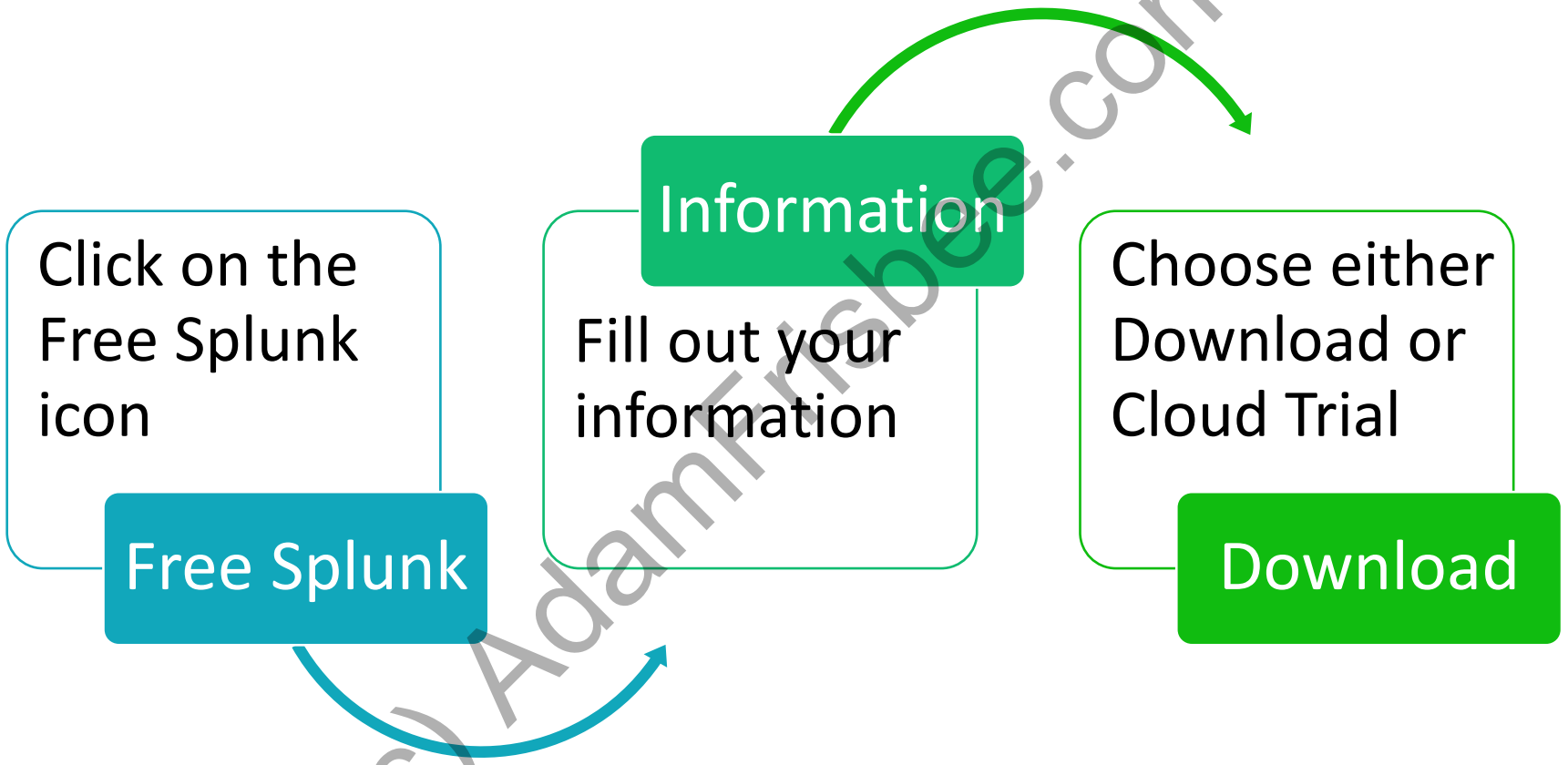
Splunk Admin Basics

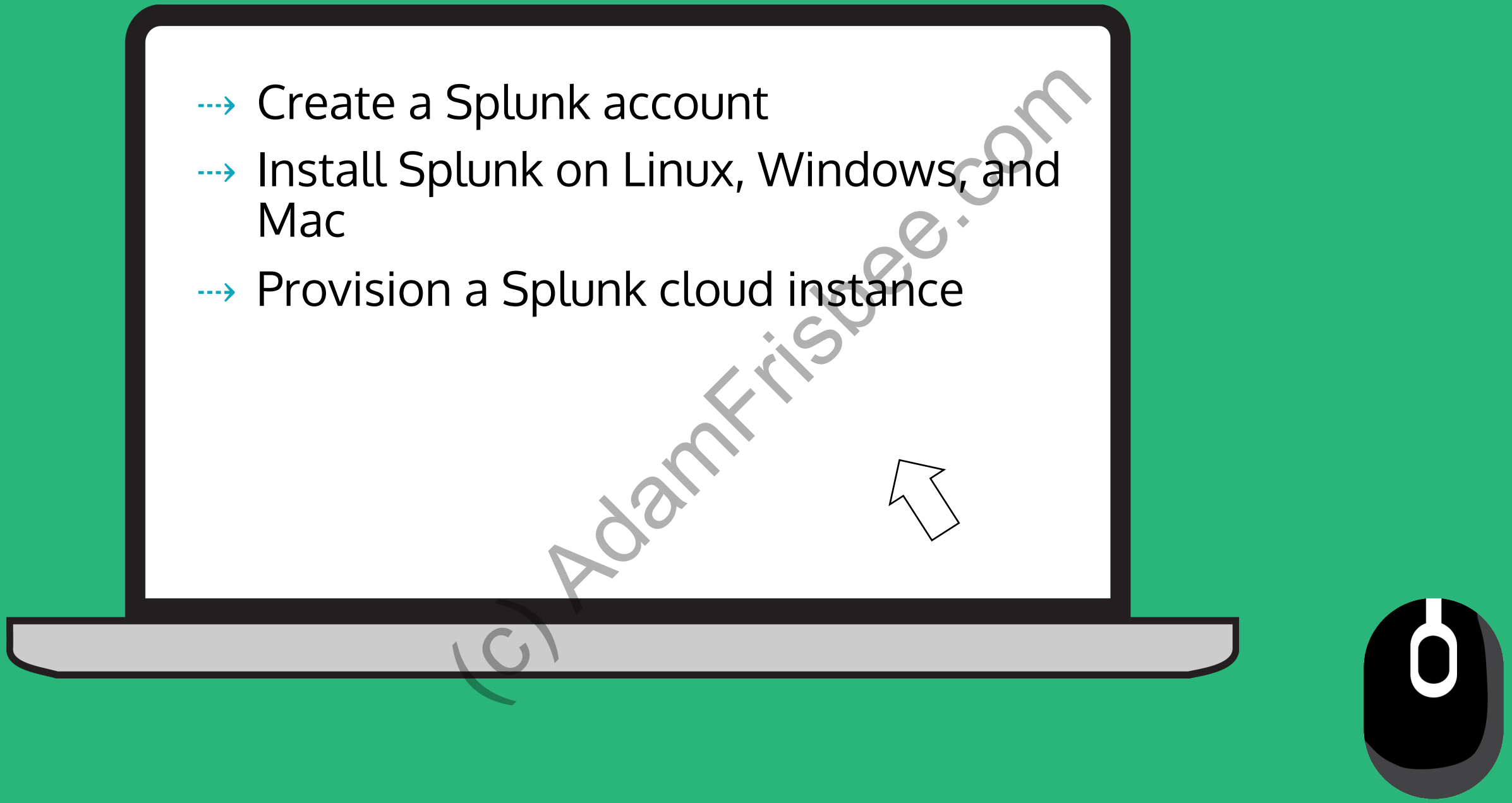
- Creating a Splunk account
- Installing Splunk
- Identifying Splunk components

Creating a Splunk Account and Installing Splunk



Installation Options

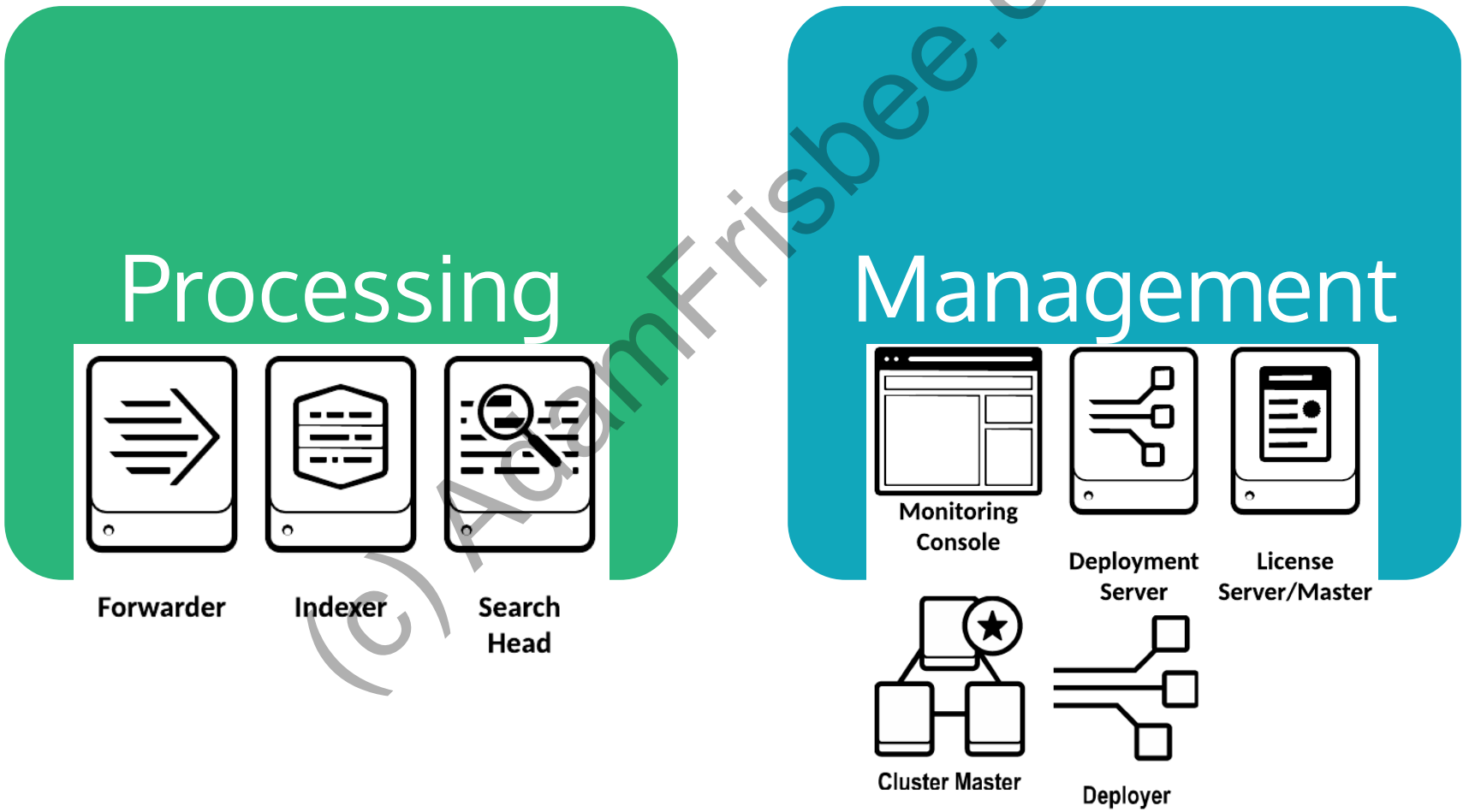
- Download
  - Windows, Linux, or Mac
- Cloud
  - Self service
  - Managed



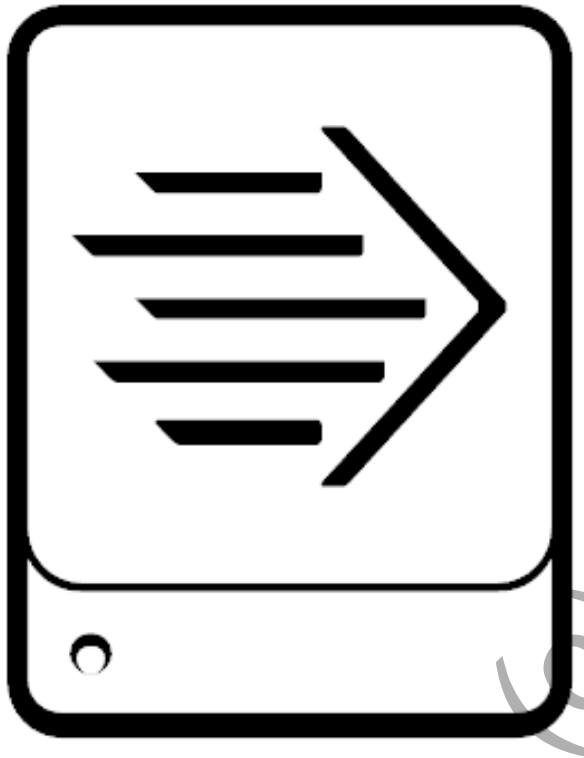
Identifying Splunk Components

Identifying Splunk Components

- Splunk only does three things
  1. Ingests data
  2. Parses, indexes, and stores data
  3. Runs searches on indexed data

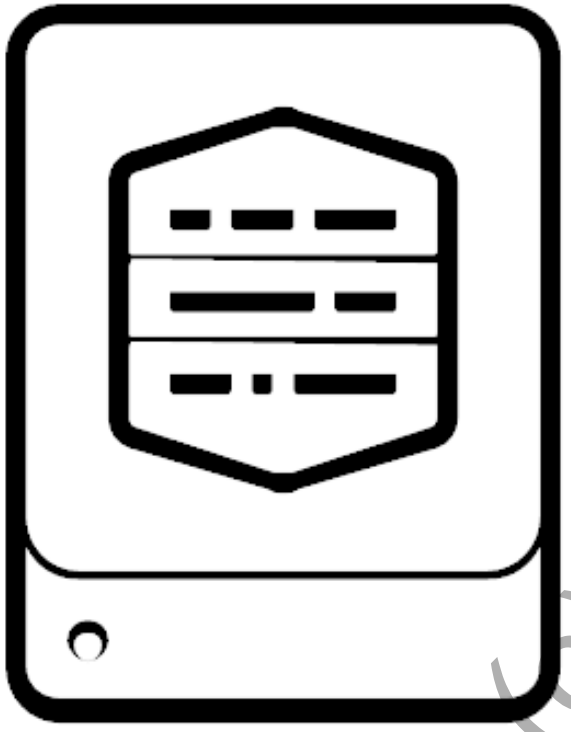


Forwarder



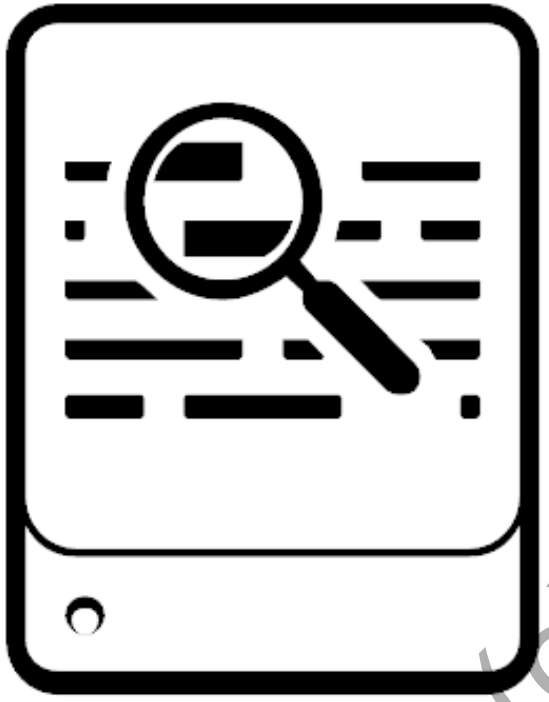
- Forwarders forward data from one Splunk component to another
  - From a source system to an indexer or indexer cluster
  - From a source system directly to a search head

Indexer



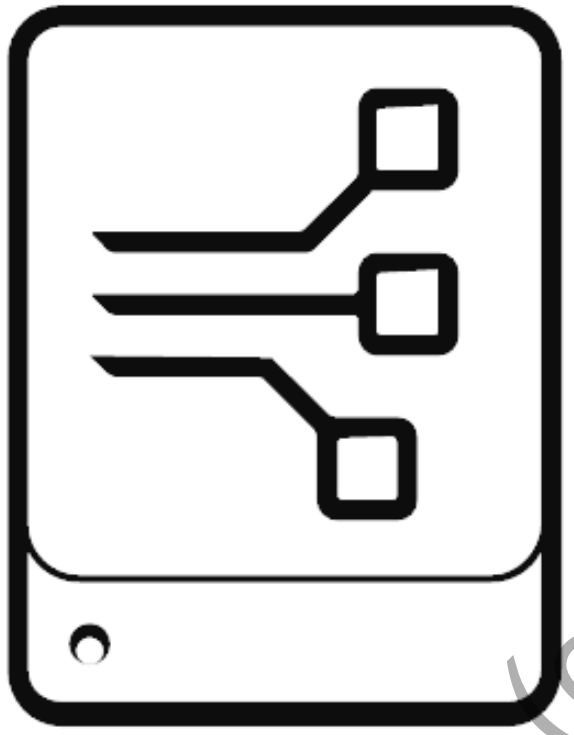
- Indexers index and store data
- In a distributed environment
  - Reside on dedicated machines
  - Can be clustered or independent
  - Clustered indexers are known as peer nodes

Search Head



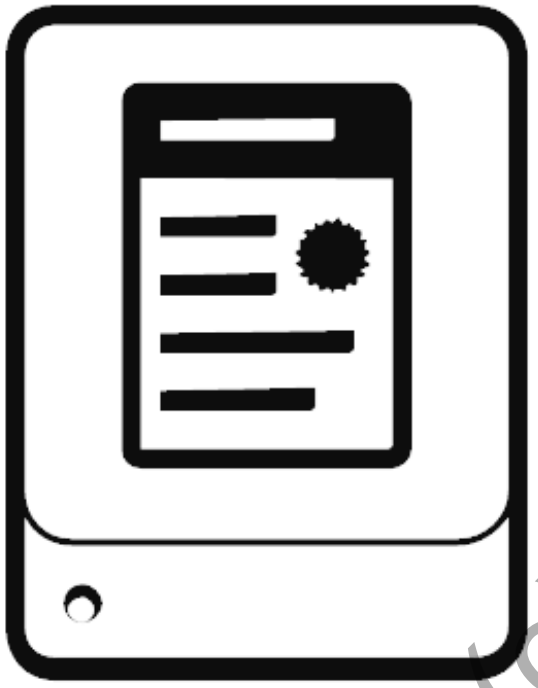
- Search heads manage search requests from users
  - Distributes searches across indexers
  - Consolidates the results from the indexers

Deployment Server



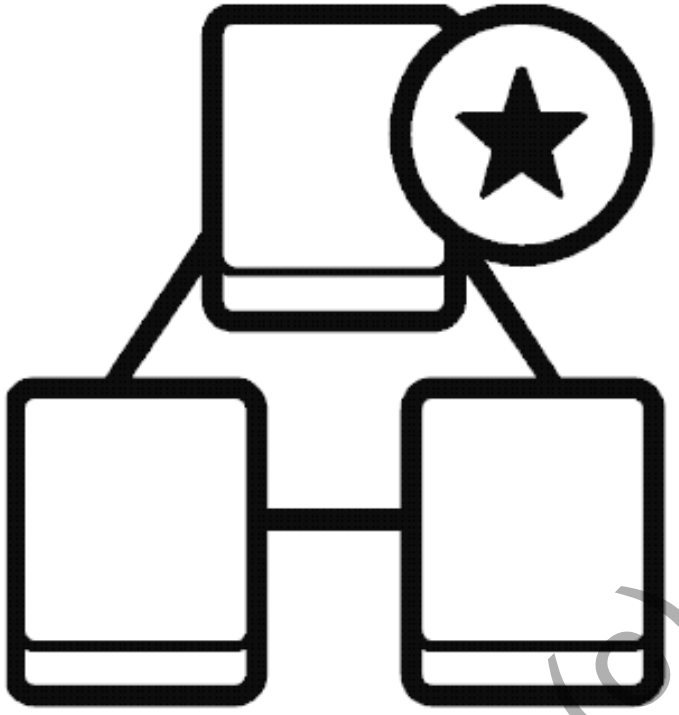
- Centralized configuration manager
- Manages deployment apps for clients
- Configured through the forwarder management interface

License Master



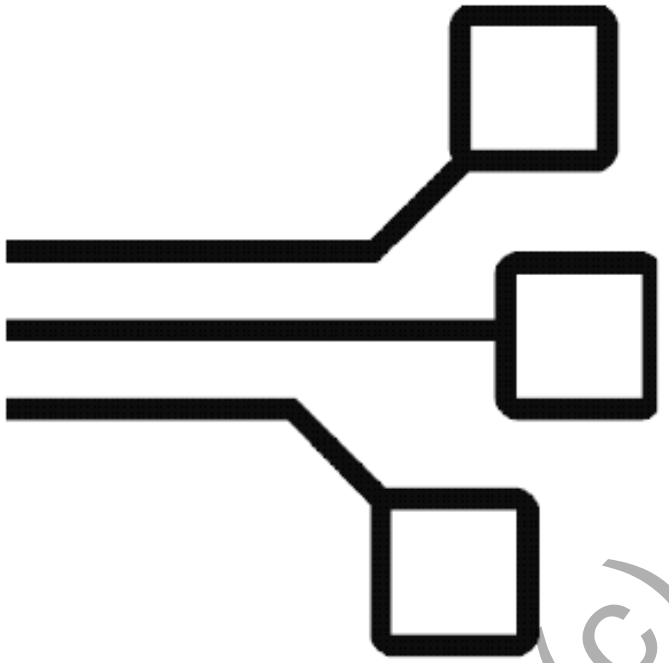
- Centralized license manager
- Clients are called license slaves
- Manages license pools and stacks

Indexer Cluster Master



- Manages indexer clusters
  - Coordinates the activities within the cluster
  - Manages data replication
  - Manages buckets (storage) for the cluster
  - Handles updates for the indexer cluster

Search Head Cluster Deployer



- Manages baselines and apps for search head cluster members
  - This is how Splunk scales
  - Not a member of the cluster

Every Splunk component is built using Splunk Enterprise. It's only a matter of configuration!

Except the Universal Forwarder, which is a specialized "light" Splunk Enterprise installation

Summary

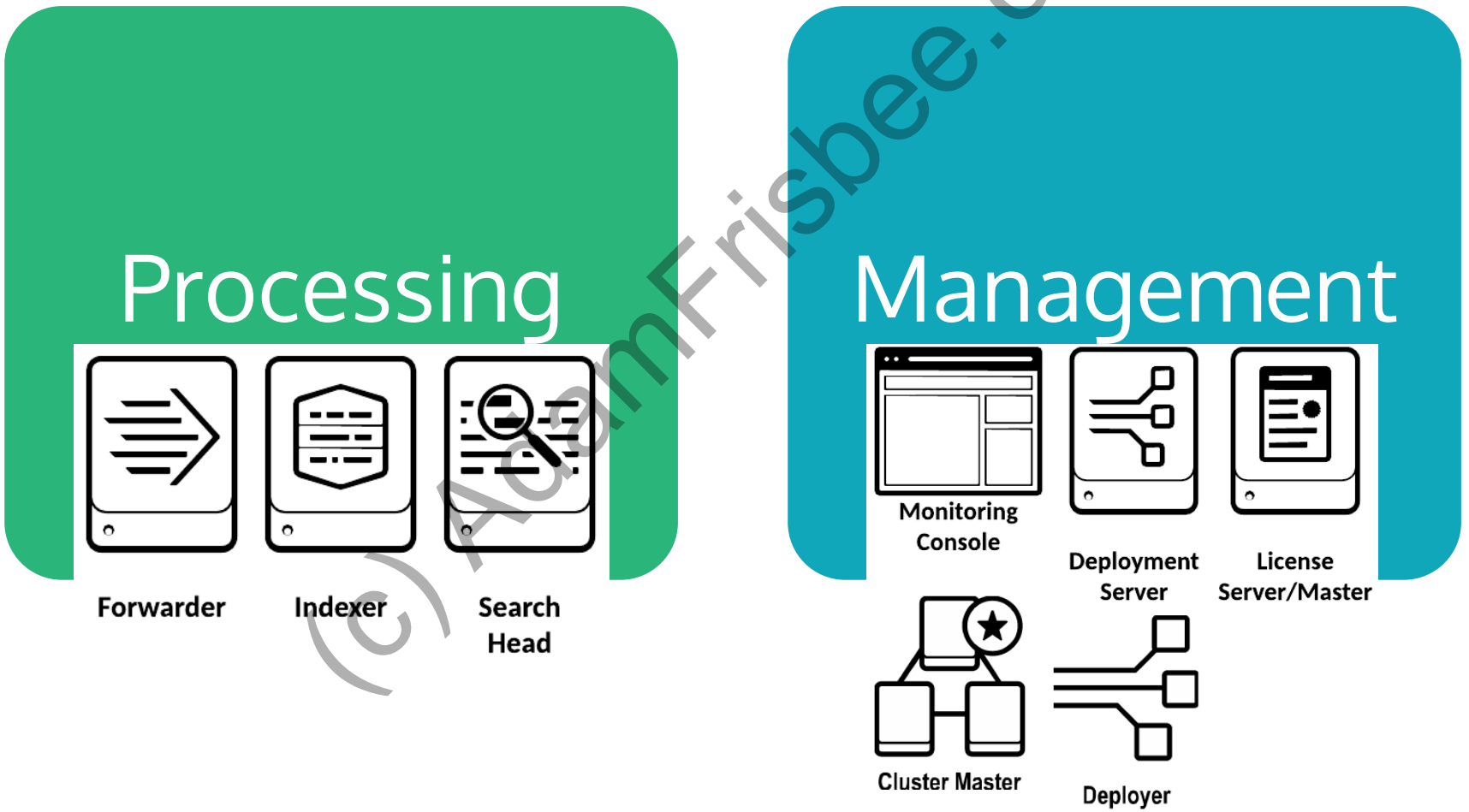
- Identified Splunk components
- Remember: every Splunk component is just an installation of Splunk Enterprise

Identifying Splunk Components

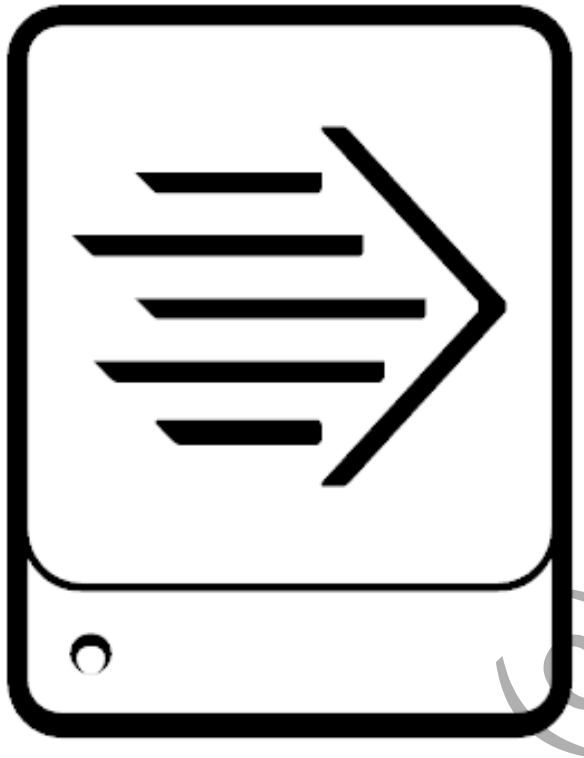
- 1.1 Identifying Splunk components

Identifying Splunk Components

- Splunk only does three things
  1. Ingests data
  2. Parses, indexes, and stores data
  3. Runs searches on indexed data

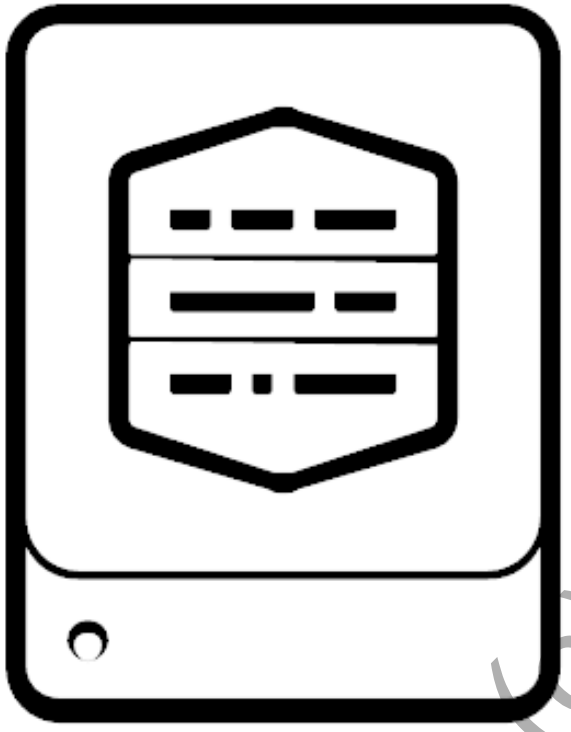


Forwarder



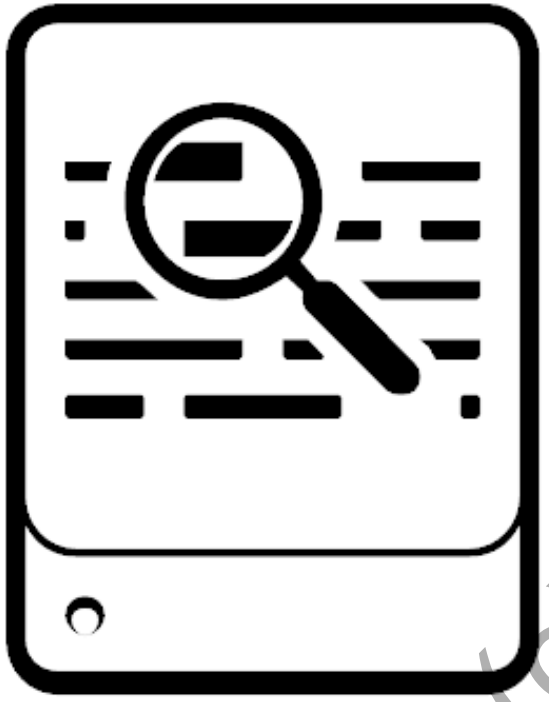
- Forwarders forward data from one Splunk component to another
  - From a source system to an indexer or indexer cluster
  - From a source system directly to a search head

Indexer



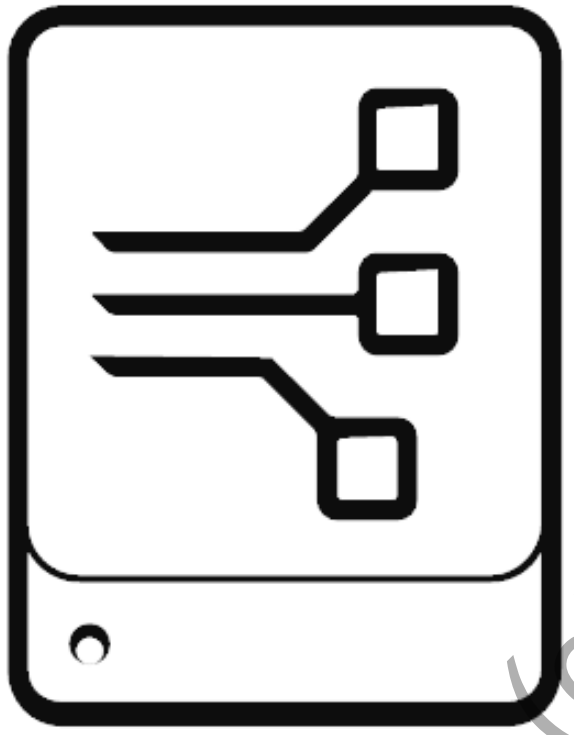
- Indexers index and store data
- In a distributed environment
  - Reside on dedicated machines
  - Can be clustered or independent
  - Clustered indexers are known as peer nodes

Search Head



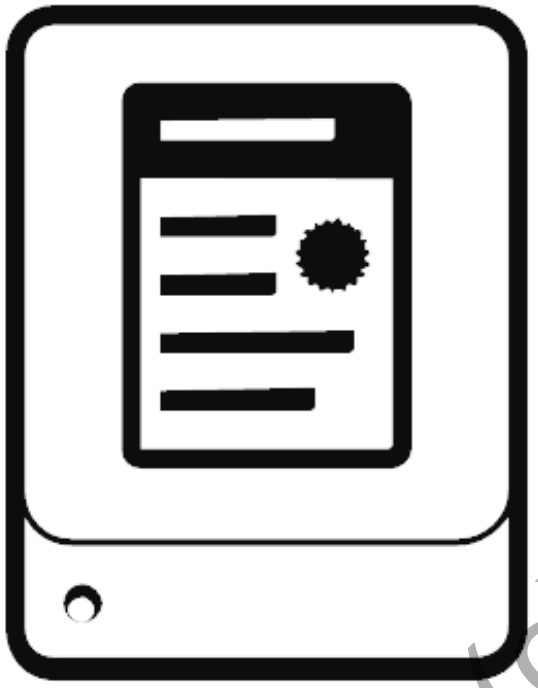
- Search heads manage search requests from users
  - Distributes searches across indexers
  - Consolidates the results from the indexers

Deployment Server



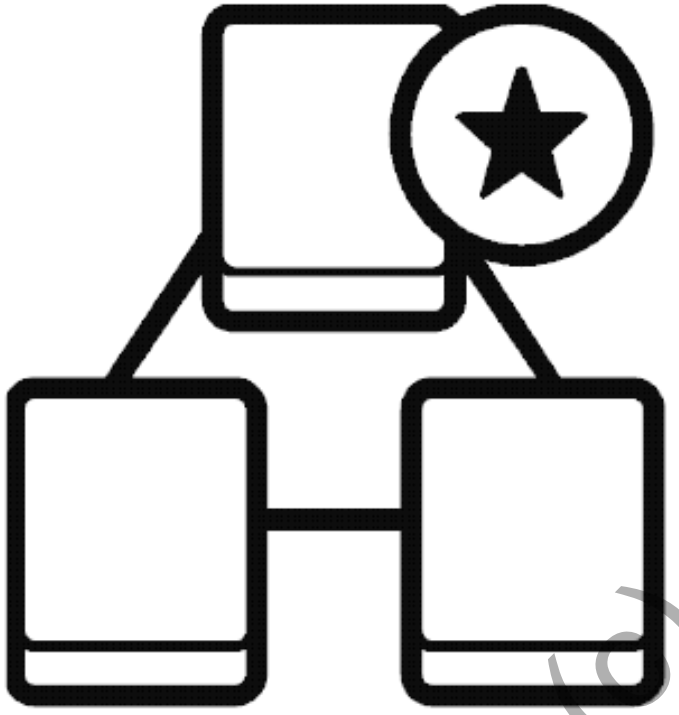
- Centralized configuration manager
- Manages deployment apps for clients
- Configured through the forwarder management interface

License Master



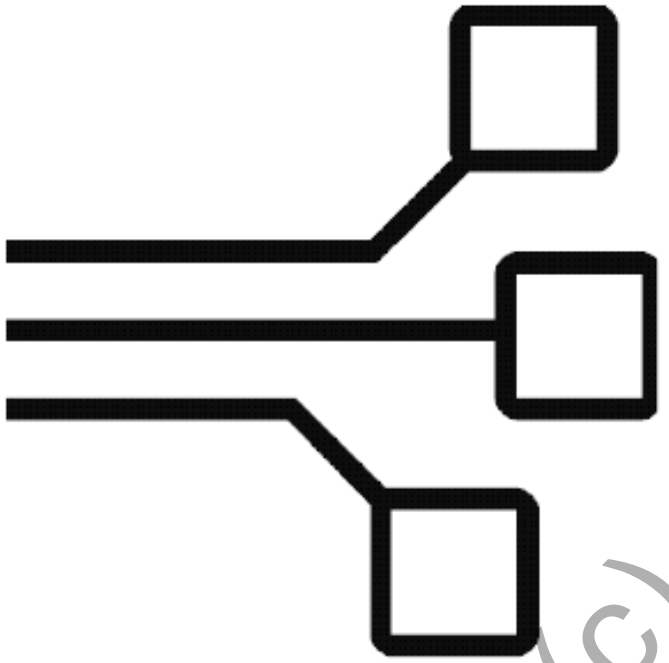
- Centralized license manager
- Clients are called license slaves
- Manages license pools and stacks

Indexer Cluster Master



- Manages indexer clusters
  - Coordinates the activities within the cluster
  - Manages data replication
  - Manages buckets (storage) for the cluster
  - Handles updates for the indexer cluster

Search Head Cluster Deployer



- Manages baselines and apps for search head cluster members
  - This is how Splunk scales
  - Not a member of the cluster

Every Splunk component is built using Splunk Enterprise. It's only a matter of configuration!

Except the Universal Forwarder, which is a specialized "light" Splunk Enterprise installation

Summary

- Identified Splunk components
- Remember: every Splunk component is just an installation of Splunk Enterprise

License Management

- Identifying license types
- Understanding license violations
- Distributed licensing

License Types

Splunk Licensing

- You license data ingested per day, not data stored
- Daily indexing volume is measured from midnight to midnight by the clock on the license master

License Types

Standard

Enterprise Trial

Sales Trial

Dev/Test

Free

Industrial IoT

Forwarder

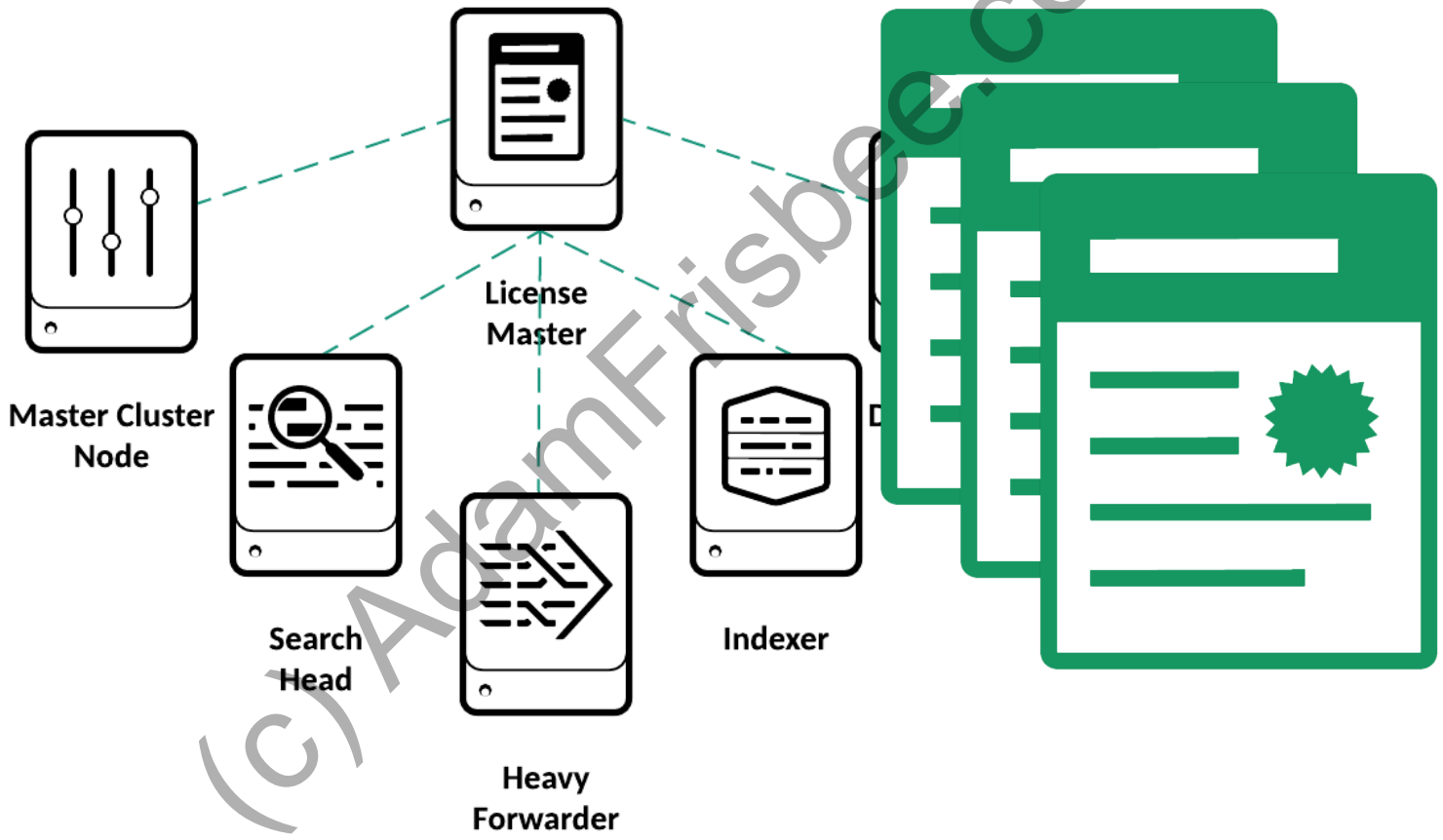
License Violations

License Violations

Starting with version 6.5, Splunk Enterprise no longer disables search when you exceed your licenses data ingestion quota.

Distributed Licensing

Distributed Licensing

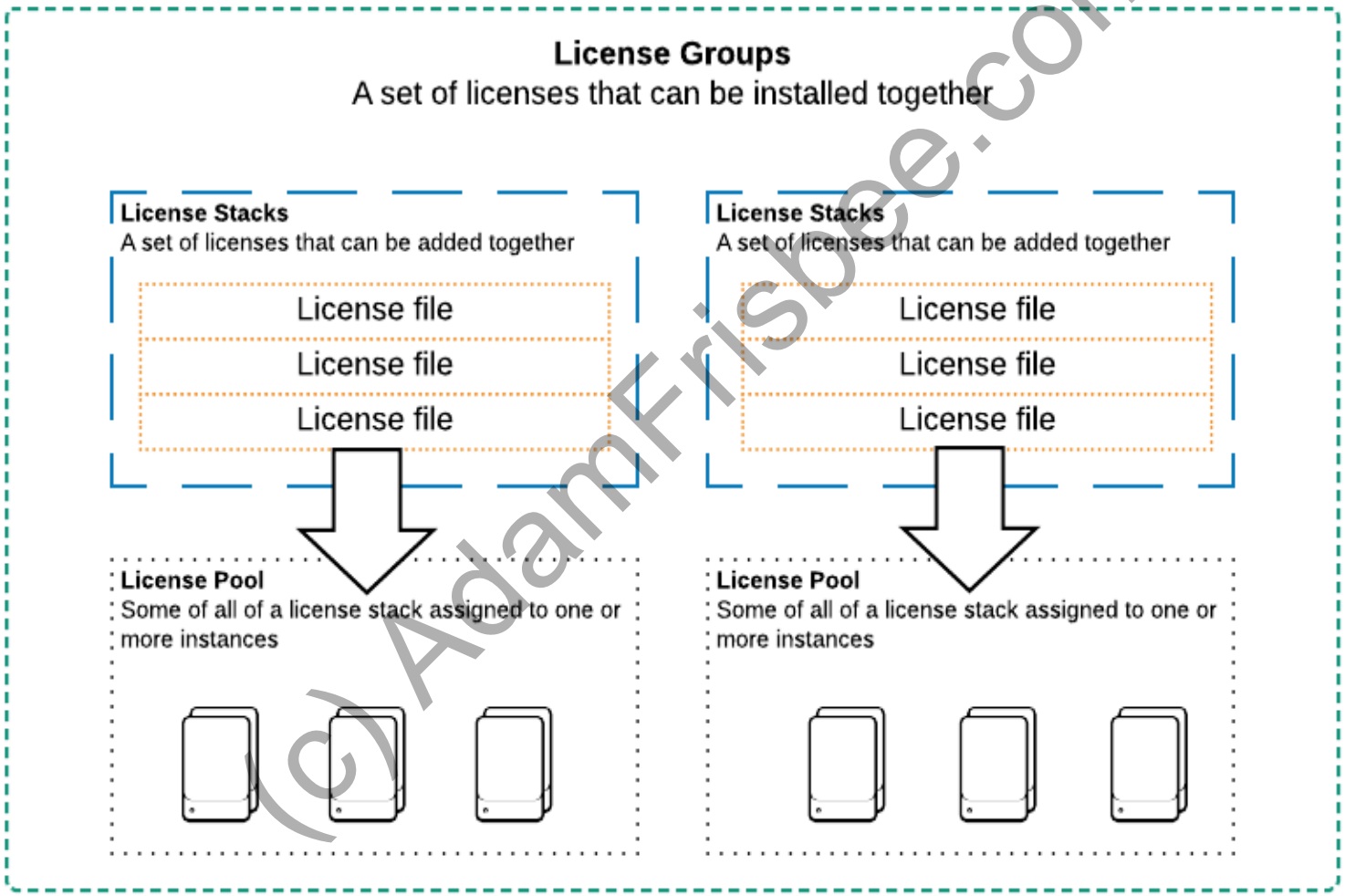


Distributed Licensing

- License pools are created from license stacks
- Pools are sized for specific purposes
- Managed by the license master
- Indexers and other Splunk Enterprise instances are assigned to a pool

Pool

Distributed Licensing



- Explore the licensing console
  - License groups
    - Forwarder license
  - Adding a license
  - Creating a license master or slave

Summary

- Learned how Splunk licensing works
  - License types
  - Violations and warnings
  - Distributed licensing
  - Pools, stacks, and groups

Configuration Files

- Describe Splunk configuration directory structure
- Understand configuration layering
- Understand configuration precedence
- Use `btool` to examine configuration settings

Splunk Configuration Directory Structure

Configuration Files

- Splunk runs on configuration (.conf) files
  - Every behavior and function within Splunk is defined in a .conf file
- Multiple copies of the same configuration file
  - Evaluated by Splunk based on precedence

Common Configuration Files

`inputs.conf`

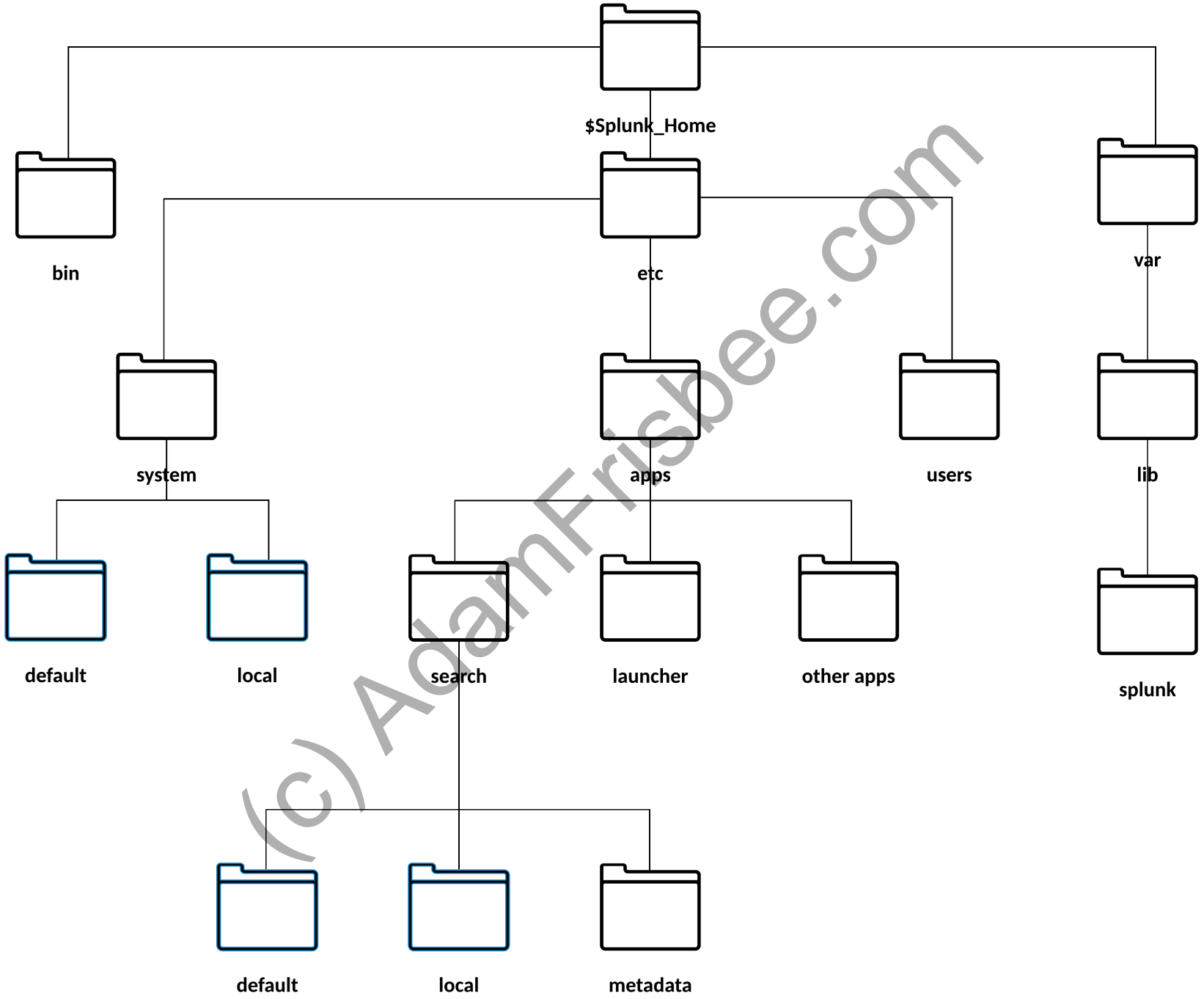
- Governs data inputs such as forwarders and file system monitoring

`props.conf`

- Governs indexing property behavior

`transforms.conf`

- Settings and values that govern data transformation



Configuration Layering and Precedence

Configuration file context

Global

App or user specific

Global context

System **local** directory

⇢App **local** directories

⇢App **default** directories

⇢System **default** directory

App or User Context

User directories for **current user**

- App directories for **currently running app** (local, followed by default)
  - App directories for **all other apps** (local, followed by default)
    - **System** directories (local, followed by default)

What's Inside?

- Stanzas
- Attribute = value pairs

app.conf

```
[id]
group = <group-name>
name = <app-name>
version = <version-number>
```

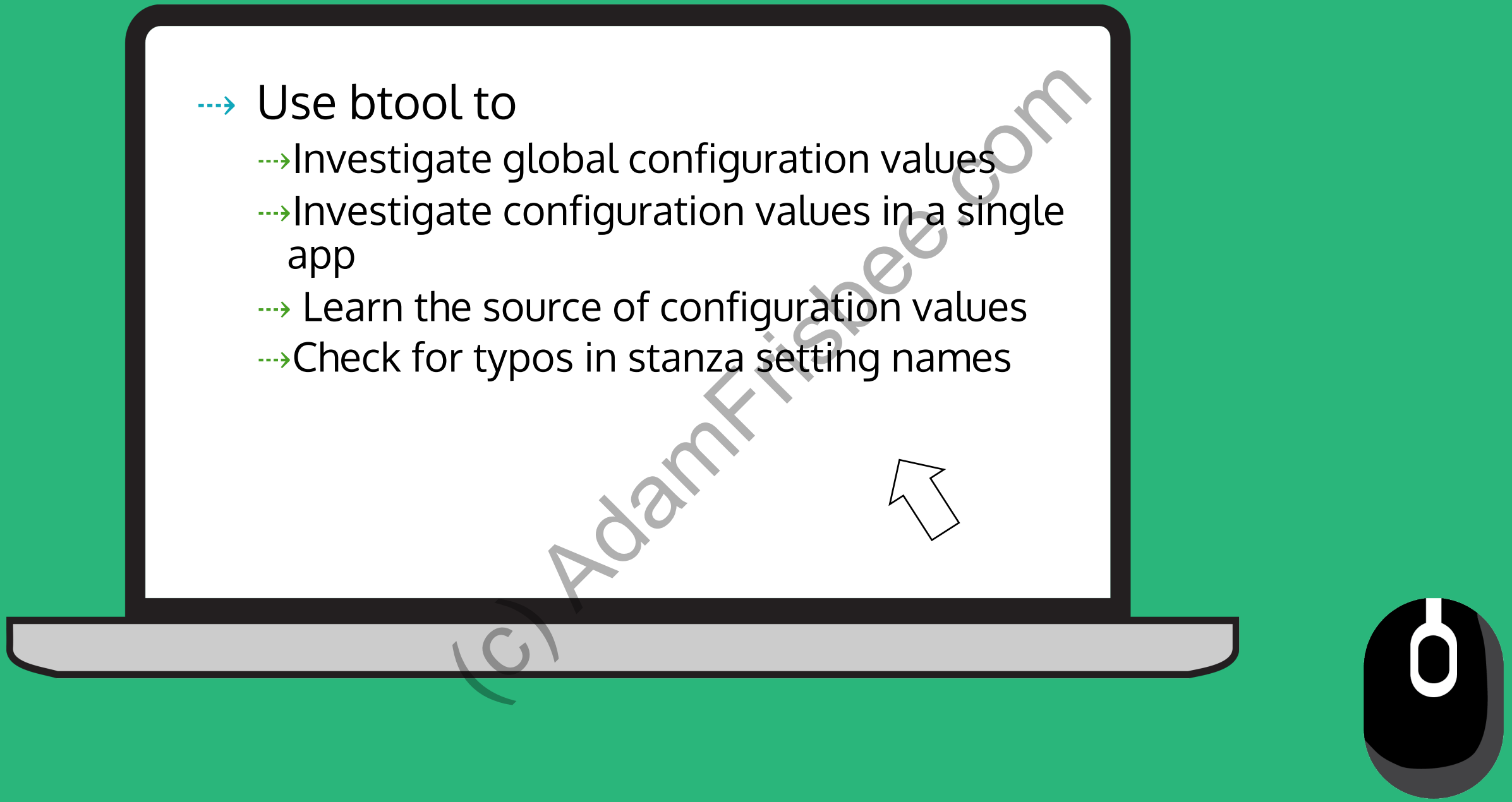
Use `btool` to Examine Configuration Settings

Btool

Troubleshoot

```
$Splunk_Home/bin
./splunk cmd btool <configuration file prefix> list
```

Merged configurations


Use btool to
Investigate global configuration values
Investigate configuration values in a single app
Learn the source of configuration values
Check for typos in stanza setting names
(c) AdamFrisbee.com

Summary

- Learned about configuration files
  - Govern almost every aspect in Splunk
  - What you do in the GUI edits the .conf files
- Discussed conf file precedence
  - Remember, don't touch the default folder no matter what context
- Use btool to verify conf file stanza values that are being used

Indexes

- Describing index structure
- Listing types of index buckets
- Checking data integrity
- Describing `indexes.conf` options
- Describing the fishbucket
- Applying data retention policy

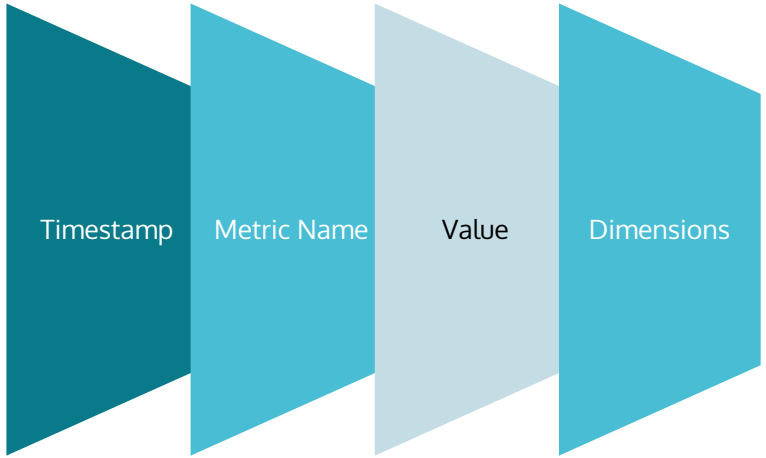
Describe Index Structure

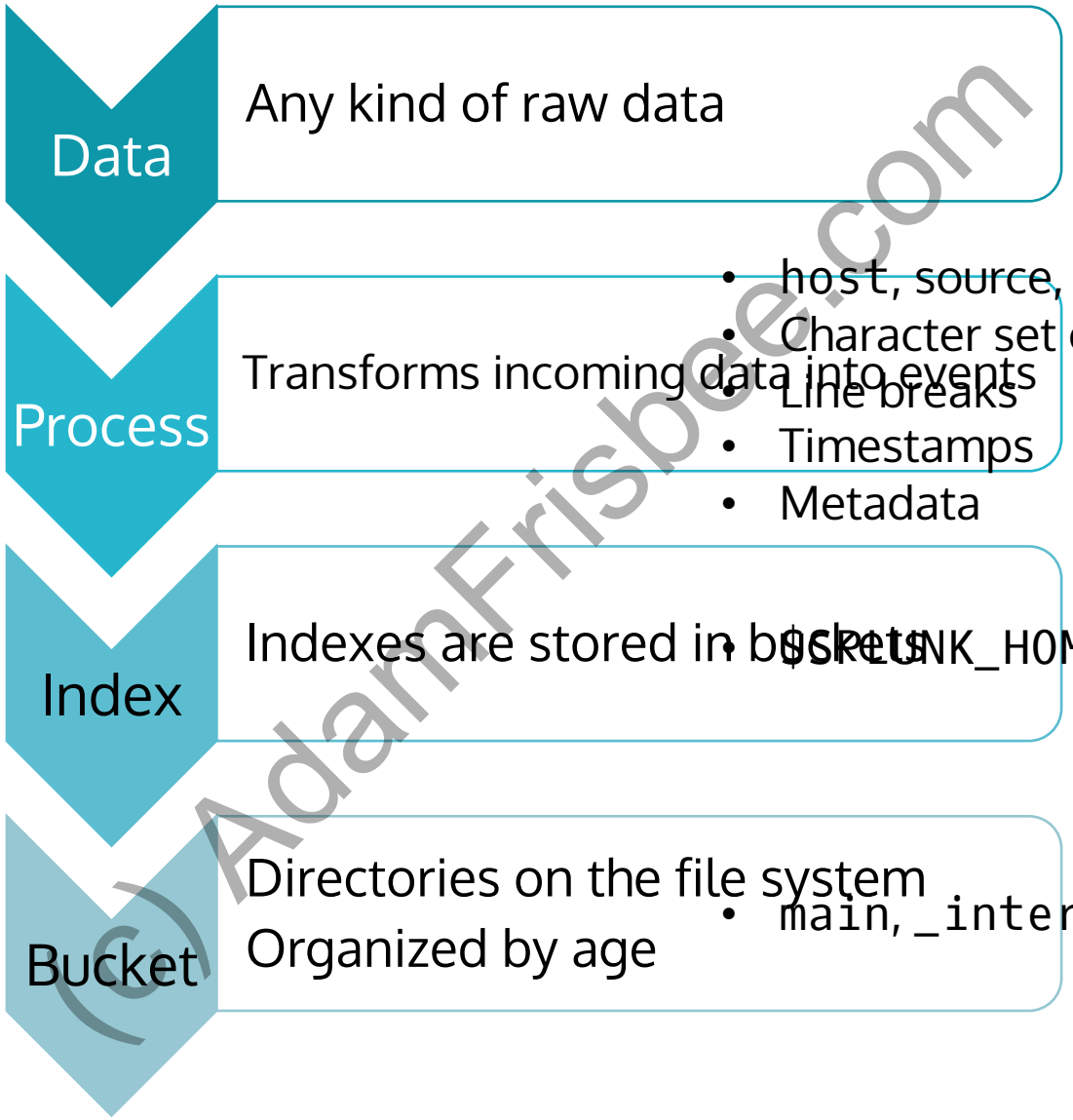
Indexes

- Repository of Splunk events
- Built-in or custom
- Indexes contain three types of data
  1. Raw data in compressed form
  2. Indexes that point to the raw data
  3. Metadata

Index Types

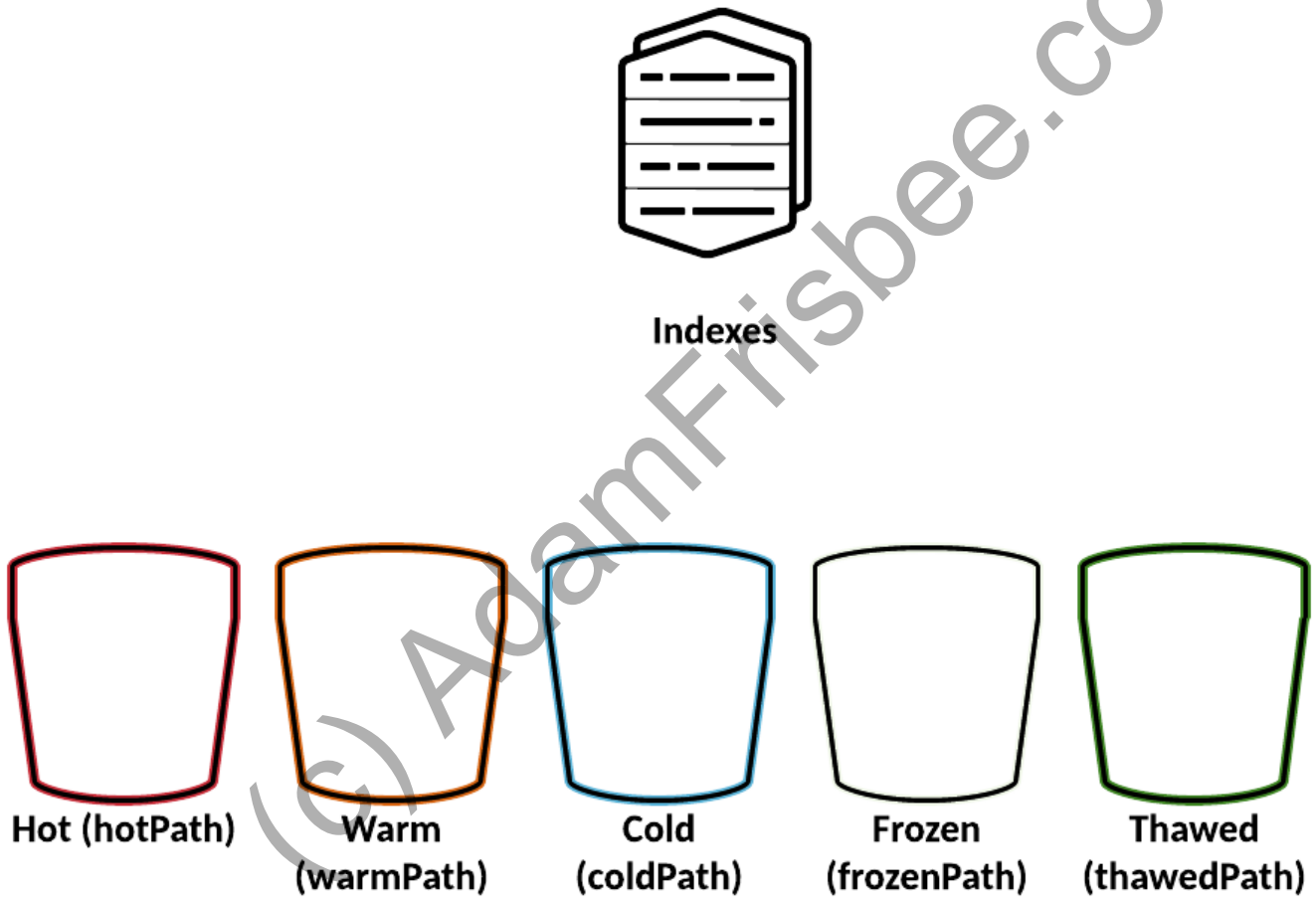
- Event
  - The default type
  - Can handle any type of data
- Metrics
  - Optimized to store and retrieve *metrics data*





Types of Index Buckets

Buckets

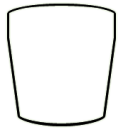


Buckets

`$SPLUNK_HOME/var/lib/splunk/defaultdb/db/*`

`$SPLUNK_HOME/var/lib/splunk/defaultdb/db/*`

`$SPLUNK_HOME/var/lib/splunk/defaultdb/colddb/*`

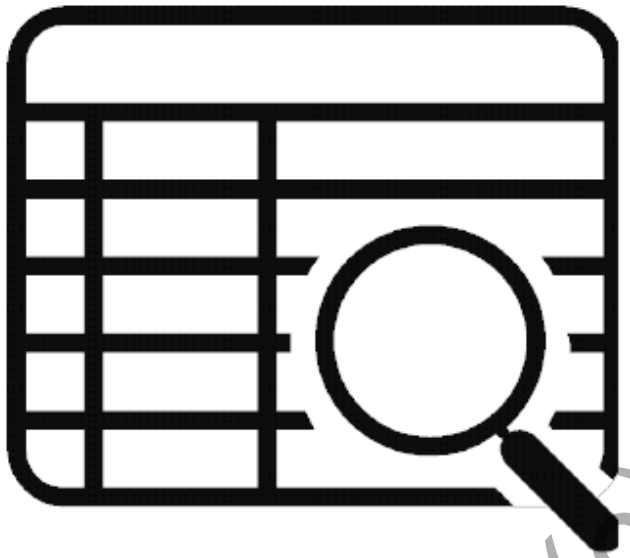


Location that you specify

`$SPLUNK_HOME/var/lib/splunk/defaultdb/thaweddb/*`

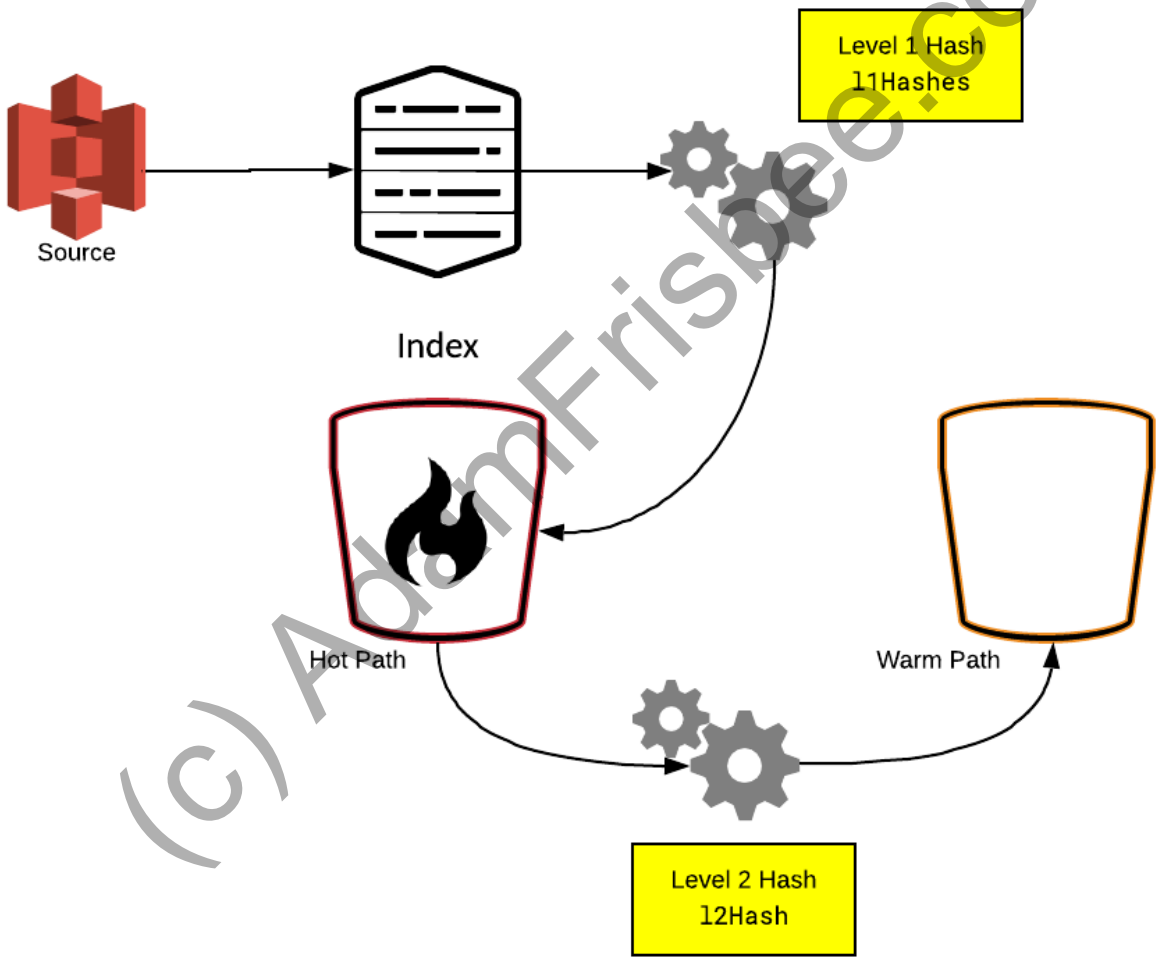
Check Data Integrity

How Data Integrity Works



- Splunk's double hash
  - Computes a hash on newly indexed data
  - Computes another hash on the same data when it moves buckets
  - Stores both hash files in the /rawdata directory

How Data Integrity Works



Command Line Options

- Check hashes to validate data

```
./splunk check-integrity -bucketPath [ bucket path ] [ -verbose ]
```

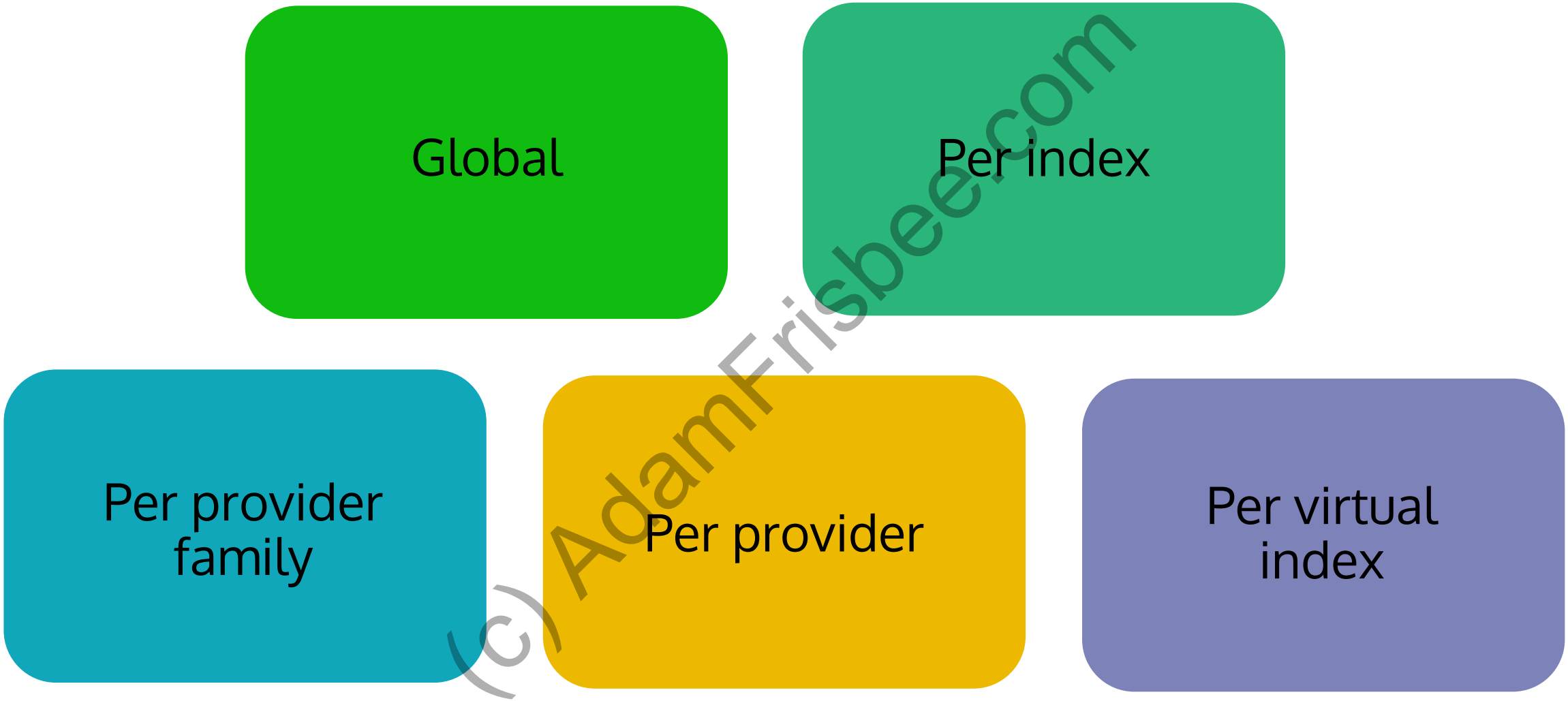
- Configure data integrity control

```
enableDataIntegrityControl=true
```

- Regenerate hashes

```
./splunk generate-hash-files -bucketPath [ bucket path ] [ verbose ]
```

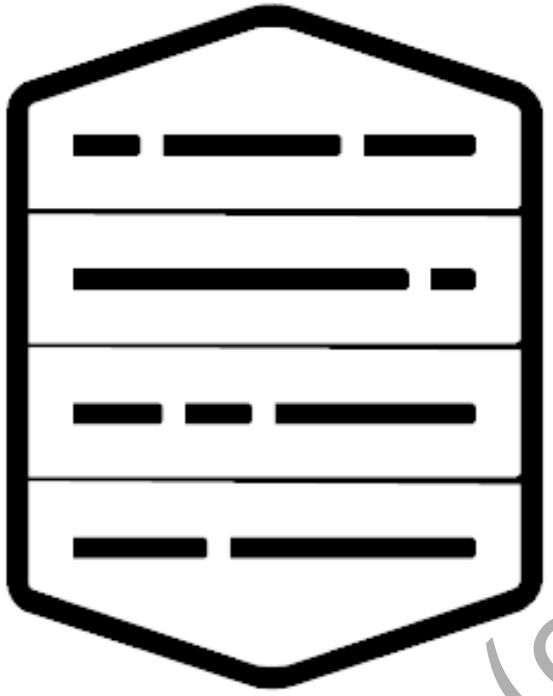
Indexes.conf Options



Global Settings

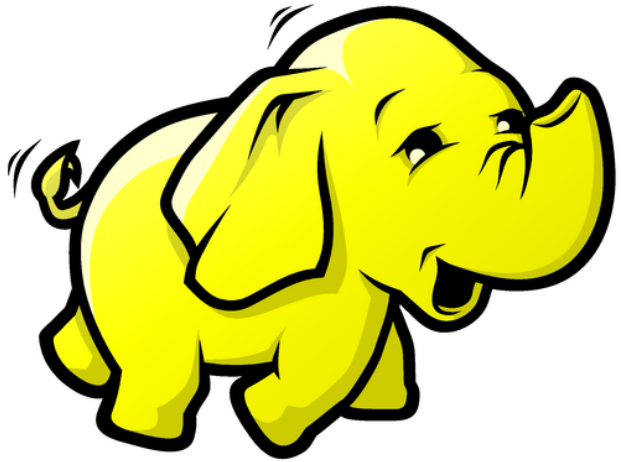
- Defined either at the beginning of the file or in the `[default]` stanza
- Each index.conf file has only one `[default]` stanza

Per Index Options



- Options under an `[<index>]` stanza
- A few of the many options
  - Set bucket paths
  - Set database sizes
  - Specify event or metric data types

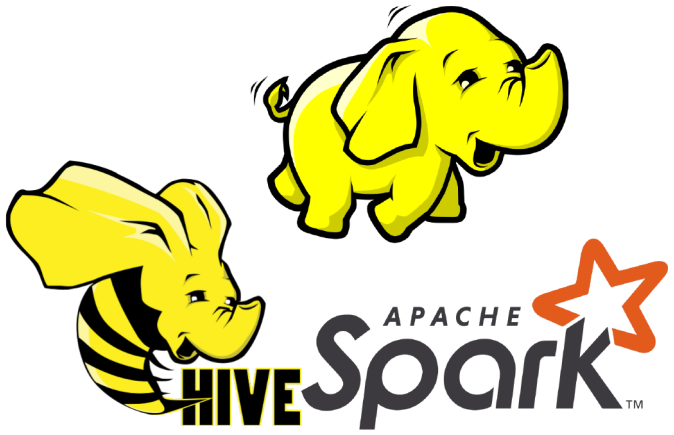
Per Provider Options



- Options for External Resource Providers (ERPs)
- All provider stanzas begin with

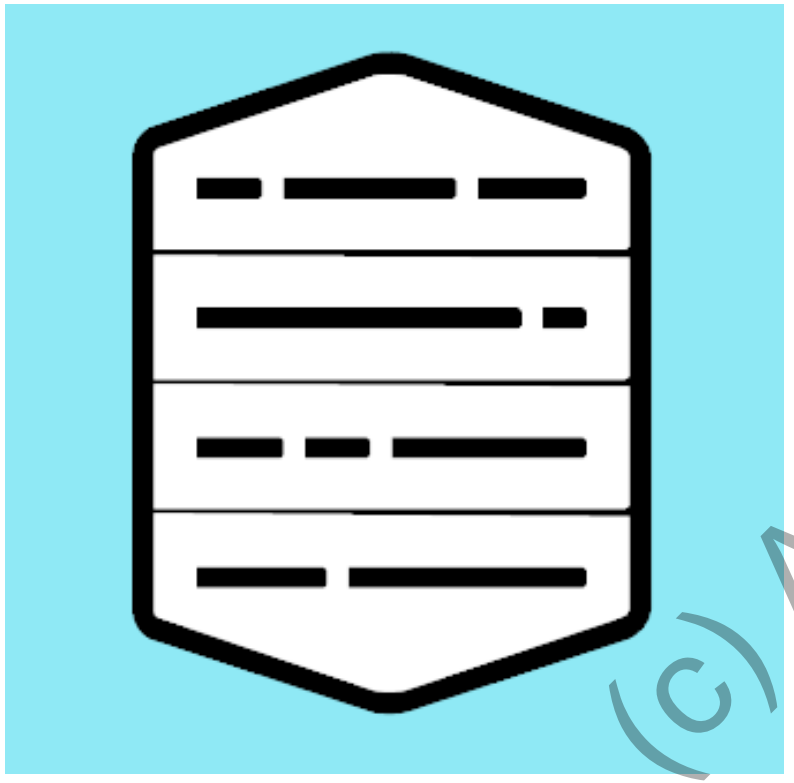
```
[provider: ]
```

Per Provider Family Options



- Properties that are common to multiple providers
- All properties that can be used in a family can be used in a provider
- Stanzas for provider families begin with `[provider-family:  ]`

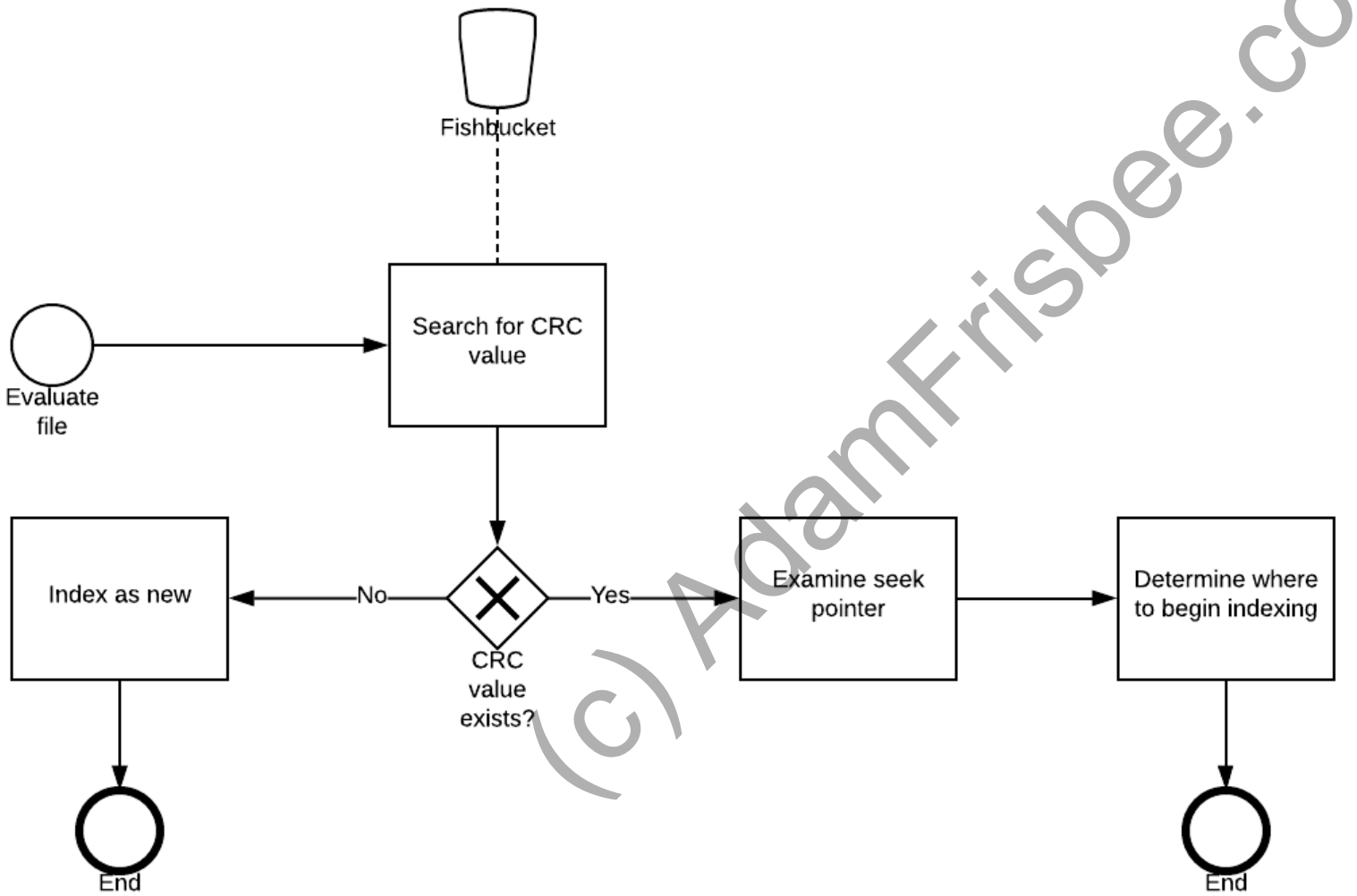
Per Virtual Index Options

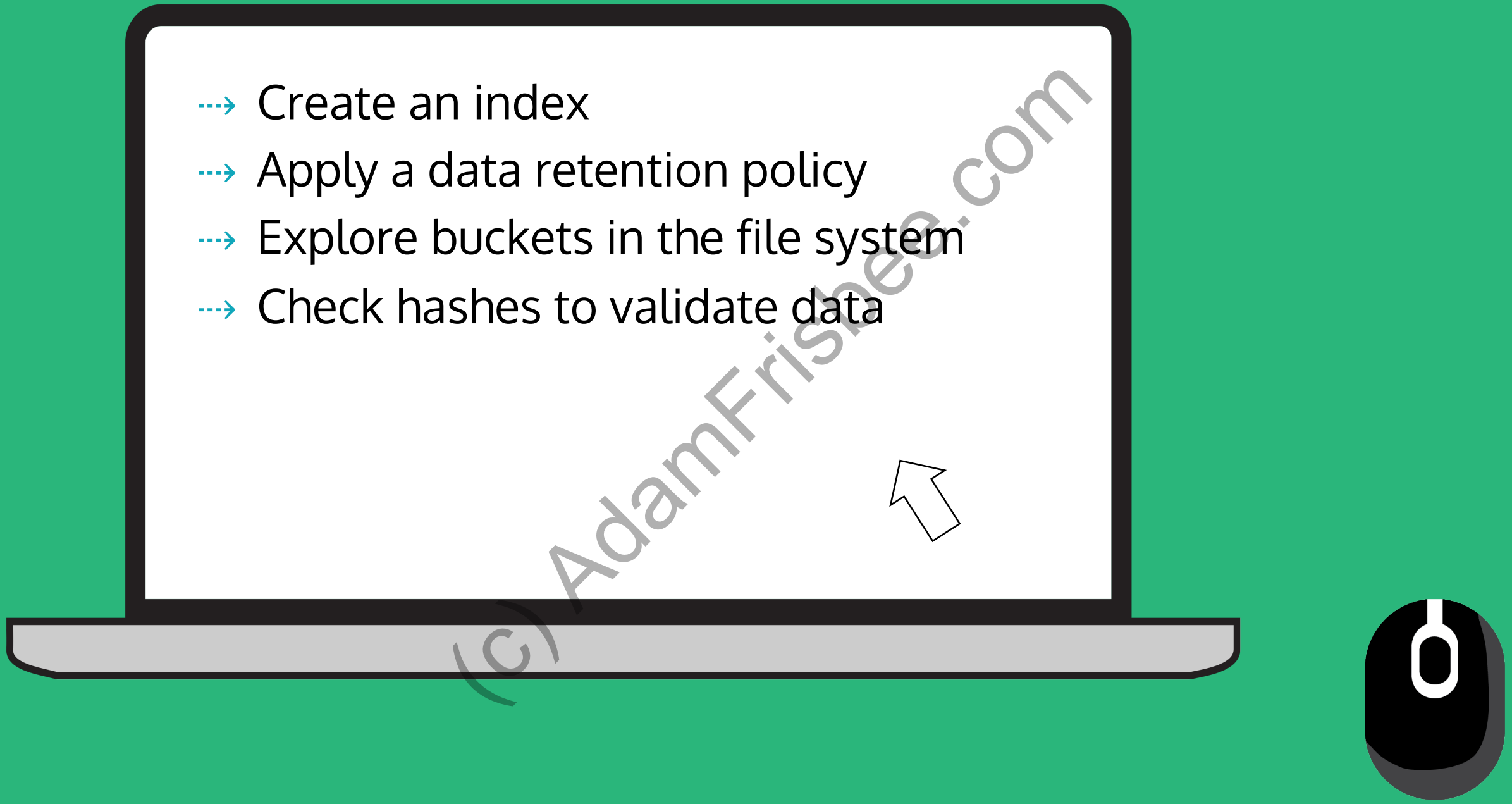


- Common for the Hadoop family
- Let Splunk access data stored in external systems and push computations to those systems

The Fish Bucket

The Fish Bucket





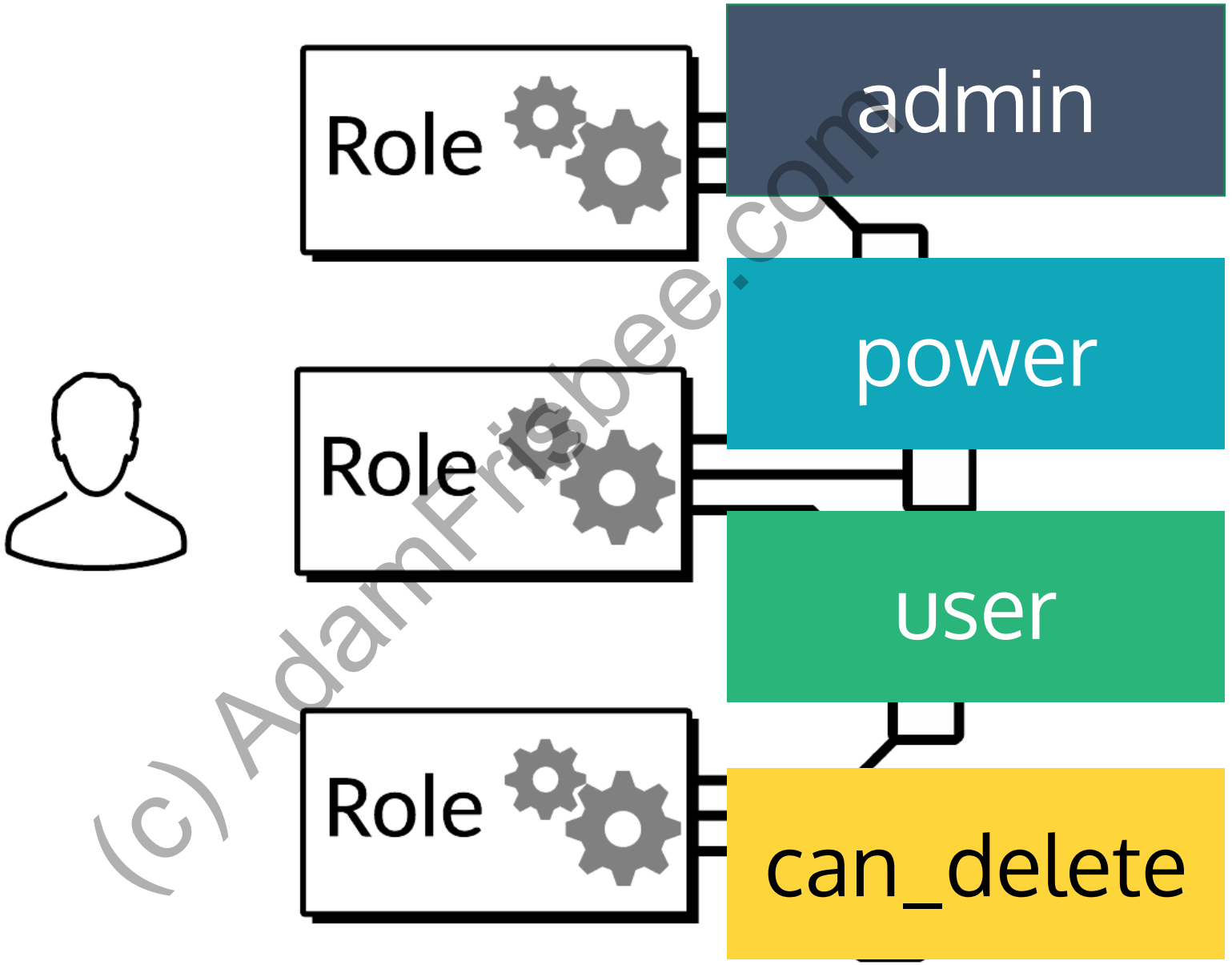
Summary

- Discussed indexes and buckets
- Learned what the fish bucket is for
- Learned about data integrity and retention
  - How to set a retention policy
  - How to check for data integrity

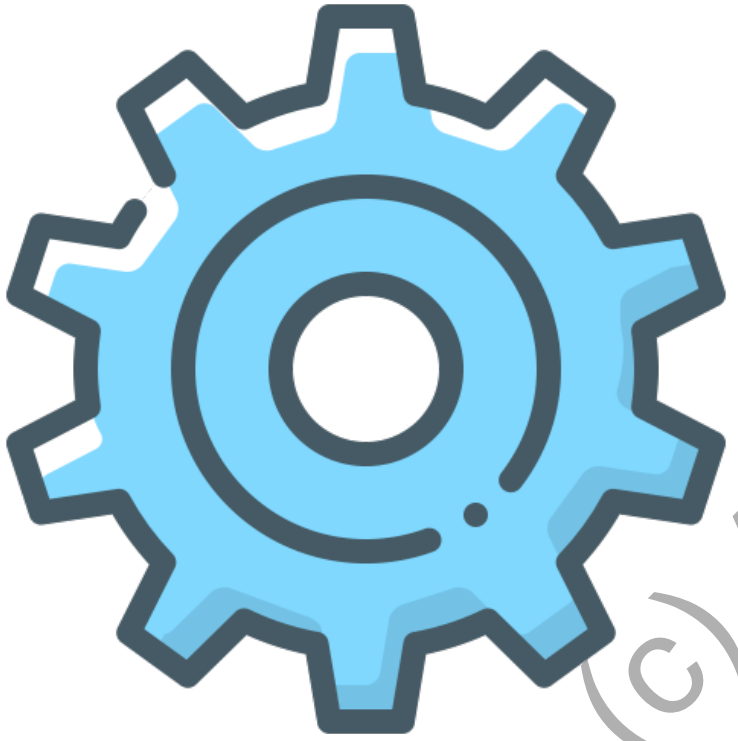
Authentication Management

- Describe user roles
- Create a custom role
- Add Splunk users

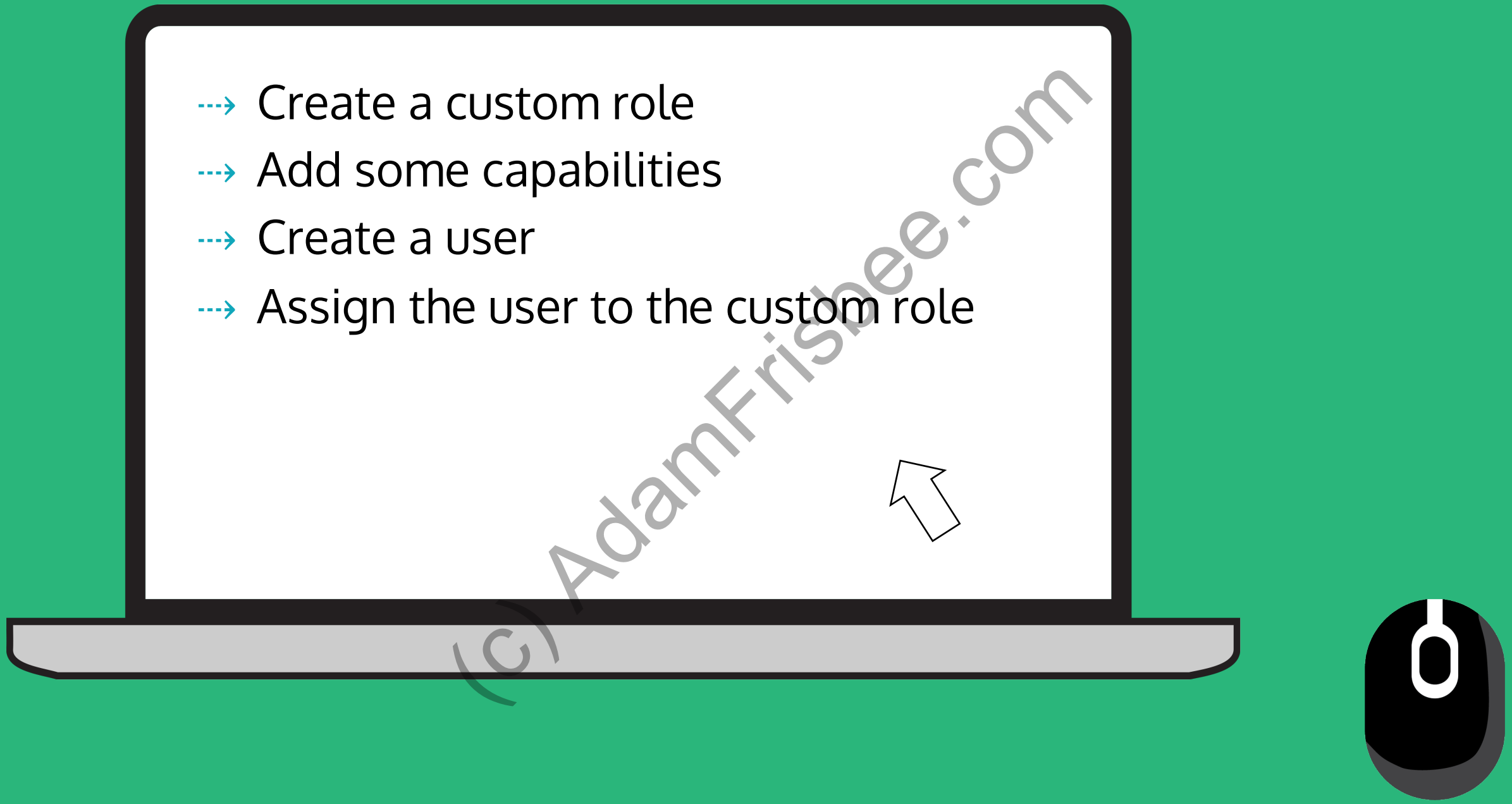
Describe User Roles



Capabilities



- Assigned to roles
- Additive in nature
- Can be used to granularly manage users in Splunk
- Can be added or removed in Splunk web, or in authorize.conf



Summary

- Described user roles
- Created a custom role
- Created a new user
- Added that user to our custom role

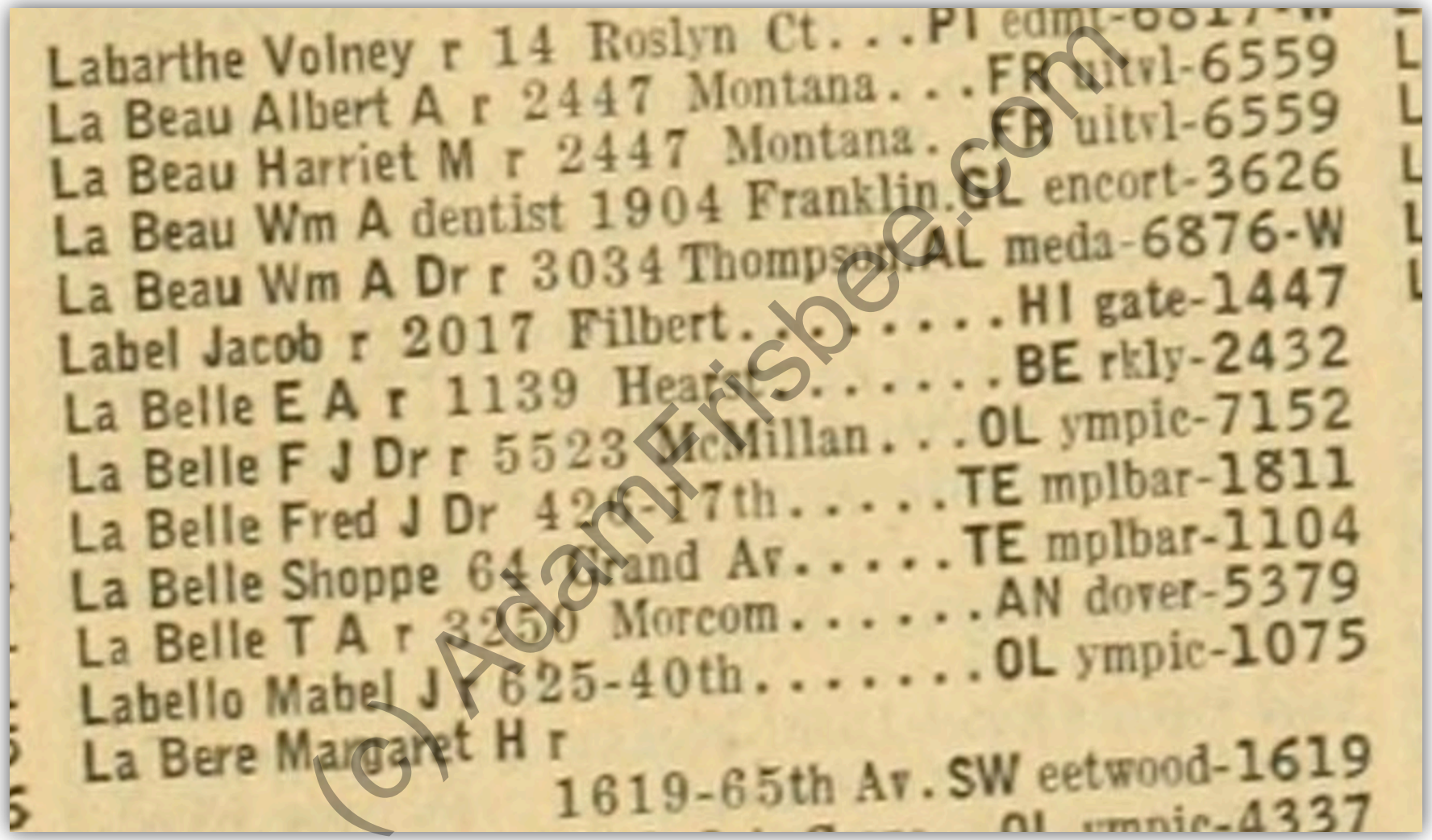
Splunk Authentication Management

- Integrate Splunk with LDAP
- List other user authentication options
- Describe the steps to enable multifactor authentication in Splunk

Integrate Splunk with LDAP

LDAP Quick Review

- Defines a protocol to authenticate to, access, and update objects in an `X.500` style directory



Labarthe Volney r 14 Roslyn Ct. . . .PI edmt-6817-W
La Beau Albert A r 2447 Montana. . . .FR uitvl-6559
La Beau Harriet M r 2447 Montana. . . .FR uitvl-6559
La Beau Wm A dentist 1904 Franklin. . .GL encort-3626
La Beau Wm A Dr r 3034 Thompson. . . .AL meda-6876-W
Label Jacob r 2017 Filbert. . . . . .HI gate-1447
La Belle E A r 1139 Hearst. . . . . .BE rkly-2432
La Belle F J Dr r 5523 McMillan. . . .OL ympic-7152
La Belle Fred J Dr 426-17th. . . . . .TE mplbar-1811
La Belle Shoppe 64 Grand Av. . . . . .TE mplbar-1104
La Belle T A r 3250 Morcom. . . . . .AN dover-5379
Labello Mabel J r 625-40th. . . . . .OL ympic-1075
La Bere Margaret H r
1619-65th Av. SW eetwood-1619
OL ympic-4337

LDAP Attribute	Meaning
DN	Distinguished Name, an entry's unique identifier. Composed of multiple attributes
CN	Common/canonical Name
OU	Organizational Unit

```
dn: cn=John Doe,dc=business,dc=lcl
cn: John Doe
sn: Doe
```

Integrate Splunk with LDAP

- Through the Splunk web interface
- By editing the `authentication.conf` file directly

Integrate Splunk with LDAP with Splunk Web

- Three easy steps

1. Create an LDAP strategy
2. Map LDAP groups to Splunk roles
3. Specify the connection order

LDAP connection settings

- Host
- Port
- Bind DN

User settings

- User base DN
- User base filter
- User name attribute
- Real name attribute
- Email attribute
- Group mapping attribute

Group settings

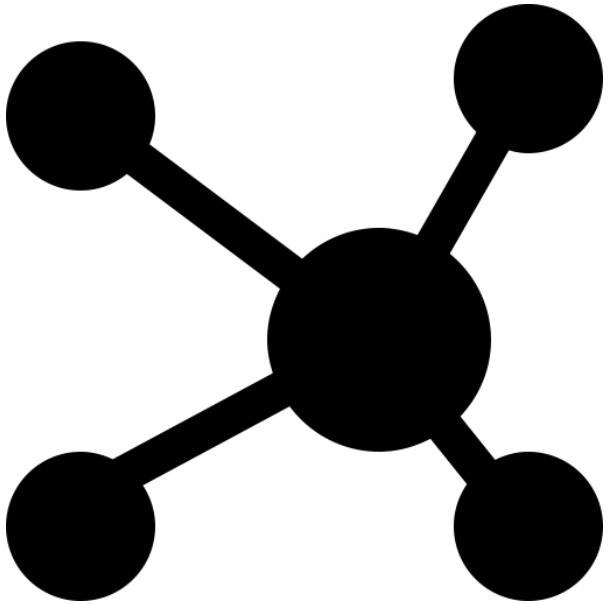
- Group base DN
- Static group search filter
- Group name attribute
- Static member attribute

Dynamic group settings

- Dynamic member attribute
- Dynamic group search filter

Other User Authentication Options

Splunk Authentication



- Provides the following default accounts
  - Admin
  - Power
  - User
- You can define you own roles and capabilities

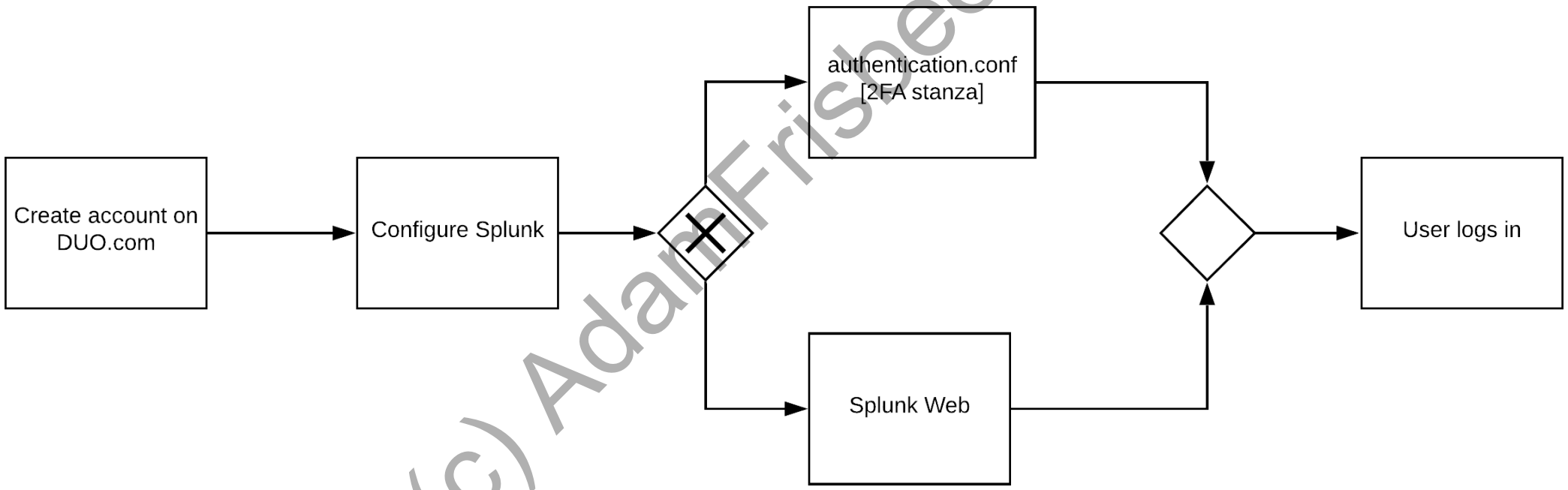
Scripted Authentication API

- Integrate Splunk authentication with an external authentication system

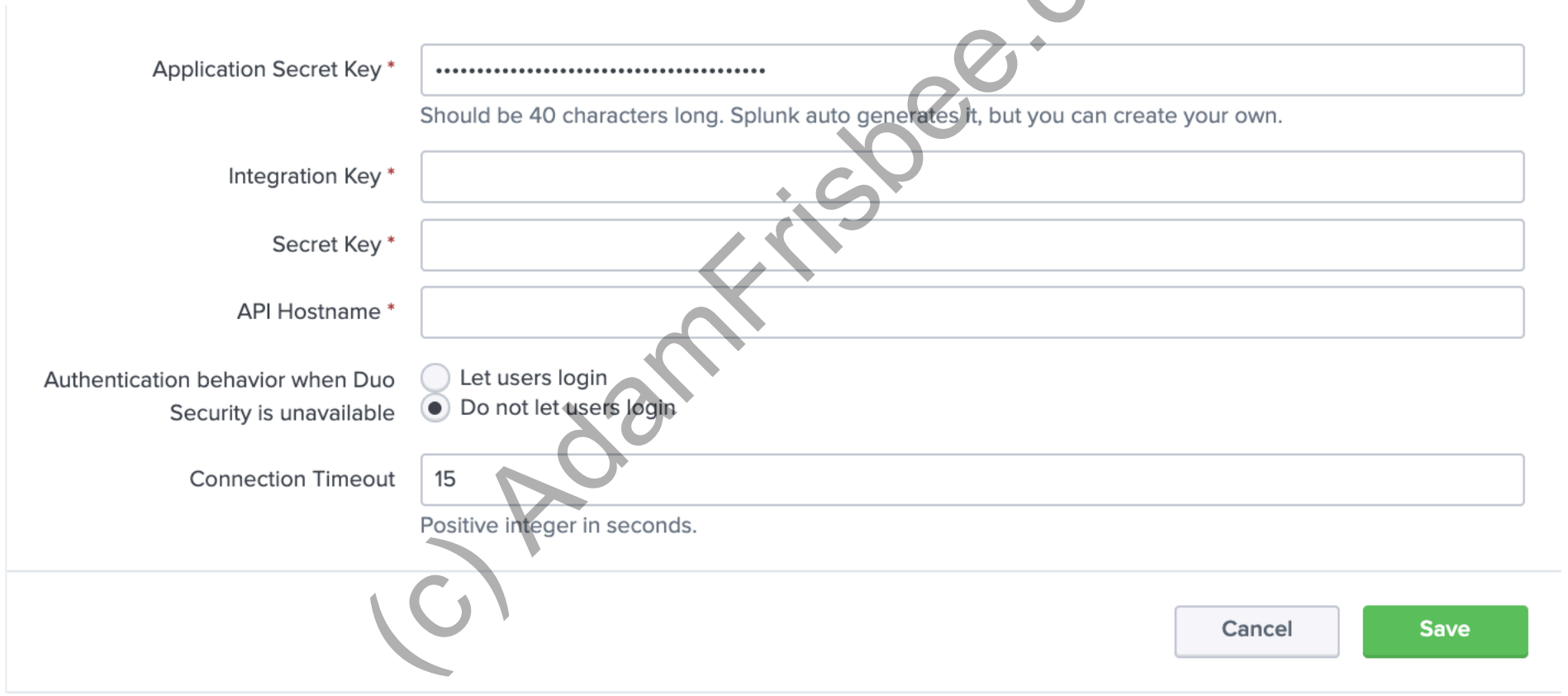
Multifactor Authentication

MFA Products Supported by Splunk

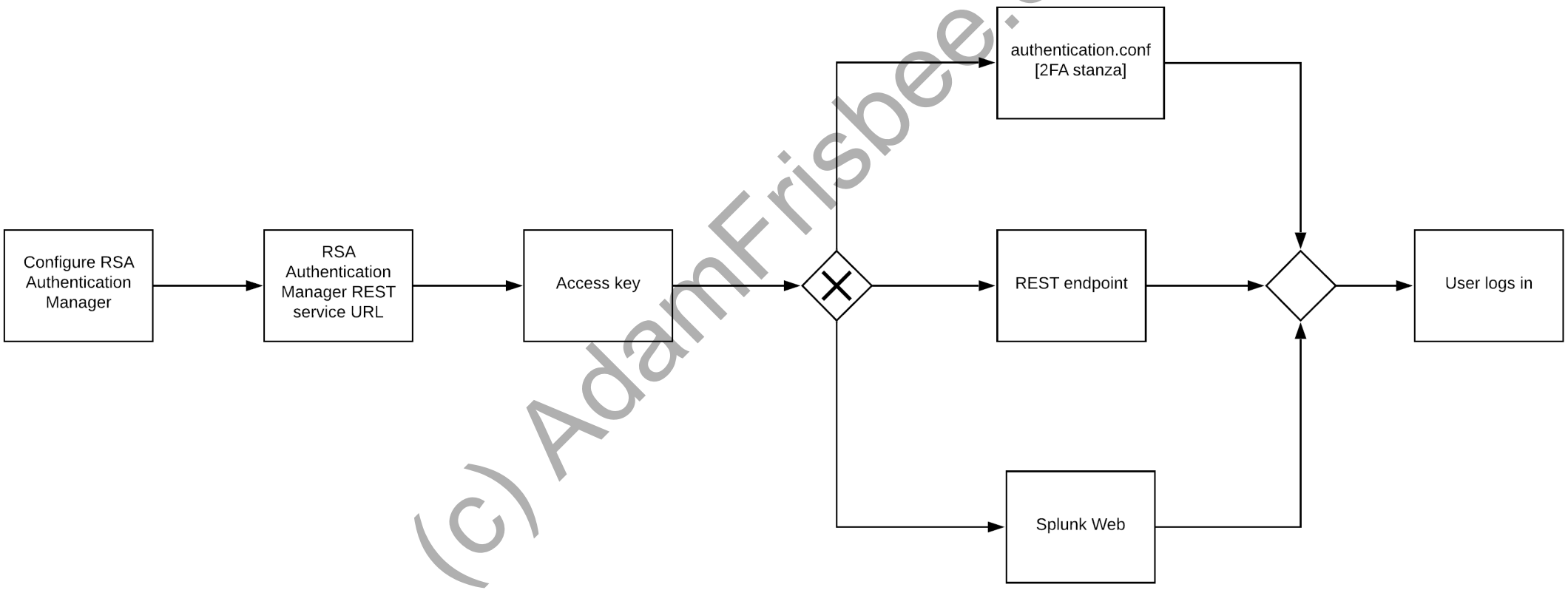
Configuring DUO



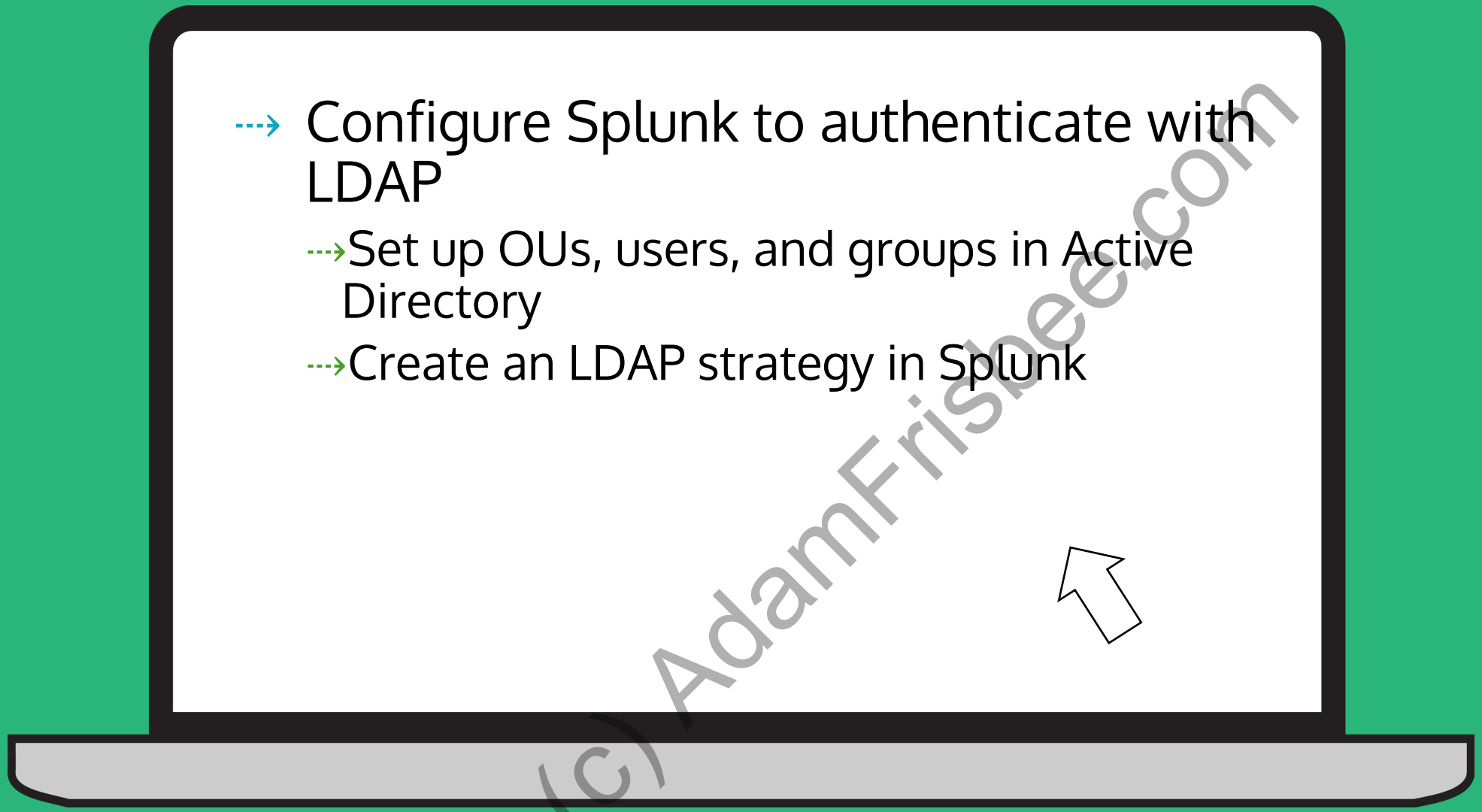
Configuring DUO



Configuring RSA



- Configure Splunk to authenticate with LDAP
  - Set up OUs, users, and groups in Active Directory
  - Create an LDAP strategy in Splunk



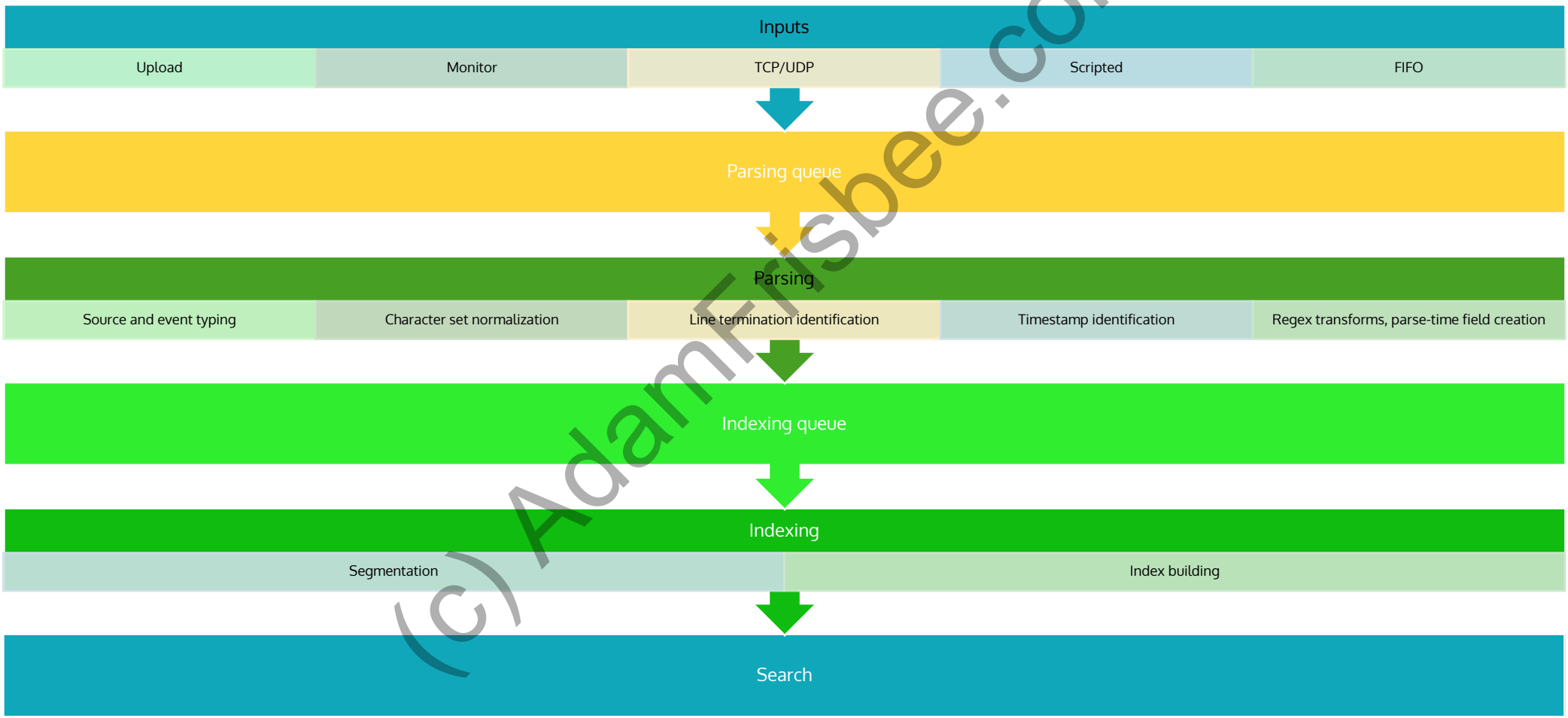
Summary

- Learned how to integrate Splunk with LDAP
- Discussed other authentication options
- Discussed two-factor authentication

Getting Data in

- Describe the basic settings for an input
- List Splunk forwarder types
- Configure a forwarder
- Add an input to a Universal Forwarder using the CLI

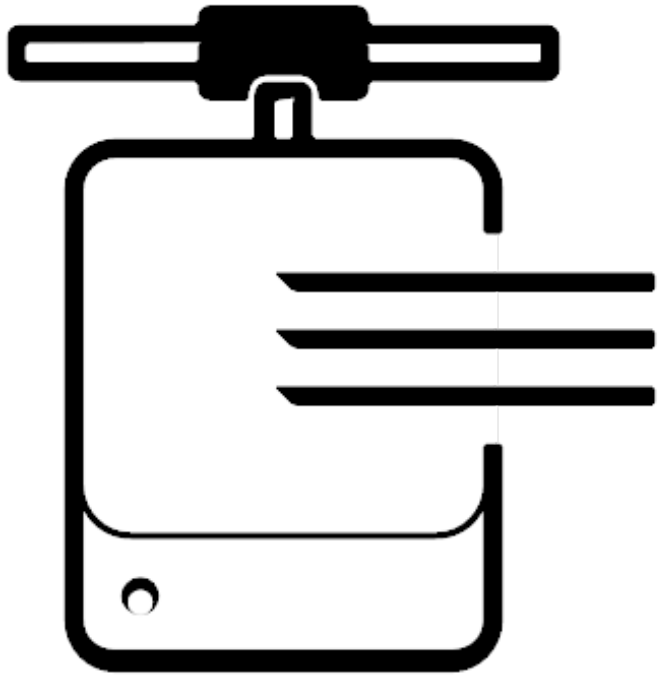
The Splunk Data Pipeline



General Input Categories

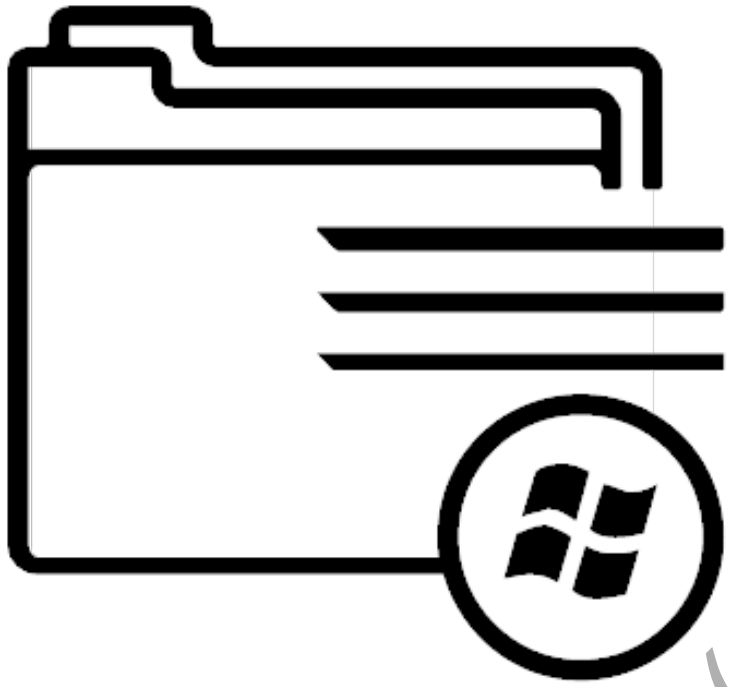
- File and directory inputs
  - Monitor files and directories
    - Locally and remotely
    - Monitor compressed files
  - Upload
    - Upload files to Splunk
    - Used for one-time analysis
  - MonitorNoHandle
    - Available for Windows hosts only
    - Monitors files and directories that the system rotates automatically

General Input Categories



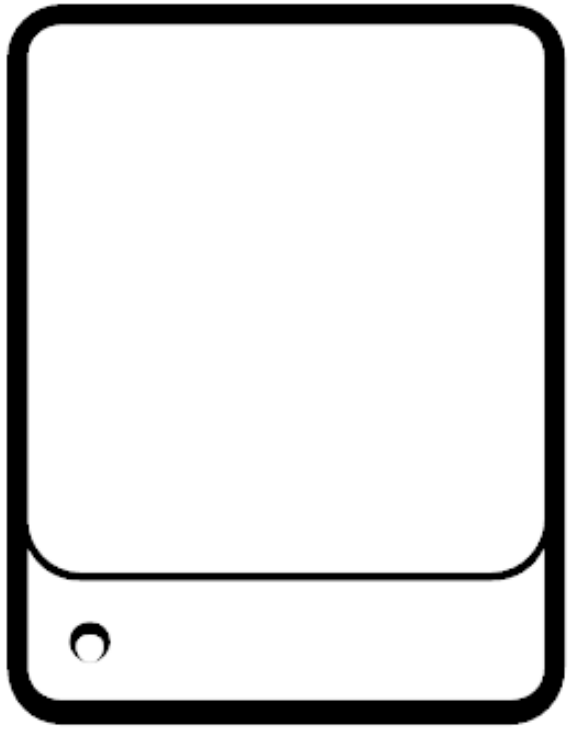
- Network inputs
  - Data from TCP and UDP
    - syslog
  - Data from SNMP events

General Input Categories



- Windows inputs
  - Windows event logs
  - Registry
  - Active Directory
  - WMI
  - Performance monitoring (perfmon)

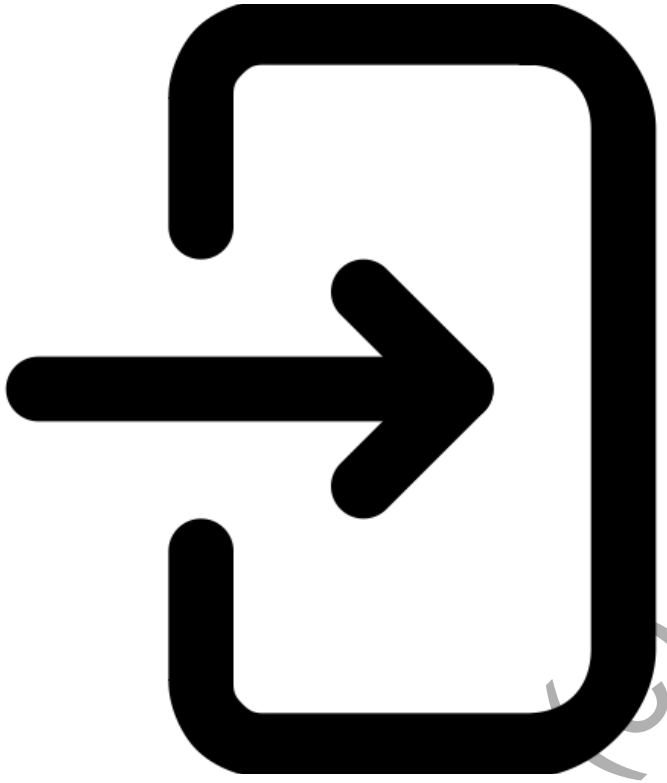
Other Data Sources



- Metrics
- FIFO queues
- Scripted inputs
- Modular inputs
- HTTP event collector

Basic Settings for an Input

Ways to Configure Inputs



- Through an app
  - Many apps have preconfigured inputs
- Splunk web
  - Settings > data inputs
  - Settings > add data
- CLI
  - `./splunk add monitor <path>`

Ways to Configure Inputs

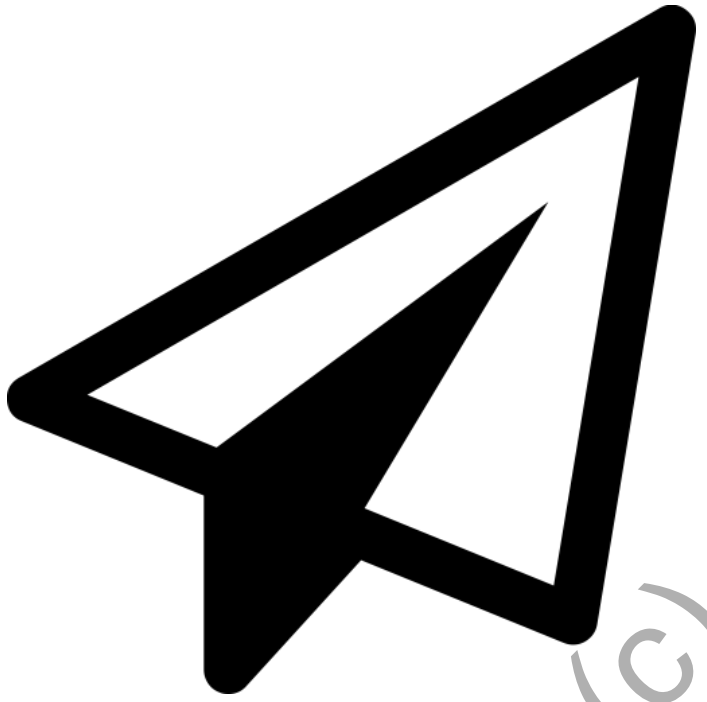
- Through `inputs.conf`
  - Add a stanza for each input
- Guided Data Onboarding (GDO)
  - Data input wizard

Splunk Forwarder Types

Universal

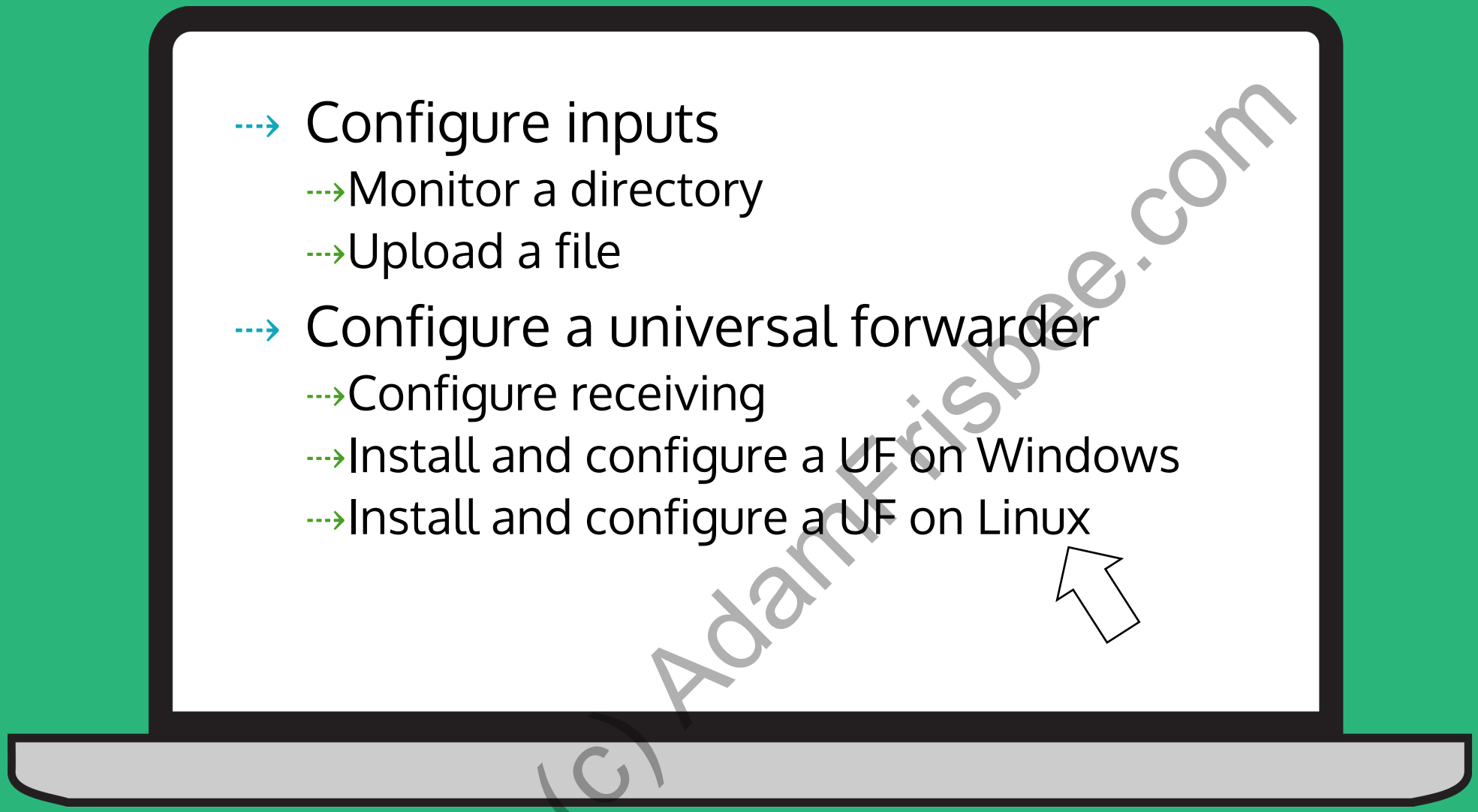
Heavy

Universal Forwarder Configuration Steps



- Configure receiving on a Splunk Enterprise instance
- Download and install the UF
- Start the UF
- Configure the UF to send data
- Configure the UF to collect data from the host system

- Configure inputs
  - Monitor a directory
  - Upload a file
- Configure a universal forwarder
  - Configure receiving
  - Install and configure a UF on Windows
  - Install and configure a UF on Linux



Summary

- Splunk data pipeline
- Configuring inputs
- Types of forwarders
- Configuring universal forwarders

Distributed Search

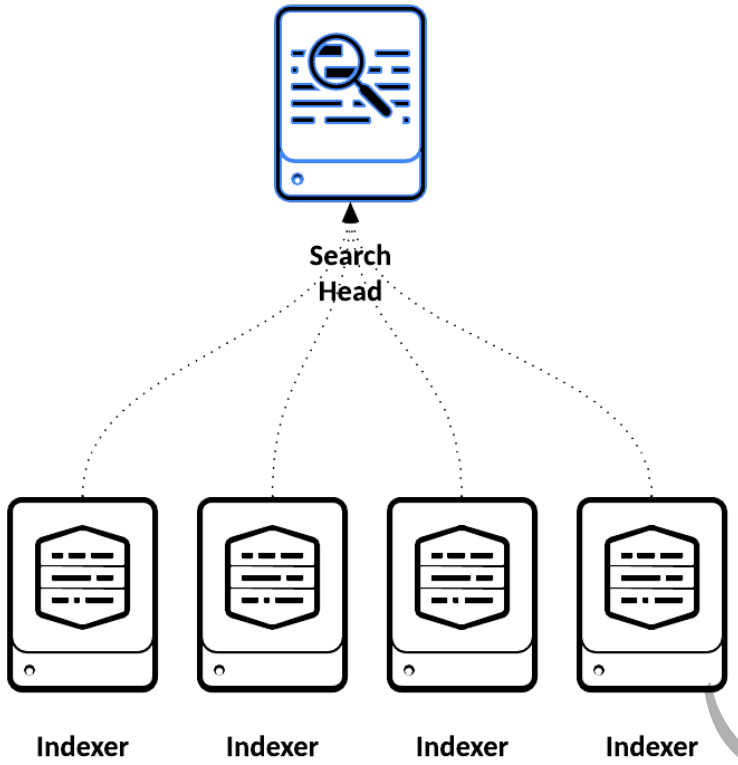
- Describe how distributed search works
- Explain the roles of the search head and search peers
- Search head scaling options
- Configure a distributed search group

How Distributed Search Works

Distributed search provides a way to scale your deployment by separating the search management and presentation layer from the indexing and search retrieval layer

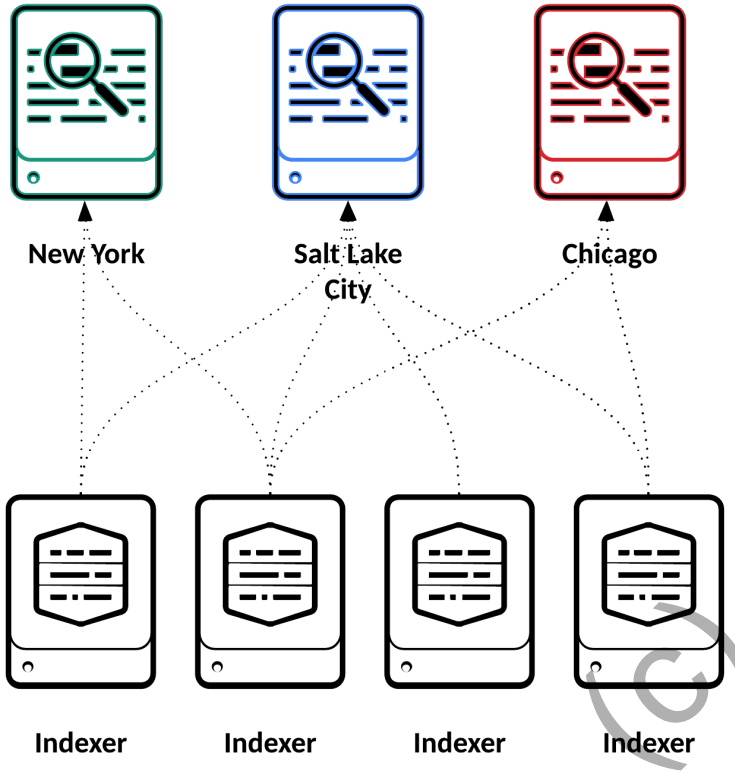
-Splunk Documentation

Components of Distributed Search



- Two necessary components for distributed searching
  - Search heads
  - Indexers, also called search peers

Why Distribute Searches?



- Horizontal scaling
- Access control
- Geo-dispersed data

Search Heads and Search Peers

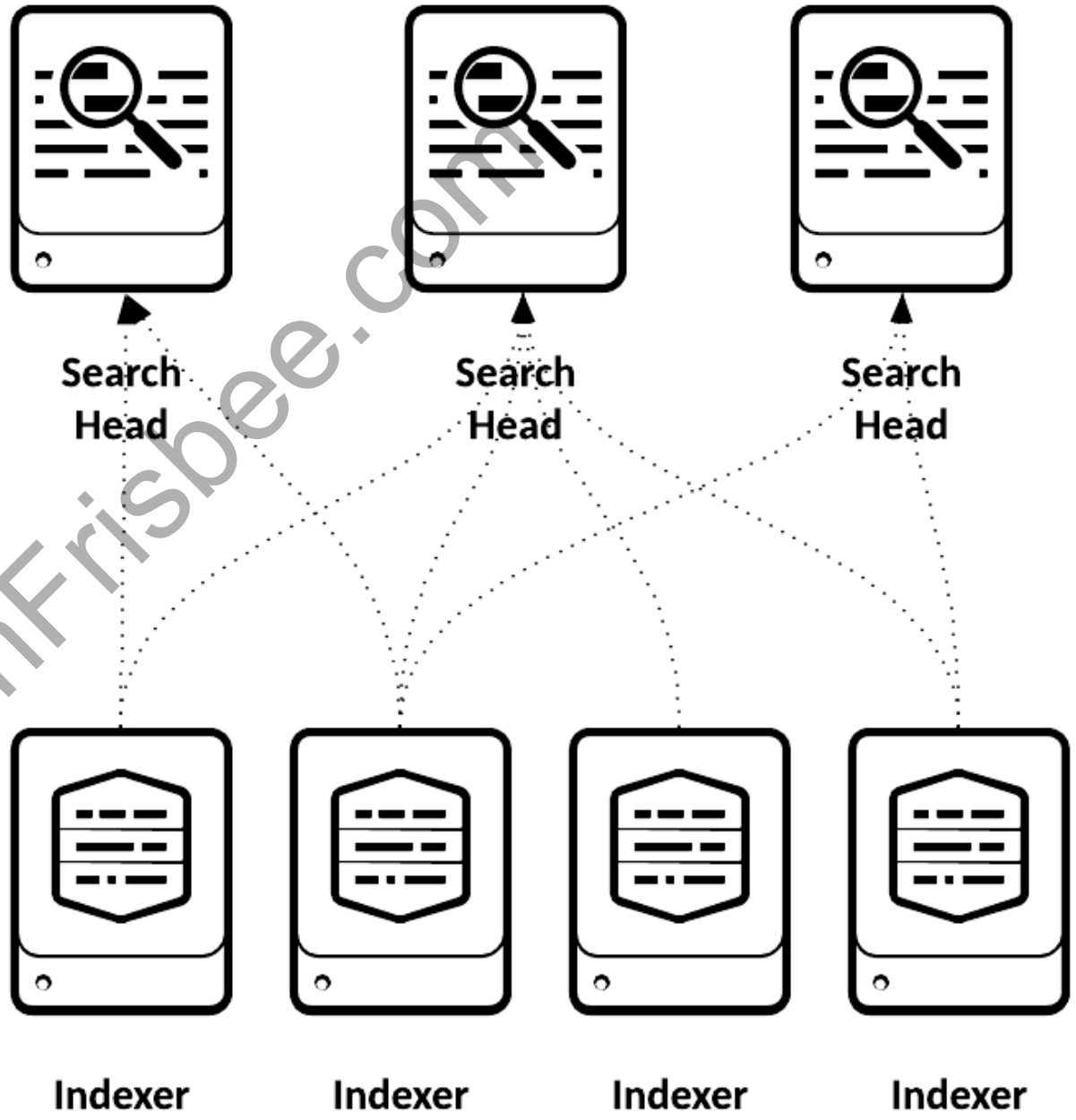
Independent Search Heads

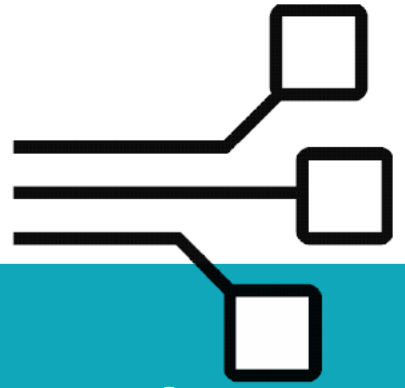
Search Head Cluster

Individual Indexers

Indexer Cluster

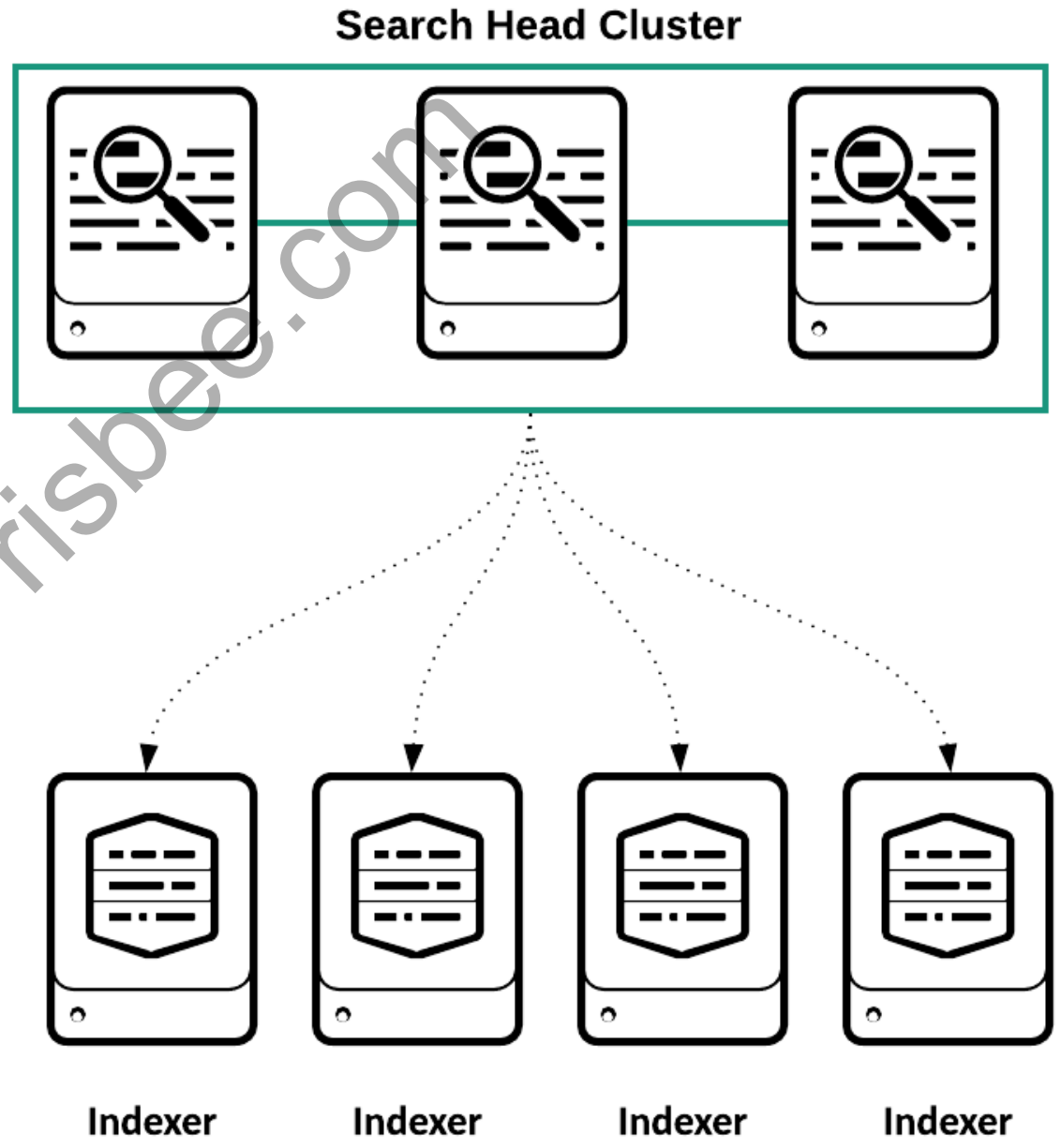
Independent Search Heads



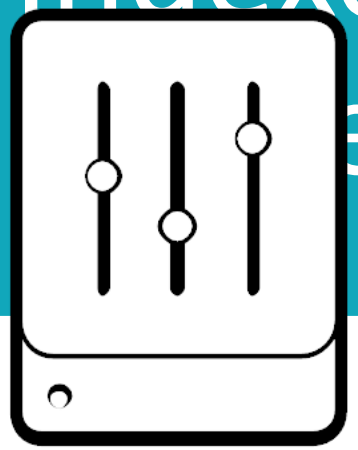


Search Head Cluster

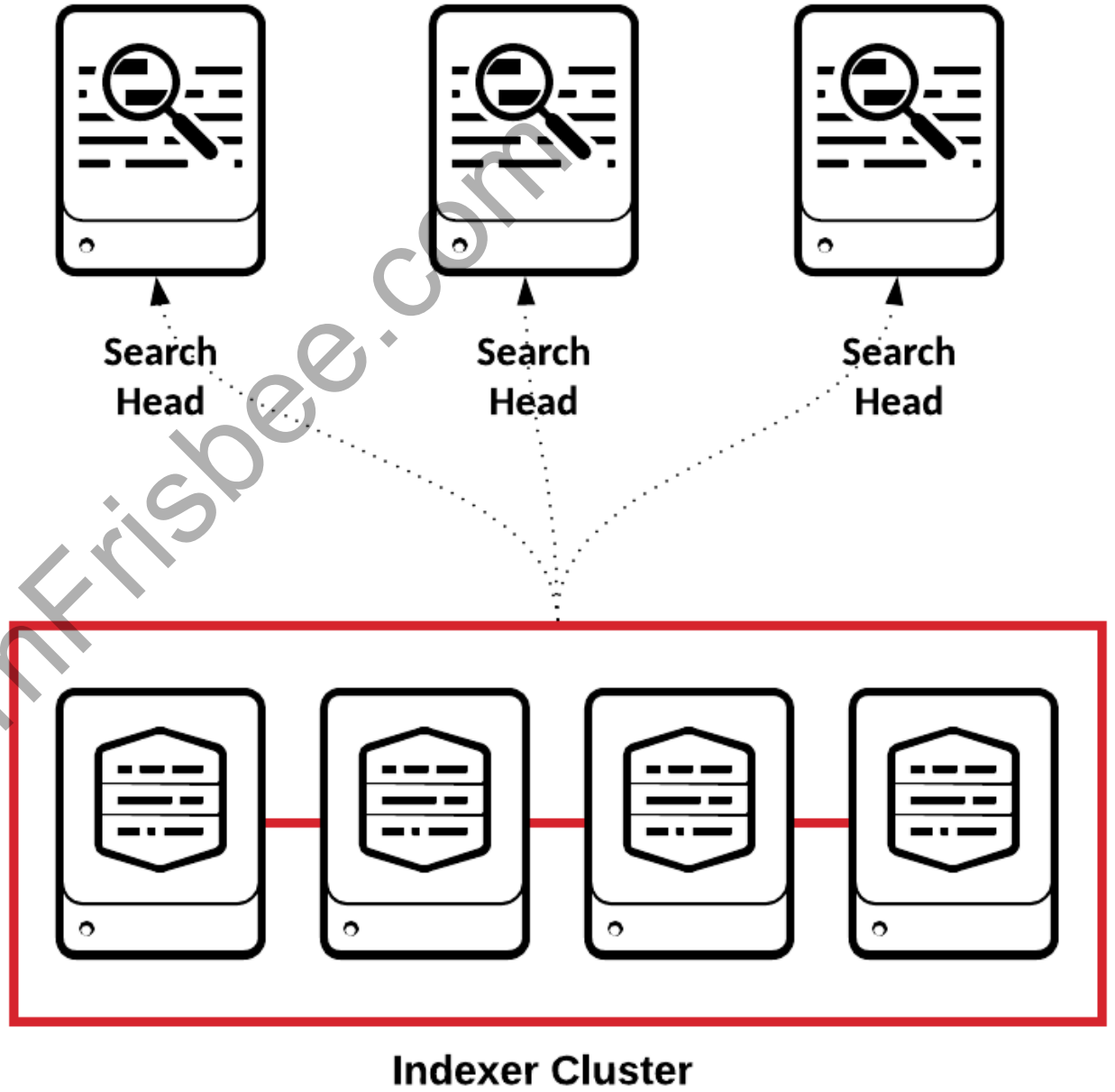
Deployer



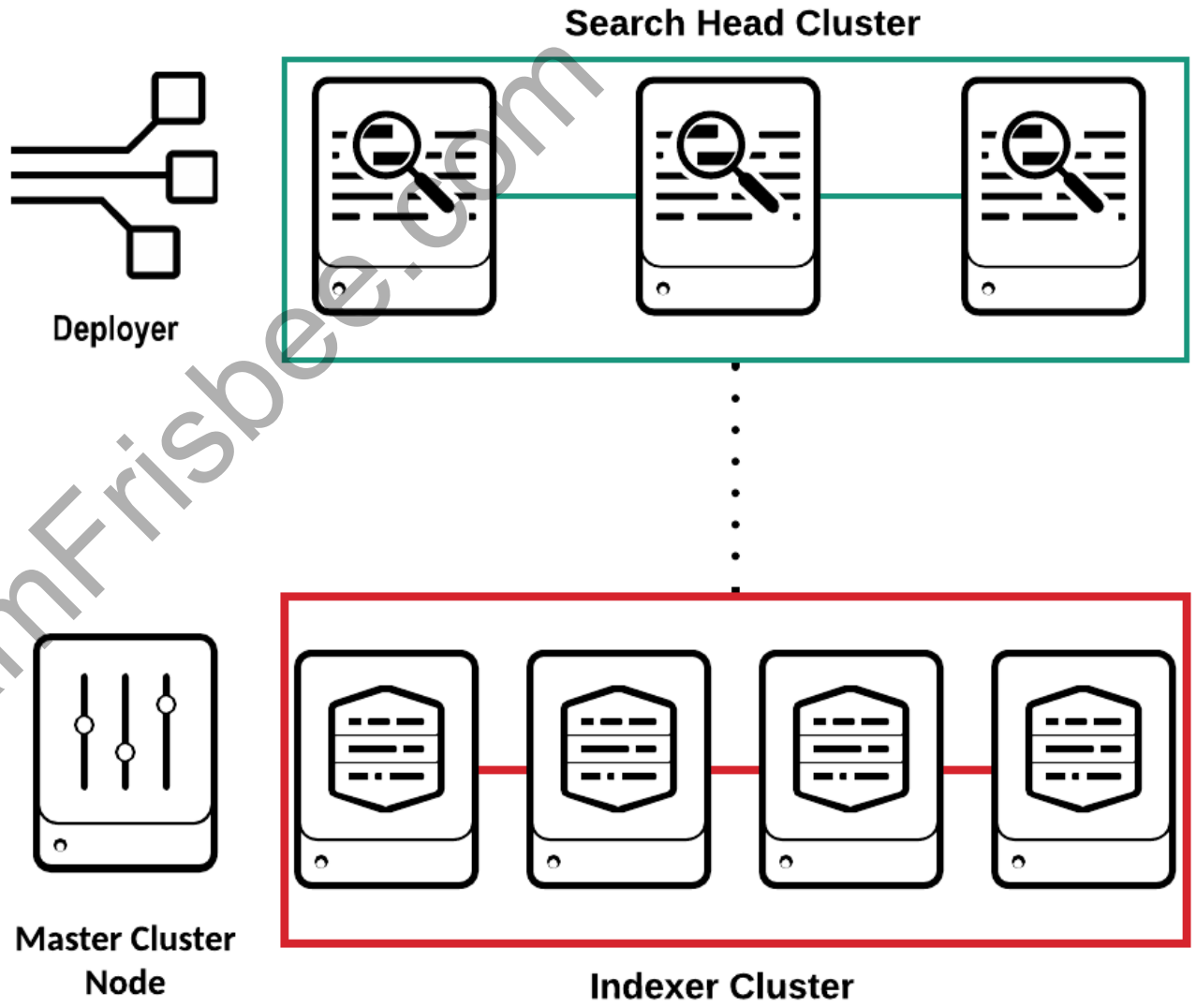
Indexer
er



Master Cluster
Node



Cluster Indexes and Search Heads

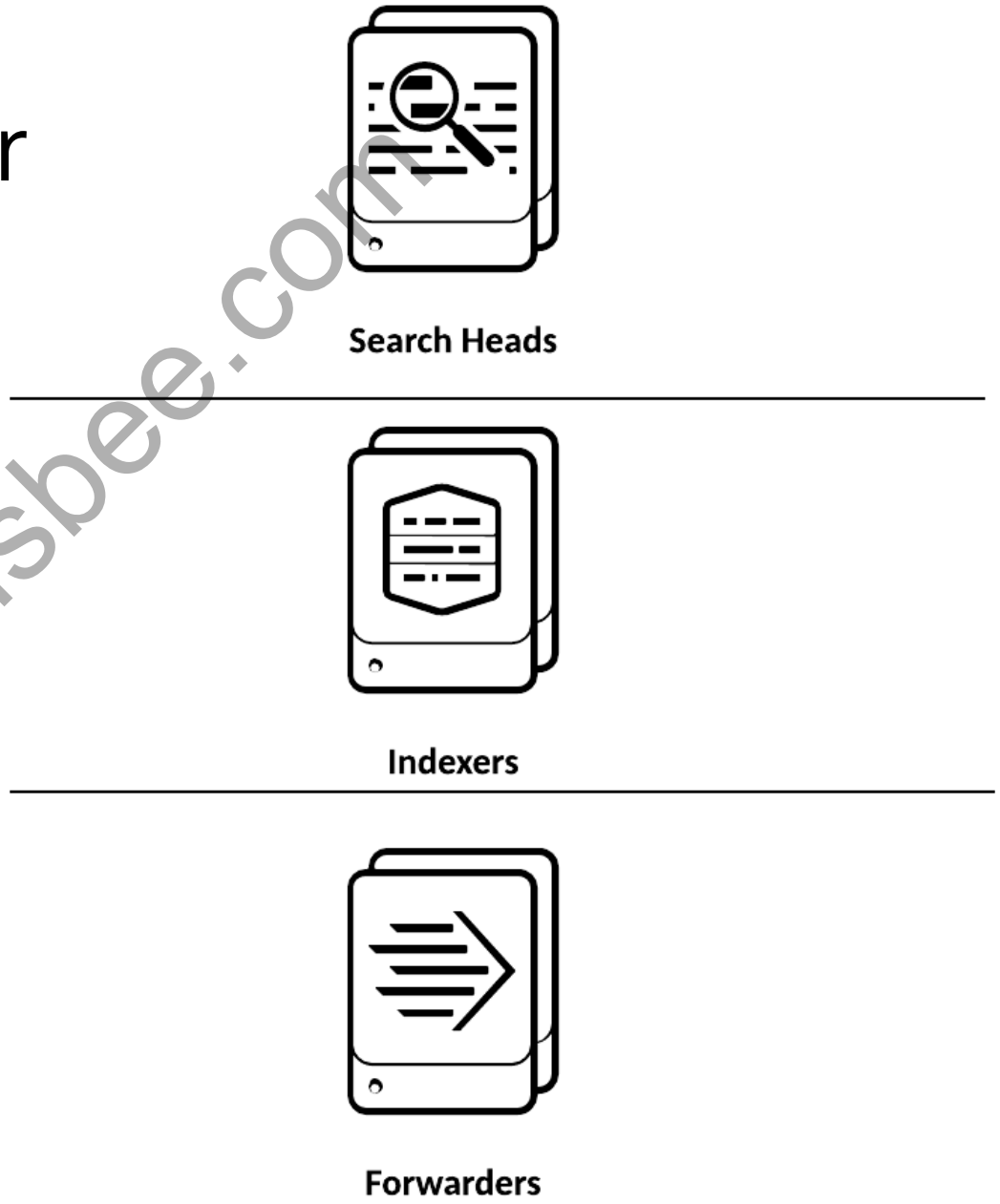


Scaling Options

Search Management Tier

Indexing Tier

Data Input Tier



Splunk Scaling Recommendations

Daily Indexing Volume						
	< 2GB/day	2 to 300 GB/day	300 to 600 GB/day	600GB to 1TB/day	1 to 2TB/day	2 to 3TB/day
Total Users: less than 4	1 combined instance	1 combined instance	1 Search Head, 2 Indexers	1 Search Head, 3 Indexers	1 Search Head, 7 Indexers	1 Search Head, 10 Indexers
Total Users: up to 8	1 combined instance	1 Search Head, 1 Indexers	1 Search Head, 2 Indexers	1 Search Head, 3 Indexers	1 Search Head, 8 Indexers	1 Search Head, 12 Indexers
Total Users: up to 16	1 Search Head, 1 Indexers	1 Search Head, 1 Indexers	1 Search Head, 3 Indexers	2 Search Heads, 4 Indexers	2 Search Heads, 10 Indexers	2 Search Heads, 15 Indexers
Total Users: up to 24	1 Search Head, 1 Indexers	1 Search Head, 2 Indexers	2 Search Heads, 3 Indexers	2 Search Heads, 6 Indexers	2 Search Heads, 12 Indexers	3 Search Heads, 18 Indexers
Total Users: up to 48	1 Search Head, 2 Indexers	1 Search Head, 2 Indexers	2 Search Heads, 4 Indexers	2 Search Heads, 7 Indexers	3 Search Heads, 14 Indexers	3 Search Heads, 21 Indexers

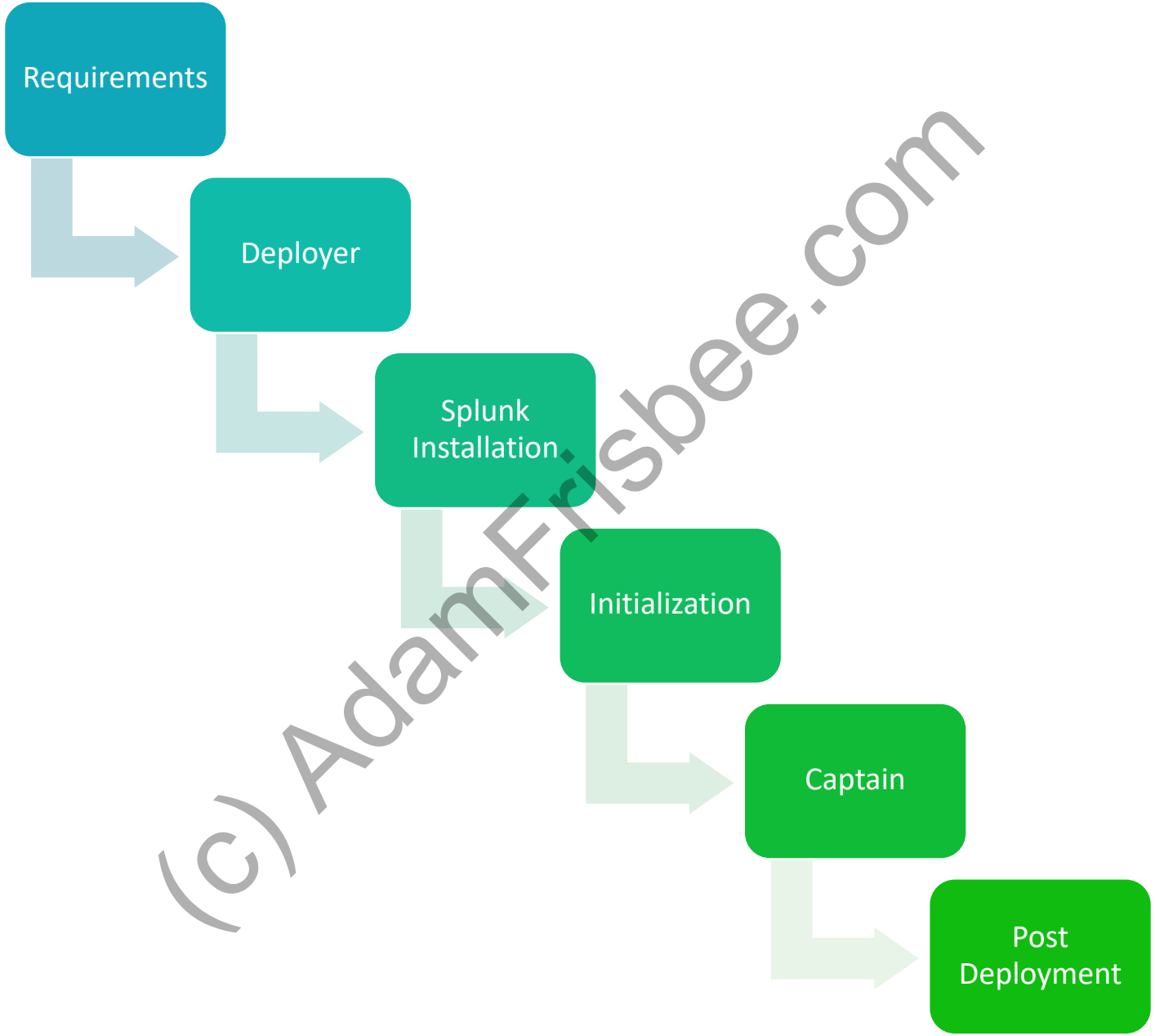
Configuring Distributed Search

What we Need to get Started

- Machines
  - Virtual, physical, or cloud machines
  - Splunk Enterprise installed on all machines
- Network
  - All machines must be able to communicate with each other
  - All machines need to have a FQDN
- CLI commands

Advisory

- Your networking must be **impeccable** for these clusters to work
  - Static IP addresses recommended
  - DNS server, or manual hostname/hosts entries
  - Firewalls configured properly
  - Minimum of one deployer and three cluster members



Configuring the Deployer

Set security key

Label the cluster

Edit `servers.conf` in `$SPLUNK_HOME/opt/etc/system/local`

Add stanza and values:

```
[shclustering]
pass4SymmKey = <security key>
shcluster_label = <name your cluster>
```

Initialize Cluster Members

Run the initialization command on each cluster member

```
sudo ./splunk init shcluster-config
-auth <username>:<password>
-mgmt_uri <URI>:<management_port>
-replication_port <replication_port>
-replication_factor <n>
-conf_deploy_fetch_url <URL>:<management_port>
-secret <security_key>
-shcluster_label <label>
```

Bootstrap the Captain

Run this bootstrap command to bring up the captain

```
splunk bootstrap shcluster-captain
-servers_list "<URI>:<management_port>,… >”
-auth <username>:<password>
```

Add Other Members

To add cluster members later

```
splunk add shcluster-member -current_member_uri
<URI>:<management_port>
```

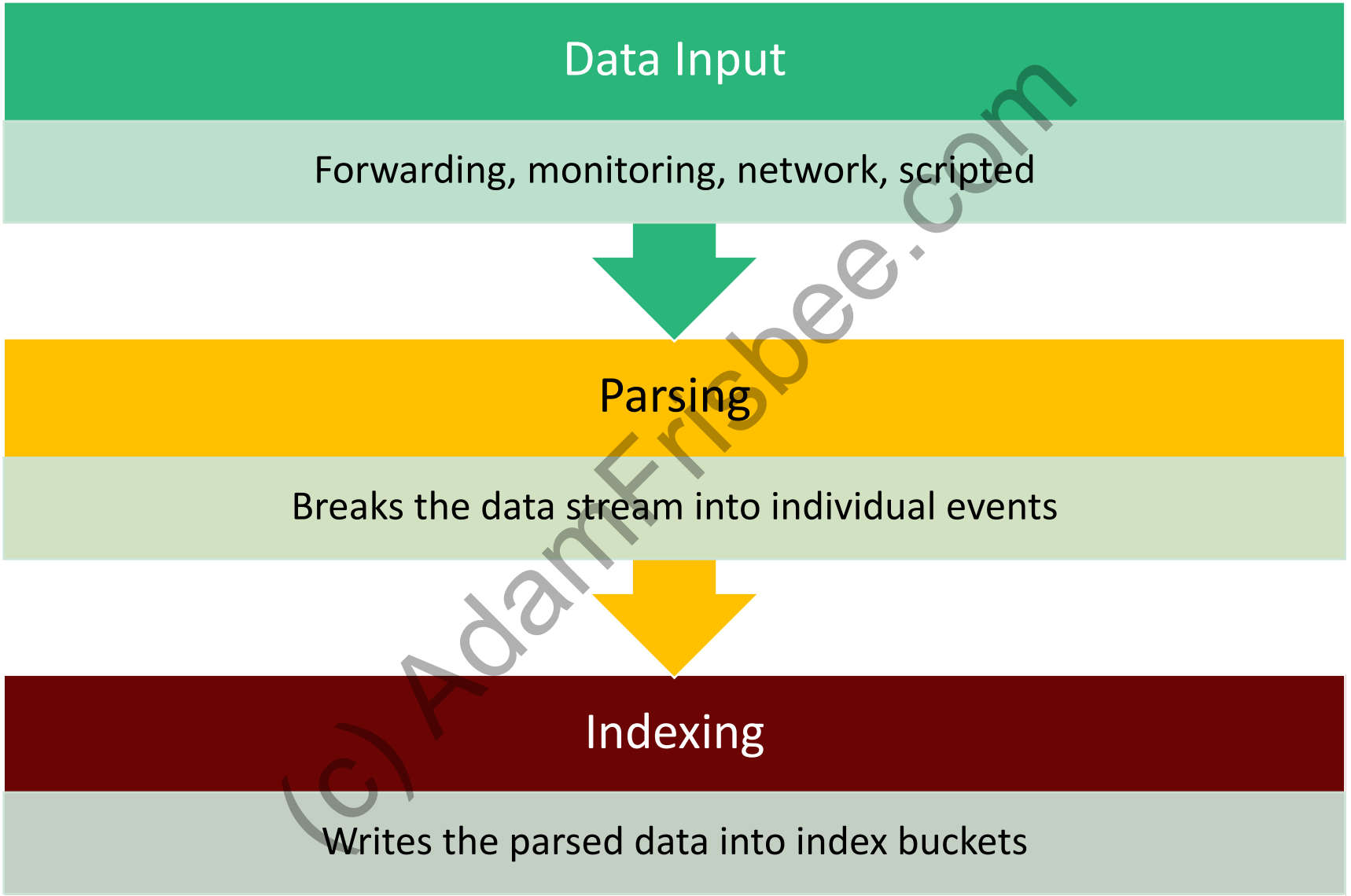
- Configure a distributed search head cluster
  - Configure a deployer
  - Add members to the cluster
  - Check the status of the cluster

Summary

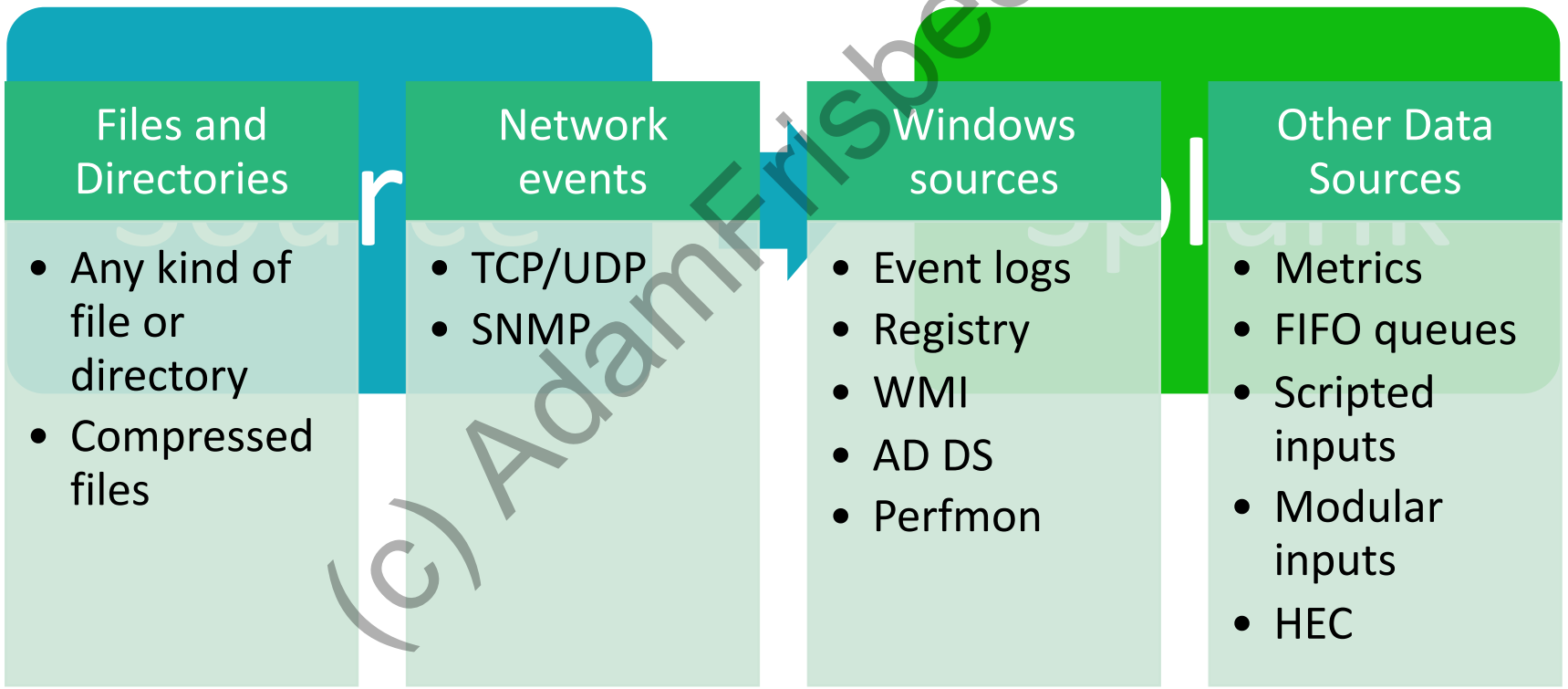
- Discussed Splunk distributed environments
- Scaling Splunk
- Built a search head cluster

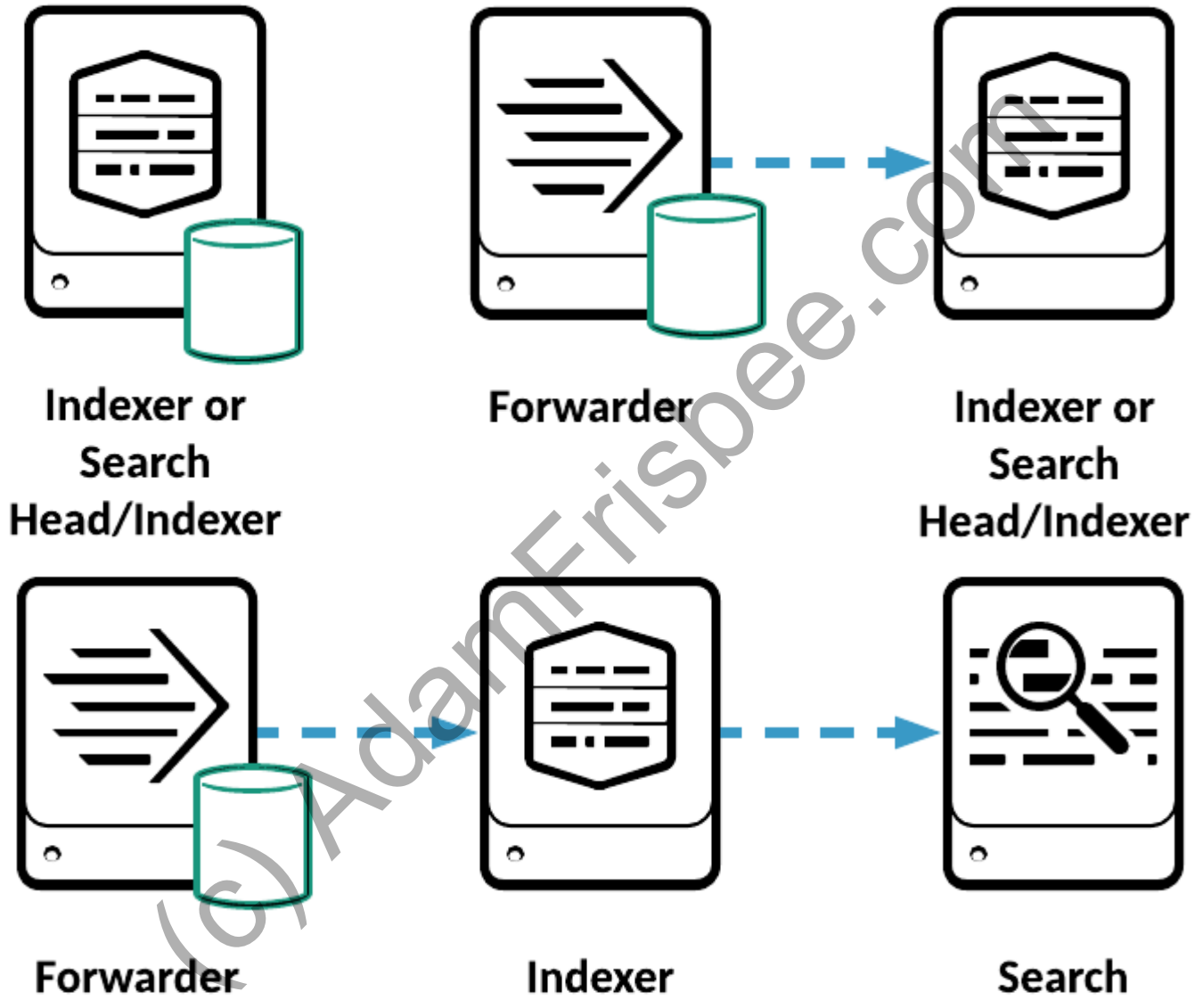
Staging Data

- Understand the three phases of the Splunk indexing process
- Discuss input options



Input Options





Summary

- Discussed the three phases of the Splunk indexing process
- Explored input options

Configuring Forwarders

- Configure universal forwarders
- Identify additional forwarder options
- Configure heavy forwarders

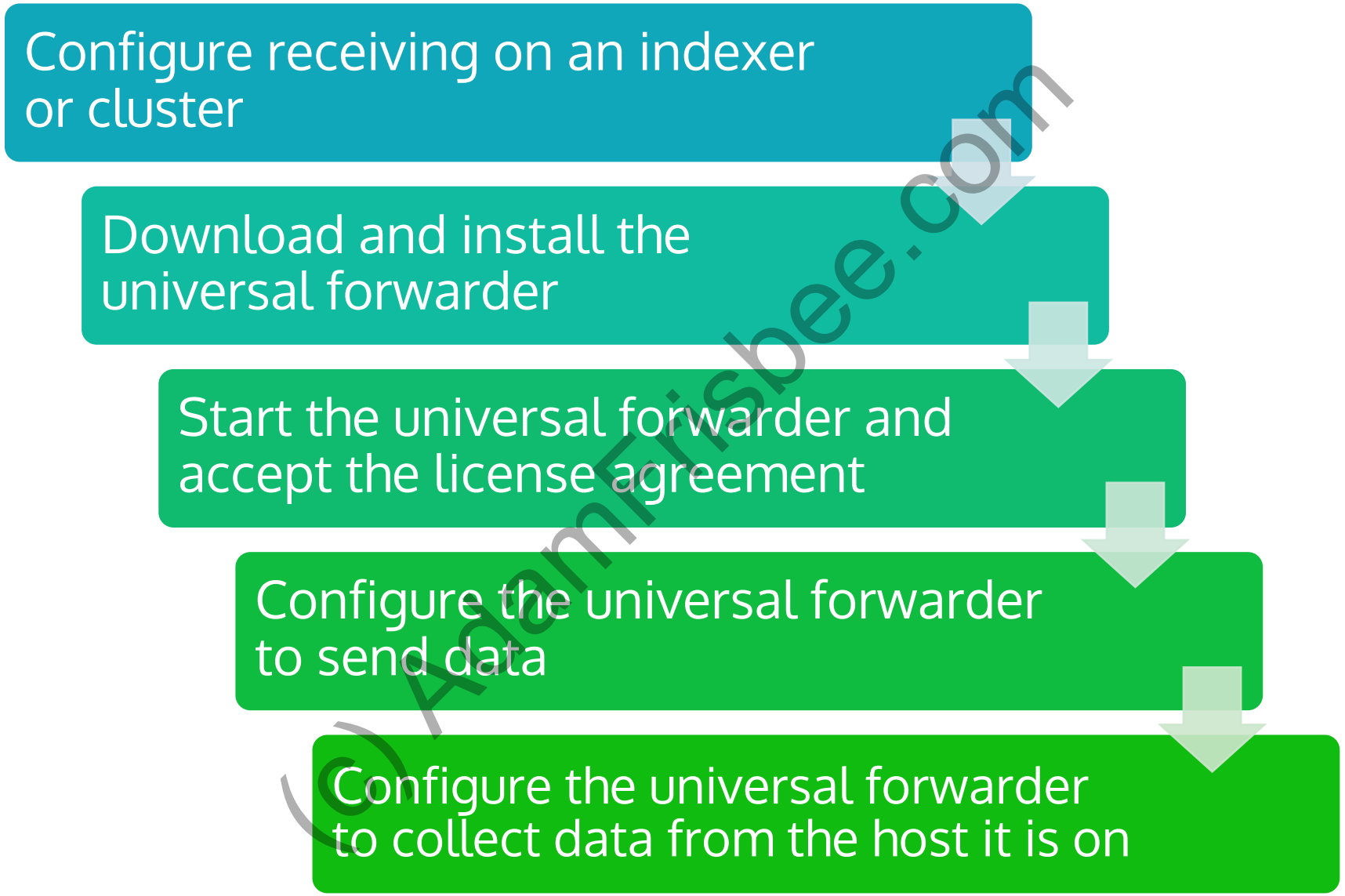
Universal Forwarders

- Only forwards
- Does not search or parse data

Heavy Forwarders

- Full installation of Splunk Enterprise with a forwarder license
- Can parse and route data
- Can index data locally

Configuring Universal Forwarders



GUI (Windows)

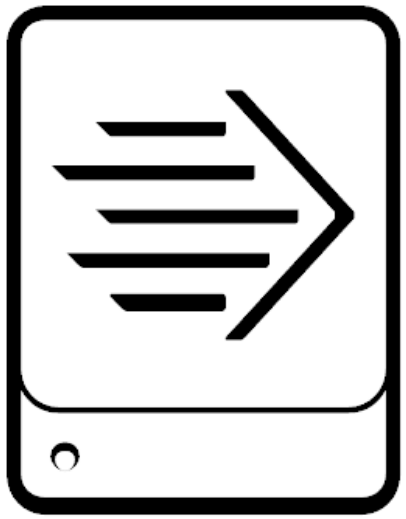
CLI

Configuration files

Deployment server

Configuring Heavy Forwarders

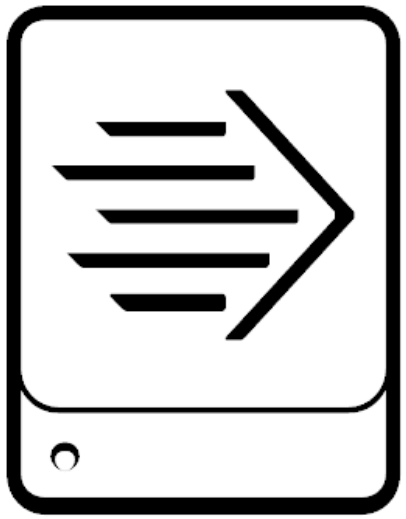
Configuring Heavy Forwarders



Forwarder

- Basic configuration:
  - Install Splunk Enterprise
  - Apply a forwarder license
  - Set up forwarding
    - `Settings > Forwarding and Receiving`
    - `Add new > Configure Forwarding`
  - Or through `outputs.conf`

Configuring Heavy Forwarders

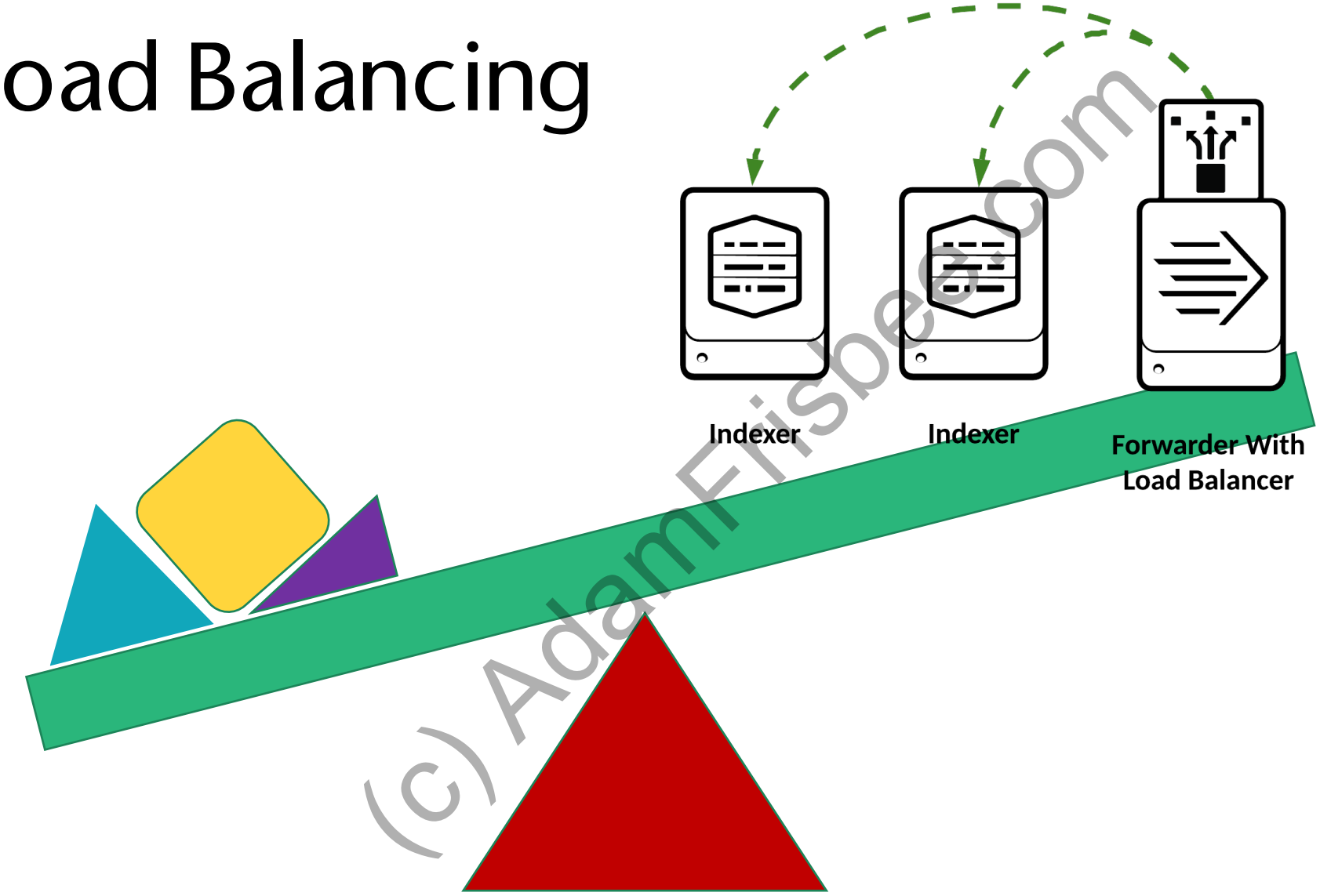


Forwarder

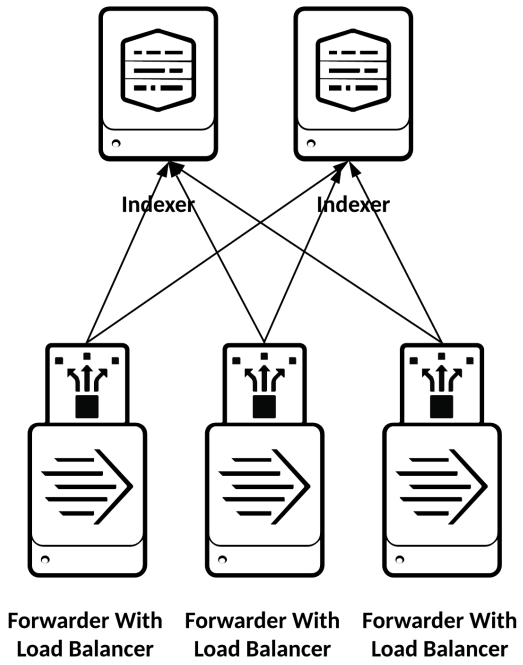
- Advanced configuration
- Heavy forwarders can index data before they forward it
- Through Splunk Web
  - `Settings > Forwarding and Receiving`
  - `Check the box`
- Or through `outputs.conf`

Additional Forwarder Options

Load Balancing

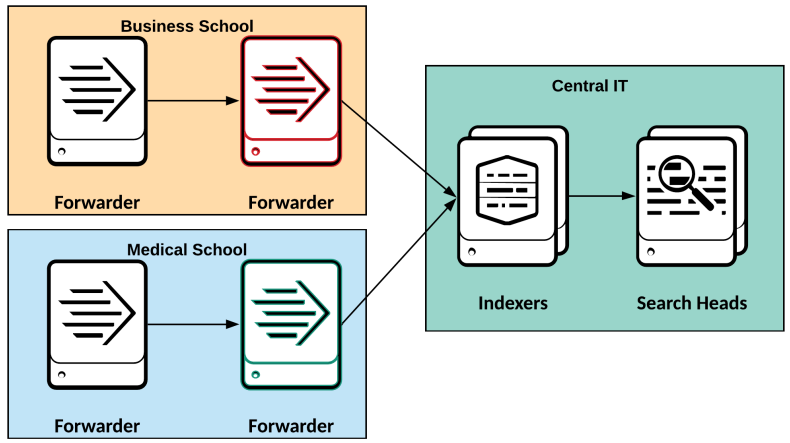


Load Balancing



- Forwarders route data based on specified time or volume interval
- Targets are configured for either DNS or static lists

Intermediate Forwarders

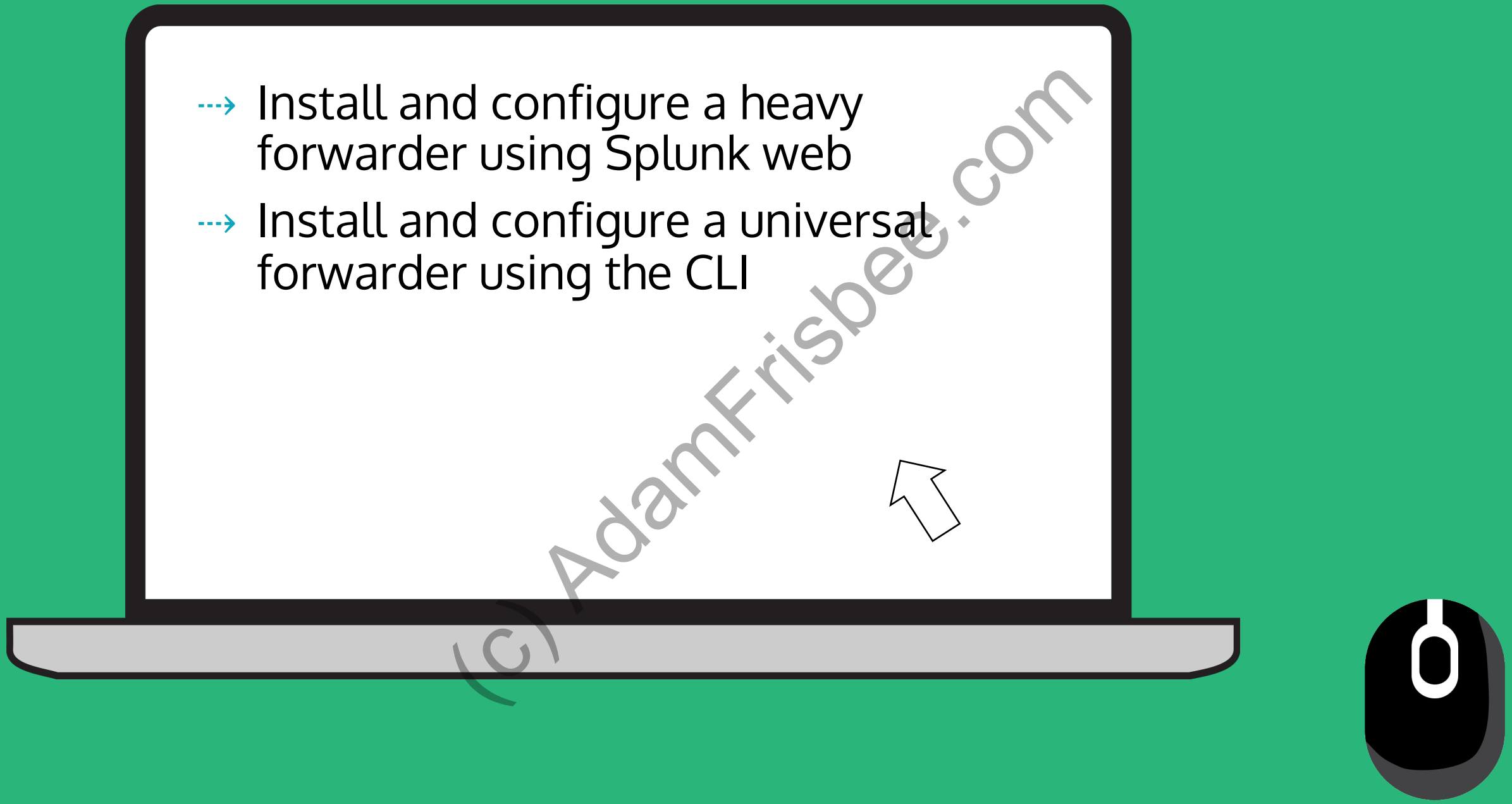


- Forward to other forwarders
- Useful for large, distributed environments
- Set up
  - Configure the forwarder to forward to another forwarder

Multiple Pipeline Sets

- Multiple pipeline sets means the forwarder can process multiple events at the same time
- Useful for machines with more than one core
- Setup
  - Edit /etc/system/local/server.conf
  - Add stanza

```
[general]
parallelIngestionPipelines = 2
```


Install and configure a heavy forwarder using Splunk web
Install and configure a universal forwarder using the CLI
(c) AdamFrisbee.com

Summary

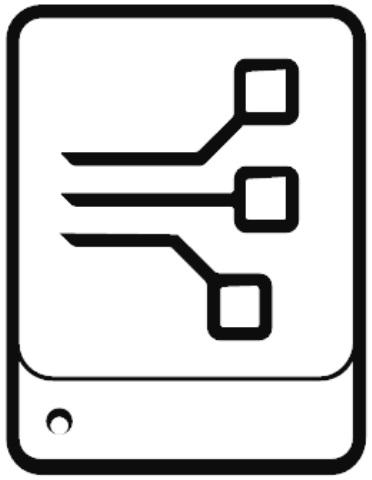
- Install and configure universal forwarders
- Install and configure heavy forwarders
- Identify additional forwarder options

Forwarder Management

- Explain the use of deployment management
- Describe Splunk deployment servers
- Manage forwarders using deployment apps
- Configure deployment clients
- Configure client groups
- Monitor forwarder management activities

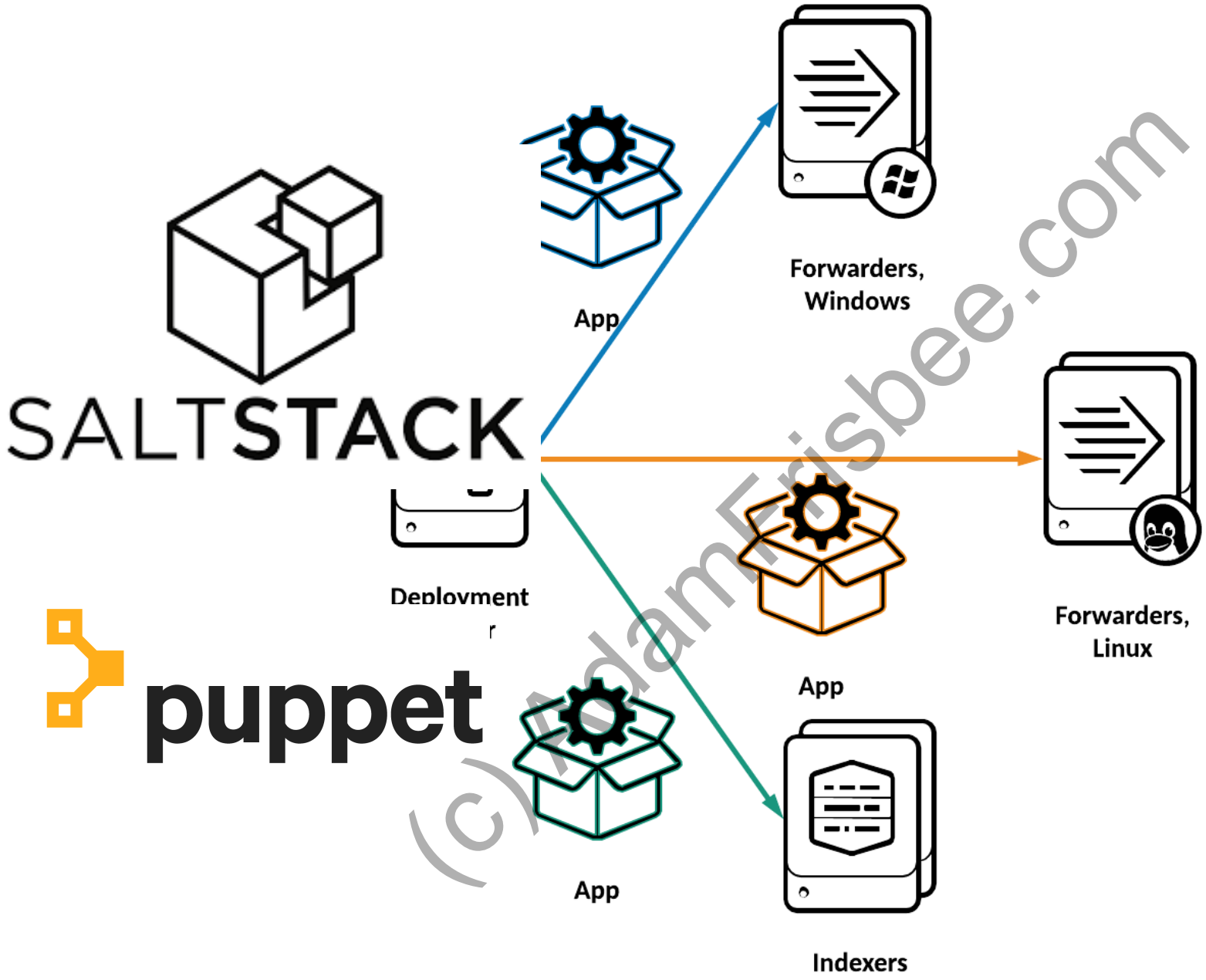
Deployment Servers

Deployment Servers



Deployment Server

- Group Splunk components by common characteristics
- Distribute content based on those groups
  - Windows server group
  - Linux server group
  - Email group

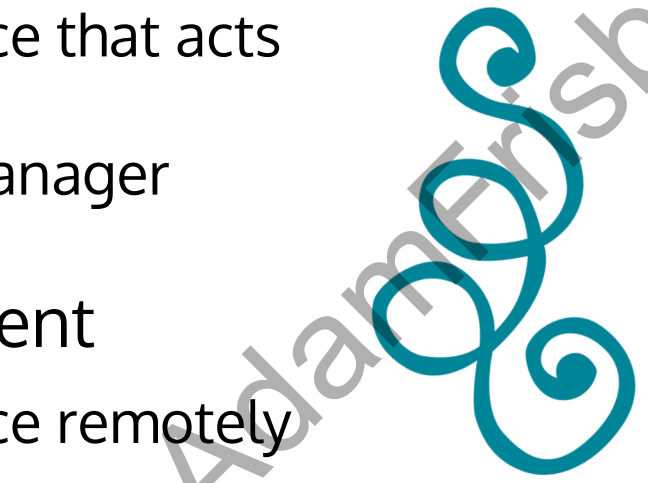


When not to use a Deployment Server

- Indexer clusters
  - You can use it to update the master node only
  - The master node then handles the cluster configuration
- Search head clusters
  - Use the search head cluster deployer

Pieces of Deployment Server Architecture

- Deployment server
  - A Splunk instance that acts as a centralized configuration manager
- Deployment client
  - A Splunk instance remotely configured by a deployment server
- Deployment app
  - a set of content (including configuration files) maintained on the deployment server
- Server class
  - Groups of Splunk instances that receive content from deployment servers



Each deployment client calls home periodically

The deployment server determines the set of deployment apps for the client, based on which **server classes** the client belongs to

The deployment server gives the client the list of apps that belong to it

The client compares the app info from the deployment server with its own app info, to determine whether there are any new or changed apps that it needs to download

If there are new or updated apps, the deployment client downloads them

Depending on the configuration for a given app, the client might restart itself before the app changes take effect

Deployment Apps

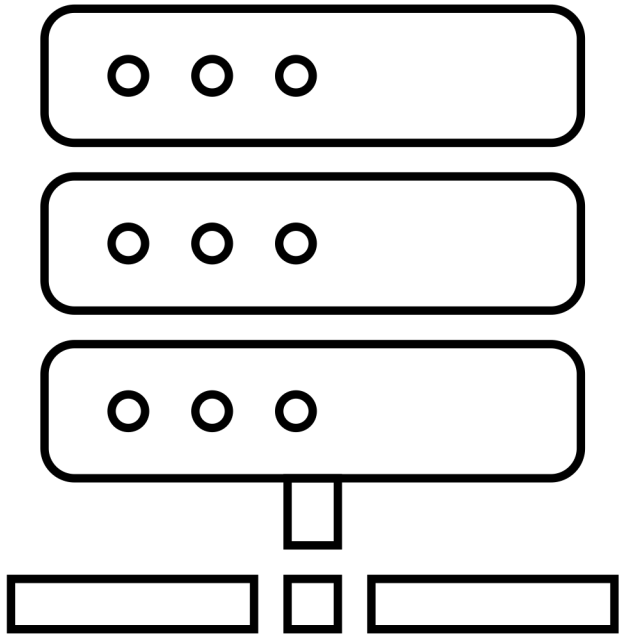
Deployment Apps

- Not necessarily traditional Splunk apps
- Arbitrary content that you want to distribute to deployment clients
  - Apps
  - Configurations
  - Scripts and supporting files

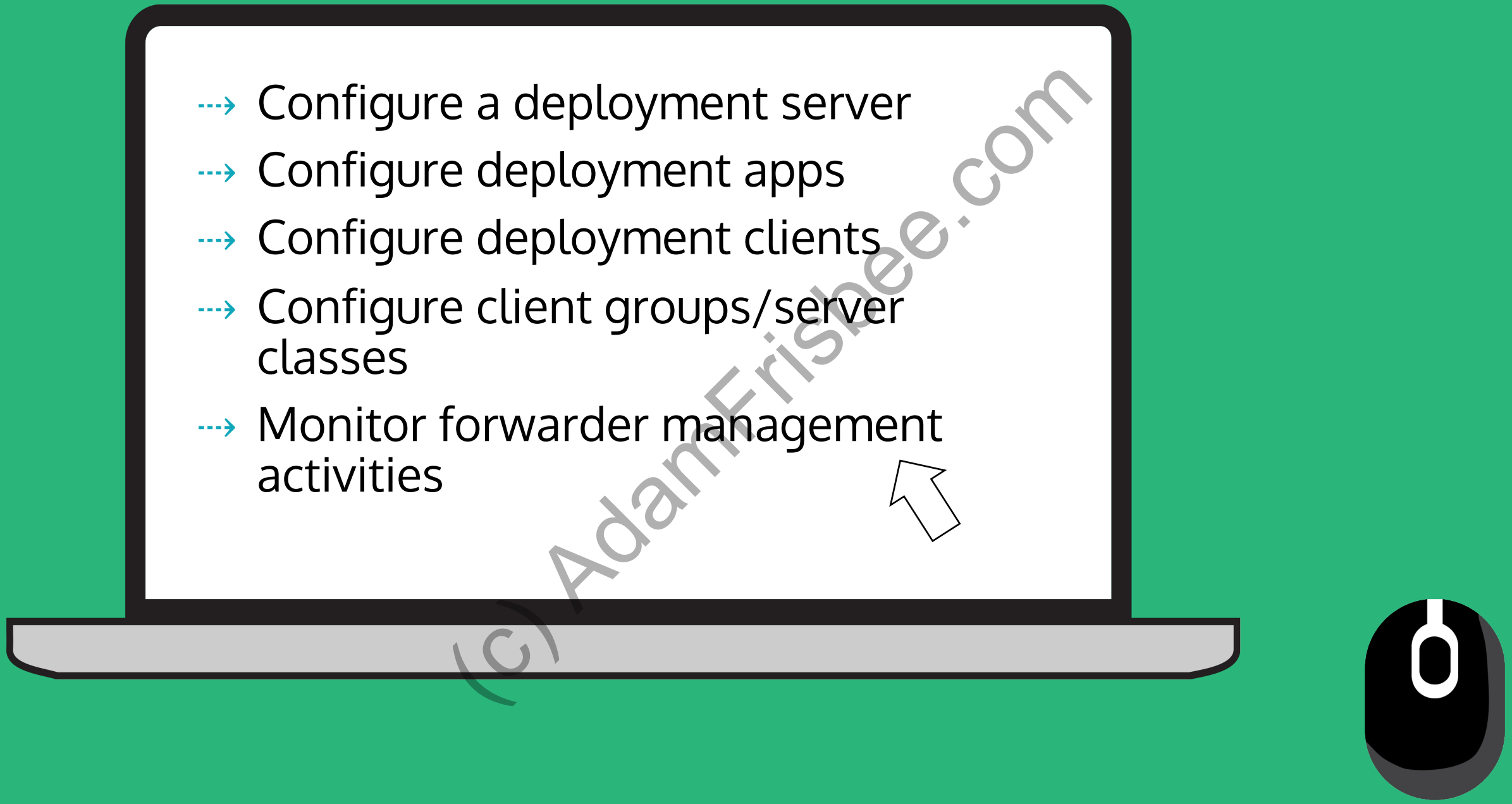
Apps managed with the deployment server can not be later managed without the deployment server

Deployment Clients

Server Classes



- A group of deployment clients that receive particular deployment apps
- Grouping can be based on many different criteria
- A client can belong to more than one class



Summary

- Explain the use of deployment management
- Splunk deployment servers
- Configure a deployment server
- Configure deployment apps
- Configure deployment clients
- Configure client groups/server classes
- Monitor forwarder management activities

Monitor Inputs

- Create file and directory monitor inputs
- Use optional settings for monitor inputs

Monitor

MonitorNoHandle

Upload

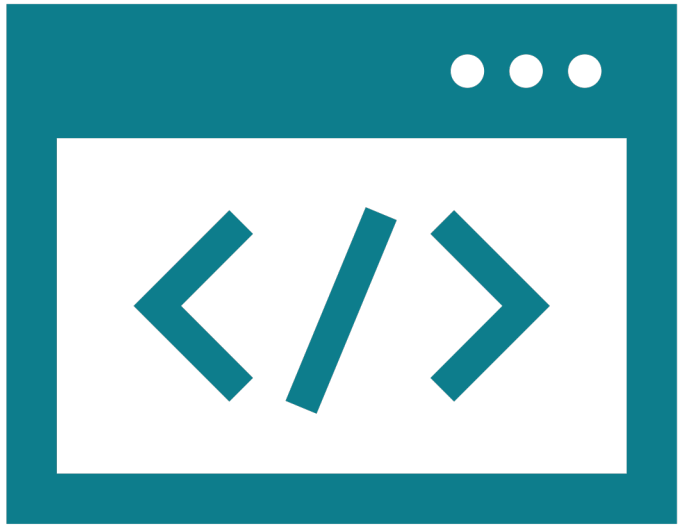
Restart

Compressed

Nonwritable Files

Restrictions

Configure monitoring through `inputs.conf`



- Add a stanza that has the path to the files or directories

```
[monitor:///var/log/messages]
disabled = 0
```

- This will monitor the /var/log/messages directory
- Add other, optional settings

Configure monitoring through CLI

- Use `splunk add monitor`

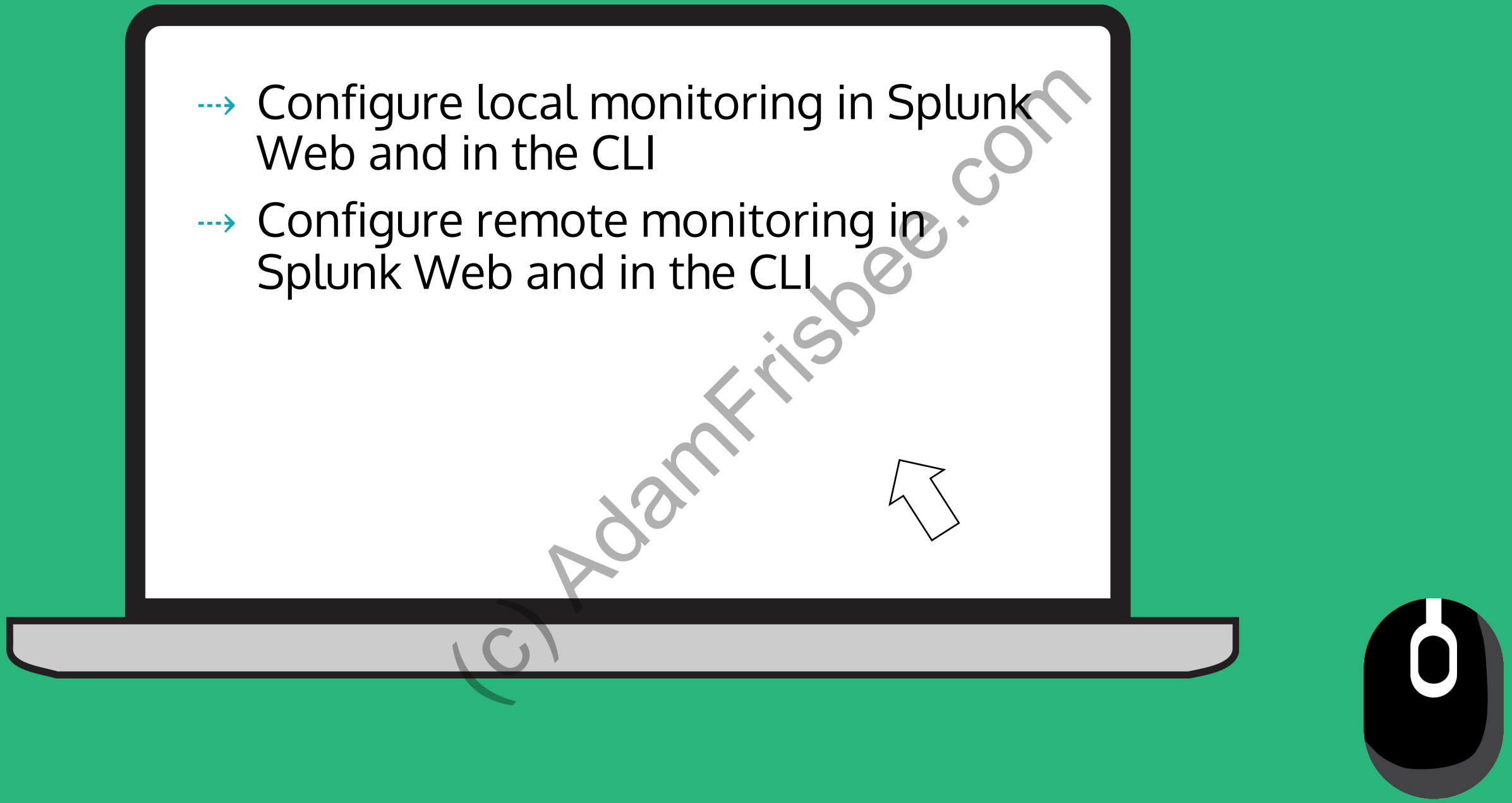
```
./splunk  add monitor /var/log/messages
```

- This will monitor the /var/log/messages directory
- Add other, optional settings

Batch

- Batched files are uploaded, ingested, then **deleted**
- Batch input types can be added to `inputs.conf`

```
[batch://<path>]
move_policy = sinkhole
```


Configure local monitoring in Splunk Web and in the CLI
Configure remote monitoring in Splunk Web and in the CLI
(c) AdamFrisbee.com

Summary

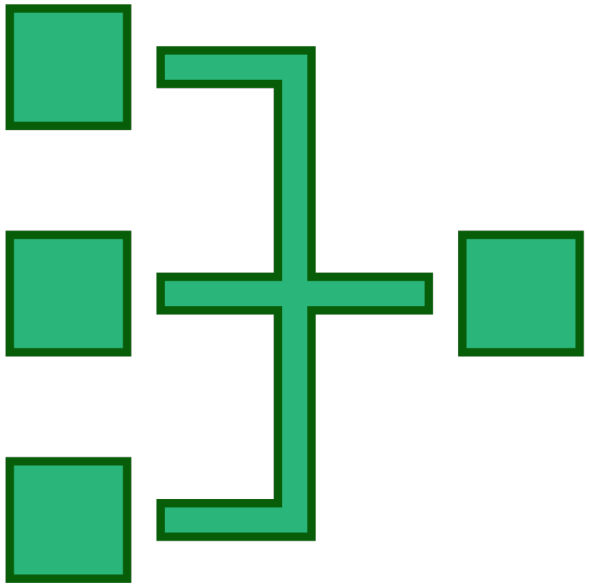
- Created file and directory monitor inputs
- Discussed optional settings for monitor inputs
- Deployed local and remote monitors

Network and Scripted Inputs

- Create network inputs (TCP/UDP)
- Describe optional settings for network inputs
- Create a basic scripted input

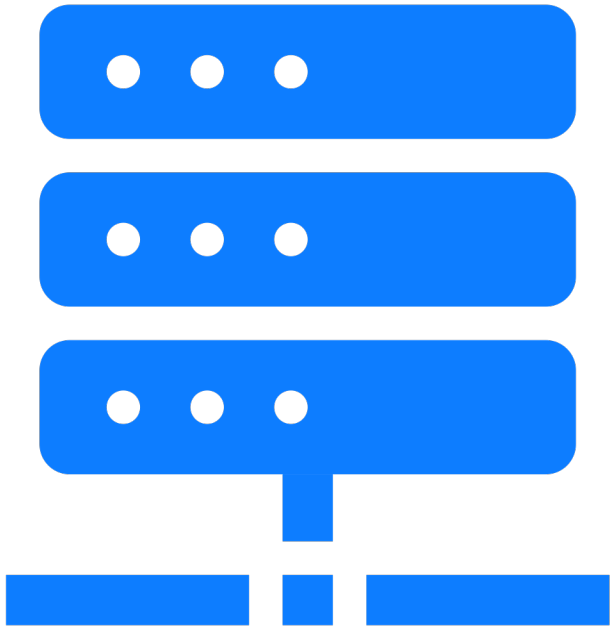
Network Inputs

Network Inputs



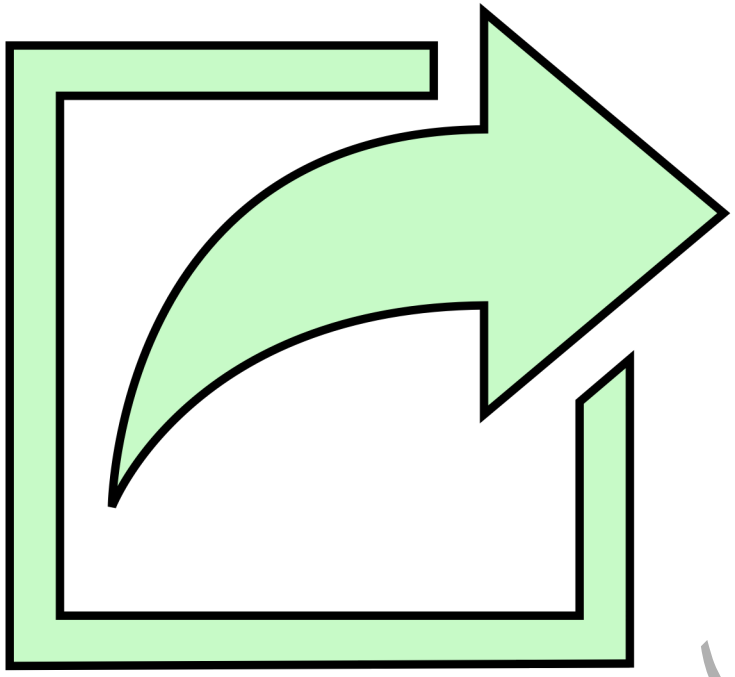
- Configure Splunk to accept network inputs on any port
- Splunk Enterprise consumes any data that arrives on these ports
- TCP is recommended
- Splunk Cloud only accepts inputs from forwarders with SSL certificates

Syslog over UDP



- Splunk can act as a syslog server or syslog message sender
- Splunk recommends configuring an intermediate universal forwarder to listen on the UDP port
  - This is required in Splunk Cloud

SNMP



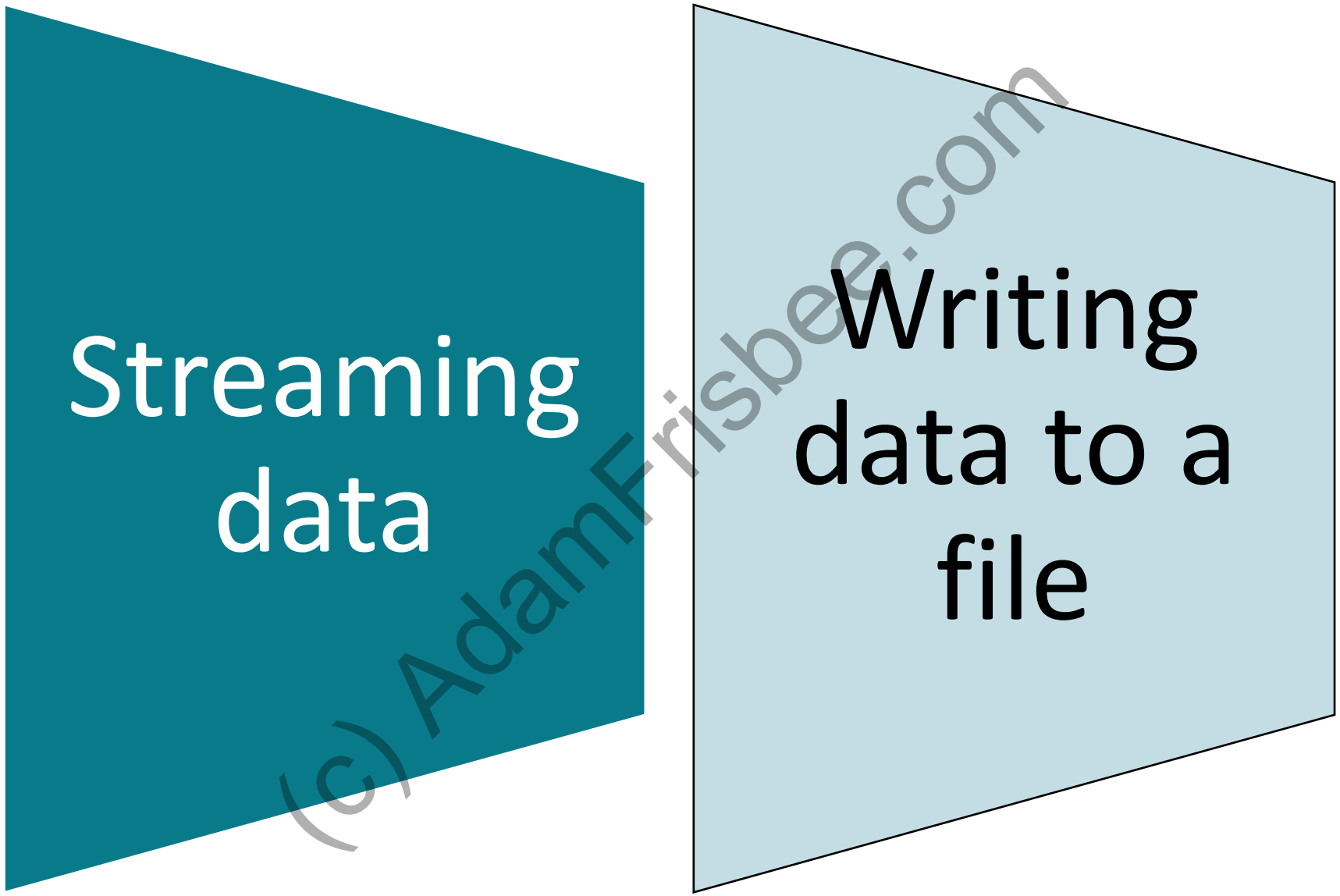
- Simple Network Management Protocol
- Write it to a file on the Splunk Enterprise server, then monitor that file

Scripted Inputs

Scripted Inputs

Scripted Input

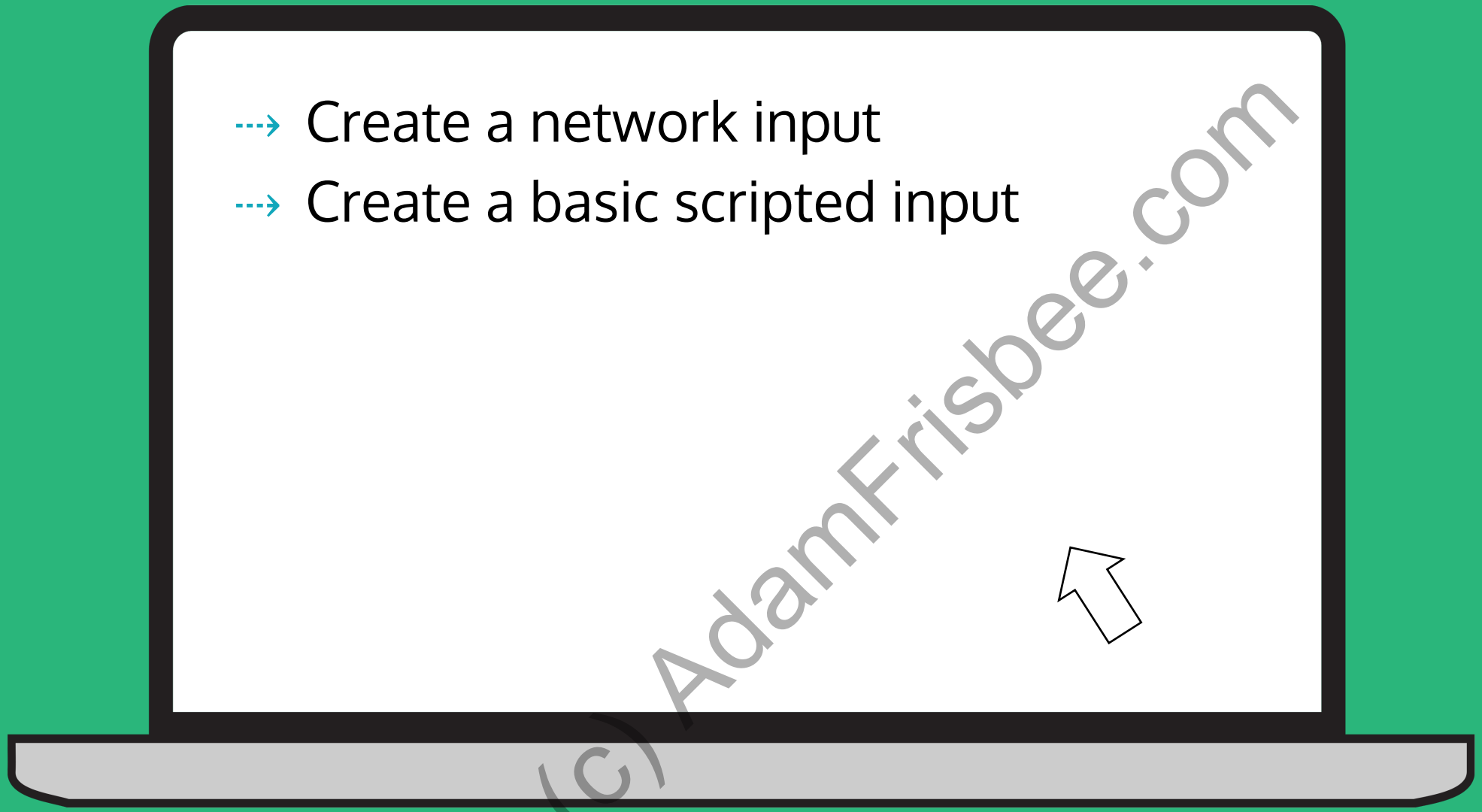
- Allow you to prepare data from a nonstandard source so Splunk properly parses it
  - Especially useful for app builders
- You can use Python, Windows batch files, PowerShell, Bash, and many other scripting tools



Other Scripting Use Cases

- Access data that cannot be sent using TCP or UDP
- Stream data from command-line tools
- Poll a database, web service, or API
- Reformat complex data to more easily parse the data
- Provide special or complex handling
- Scripts that manage passwords and credentials

- Create a network input
- Create a basic scripted input



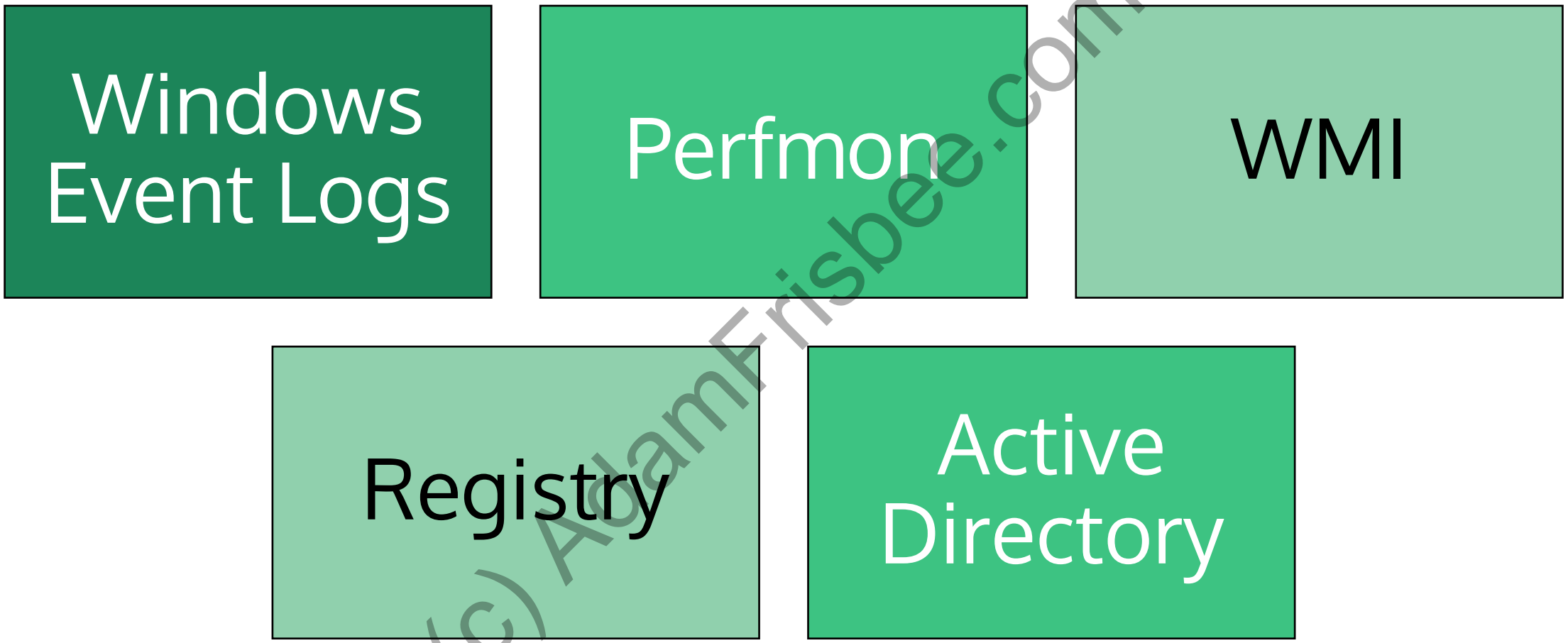
Summary

- Discussed and created network inputs
- Discussed scripted inputs

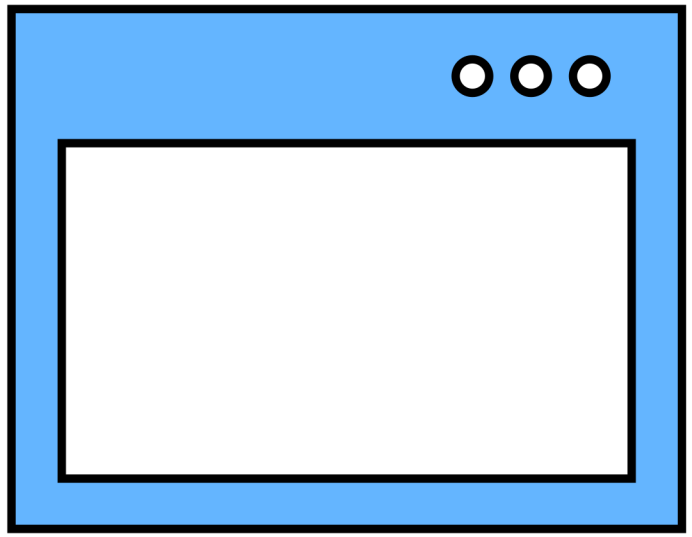
Agentless Inputs

- Identify Windows input types and uses
- Describe the HTTP Event Collector (HEC)

Windows Input Types



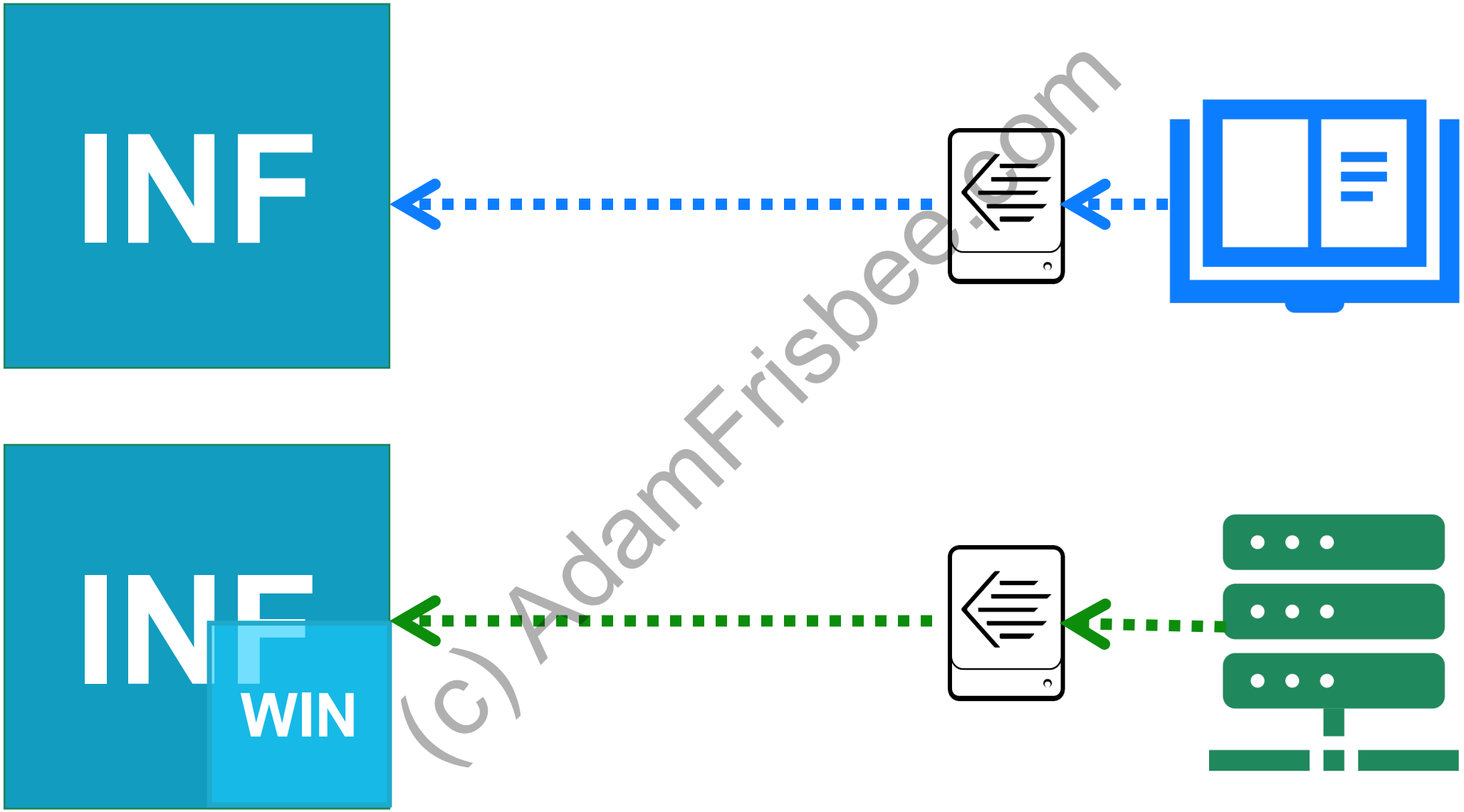
Other Windows Considerations



- Windows Firewall
- Active Directory authentication
- Antivirus software

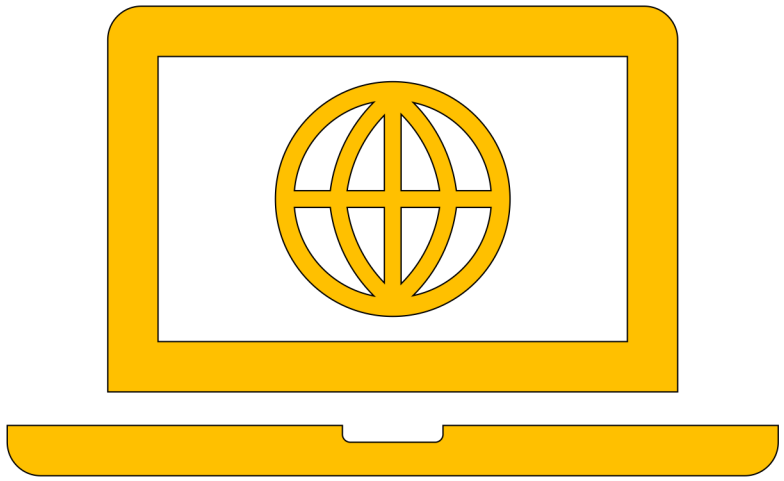
Splunk App for Windows Infrastructure

- Pre-built data inputs, searches, reports, and dashboards for both Windows Servers and Desktops
- "First time run experience"



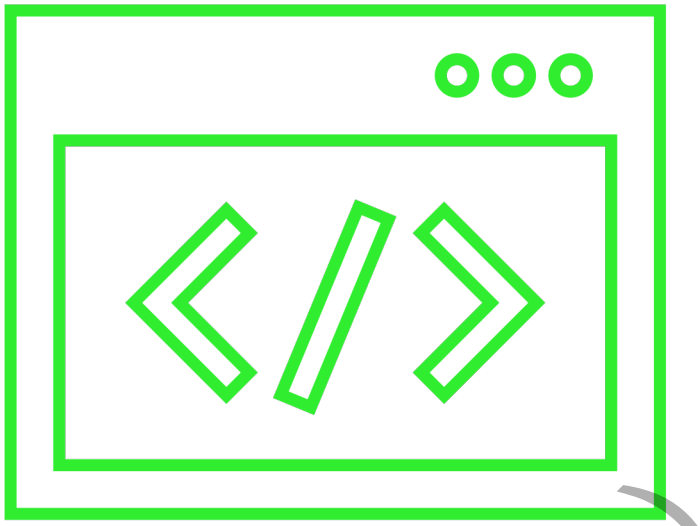
The HTTP Event Collector

The HEC



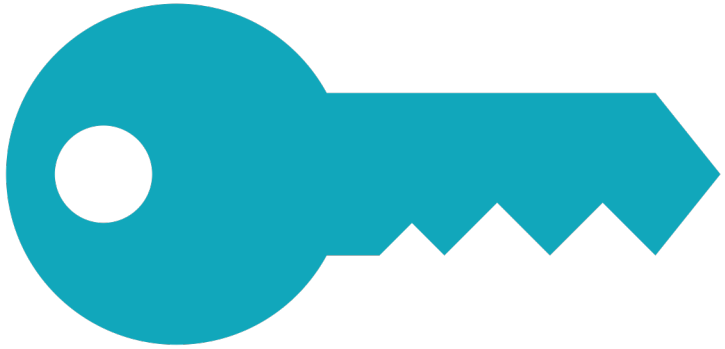
- Allows you to send data directly to Splunk over HTTP(S)
- Especially useful for client-side web application monitoring (SPA)
- Token based authentication

The HEC



- Format for sending data to the HEC
- <protocol>://<host>:<port>/<endpoint>
- Two endpoints
  - /services/collector
  - /services/collector/raw

The HEC



- <protocol>://<host>:<port>/<endpoint> -H “Authorization:Splunk <token>” -d


HEC demo
(c) AdamFrisbee.com

Summary

- Windows input types
- The HTTP Event Collector (HEC)

Fine Tuning Inputs

- Understand the default processing that occurs during the input phase
- Configure input phase options, such as sourcetype fine-tuning and character set encoding

Input Phase

Fine-tuning Inputs

- Setting the sourcetype
  - When you add data, you can specify or create a new sourcetype
- Application context
  - Which app Splunk will write the .conf files to
- Configure the host value
  - The default host value is the DNS name of the machine
  - You can set it to use the IP address, or specify your own hostname
- Index
  - You can use a built-in index, or create a new index

Character Set Encoding

- Splunk defaults to UTF-8 character encoding
- If the source does not use UTF-8, Splunk attempts to convert it
- You can specify a character set to use in `props.conf` using `charset=<string>`

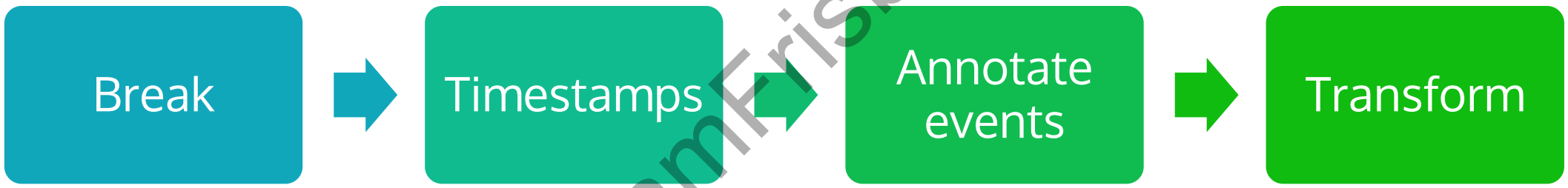
Summary

- Discussed the default processing that occurs during the input phase
- Discussed configuration of input phase options, such as sourcetype fine-tuning and character set encoding

Parsing Phase and Data

- Understand the default processing that occurs during parsing
- Optimize and configure event line breaking
- Explain how timestamps and time zones are extracted or assigned to events
- Use Data Preview to validate event creation during the parsing phase

Parsing Phase



Event Line Breaking

How Splunk Determines Event Boundaries

- LINE_BREAKER regular expression

`([\r\n]+)`

  - Looks for
    - `\r` Escaped character – carriage return
    - `\n` Escaped character – line Feed
    - `[ ]` character set
    - `+` Match one of more of the preceding token(s)

Configure Custom Event Boundaries

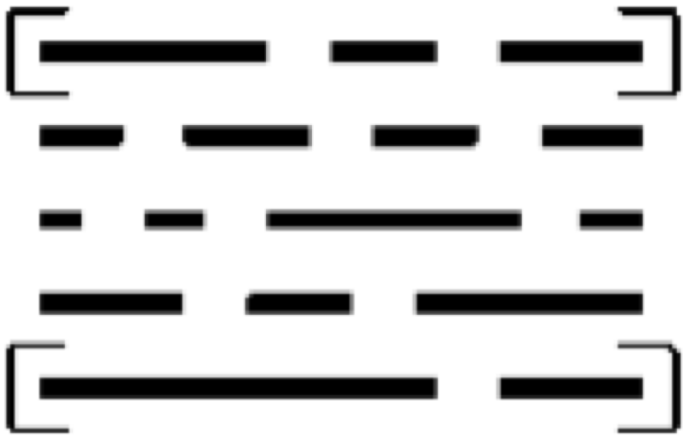
- Set custom rules in `props.conf`

Timestamps and Time Zones

Timestamps

- Timestamps are converted to UNIX time and stored in the `_time` field
- Splunk assumes that any data indexed is in the time zone of the Splunk instance

The `_time` Field



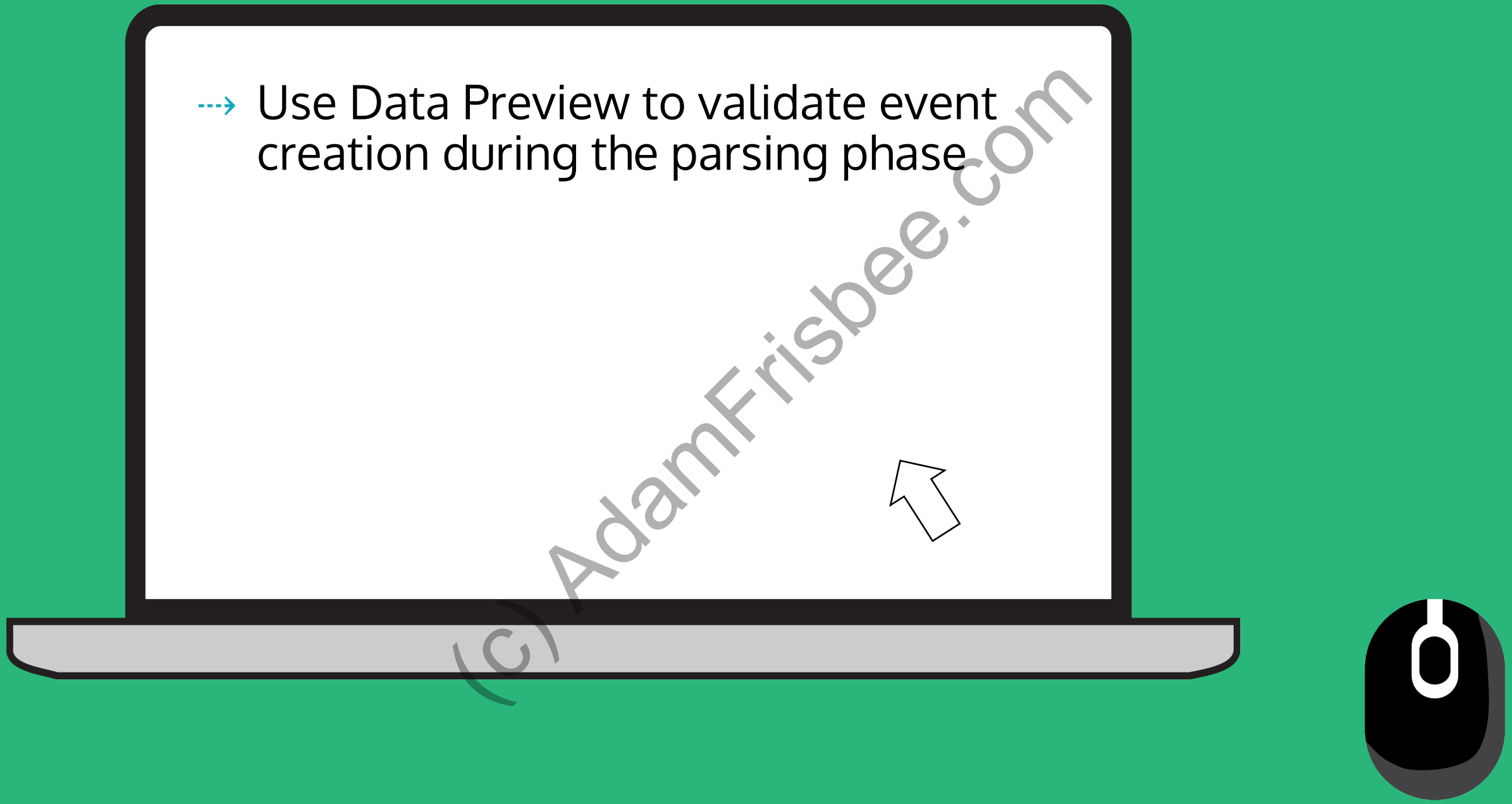
- A default, and essential, field
- Values in the _time field are stored in UNIX time
- In Splunk web, the _time field appears in human-readable format
- Use search commands to manipulate the time format

Looks for timestamp in the event

Looks at the source name or file name

File modification time

Last resort



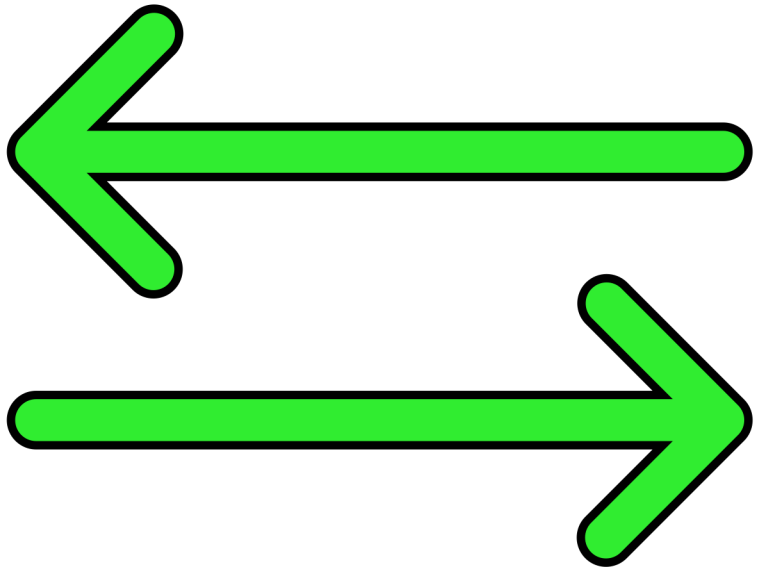
Summary

- Parsing phase processing
- Line breaking
- Timestamps
- Data preview

Manipulating Raw Data

- Explain how data transformations are defined and invoked
- Use transformations with `props.conf` and `transforms.conf` to:
  - Mask or delete raw data as it is being indexed
  - Override sourcetype or host based upon event values
  - Route events to specific indexes based on event content
  - Prevent unwanted events from being indexed
- Use SEDCMD to modify raw data

Why Transform?



- Masking sensitive data before it gets indexed
  - HIPAA data
  - PCI data
  - GDPR data
- Event routing
  - Route specific data to specific indexes

Two Methods

- SEDCMD
  - Uses `inputs.conf` and `props.conf`
  - Easier, if you are familiar with UNIX SED
- Transformations
  - Uses `inputs.conf`, `transforms.conf` and `props.conf`
  - Takes longer to set up

Transformations

Anonymizing Data

- What you need
  - `inputs.conf` that tells Splunk where the data is
  - `transforms.conf` that defines the regular expression and other masking parameters
  - `props.conf` that references the parameters in transforms.conf

inputs.conf

```
[monitor://<path>]
sourcetype = <sourcetype>
```

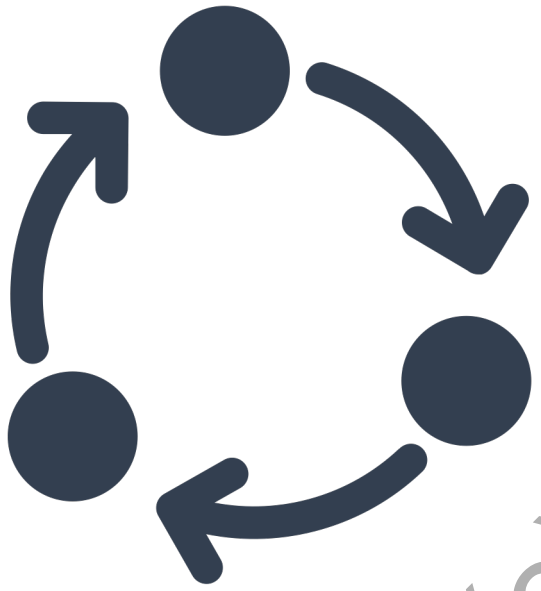
props.conf

```
[<sourcetype from inputs.conf>]
TRANSFORMS-anonymize = <stanza_name
 from transforms>
```

transforms.conf

```
[<stanza_name>]
REGEX = <regular expression>
FORMAT = <format of the mask>
DEST_KEY = _raw
```

Override Sourcetype or Host



- Occurs at parse time
- Works on an indexer or heavy forwarder
- What you need
  - `transforms.conf`
  - `props.conf`

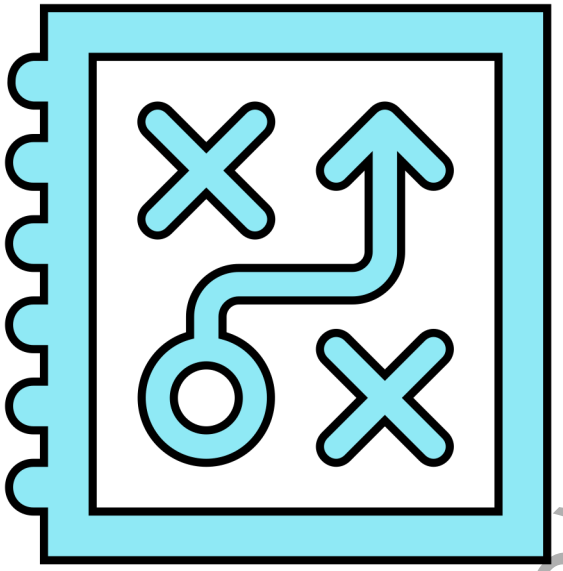
transforms.conf

```
[<stanza name>]
REGEX = <regular expression>
FORMAT = sourcetype::<your_custom_sourcetype_value>
DEST_KEY = MetaData:Sourcetype
```

props.conf

```
[<spec>]
TRANSFORMS-change_sourcetype = <unique stanza name from
      transforms>
```

Route Events to Specific Indexes



- Configured on a heavy forwarder
- What you need
  - `props.conf` to determine routing based on event data
  - `transforms.conf` to specify criteria based on regex
  - `outputs.conf` to define target groups

transforms.conf

```
[<stanza name>]
REGEX = <routing criteria>
FORMAT = <target group1, …>
DEST_KEY = _TCP_Routing
```

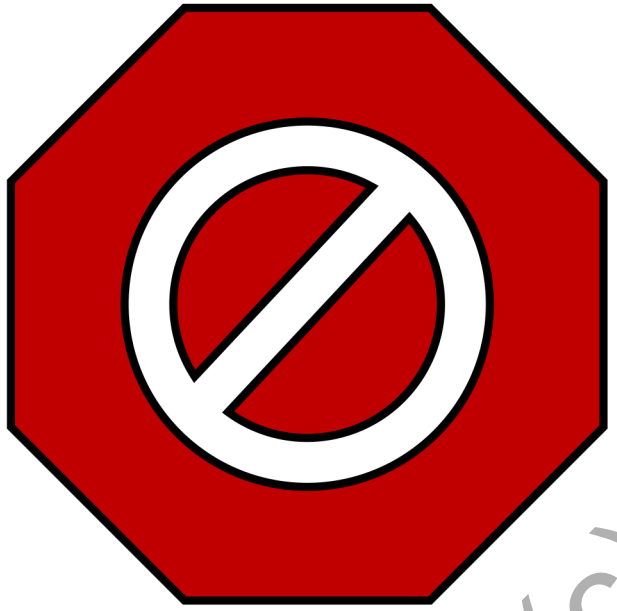
outputs.conf

```
[tcpout:<target_group>]
server=<ip>:<port>
```

props.conf

```
[<spec>]
TRANSFORMS-route = <transforms stanza>
```

Prevent unwanted events from being indexed



- Define a regex to match the events you want to keep
  - Do this in `transforms.conf`
- Send everything else to `nullqueue`
  - Do this in `props.conf`

transforms.conf

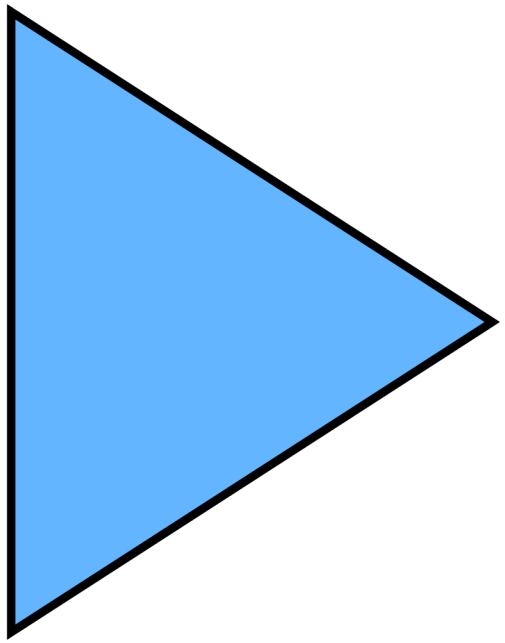
```
[<stanza name>]
REGEX = <regular expression>
FORMAT = nullqueue
DEST_KEY = queue
```

props.conf

```
[<spec>]
TRANSFORMS-accept = <transforms stanza>
```

SEDCMD

What is SEDCMD?



- SED: UNIX/Linux **S**tream **Ed**itor
- Edits data in real time, as it's coming in—streaming
  - Replace strings (s) and substitute characters (y)
- SEDCMD key-value pairs into `props.conf`
- In `inputs.conf`, we tell Splunk where the data is

inputs.conf

```
[monitor://<path>]
sourcetype = <name>
```

props.conf

```
[<sourcetype_from_inputs>]
SEDCMD-<name> = s/<regex>/<replacement>/g
```

What is SEDCMD?

```
22/Oct/1910:04:51:01 StudentID=adfr12 StudentNo=1411536551
```

Props.conf

```
[host::.../gradeserver]
SEDCMD-grades = = s/StudentNo=\d{6}(\d{6})/StudentNo=xxxxx\1/g
```

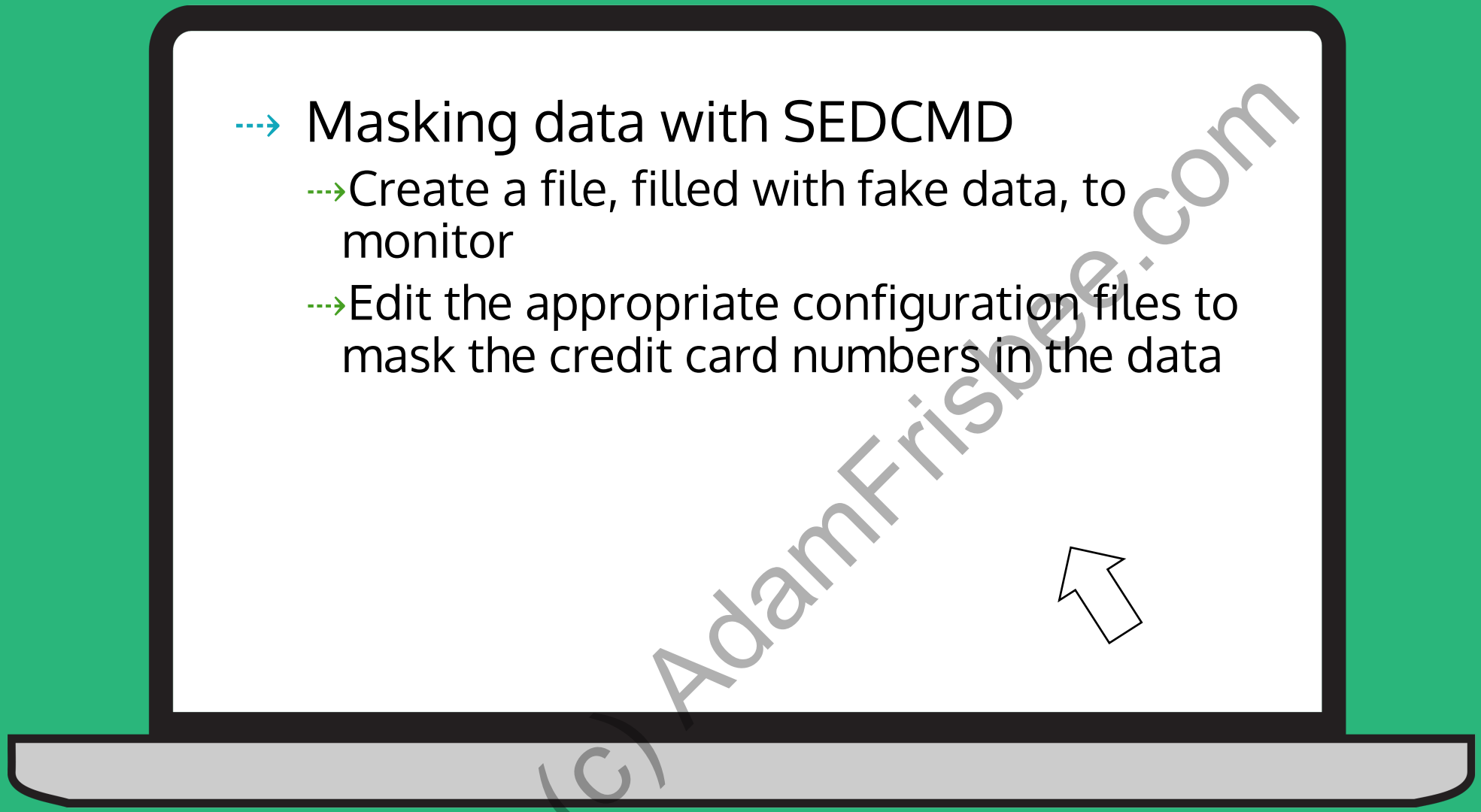
Replace

Regular Expression
Matches the first six digits
after StudentNo=

Replace with six x's

Capture group
and
global search

- Masking data with SEDCMD
  - Create a file, filled with fake data, to monitor
  - Edit the appropriate configuration files to mask the credit card numbers in the data



Summary

- Transforming raw data
  - For masking
  - For overriding
  - For routing
  - For blacklisting
- SEDCMD and SED scripts