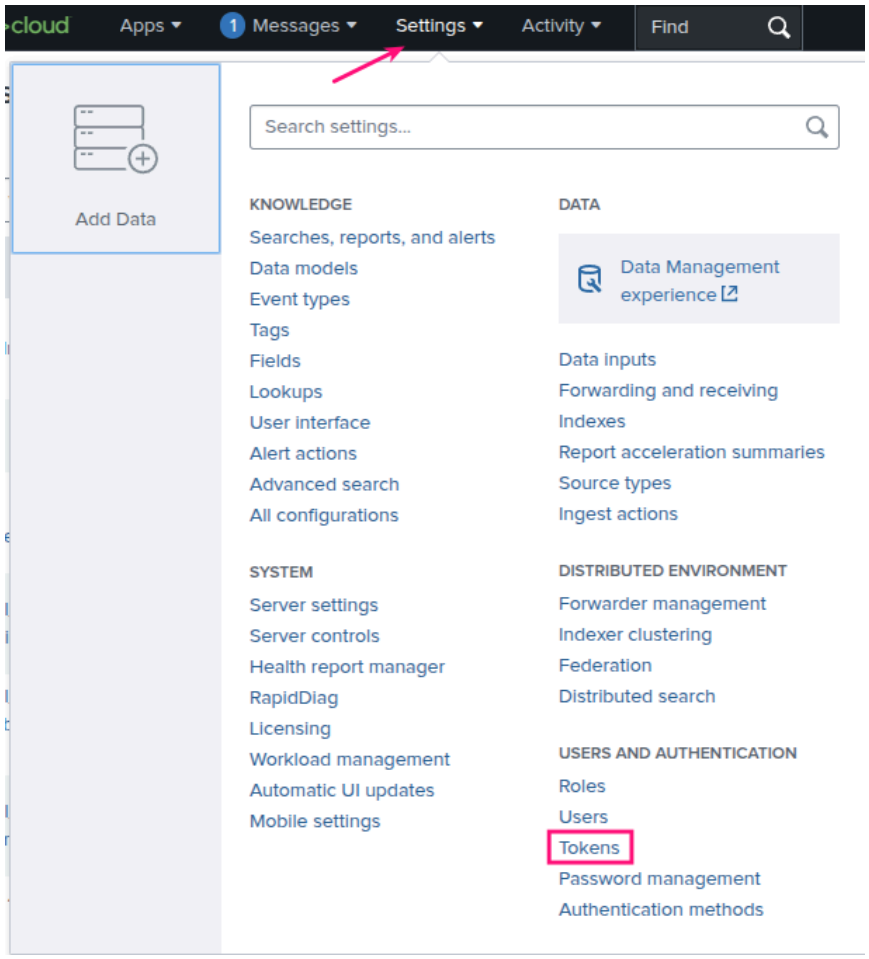


d. Click on the green ‘Create’ button to create the user ‘acs_admin’ user with the ‘sc_admin’ role.

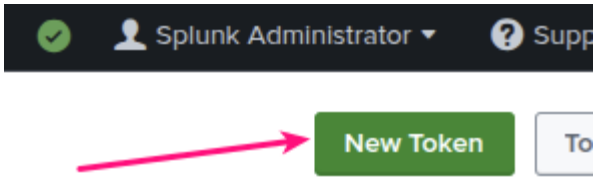
e.

4. Create the authentication token for the user

a. Click on the black bar “Settings”, then select “Tokens”



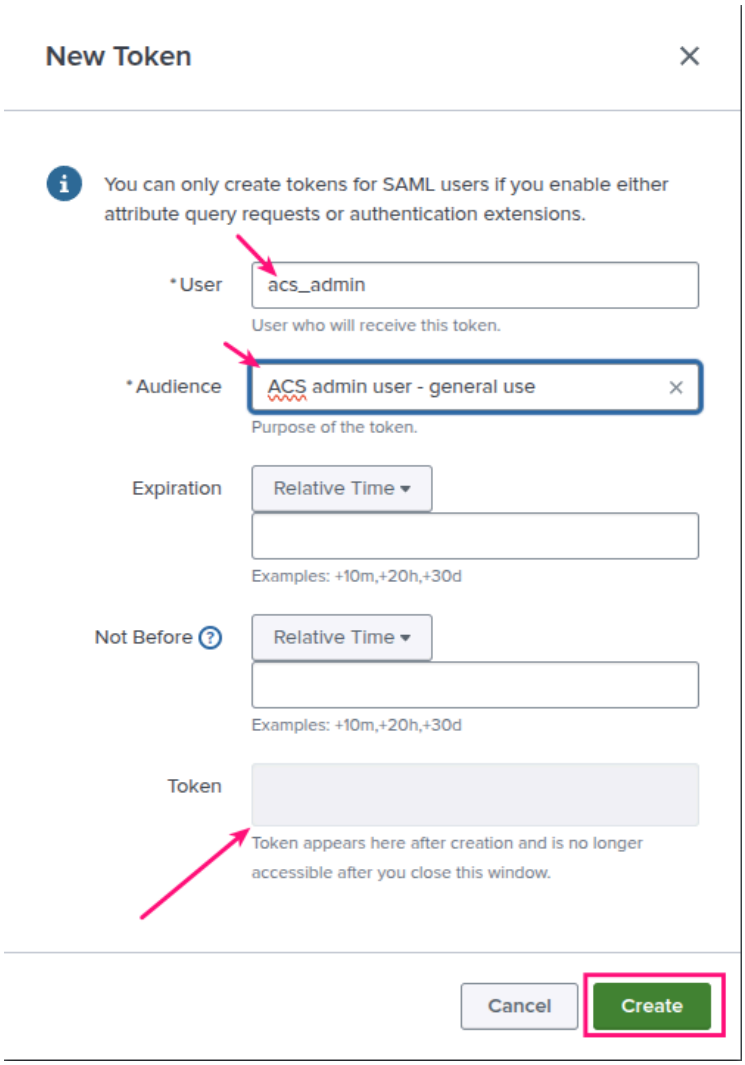
b. On the resulting page, click on the “New Token” green button in the upper right.



c. Complete the form provided

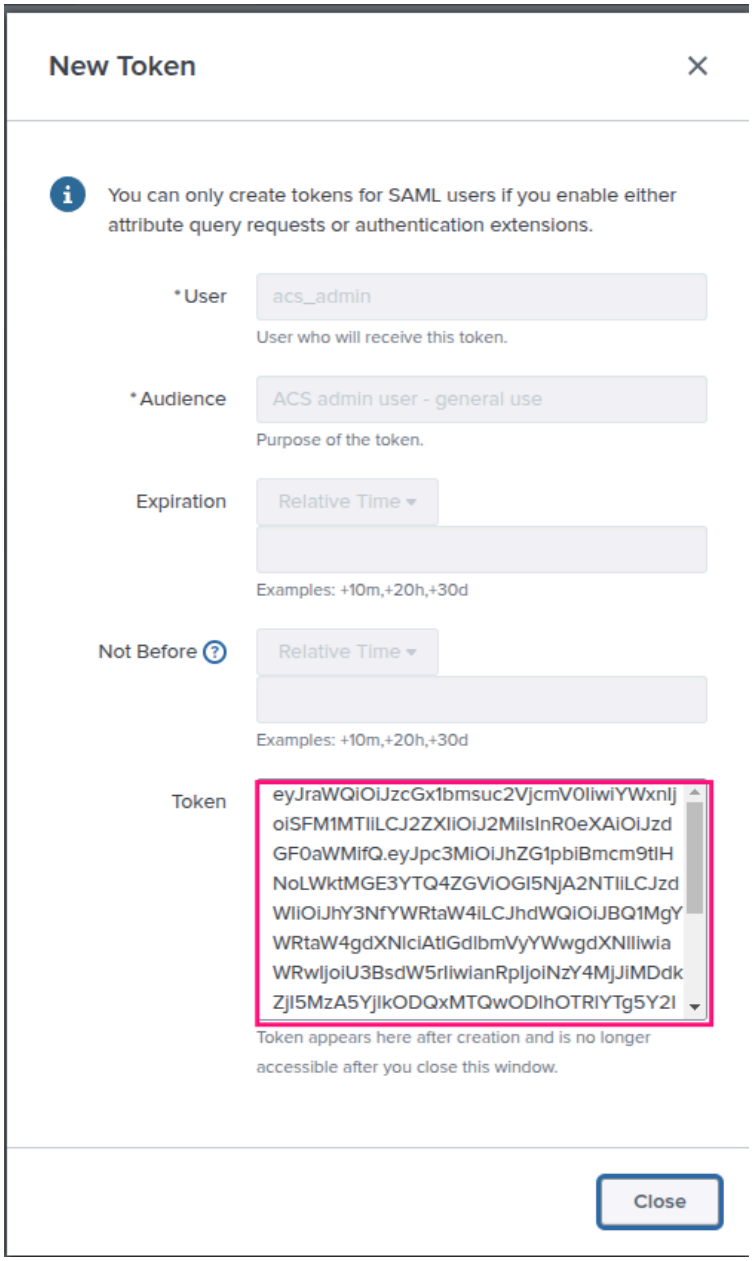
   i. User = “`acs_admin`”

   ii. Audience = “`acs`”



d. “Expiration” and “Not Before” can be used to ensure the life-cycle and life-span of the token is tightly managed. This lab will not use these parameters. The Token will appear in the ‘Token’ field once the “Create” button is pushed.

e. Push the “Create” green button.



f. Be sure to capture the token into a text editing program, or in a production setting directly into the vault system that the environment uses.

2. Configure the 'admin host'

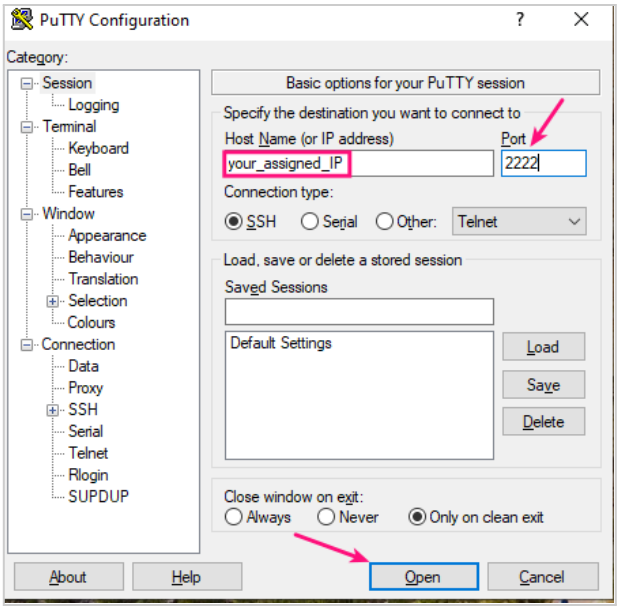
b. Access the "admin host" and then the Splunk systems command line interface

i. **The command line version for both linux and macOs [note the lowercase "p" to change the port 2222] and** change "<your_assigned_IP>" to be the actual assigned IP or hostname, with the password "`5p1unk.conf`"

```
ssh -p 2222 attendee@<your_assigned_IP>
```

```
snoop 03/03/25 > ssh -p 22 attendee@[redacted]
(attendee@3.87.255.15) Password:
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1021-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Mon Mar  3 22:10:25 UTC 2025

  System load:  0.0               Temperature:           -273.1 C
  Usage of /:   41.7% of 6.71GB   Processes:             113
  Memory usage: 34%               Users logged in:       0
  Swap usage:   0%                IPv4 address for ens5: 172.31.86.240

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

   https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

95 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


*** System restart required ***
Last login: Tue Jan 28 22:48:16 2025 from 169.197.22.130
attendee@ip-172-31-86-240:~$
```

ii. **In Windows, here with PuTTY - note the GUI** (change "<your_assigned_IP>" to be the actual assigned IP): same passwd as above "`5p1unk.conf`". Be sure to enter the login name as 'attendeee' and change the port to '2222'.



c. Install the Splunk-built tool ‘acs’ using ‘homebrew’

i. Execute ‘brew update’ to ensure the repository database is up-to-date.

```
brew update
```

ii. Execute ‘brew tap splunk/tap’ to configure the local homebrew to access the Splunk repo. The ‘tap’ command tells homebrew that to use the ‘splunk’ ‘cask’ (what homebrew calls their software sub-repos) where the ‘acs’ software is stored. This process uses a git repository with pre-built binaries and downloads the most current version.

```
brew tap splunk/tap
```

iii. Once the ‘cask’ is ‘tapped’, install the software (pour from cask through the tap).

```
brew install acs
```

```
attendee@ip-172-31-86-2:~$ brew update && brew tap splunk/tap && brew install acs
==> Updating Homebrew...
==> Homebrew collects anonymous analytics.
Read the analytics documentation (and how to opt-out) here:
  https://docs.brew.sh/Analytics
No analytics have been recorded yet (nor will be during this `brew` run).

==> Homebrew is run entirely by unpaid volunteers. Please consider donating:
  https://github.com/Homebrew/brew#donations

Already up-to-date.
==> Tapping splunk/tap
Cloning into '/home/linuxbrew/.linuxbrew/Homebrew/Library/Taps/splunk/homebrew-tap'...
remote: Enumerating objects: 282, done.
remote: Counting objects: 100% (282/282), done.
remote: Compressing objects: 100% (188/188), done.
remote: Total 282 (delta 110), reused 259 (delta 94), pack-reused 0 (from 0)
Receiving objects: 100% (282/282), 52.63 KiB | 10.52 MiB/s, done.
Resolving deltas: 100% (110/110), done.
Tapped 3 formulae (16 files, 128.0KB).
==> Fetching splunk/tap/acs
==> Downloading https://github.com/splunk/acs-cli/releases/download/v2.16.0/acs_v2.16.0_linux_amd64.tar.gz
==> Downloading from https://objects.githubusercontent.com/github-production-release-asset-2e65be/451658559/d
##################################################################################################################
==> Installing acs from splunk/tap
🍺  /home/linuxbrew/.linuxbrew/Cellar/acs/2.16.0: 4 files, 12.9MB, built in 3 seconds
==> Running `brew cleanup acs`...
Disable this behaviour by setting HOMEBREW_NO_INSTALL_CLEANUP.
Hide these hints with HOMEBREW_NO_ENV_HINTS (see `man brew`).
```

d. Configure the tool (acs) for access to the Splunk Cloud stack.

i. Use a shell memory variable to set the stack for reuse, in the example the stack name is ‘s4a-acs-02’. then make acs aware of the stack to operate against with the ‘config add-stack’, then ensure that acs is set to use the stack in the following commands with ‘config use-stack’. Then, configure acs to be able to operate against the stack by using the ‘login’ to pull the token into the local configuration.

```
export SPLUNKCLOUD_STACK_NAME=s4a-acs-sc01
acs config add-stack ${SPLUNKCLOUD_STACK_NAME}
acs config use-stack ${SPLUNKCLOUD_STACK_NAME}
acs login --token-user=acs_admin
```

ii. Username -= ‘`acs_admin`’

iii. Password = ‘`5p1unk.conf`’

iv. The output will be a “Token successfully cached …”

```
attendee@ip-172-31-86-2:~$ export SPLUNKCLOUD_STACK_NAME="s4a-acs-02"
acs config add-stack ${SPLUNKCLOUD_STACK_NAME}
acs config use-stack ${SPLUNKCLOUD_STACK_NAME}
acs login --token-user=acs_admin

Stack added: s4a-acs-02, Type: victoria

Please run "acs config use-stack" to use this stack for further CLI operations.

current-stack is now set to s4a-acs-02

Please run "acs login" command to create and cache token to authenticate against ACS for further commands on this stack.
Enter Username: acs_admin
Enter Password:
Token successfully cached to authenticate against ACS for further operations
Populating stack type...
{
    "user": "acs_admin",
    "audience": "acs_admin",
    "id": "7e718b4018ac830572b435359a341e5084742bf30dac3d3c1bac249d10043105",
    "token": "eyJraWQiOiJzcGx1bmsuc2VjcmV0IiwiYWxnIjoiSFM1MTIiLCJ2ZXIiOiJ2MiIsInR0eXAiOiJzdGF0aWMifQ.eyJpc3MiOiJhY3NfYWRtaW4g2
nRpIjoiN2U3MThiNDAxOGFjODMwNTcyYjQzNTM1OWEzNDF1NTA4NDc0MmJmMzBkYWMzZDNjMWJhYzI0OWQxMDA0MzEwNSIsImlhdCI6MTczODEwNjA4MiwiZXhwIjo
ktUkMUHn3TNdFWLQ",
    "status": "enabled",
    "expiresOn": "2025-02-27T23:14:42Z",
    "notBefore": "2025-01-28T23:14:42Z"
}

attendee@ip-172-31-86-2:~$
```

Exercise '-' Help!

Description

- Experience command line help with the Splunk-built 'acs', and, at the same time, learn the command structure of the 'acs' application.

Steps

1. On the command line of the admin host

a. Type 'acs'.

i. This will execute the command without parameters or options, which the application interprets as '`acs --help`'. That is two dashes before help.

```
attendee@ip-172-31-86-240:~$ acs
API for managing splunk cloud stacks.
The Admin Config Service (ACS) is a cloud-native API that provides programmatic self-servi
ce administration capabilities for Splunk Cloud Platform.
Splunk Cloud Platform administrators can use the ACS API to perform common administrative
tasks without assistance from Splunk Support.
Example usage:
Begin by setting up and logging into an ACS stack:   "acs setup <stack-name>"
Alternatively, to perform these actions one at a time, begin by running the "acs config" c
ommand:
Add a stack to CLI config:                           "acs config add-stack <stack-name>"
Switch to using this stack for further operations:   "acs config use-stack <stack-name>"
Login to the stack to cache authentication info:     "acs login"
Run further commands for ACS APIs, for ex:           "acs token", "acs apps", etc

Exit Codes:
        1 - General Error
        2 - HTTP Request completed with unsuccessful return code.

Usage:
  acs [command]

Available Commands:
  apps                Manage apps
  capabilities        Manage capabilities
  completion          Generate the autocompletion script for the specified shell
  config              Configure the Admin Config Service CLI
  deployment          Manage task deployment
  hec-token           Manage HEC tokens
  help                Help about any command
```

ii. Just ‘acs’ provides information about the command, including:

1. What acs is and how it relates to Splunk Cloud
2. The setup process (as the previous lab demonstrated)
3. Exit and error codes
4. A comprehensive list of available commands (top-level). This correlates to the REST endpoints on the configured stack.

b. Type ‘acs hec-token’.

i. This will execute the sub-command ‘`acs hec-token --help`’ help and provide information about what can be done to manage HEC tokens on the defined stack.

```
attendee@ip-172-31-86-240:~$ acs hec-token
The HTTP Event Collector (HEC) lets you send data and application events to your Splunk
 deployment over HTTP protocol using token-based authentication.
You can use the Admin Config Service (ACS) CLI to create and manage HEC tokens for your
 Splunk Cloud Platform deployment programmatically.
The ACS CLI is available for HEC token management on Splunk Cloud Platform deployments
on Victoria Experience only.
If your Splunk Cloud Platform deployment is on Classic Experience, you can alternativel
y use Splunk Cloud Classic endpoints for HEC token management, please refer to Splunk d
ocumentation for more details.

To manage HEC tokens using the ACS API, you must:
- Have the sc_admin (Splunk Cloud Platform Administrator) role.
- Have Splunk Cloud Platform version 8.1.2101 or higher.

WARNING: It can take several seconds for HEC token CRUD requests to complete.

Usage:
  acs hec-token [command]

Available Commands:
  bulk-create Create HEC tokens in bulk
  bulk-delete Delete HEC tokens in bulk
  bulk-update Update HEC tokens in bulk
  create      Create a new HEC token
  delete      Delete a HEC token
  describe    Describe an individual HEC token
  list        List existing HEC tokens in your environment
  update      Update an existing HEC token

Flags:
  -h, --help   help for hec-token

Global Flags:
  -f, --format string   Controls the output format of the CLI.
                        The "pretty" format is optimized for interactive use.
                        The "structured" format is more parsable for automation. (defau
lt "pretty")
      --server string   Override server URL (default "https://staging.admin.splunk.com"
)
  -v, --verbose         Configures the client to print detailed messages. Extra message
s printed in verbose mode respect the current format setting.

Use "acs hec-token [command] --help" for more information about a command.
```

ii. This provides action commands for the ‘hec-token’ endpoint.

1. bulk-create
2. bulk-delete
3. bulk-update
4. create
5. delete
6. describe
7. list
8. update

iii. With information relevant about what each of the different action commands do within the REST interface.

c. **Type ‘acs hec-token create –help’.**

i. This executes the action command ‘create’ (`acs hec-token create --help`) for the endpoint/sub-command ‘hec-token’ and provides the parameters and options that can be used on the action

```
attendee@ip-172-31-86-240:~$ acs hec-token create --help
Create a new HEC token. You must specify a unique token name and a default index.
You can optionally specify an existing token.
You can also optionally specify the useACK option, as a boolean value (true/false), to
enable or disable indexer acknowledgement.
Splunk Cloud Platform currently supports indexer acknowledgement for Amazon Web Service
s (AWS) Kinesis Firehose only.
Make sure the specified default index exists on your system. Specifying an index that d
oes not exist can cause data loss.

Once the request completes successfully, ACS will reply with the new HEC token. Note th
at it can take several seconds for the token creation request to complete.

Usage:
  acs hec-token create [flags]

Flags:
      --name string                  HEC token name (required)
      --default-index string         default index
      --default-source string        default source
      --default-source-type string   default source type
      --disabled                     specify '--disabled=true to disable
      --default-host string          default host
      --token string                 existing token
      --use-ack                      specify '--use-ack=true' to enable indexer acknowl
edgement
      --allowed-indexes strings      allowed indexes
  -h, --help                         help for create

Global Flags:
  -f, --format string   Controls the output format of the CLI.
                        The "pretty" format is optimized for interactive use.
                        The "structured" format is more parsable for automation. (defau
lt "pretty")
      --server string   Override server URL (default "https://staging.admin.splunk.com"
)
  -v, --verbose         Configures the client to print detailed messages. Extra message
s printed in verbose mode respect the current format setting.
```

ii. The available options:

1. name
2. default-index
3. default-source
4. default-source-type
5. disabled
6. token
7. use-ack

iii. With information relevant to each of the parameters and how they should be executed.

Exercise 2 – ACS: Create indexes

Description

- Explore creating an index to learn more about how the ACS REST commands interact with the Splunk Cloud instance.

Steps

1. On the command line of the admin host

a. Execute the command to add an index:

```
acs indexes create --name syslog --data-type event --max-data-size-mb 1000
--searchable-days 50
```

i. This will execute the command to create a new index with the following configuration parameters:

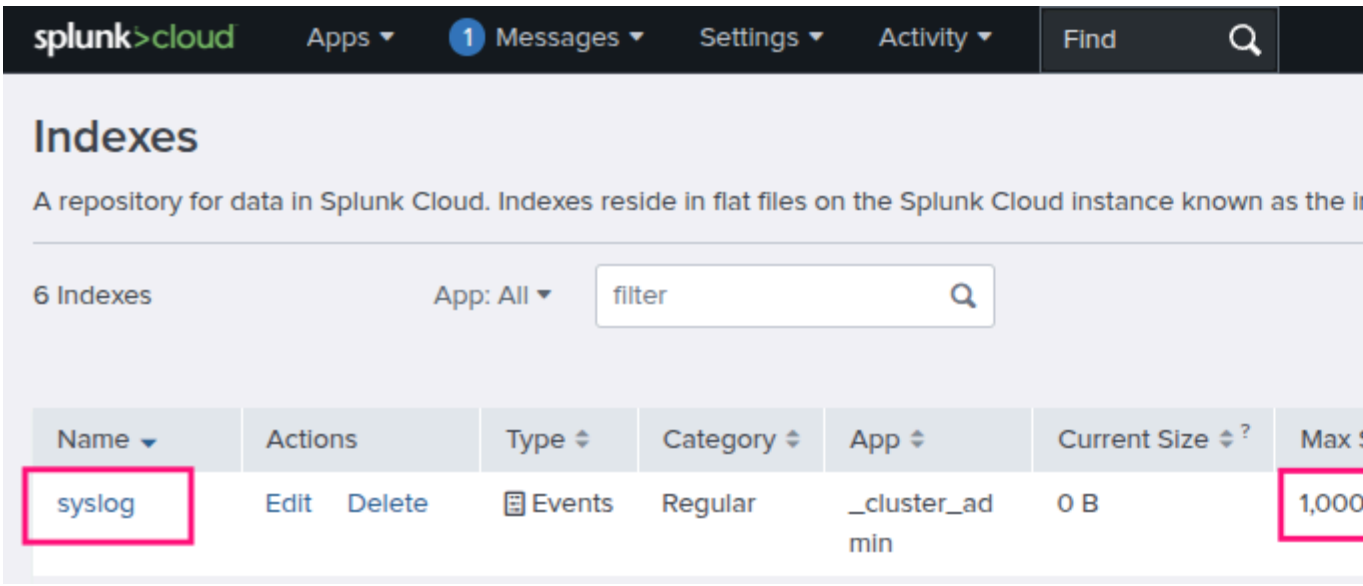
1. Name = syslog (the label of the index)
2. Data-type = event (what type of data will be stored, either *event* or *metrics*)
3. Max-data-size-mb = 1000 (in megabytes, maximum size of the index)
4. searchable-days = 50 (window of time in which events will be retained in a searchable state within the Splunk Cloud instance)

ii.

```
attendee@ip-172-31-86-2:~$ acs indexes create --name syslog --data-type event --max-data-size-mb 1000 --searchable-days 50
Index creation in progress (52c0073c-e9ae-9734-b4df-c1f3d6558baa) .....
{
    "datatype": "event",
    "maxDataSizeMB": 1000,
    "name": "syslog",
    "searchableDays": 50,
    "totalEventCount": "0",
    "totalRawSizeMB": "0"
}

attendee@ip-172-31-86-2:~$
```

iii. Output, in JSON, shows a successful response. The initial text “Index creation in progress …” will show once the REST connection has been made and a successful request for creation has been initiated on the Cloud stack. Success will show in the JSON output.

b. **Validate on Cloud**

c. 

2. Use Curl

   a. Use the common ‘curl’ to access the interface. With ‘curl’, knowledge of the endpoints and the parameter and options is required. Additionally, the token for access must be known and included on the command line

   b. To get a list of existing indexes: (the command below is all one line, be sure to copy it entirely)

```
curl 'https://staging.admin.splunk.com/s4a-acs-02/adminconfig/v2/indexes' --header
'Authorization: Bearer <your_token>' | jq
```

   c. The URL includes ‘staging’ as this is an environment dedicated to this lab. For customer environments, the URL is ‘admin.splunk.com’.

   d. The name of the stack is ‘s4a-acs-02’, the remainder of the URL is fixed, and then the endpoint is ‘indexes’

   e. The ‘--header’ includes the ‘Authorization: Bearer <token>’ is the token extracted from the lab stack. This will be different for each stack. Each attendee has a different stack. The token will need to have been extracted from the the GUI interaction in Lab 1, or, the token is available at ‘`~/.acs/acs_config.json`’.

```
attendee@ip-172-31-86-240:~/.acs$ more acs_config.json
{
  "current-stack": "s4a-acs-02",
  "stacks": {
    "s4a-acs-02": {
      "current-search-head": "",
      "token": "eyJraWQiOiJzcGx1bmsuc2VjcmV0IiwiYWxnIjoiSFM1MTIiLCJ2ZXIiOiJ2MiIsInR0eXAiOiJzdGF0a
WMifQ.eyJpc3MiOiJhY3NfYWRtaW4gZnJvbSBzaC1pLTA1ZDVlMzgwYmQ2NzE1NDg1Iiwic3ViIjoiYWNzX2FkbWluIiwiYXV
kIjoiYWNzX2FkbWluIiwiaWRwIjoiU3BsdW5rIiwianRpIjoiNjNkNWU3NTVlZmEwNzRjNTQ3YTI1ODQ5MGNmMTU4YmQ2NjZh
YTkzMWMxNGJmNGE1N2QyODY5Njk4ZDJjY2I3MyIsImlhdCI6MTczODM0ODI1MSwiZXhwIjoxNzQwOTQwMjUxLCJuYnIiOjE3M
zgzNDgyNTF9.mgdMHu5glD_yOnU3CAQG__cxVtpZlItTfpLOwN-HnPO2OoiUaDuMfswnXGxxXvjk5V0pfZ3TEtOZVxbOEiFnX
w",
      "token-id": "63d5e755efa074c547a258490cf158bd666aa931c14bf4a57d2869698d2ccb73",
      "type": "classic"
    }
  }
}
```

f.

Exercise 3 – Create IP Allow list

Description

Experience and understand how creating an IP Allow list can be executed from the command line.

Steps

1. **On the admin host, create an IP allow list**

   a. Execute the command: (Be sure to change the final octet to 0)

```
acs ip-allowlist create search-api --subnets <your_first_three_octets.0/24>
```

```
attendee@ip-172-31-86-2:~$ acs ip-allowlist create search-api --subnets 18.212.166.0/24
IP allow list subnets creation request submitted successfully for feature:search-api
Note that it can take several minutes for the subnet update to be applied to your Splunk Cloud Platform stack.
To verify the status of your stack after subnet update request, please run the "acs status current-stack" command.
{
    "warnings": [
        "IP allow list subnets creation request submitted successfully. Note that it can take several minutes for the subnet update to be applied to your Splunk Cloud Platform st
    ]
}
```

   b. This shows success. The subnet will be checked for validity and an error message returned if the subnet and subnet mask are impossible. Get his information from the networking team responsible. As noted, this action can take time, especially if this is operating against many other hosts, such as a set of indexers for either HEC allow lists or S2S.

2. **Validate the creation of the Allow list from the admin host**

   a. Execute the command:

```
acs ip-allowlist describe search-api
```

```
attendee@ip-172-31-86-2:~$ acs ip-allowlist describe search-api
{
    "subnets": [
        "[redacted]/24"
    ]
}
```

   b. Here the command is ‘ip-allowlist describe search-api’ is executed to show the configured IP Allow list.

3. Validate in the Cloud GUI

Indexing Performance by Instance

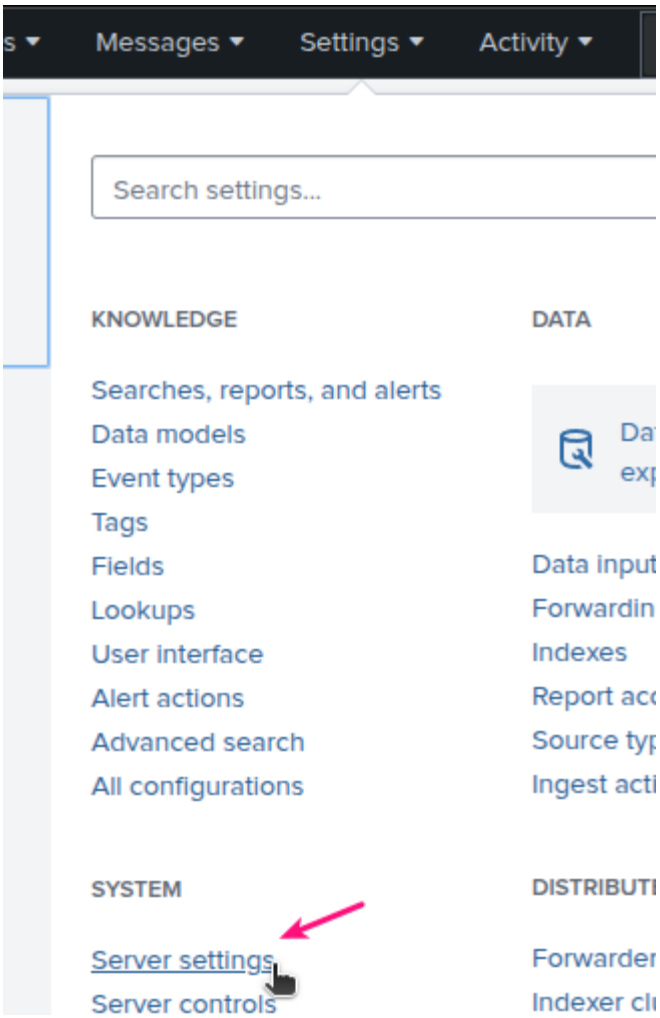
3 Instances

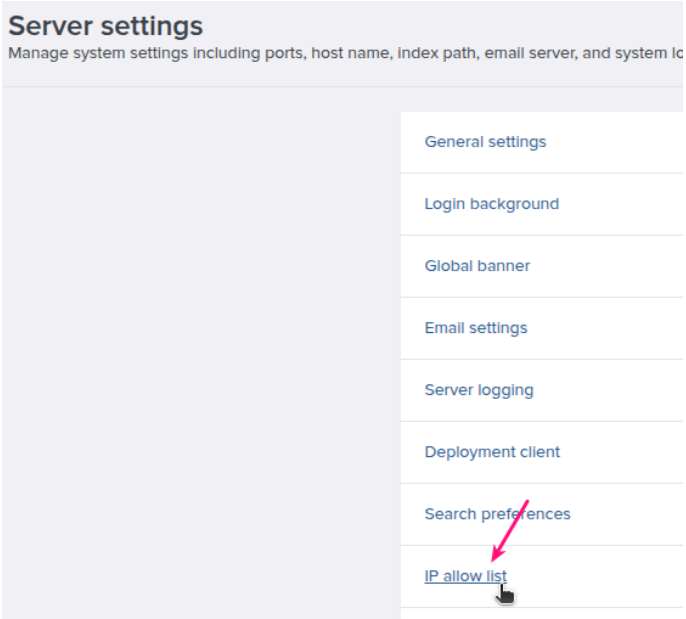
Instance	Pipeline Set Count	Indexing Rate (KB/s)	Status	Parsing Queue Fill Ratio (%)	Aggregation Queue Fill Ratio (%)	Typing Queue Fill Ratio (%)	Indexing Queue Fill Ratio (%)
s4a-ip-idx02	2	854	normal	pset0: 95.28 pset1: 0.00	pset0: 86.67 pset1: 0.00	pset0: 54.92 pset1: 0.00	pset0: 85.29 pset1: 0.04
s4a-ip-idx03	1	583	normal	99.17	99.96	99.95	100.00
s4a-ip-idx01	1	504	normal	83.25	57.82	70.87	66.32

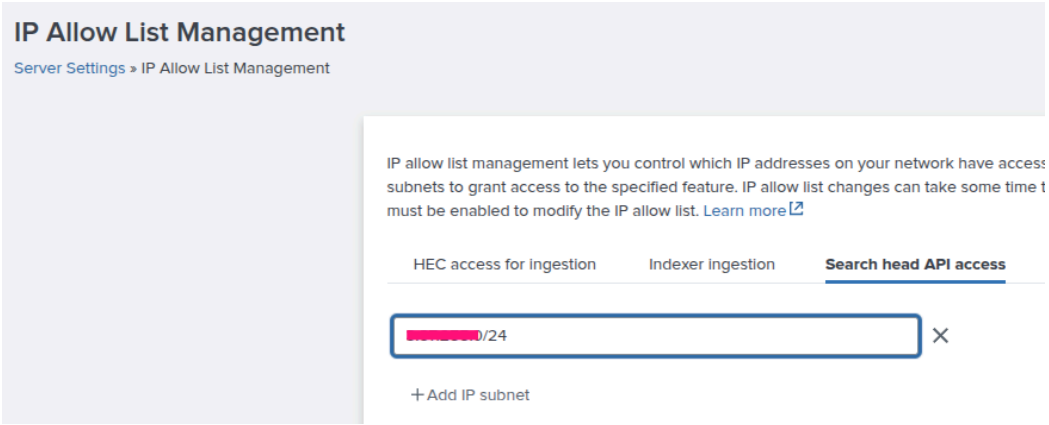
a. Given that each of these instances is identical, the behavior should be expected to be roughly similar. This tends to lead to a hypothesis that the issue is an underlying storage issue

4. Go to the Indexer Performance:Advanced dashboard

a. Select Black-bar ‘Settings’->’Server Settings’->IP allow list’->’Search head API access’







Exercise 4 – Manage HEC token - create

Description

Explore and experience managing HTTPS Event Collector (HEC) tokens via the 'acs' command interface. This demonstrates basic operations, with more complex operations available as well.

Steps

1. **On the admin host, create a HEC token**
    a. Execute the command: (copy both commands in one copy and paste directly onto the CLI)

```
acs hec-token create --name syslog_hec_sys01
acs hec-token create --name syslog_hec_sys02
```

```
attendee@ip-172-31-82-162:~$ acs hec-token create --name syslog_hec_sys01
Hec creation in progress (efd01505-a2b8-94f3-a7eb-233dbe07e308) ...

{
    "http-event-collector": {
        "spec": {
            "allowedIndexes": null,
            "defaultHost": "",
            "defaultIndex": "",
            "defaultSource": "",
            "defaultSourcetype": "",
            "disabled": false,
            "name": "syslog_hec_sys01",
            "useAck": false
        },
        "token": "7F04BCBE-8EDE-4633-83D1-4844DF13F2EA"
    }
}
```

b. Here the output shows a successful creation.

```
attendee@ip-172-31-82-162:~$ acs hec-token create --name syslog_hec_sys02
A deployment task is still in progress. Please try again later. Please refer https://docs.splunk
.com/Documentation/SplunkCloud/latest/Config/ACSerrormessages for general troubleshooting tips.
{
    "code": "424-dependency-incomplete",
    "message": "A deployment task is still in progress. Please try again later. Please refer htt
ps://docs.splunk.com/Documentation/SplunkCloud/latest/Config/ACSerrormessages for general troubl
eshooting tips."
}

Error: Create hec token failed: POST request to "https://staging.admin.splunk.com/s4a-acs-02/adm
inconfig/v2/inputs/http-event-collectors" failed, code: 424 Failed Dependency
```

c. But the second command demonstrates what happens when two commands are executed one after the other before and the first has not completed. Splunk Cloud will respond with a ‘424’ which is effectively ‘wait a moment’.

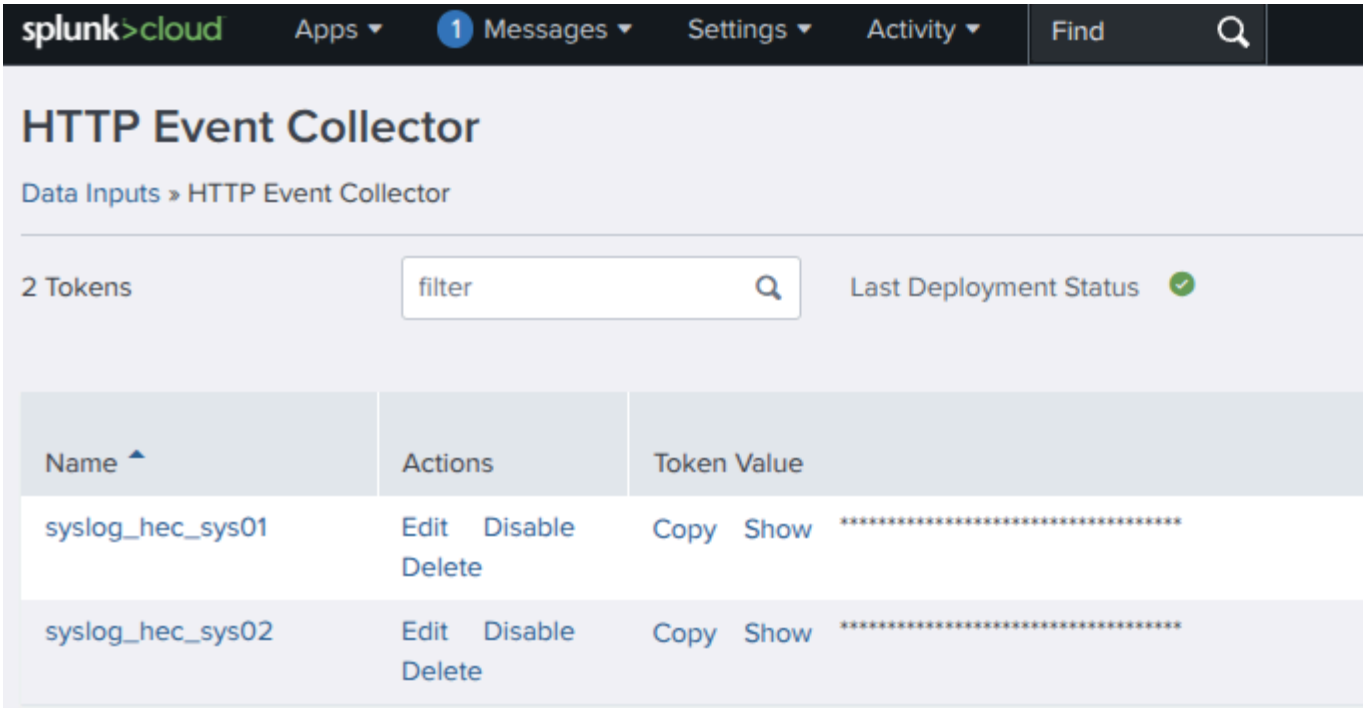
d. Execute the second command again (or just repaste the original), and an error is thrown that *‘syslog_hec_01’* already exists (a **‘429’**), but then the second command will complete.

e.

```
attendee@ip-172-31-86-240:~$ acs hec-token create --name syslog_hec_sys01
acs hec-token create --name syslog_hec_sys02
Hec name syslog_hec_sys01 already exists. To update an existing HEC token, please use PUT/PATCH. P
lease refer https://docs.splunk.com/Documentation/SplunkCloud/latest/Config/ACSerrormessages for g
eneral troubleshooting tips.
{
    "code": "409-object-already-exists",
    "message": "Hec name syslog_hec_sys01 already exists. To update an existing HEC token, please
use PUT/PATCH. Please refer https://docs.splunk.com/Documentation/SplunkCloud/latest/Config/ACSerr
ormessages for general troubleshooting tips."
}

Error: Create hec token failed: POST request to "https://staging.admin.splunk.com/s4a-acs-02/admin
config/v2/inputs/http-event-collectors" failed, code: 409 Conflict
Hec creation in progress (e2a87cae-59f9-9bbc-bba0-d4856607ef4b) ...

{
    "http-event-collector": {
        "spec": {
            "allowedIndexes": null,
            "defaultHost": "",
            "defaultIndex": "",
            "defaultSource": "",
            "defaultSourcetype": "",
            "disabled": false,
            "name": "syslog_hec_sys02",
            "useAck": false
        },
        "token": "97E1C694-B015-4653-8CC7-CF1DDC887E35"
    }
}
```

2. Validate in the GUI

a. On the assigned Splunk Cloud instance, black-bar ‘Settings’->’Data Inputs’->’HTTP Event Collector’



Exercise 5 – Manage HEC token - bulk create

Description

Explore and experience managing HTTPS Event Collector (HEC) tokens via the ‘acs’ command interface in bulk using an input JSON file. This demonstrates basic operations, with more complex operations available as well.

Steps

1. **On the admin host, use the existing file ‘additional_syslog_tokens.json’**
    a. Observe the file - ‘‘additional_syslog_tokens.json’ with the command below:

```
more additional_syslog_tokens.json
```

    b. The file should be in the home directory of ‘attendee’.
    c. Execute the command to create the tokens from the file:

```
acs hec-token bulk-create --file ./additional_syslog_tokens.json
```

```
attendee@ip-172-31-82-162:~$ acs hec-token bulk-create --file ./additional_syslog_tokens.json

HEC creation (syslog_token_sys01) in progress (e6b7510b-5296-97be-bb02-93ed7ee3ed00) ...
{
    "identifier": "syslog_token_sys01",
    "requestID": "e6b7510b-5296-97be-bb02-93ed7ee3ed00",
    "statusCode": 202,
    "message": "Token creation request submitted successfully. Note that it can take several minutes for the token creation to be applied to your Splunk Clou
d Platform stack.",
    "response": "{\"http-event-collector\":{\"spec\":{\"name\":\"syslog_token_sys01\"}}}\n",
    "content-type": "application/json; charset=UTF-8"
}
HEC creation (syslog_token_sys02) in progress (d5d00965-abbc-959e-8f51-8cc9bffd765d) ...
{
    "identifier": "syslog_token_sys02",
    "requestID": "d5d00965-abbc-959e-8f51-8cc9bffd765d",
    "statusCode": 202,
    "message": "Token creation request submitted successfully. Note that it can take several minutes for the token creation to be applied to your Splunk Clou
d Platform stack.",
    "response": "{\"http-event-collector\":{\"spec\":{\"name\":\"syslog_token_sys02\"}}}\n",
    "content-type": "application/json; charset=UTF-8"
}
HEC creation (syslog_token_sys03) in progress (ae1b3379-39f3-9e01-8f84-d85fae5ab9f2) ...
{
    "identifier": "syslog_token_sys03",
    "requestID": "ae1b3379-39f3-9e01-8f84-d85fae5ab9f2",
    "statusCode": 202,
    "message": "Token creation request submitted successfully. Note that it can take several minutes for the token creation to be applied to your Splunk Clou
d Platform stack.",
    "response": "{\"http-event-collector\":{\"spec\":{\"name\":\"syslog_token_sys03\"}}}\n",
    "content-type": "application/json; charset=UTF-8"
}
HEC creation (syslog_token_sys04) failed (e3e3d909-d33b-9252-b9d9-ef76ba919255) ...
{
    "identifier": "syslog_token_sys04",
    "requestID": "e3e3d909-d33b-9252-b9d9-ef76ba919255",
    "statusCode": 400,
    "message": "syslog_finance in allowedIndexes is an internal index or is not a valid index. Please refer https://docs.splunk.com/Documentation/SplunkCloud
/latest/Config/ACSerrormessages for general troubleshooting tips.",
    "response": "{\"code\":\"400-bad-request\",\"message\":\"syslog_finance in allowedIndexes is an internal index or is not a valid index. Please refer http
s://docs.splunk.com/Documentation/SplunkCloud/latest/Config/ACSerrormessages for general troubleshooting tips.\"}\n",
    "content-type": "application/json; charset=UTF-8"
}
Bulk Creation Results (success=3, failures=1, skipped=0)
Report for hec_bulk_create is saved at: /home/attendee/.acs/reports/s4a-acs-02/hec_bulk_create_1738198899.json
Error: Bulk create HEC token failed: POST request to "https://staging.admin.splunk.com/s4a-acs-02/adminconfig/v2/inputs/http-event-collectors" failed, code:
400 Bad Request
```

d. The command takes time, as the events occur sequentially, but eventually: (This command, similarly to the prior commands - and others - can take time, watch the HEC Data Inputs page to see a progress bar)
    i. ‘syslog_token_sys01’ is created successfully
    ii. ‘syslog_token_sys02’ is created successfully, with a default index of ‘main’
    iii. ‘syslog_token)sys03’ is created successfully with allowed indexes of ‘syslog’ and ‘main’
    iv. ‘syslog_token_sys04’ is NOT created successfully, but why? The command executes against the appropriate cloud instance and validates that the resources that are called for in the command are available. Here, the index (‘syslog_finance’) that is meant to be both the ‘allowedIndexes’ and the ‘defaultIndex’ does not exist, ‘syslog_finance’.

2. Validate from the CLI the created tokens
    a. Execute the command:

```
acs hec-token list | jq
```

```
attendee@ip-172-31-86-240:~$ acs hec-token list | jq
{
  "http-event-collectors": [
    {
      "spec": {
        "allowedIndexes": null,
        "defaultHost": "",
        "defaultIndex": "main",
        "defaultSource": "",
        "defaultSourcetype": "",
        "disabled": false,
        "name": "syslog_token_sys02",
        "useAck": false
      },
      "token": "CB0A6FF5-EEFA-43E6-A7C7-518250EF4E26"
    },
    {
      "spec": {
        "allowedIndexes": [
          "syslog",
          "main"
        ],
        "defaultHost": "",
        "defaultIndex": "syslog",
        "defaultSource": "",
        "defaultSourcetype": "",
        "disabled": false,
        "name": "syslog_token_sys03",
```

b. Or - filter down a bit - (mulitline, be sure copy both lines)

```
acs hec-token list | jq -r '.["http-event-collectors"][] | "Name: \(.spec.name) ||
Token:\(.token)"'
```

```
attendee@ip-172-31-86-240:~$ acs hec-token list | jq -r '.["http-event-
collectors"][] | "Name: \(.spec.name) || Token:\(.token)"'
Name: syslog_hec_sys01 || Token:5A79C92E-A719-4019-B7D9-C3A51D96CE54
Name: syslog_hec_sys02 || Token:97E1C694-B015-4653-8CC7-CF1DDC887E35
Name: syslog_token_sys01 || Token:22FA1F7F-055F-4E23-93BE-594FB1078022
Name: syslog_token_sys02 || Token:CB0A6FF5-EEFA-43E6-A7C7-518250EF4E26
Name: syslog_token_sys03 || Token:5EB5D9E0-285B-4A25-8D02-0F4F834E08D2
```

Exercise 6 – Manage Roles

Description

ACS can be used to manage roles, with the actions create, delete, modify, individually and in bulk. Experiencing and understanding some of these features helps understand how to build the CLI interface.

Steps

1. On the admin host, create a new role

a. Execute the following command on the admin host CLI to create a new role with a specific capability:

```
acs roles create --name "index-syslog" --srch-indexes-allowed syslog
```

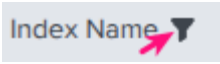
b. This commend creates a role labelled ‘index-syslog’ with the ability to search in the index ‘syslog’.

```
attendee@ip-172-31-86-240:~$ acs roles create --name "index-syslog" --srch-indexes-allowed "syslog"
{
    "cumulativeRTSrchJobsQuota": 100,
    "cumulativeSrchJobsQuota": 50,
    "defaultApp": "",
    "imported": {
        "capabilities": [],
        "rtSrchJobsQuota": 0,
        "srchDiskQuota": 0,
        "srchFilter": "",
        "srchIndexesAllowed": [],
        "srchIndexesDefault": [],
        "srchJobsQuota": 0,
        "srchTimeEarliest": -1,
        "srchTimeWin": -1,
        "roles": []
    },
    "name": "index-syslog",
    "capabilities": [],
    "rtSrchJobsQuota": 6,
    "srchDiskQuota": 100,
    "srchFilter": "",
    "srchIndexesAllowed": [
        "syslog"
    ],
    "srchIndexesDefault": [],
    "srchJobsQuota": 3,
    "srchTimeEarliest": -1,
    "srchTimeWin": -1
}
```

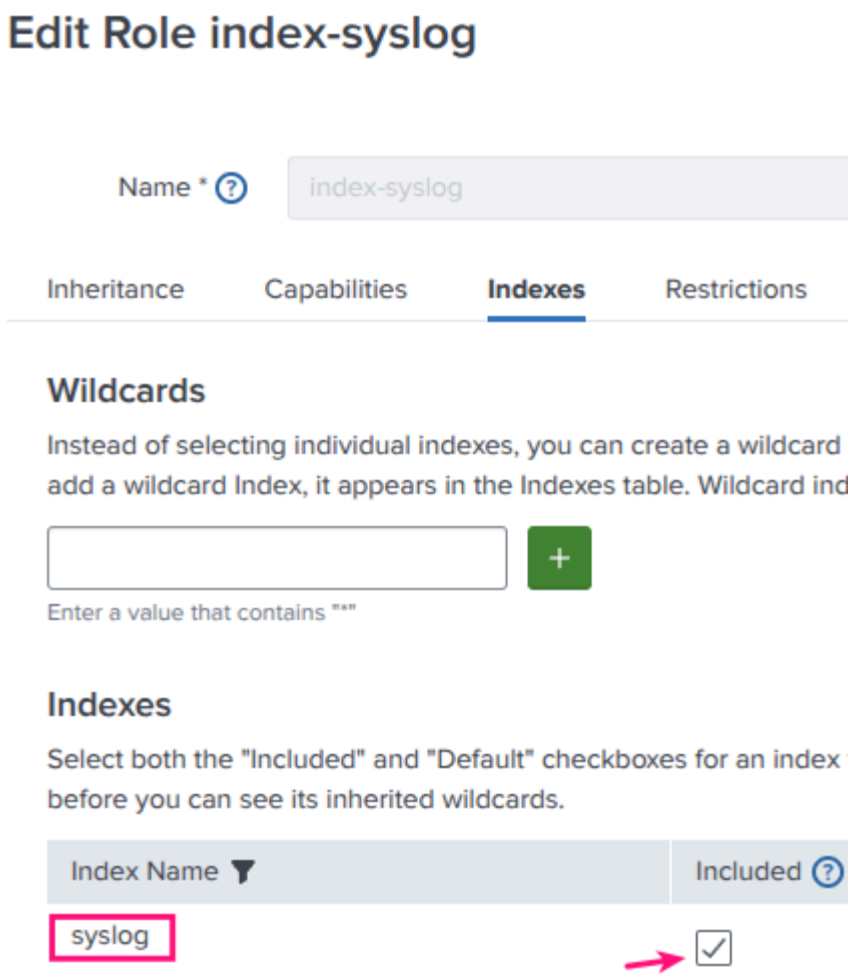
c. The JSON output demonstrates that the role with the correct ‘srchIndexesAllowed’ setting (‘syslog’) is built.

2. Validate via the GUI

a. Select ‘Black-bar Settings’->’Roles’->’index-syslog’->’Indexes’. Scroll down (or filter using the



and type in ‘syslog’) to see the index as ‘Included’ via the checkbox.



3. **Experience an error:**
   a. Execute the command below:

```
acs roles create --name index-syslog1 --srch-indexes-allowed syslog1
```

```
attendee@ip-172-31-86-240:~$ acs roles create --name "index-syslog1" --srch-indexes-allowed "sysl
cannot set index name 'syslog1': index does not exist. Please refer https://docs.splunk.com/Docum
n/SplunkCloud/latest/Config/ACSerrormessages for general troubleshooting tips.
{
    "code": "400-bad-request",
    "message": "cannot set index name 'syslog1': index does not exist. Please refer https://docs.
com/Documentation/SplunkCloud/latest/Config/ACSerrormessages for general troubleshooting tips."
}

Error: Create role failed: POST request to "https://staging.admin.splunk.com/s4a-acs-02/admincon
oles" failed, code: 400 Bad Request
```

   b. This command attempts to create a new role ‘index-syslog1’ with the ability to search against an index labelled ‘syslog1’, except the index doesn’t exit, so a **‘400’** code is returned.

4. Update a role
   a. Execute the command below:

```
acs roles update index-syslog --srch-indexes-allowed wineventlog
```

```
attendee@ip-172-31-86-240:~$ acs roles update index-syslog
--srch-indexes-allowed wineventlog


{
    "cumulativeRTSrchJobsQuota": 100,
    "cumulativeSrchJobsQuota": 50,
    "defaultApp": "",
    "imported": {
        "capabilities": [],
        "rtSrchJobsQuota": 0,
        "srchDiskQuota": 0,
        "srchFilter": "",
        "srchIndexesAllowed": [],
        "srchIndexesDefault": [],
        "srchJobsQuota": 0,
        "srchTimeEarliest": -1,
        "srchTimeWin": -1,
        "roles": []
    },
    "name": "index-syslog",
    "capabilities": [],
    "rtSrchJobsQuota": 6,
    "srchDiskQuota": 100,
    "srchFilter": "",
    "srchIndexesAllowed": [
        "wineventlog"
    ],
    "srchIndexesDefault": [],
    "srchJobsQuota": 3,
    "srchTimeEarliest": -1,
    "srchTimeWin": -1
}
```

b. Note what has occurred. The previous setting for ‘srchIndexesAllowed’ included ‘syslog’ but this index is not represented here.

c. Important information: ‘update’ always over-writes. It is not an ‘add’.

d. Execute the following command to add the index ‘wineventlog’:

```
acs roles update index-syslog --srch-indexes-allowed wineventlog,syslog
```

e.

```
attendee@ip-172-31-86-240:~$ acs roles update index-syslog
 --srch-indexes-allowed "wineventlog,syslog"
{
    "cumulativeRTSrchJobsQuota": 100,
    "cumulativeSrchJobsQuota": 50,
    "defaultApp": "",
    "imported": {
        "capabilities": [],
        "rtSrchJobsQuota": 0,
        "srchDiskQuota": 0,
        "srchFilter": "",
        "srchIndexesAllowed": [],
        "srchIndexesDefault": [],
        "srchJobsQuota": 0,
        "srchTimeEarliest": -1,
        "srchTimeWin": -1,
        "roles": []
    },
    "name": "index-syslog",
    "capabilities": [],
    "rtSrchJobsQuota": 6,
    "srchDiskQuota": 100,
    "srchFilter": "",
    "srchIndexesAllowed": [
        "syslog",
        "wineventlog"
    ],
    "srchIndexesDefault": [],
    "srchJobsQuota": 3,
    "srchTimeEarliest": -1,
    "srchTimeWin": -1
}
```

f. Successfully added ‘syslog’ back in to the list of searchable indexes!

Exercise 7 – Manage Users

Description

ACS can be used to manage users, with the actions create, delete, modify, individually and in bulk. Experiencing and understanding some of these features helps understand how to build the CLI interface.

Steps

1. **On the admin host, create a new user**

    a. Execute the following command on the admin host to create a new user: (multiline in below, be sure to capture the entire line)

```
acs users create --name "soc_user_1" --password "5p1unk.conf" --force-change-pass --roles
user,power
```

b.

```
attendee@ip-172-31-86-240:~$ acs users create --name "soc_user_1"
--password "5p1unk.conf" --force-change-pass --roles "user,power"
{
    "capabilities": [
        "accelerate_search",
        "change_own_password",
        "edit_log_alert_event",
        "edit_messages",
        "edit_own_objects",
        "edit_search_schedule_window",
        "edit_sourcetypes",
        "edit_statsd_transforms",
        "embed_report",
        "export_results_is_visible",
        "get_metadata",
        "get_typeahead",
        "input_file",
        "list_accelerate_search",
        "list_all_objects",
        "list_field_filter",
        "list_inputs",
        "list_metrics_catalog",
        "metric_alerts",
        "output_file",
        "pattern_detect",
        "read_o11y_content",
        "request_remote_tok",
        "rest_access_server_endpoints",
        "rest_apps_view",
        "rest_properties_get",
        "rest_properties_set",
        "run_collect",
        "run_commands_ignoring_field_filter",
        "run_custom_command",
        "run_dump",
        "run_mcollect",
        "run_msearch",
        "run_sendalert",
        "schedule_rtsearch",
        "schedule_search",
        "search",
        "upload_lookup_files",
        "write_o11y_content"
    ],
    "defaultApp": "launcher",
    "defaultAppSource": "system",
    "email": "",
    "fullName": "",
    "lastSuccessfulLogin": "",
    "lockedOut": false,
    "name": "soc_user_1",
    "roles": [
        "power",
        "user"
    ]
}
```

c. This command creates a new user '`soc_user_1`', with the password '`5p1unk.conf`' and the roles "`user,power`". Note that by default, the command does NOT set the user to be required to change their password at their first login. The '--force-change-pass' causes this to be set.

d. Validate via the CLI

```
acs users describe "soc_user_1" | jq '{name: .name, roles: .roles}'
```

e.

```
attendee@ip-172-31-86-240:~$ acs users describe
"soc_user_1" | jq '{name: .name, roles: .roles}'


{
  "name": "soc_user_1",
  "roles": [
    "power",
    "user"
  ]
}
```

f. Using jq, filter the 'describe' output to demonstrate the success.

2. Update the user to add the 'index-syslog' role

a. Execute the command:

```
acs users update soc_user_1 --roles user,power,index-syslog
```

```
attendee@ip-172-31-86-240:~$ acs users update "soc_user_1"
--roles "user,power,index-syslog"
{
    "capabilities": [
        "accelerate_search",
        "change_own_password",
        "edit_log_alert_event",
        "edit_messages",
        "edit_own_objects",
        "edit_search_schedule_window",
        "edit_sourcetypes",
        "edit_statsd_transforms",
        "embed_report",
        "export_results_is_visible",
        "get_metadata",
        "get_typeahead",
        "input_file",
        "list_accelerate_search",
        "list_all_objects",
        "list_field_filter",
        "list_inputs",
        "list_metrics_catalog",
        "metric_alerts",
        "output_file",
        "pattern_detect",
        "read_o11y_content",
        "request_remote_tok",
        "rest_access_server_endpoints",
        "rest_apps_view",
        "rest_properties_get",
        "rest_properties_set",
        "run_collect",
        "run_commands_ignoring_field_filter",
        "run_custom_command",
        "run_dump",
        "run_mcollect",
        "run_msearch",
        "run_sendalert",
        "schedule_rtsearch",
        "schedule_search",
        "search",
        "upload_lookup_files",
        "write_o11y_content"
    ],
    "defaultApp": "launcher",
    "defaultAppSource": "system",
    "email": "",
    "fullName": "",
    "lastSuccessfulLogin": "",
    "lockedOut": false,
    "name": "soc_user_1",
    "roles": [
        "index-syslog"
        "power",
        "user"
    ]
}
```

b.

3. Validate with the CLI

```
acs users describe "soc_user_1" | jq '{name: .name, roles: .roles}'
```

a.

```
attendee@ip-172-31-86-240:~$ acs users describe "soc_user_1"
 | jq '{name: .name, roles: .roles}'
{
  "name": "soc_user_1",
  "roles": [
    "index-syslog",
    "power",
    "user"
  ]
}
```

Exercise 8 – Add App - private

Description

Adding apps are possible via the acs CLI. This is an important tool to help ensure that versioning is tracked appropriately, via whichever tool is used. The process is more than just upload, however, as all apps installed onto a Splunk Cloud stack must be vetted to help prevent issues that may cause stack downtime or cause difficulties that are difficult to resolve.

There are already private apps available on the instance that will be used to demonstrate the process and what the vetting may discover.

Steps

1. On the admin server CLI, install a private app

a. Execute the command below to install a private app:

```
acs apps install private --app-package SA_hywels_dashboards222.tar.gz --acs-legal-ack=Y
```

i. The process requires a username and password at the splunk.com web page, meaning that the user must be credentialed and a user of Splunk. Anyone with a Splunk Cloud instance can receive a Splunk.com account via their Sales team.

ii. The process also requires the acceptance of license terms. For apps that are supported via the Splunk license (such as the dashboards that will be uploaded in this exercise) the default license location is required. For other private apps, the licensure may be Splunk's basic license (this is your stack with your apps) but if the app comes from another third-party who has done work for the stack owner, there may be a different URL. Most will be Splunk.

iii. The vetting process is automated and will provide feedback as to success of failure.

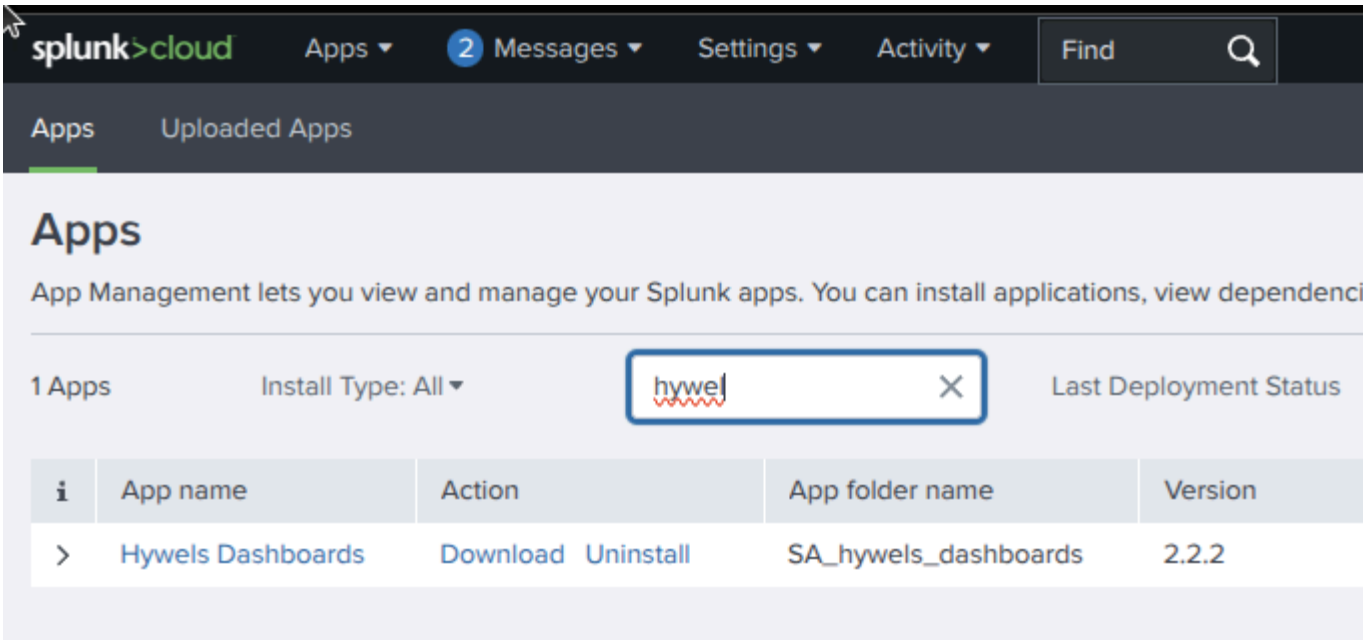
b.

```
attendee@ip-172-31-86-240:~$ acs apps install private --app-package SA_hywels_dashboards222.tar.g
z --acs-legal-ack=Y
Authenticating with the Splunk API service using your splunk.com credentials.
Enter Username: [redacted]
Enter Password:
Successfully authenticated user and assigned a token
Inspecting your private app...
Submitted app for inspection (requestId='92aadfb3-531b-4c01-82d1-7755238a1437')
Waiting for inspection to finish...
processing.....success
Vetting completed, summary:
{
    "error": 0,
    "failure": 0,
    "skipped": 1,
    "manual_check": 0,
    "not_applicable": 117,
    "warning": 1,
    "success": 77
}
Vetting successful
Installing the app...
{
    "label": "Hywels Dashboards",
    "name": "SA_hywels_dashboards",
    "package": "SA_hywels_dashboards-2.2.2.tar.gz",
    "status": "installed",
    "version": "2.2.2"
}
```

c. In this instance, the app successfully passes vetting and is installed

2. Validate the installation of the app via GUI

a. On the GUI, black-bar 'Apps'->'Manage Apps' and search for 'Hywels Dashboards' (that is pronounced "How-ell", by the way).

b. 

3. Experience an App Vetting failure

a. On the admin host, execute the following command: (multiline, be sure to get the complete)

```
acs apps install private --acs-legal-ack=Y  --app-package
SC_Fail_SA_hywels_dashboards222.tar.gz --acs-legal-ack=Y
```

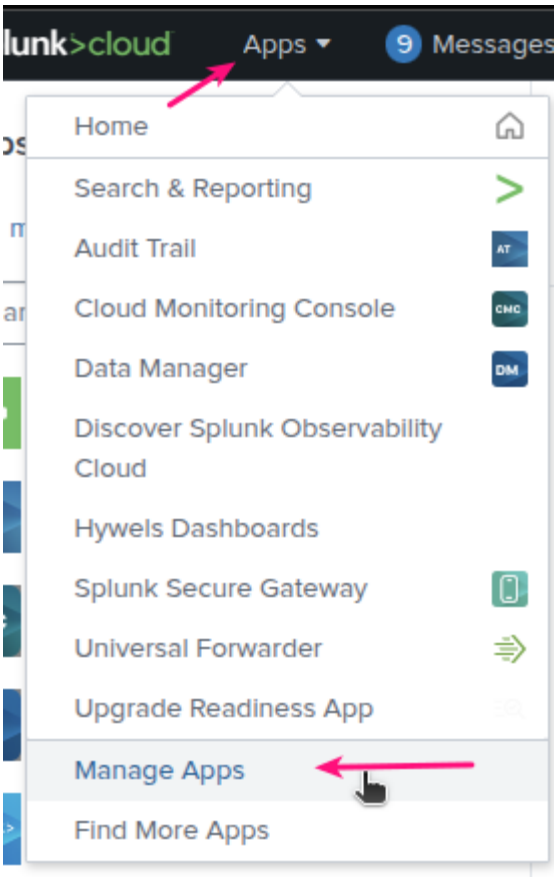
b.

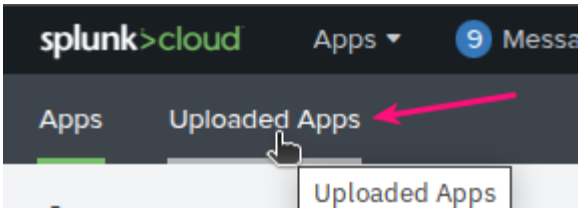
```
attendee@ip-172-31-86-240:~$ acs apps install private --acs-legal-ack=Y  --app-package
SC_Fail_SA_hywels_dashboards222.tar.gz
Authenticating with the Splunk API service using your splunk.com credentials.
Enter Username: [redacted]
Enter Password:
Successfully authenticated user and assigned a token
Inspecting your private app...
Submitted app for inspection (requestId='a437e856-4c36-4c77-8a70-02f26cfdfca7')
Waiting for inspection to finish...
processing......success
Vetting completed, summary:
{
    "error": 0,
    "failure": 0,
    "skipped": 0,
    "manual_check": 0,
    "not_applicable": 117,
    "warning": 2,
    "success": 77
}
Vetting successful
Installing the app...
This app has failed AppInspect validation. Fix the issues in the report, and retry AppI
nspect validation. Please refer https://docs.splunk.com/Documentation/SplunkCloud/lates
t/Config/ACSerrormessages for general troubleshooting tips.
{
    "code": "400-bad-request",
    "message": "This app has failed AppInspect validation. Fix the issues in the report
, and retry AppInspect validation. Please refer https://docs.splunk.com/Documentation/S
plunkCloud/latest/Config/ACSerrormessages for general troubleshooting tips."
}

Error: Install App failed: POST request to "https://staging.admin.splunk.com/s4a-acs-02
/adminconfig/v2/apps" failed, code: 400 Bad Request
```

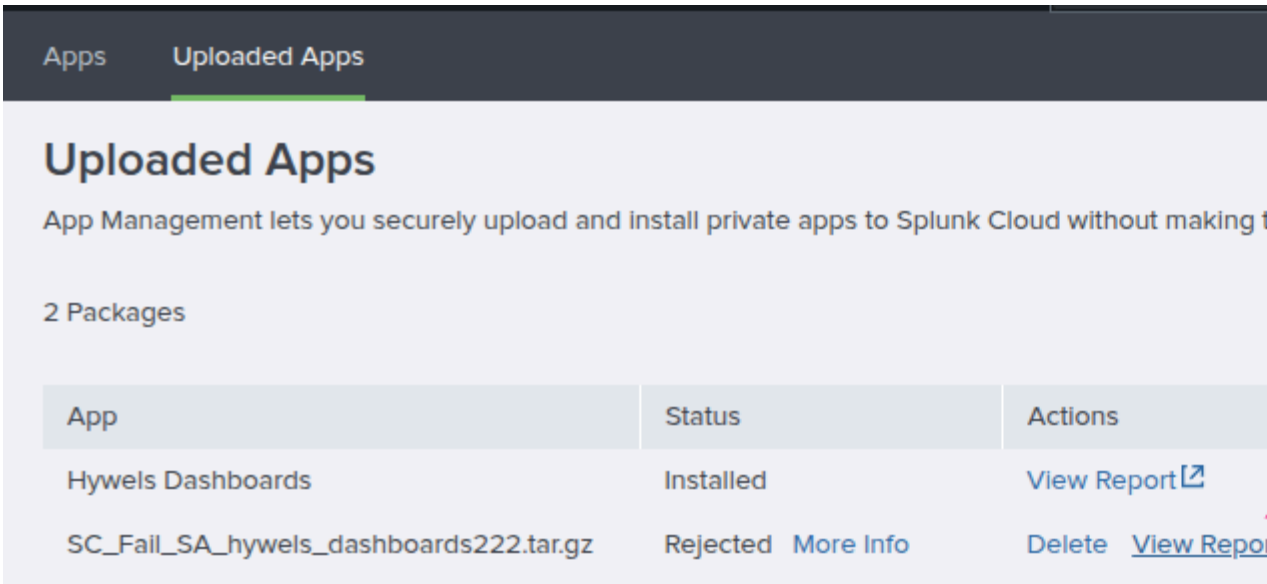
c. Here we see how the process works. Note there is a ‘success’ in this output at ‘process…success’, that just means that the app was sent to the vetting processor successfully. The output code of indicates that the apps failed vetting. The question is now “why?”.

d. To discover why the app failed vet, the GUI is required, so access the lab stack GUI assigned.

   i. Select ‘Black-bar’->’Apps’->’Manage Apps’->Gray-bar ‘Uploaded Apps’ and find the name of the app (here SC_Fail_SA_hywels_dashboards222.tar.gz) and select ‘View Report’

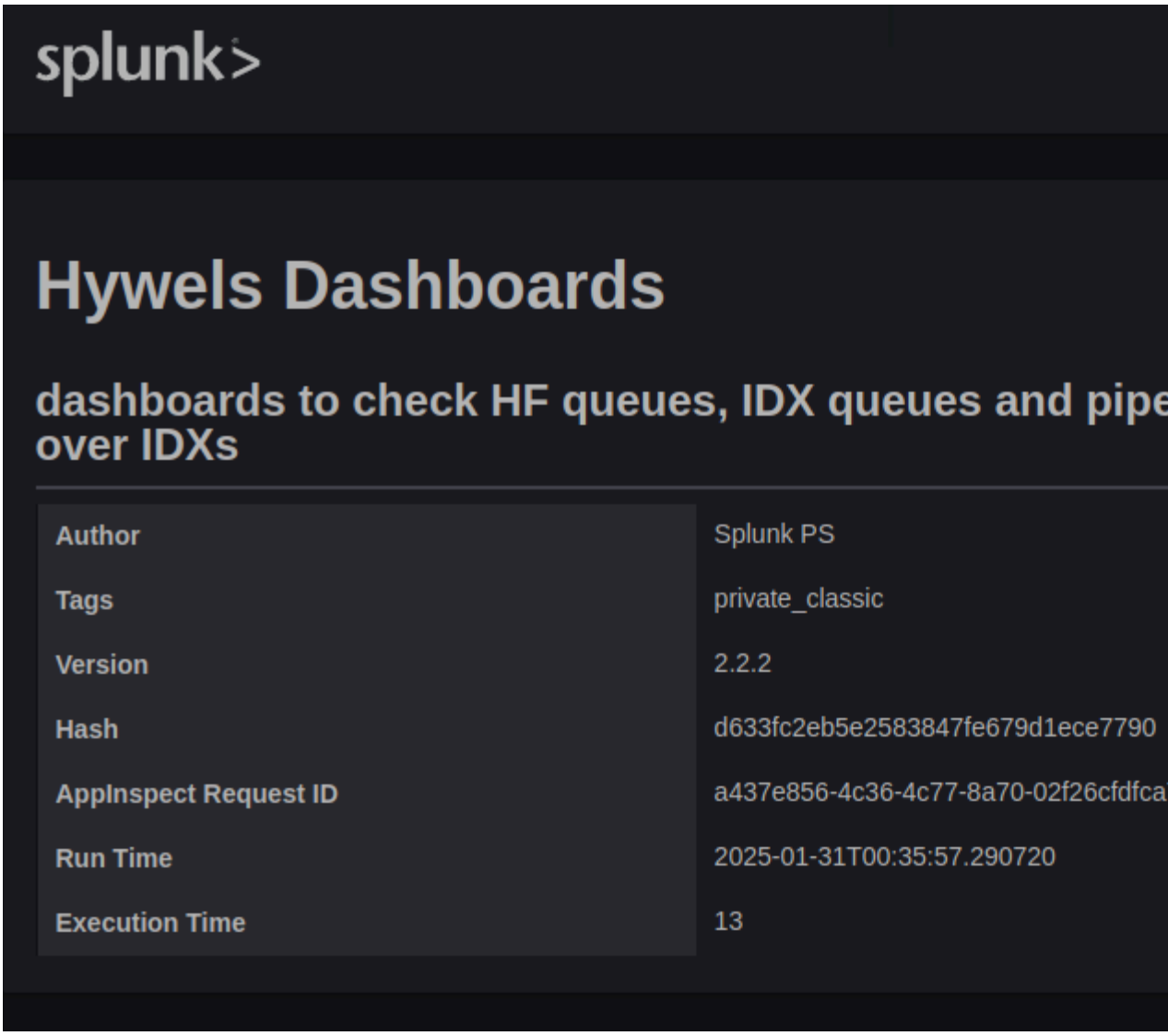




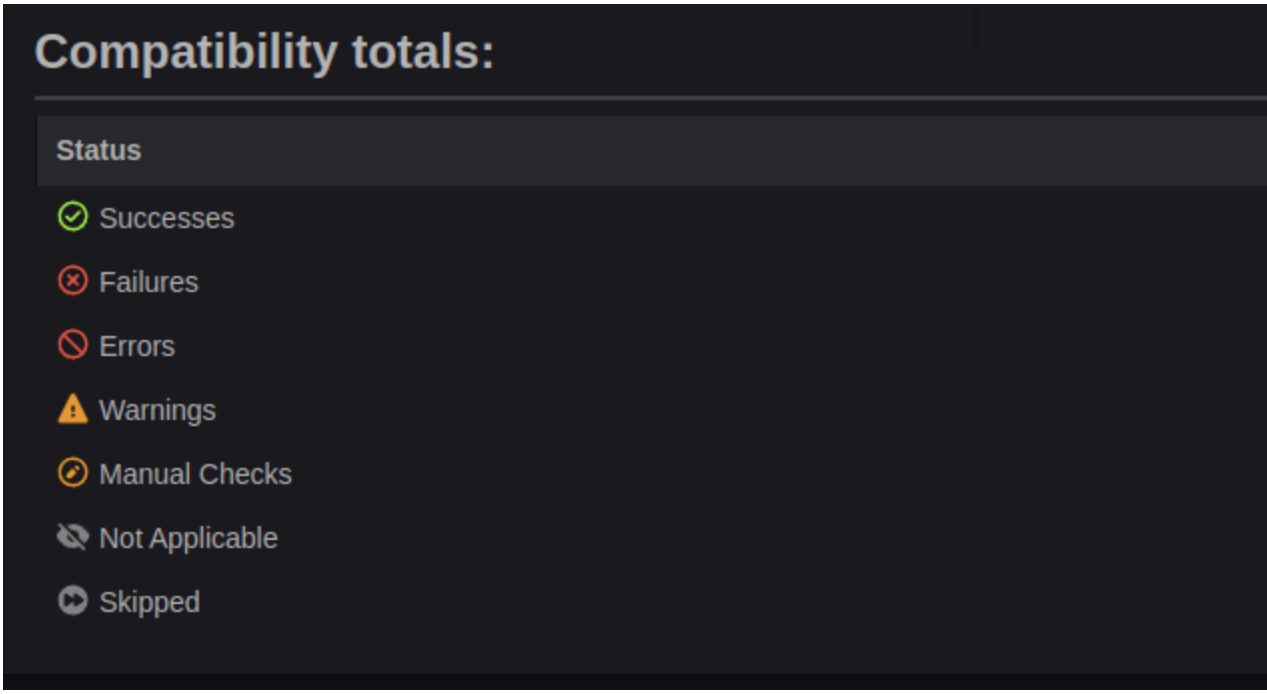


ii. Which will bring the report up of what was discovered to be wrong with the app. Remediating the deliberate mistake is beyond the scope of this lab.

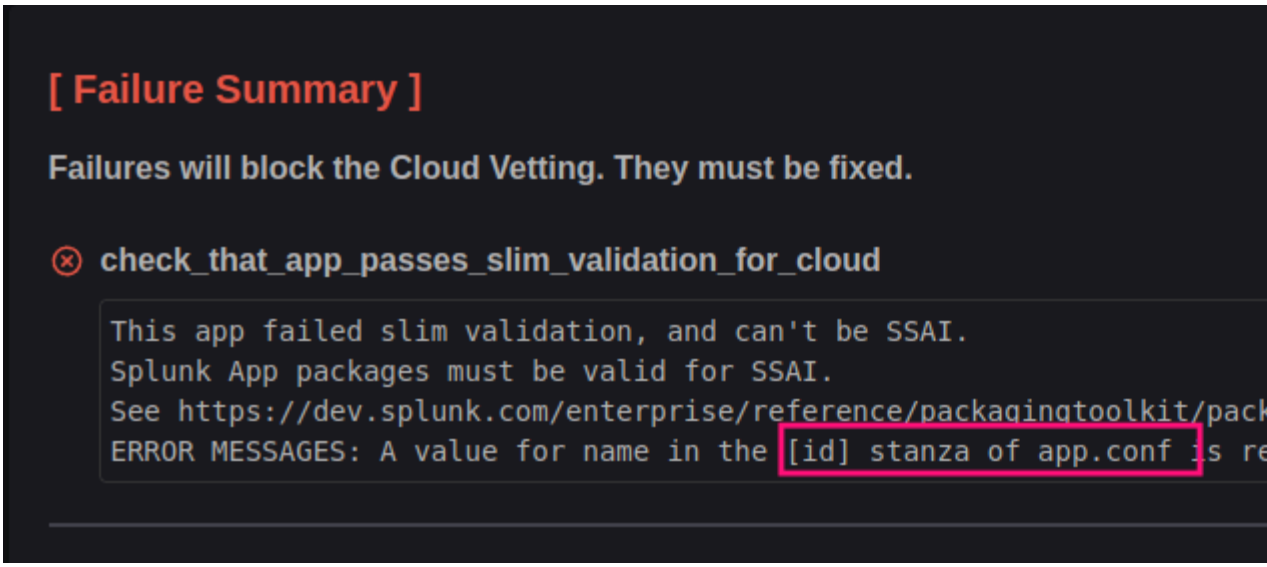
iii. The below screenshots are broken up because the page is quite large.



iv. Top of report, provides metadata about the report



v. Middle of the report, provides a summary of the checks executed.



vi. Farther down in the report, the specifics of the failure. Part of the metadata within the app is not set correctly (a stanza, 'id', in the 'app.conf' file is missing a 'name' value, which is required for Splunk Cloud.

Exercise 9 – Add App - Splunkbase

Description

Additionally, Splunkbase apps can be added via the CLI. Adding a Splunkbase app requires the Splunk admin to know the Splunkbase numerical ID of the app to be installed. Apps on Splunkbase that are listed in the Compatibility square with 'Splunk Cloud' should just install.

Steps

1. On the admin server CLI, install a Splunkbase app

a. Execute the command below to install a Splunkbase app: (we will use the Splunk Add-on for Unix and Linux app, with the number 833)

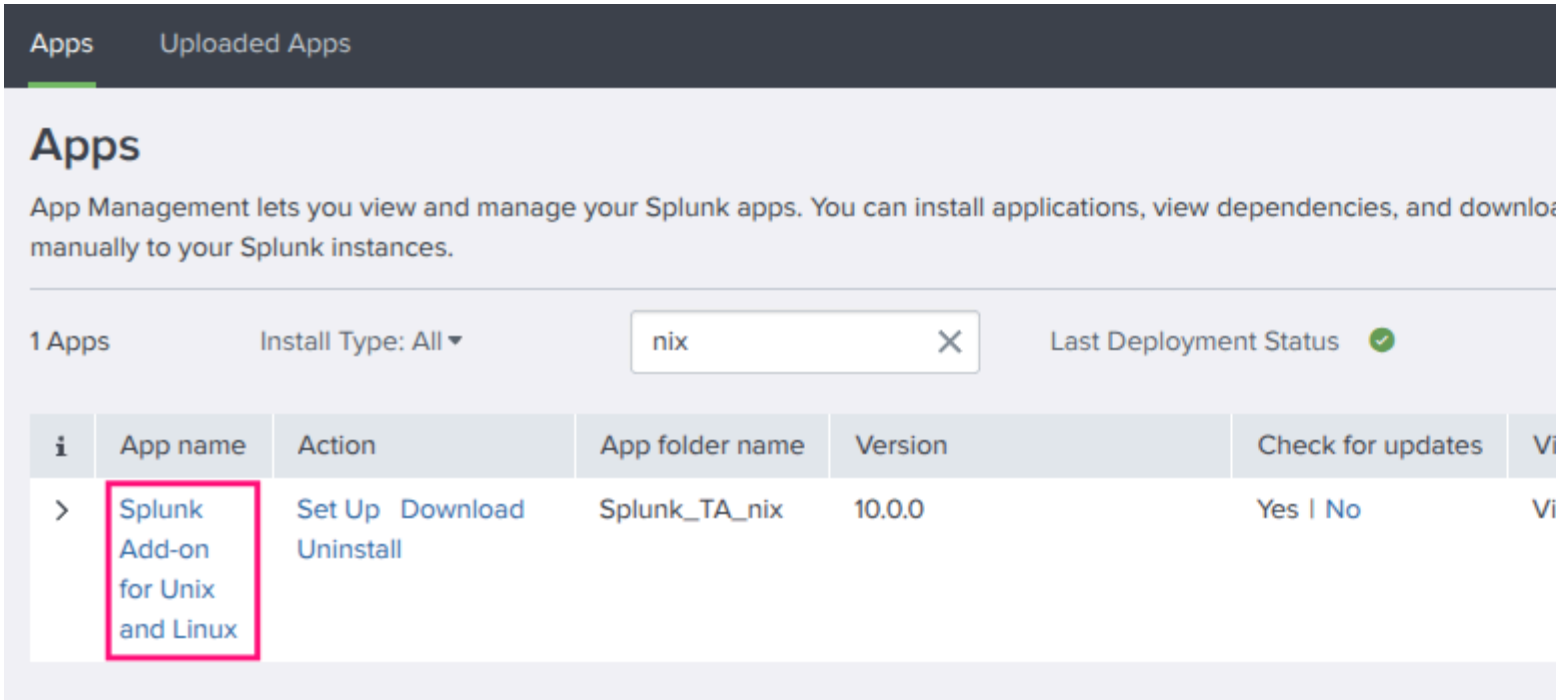
```
acs apps install splunkbase --splunkbase-id 833
```

```
attendee@ip-172-31-86-240:~$ acs apps install splunkbase --splunkbase-id 833
Authenticating with the Splunk API service using your splunk.com credentials.
Enter Username: [redacted]
Enter Password:
Retrieving license for Splunkbase app: 833

URL of the third-party license for the app to acknowledge your acceptance of any risks:
https://www.splunk.com/en_us/legal/splunk-general-terms.html
Enter 'Y' to accept: Y

Installing the app...
App installation in progress (8e2d8fa1-eb0f-922a-a885-7abc1a431962)....
App install succeeded
{
    "appID": "Splunk_TA_nix",
    "label": "Splunk Add-on for Unix and Linux",
    "name": "Splunk_TA_nix",
    "package": "",
    "splunkbaseID": "833",
    "stateChangeRequiresRestart": false,
    "status": "installed",
    "version": "9.2.0"
}
```

b. Similarly to the private app add, a login to Splunk.com is required. As with a GUI download from Splunkbase itself, this is required to access the app repository. Acceptance of the license is required, here the example is interactive, but the acceptance can be executed via a command line switch.

2. Validate the installation of Splunk Add-on for Unix and Linux with the GUI:

a. Select ‘Black-bar’->’Apps’->’Manage Apps’
b. Find the app in the page



c. Note the ‘App name’ and the ‘App Origin’

Links

	Link
Service (ACS) API	https://docs.splunk.com/Documentation/SplunkCloud/latest/Config/ACSIntro
	https://restfulapi.net/
CS) API endpoint reference	https://docs.splunk.com/Documentation/SplunkCloud/latest/Config/ACSREF
extensions to interface with your SAML	https://docs.splunk.com/Documentation/Splunk/latest/Security/ConfigureauthextensionsforSAMLtokens
s with capabilities	https://docs.splunk.com/Documentation/SplunkCloud/latest/Config/RBAC#Manage_ACS_endpoint_acc es
Service (ACS) API	https://docs.splunk.com/Documentation/SplunkCloud/latest/Config/ACSIntro
	https://restfulapi.net/
CS) API endpoint reference	https://docs.splunk.com/Documentation/SplunkCloud/latest/Config/ACSREF
oncepts for the Admin Config Service	https://docs.splunk.com/Documentation/SplunkCloud/latest/Config/ACSusage
k ACS Github Action CI/CD Starter	https://lantern.splunk.com/Splunk_Platform/Product_Tips/Administration/Introduction_to_the_Splunk_A CI%2F%2FCD_Starter
Cloud Stacks Easily and Efficiently with	Introduction to the Splunk ACS Github Action CI/CD Starter
ployment with Splunk Cloud’s Admin CS)	https://community.splunk.com/t5/Splunk-Tech-Talks/Start-Managing-Your-Deployment-with-Splunk-Clou 981
the Splunk Cloud Platform Admin CS) API	https://lantern.splunk.com/Splunk_Platform/Product_Tips/Administration/Creating_allows_lists_with_the atform_Admin_Configuration_Service_API
vice (ACS) in Splunk Cloud Platform	https://lantern.splunk.com/Splunk_Platform/Product_Tips/Administration/Using_Admin_Config_Service_ Cloud_Platform_FedRAMP_environments#
ging Splunk Cloud via ACS	https://docs.splunk.com/Documentation/SplunkCloud/latest/Config/ACSCLI#:~:text=and%20password% login%20credentials,-Some%20ACS%20CLI
kens (Splunk Cloud)	https://docs.splunk.com/Documentation/SplunkCloud/latest/Security/CreateAuthTokens
l Platform using the ACS CLI	https://docs.splunk.com/Documentation/SplunkCloud/latest/Config/ACSCLI
	https://docs.brew.sh/Homebrew-on-Linux
ux (and MacOS)	https://brew.sh/
machine	https://docs.splunk.com/Documentation/SplunkCloud/latest/Config/ACSCLI#:~:text=Install%20or%20up ac%20or%20Linux
ge a Splunk Cloud stack	https://docs.splunk.com/Documentation/SplunkCloud/latest/Config/ACSCLI#:~:text=server%20override %20the%20ACS%20CLI,-Before%20you%20can
on the command line	https://docs.splunk.com/Documentation/SplunkCloud/latest/Config/ACSCLI#Run_ACS_CLI_operations
S	https://docs.splunk.com/Documentation/SplunkCloud/latest/Config/ACSCLI#Manage_indexes
a ACS	https://docs.splunk.com/Documentation/SplunkCloud/latest/Config/ACSCLI#Configure_IP_allow_lists

ng ACS	https://docs.splunk.com/Documentation/SplunkCloud/latest/Config/ACSCLI#Manage_HTTP_Event_Col okens
ulk using ACS	https://docs.splunk.com/Documentation/SplunkCloud/latest/Config/ACSCLI#Create_HEC_tokens_in_bu
ine with jq	https://shapeshed.com/jq-json/
d capabilities in Splunk Cloud Platform	https://docs.splunk.com/Documentation/SplunkCloud/latest/Config/ManageRoles
ng ACS	https://docs.splunk.com/Documentation/SplunkCloud/latest/Config/ACSCLI#Manage_private_apps
Splunk Cloud Platform with ACS	https://docs.splunk.com/Documentation/SplunkCloud/latest/Config/ManageApps
your Splunk Cloud Platform deployment	https://docs.splunk.com/Documentation/SplunkCloud/latest/Admin/PrivateApps
e app using ACS	https://docs.splunk.com/Documentation/SplunkCloud/latest/Config/ACSCLI#Manage_Splunkbase_apps
e app using ACS API	https://docs.splunk.com/Documentation/SplunkCloud/latest/Config/ManageSplunkbaseApps
Unix and Linux	https://splunkbase.splunk.com/app/833
re Shell)?	https://www.ssh.com/academy/ssh
atest release	https://putty.org/