Working with Time

Document Usage Guidelines

- Should be used only for enrolled students
- Not meant to be a self-paced document, an instructor is needed
- Lab Exercise slides reference the hands-on lab exercise guide that will be provided by your instructor
- Do not distribute

Course Goals

- Define the **`_time`** field
- Compare time modifier and time picker
- Use index-time based modifiers
- Convert UNIX time to human readable time
- Use the **`timewrap`** command with **`timechart`**
- Describe how time zones are processed

Course Outline

- Searching with Time
- Formatting Time
- Using Time Commands
- Working with Time Zones

Searching with Time

Topic Objectives

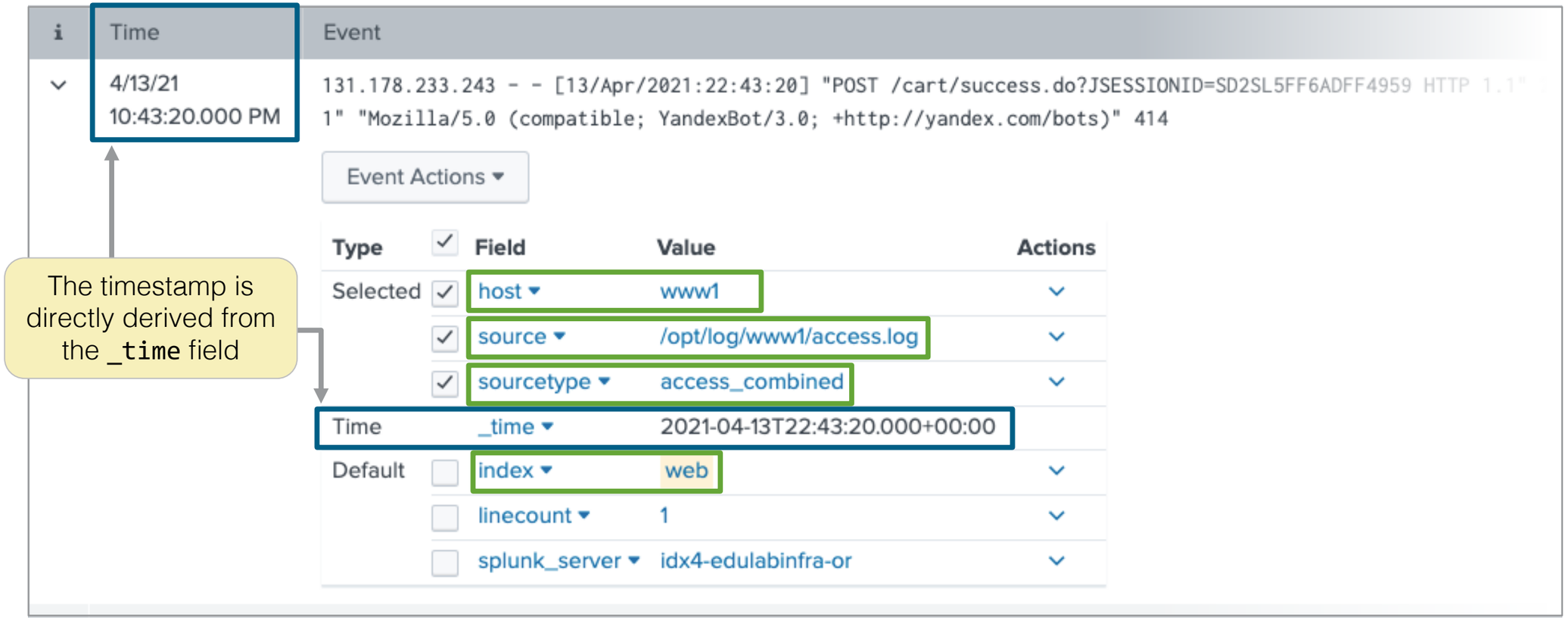
- `_time` field and timestamps
- Viewing and interacting with the Event Timeline
- Using `earliest` and `latest` time modifiers
- Using the `bin` command with `_time`

The `_time` Field

- The `_time` field contains the event's timestamp and is used to create the event timeline in Splunk Web user interface
- The `_time` field is stored with the event in the index prior to search time alongside other metadata fields:
  - Important basic default fields: `host`, `source`, `sourcetype`
  - Other default fields: `_raw`, `_time`, `index`, `timestamp`, `splunk_server`
- `_time` is expressed in UNIX time (epoch time) and translated to human-readable UNIX time during the search operation process
- All events are sorted by time, thus `_time` is the most efficient filter

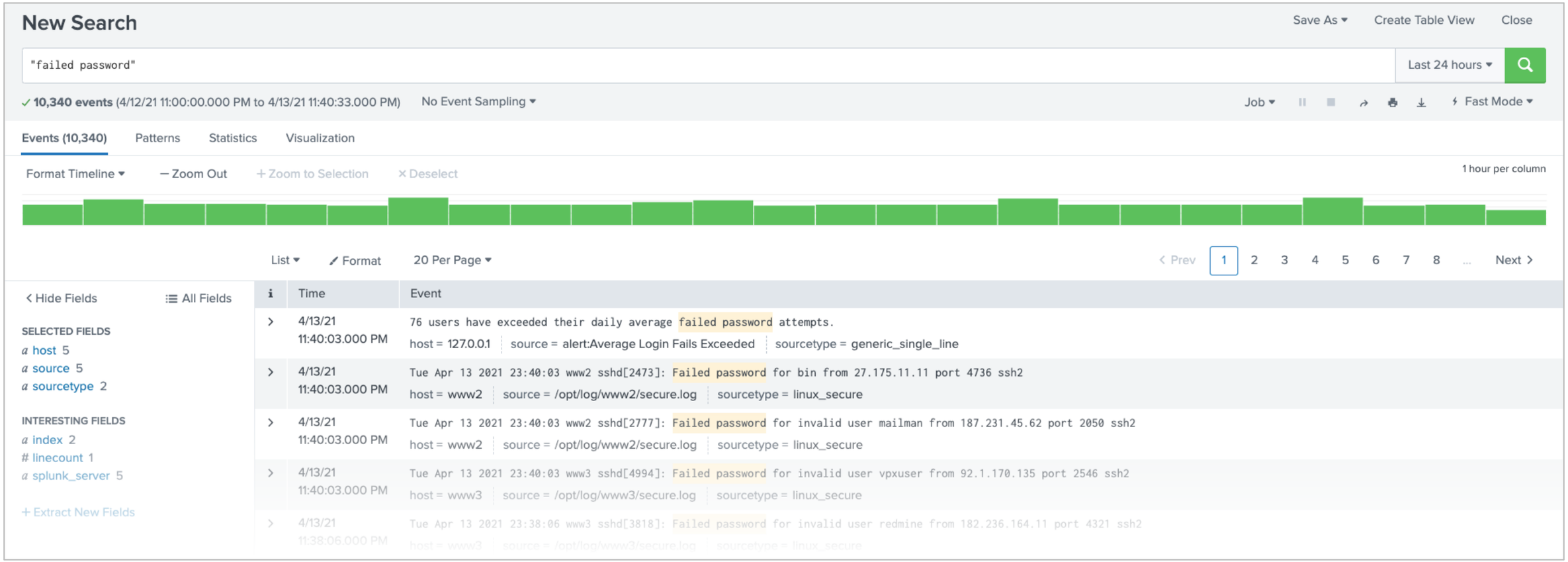
Timestamp

The **_time** (**timestamp**) field and other metadata fields (**host**, **source**, **sourcetype**, and **index**) are assigned to every event



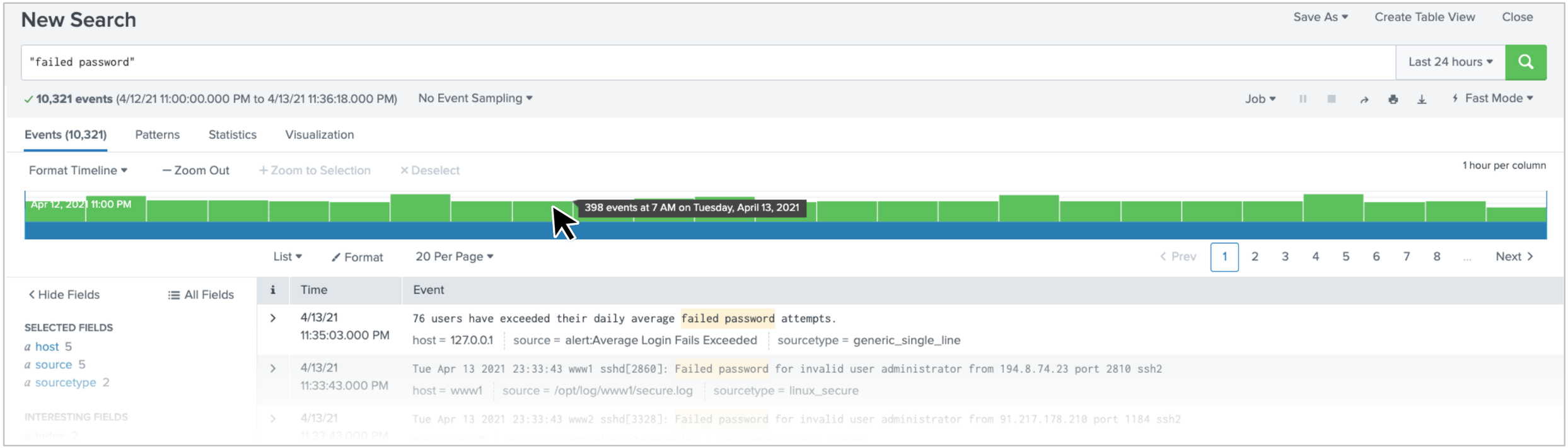
Viewing the Timeline

The timeline shows distribution of events in the time range



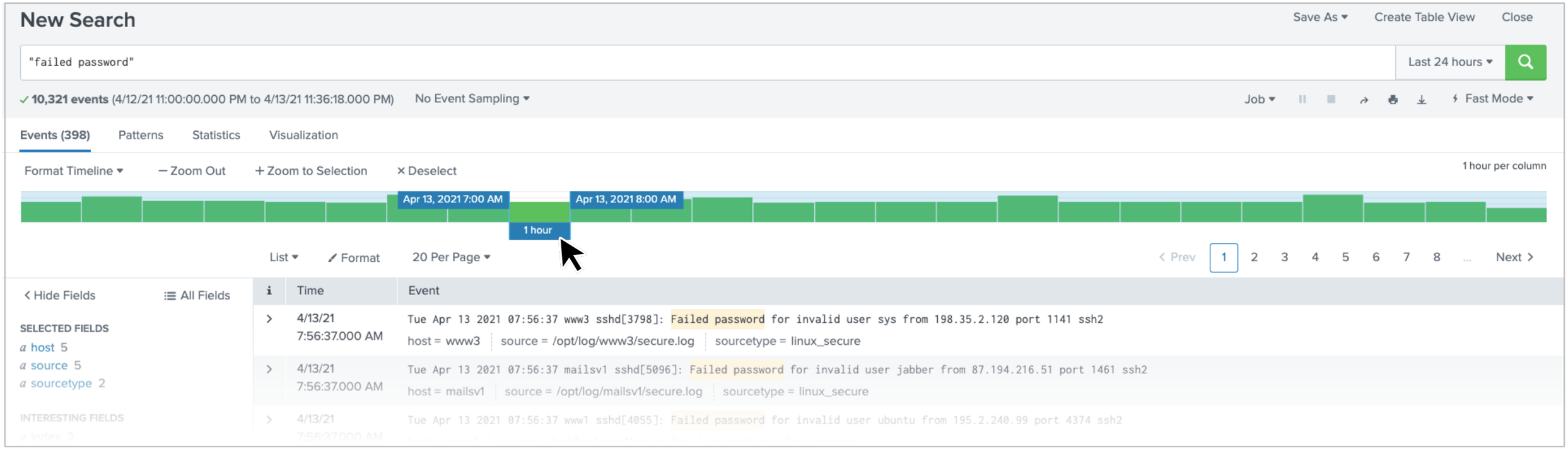
Timeline Mouse Actions

Hover to view the event count for a specific time and date



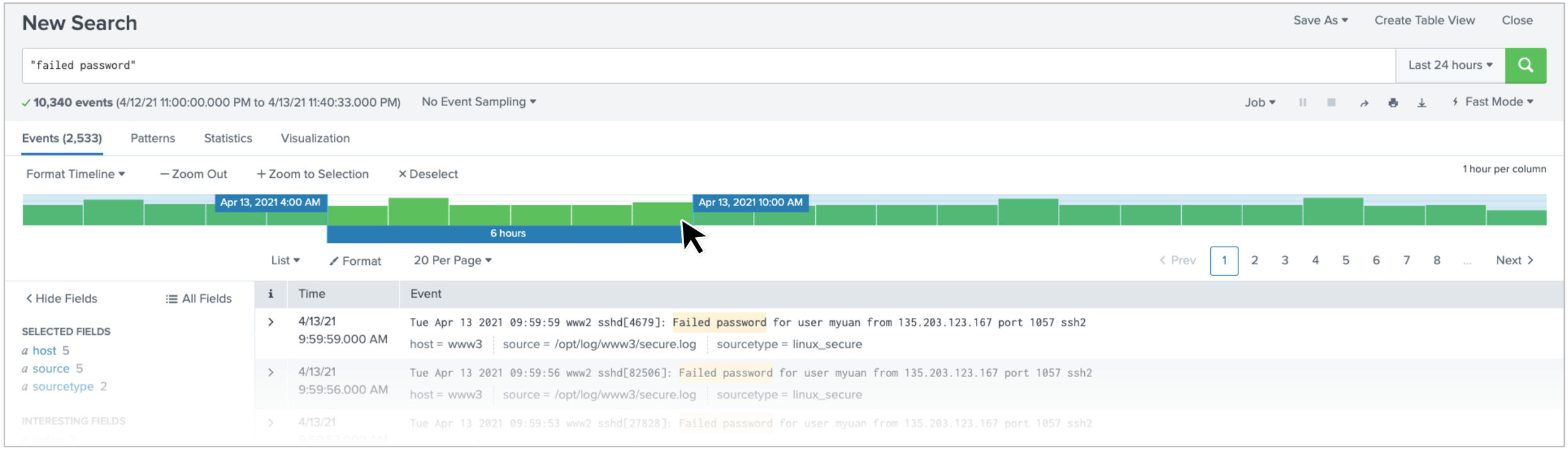
Timeline Mouse Actions (cont.)

Click on a timeline column to filter results for that time period



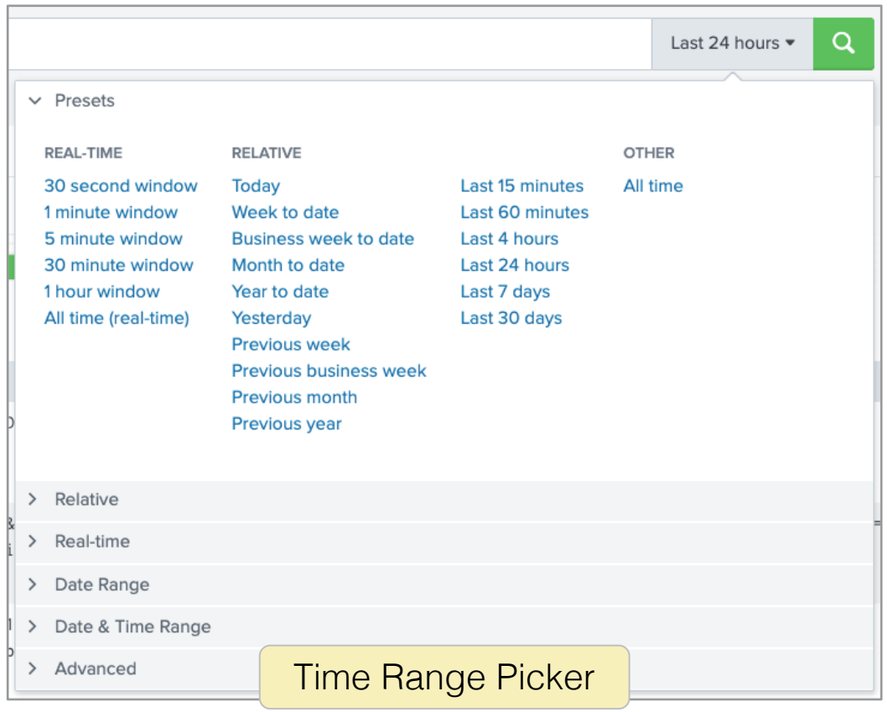
Timeline Mouse Actions (cont.)

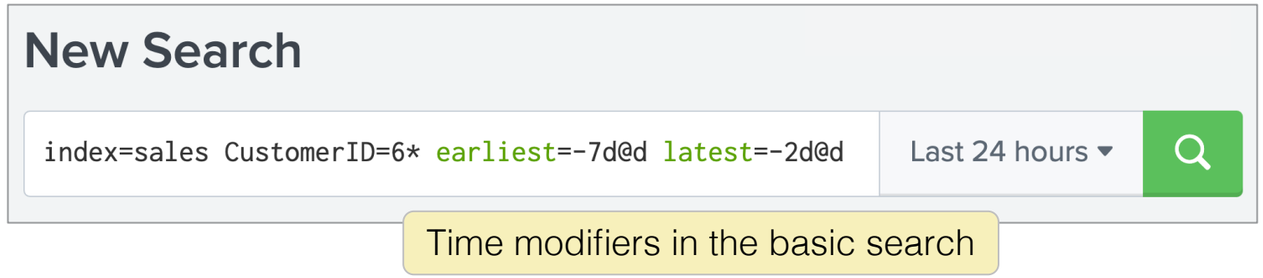
Select a narrow time range by click/dragging across multiple columns



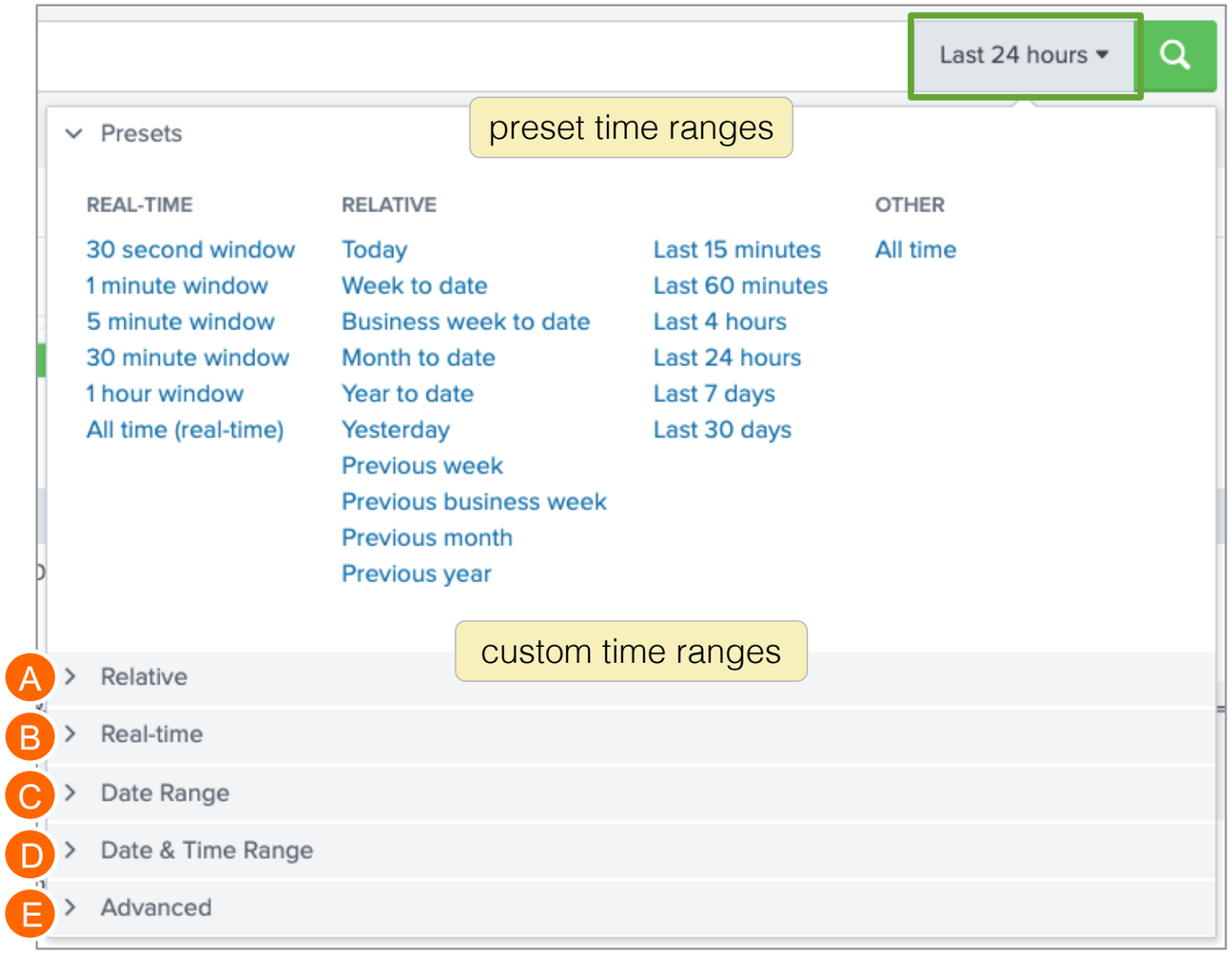
Specifying a Time Range

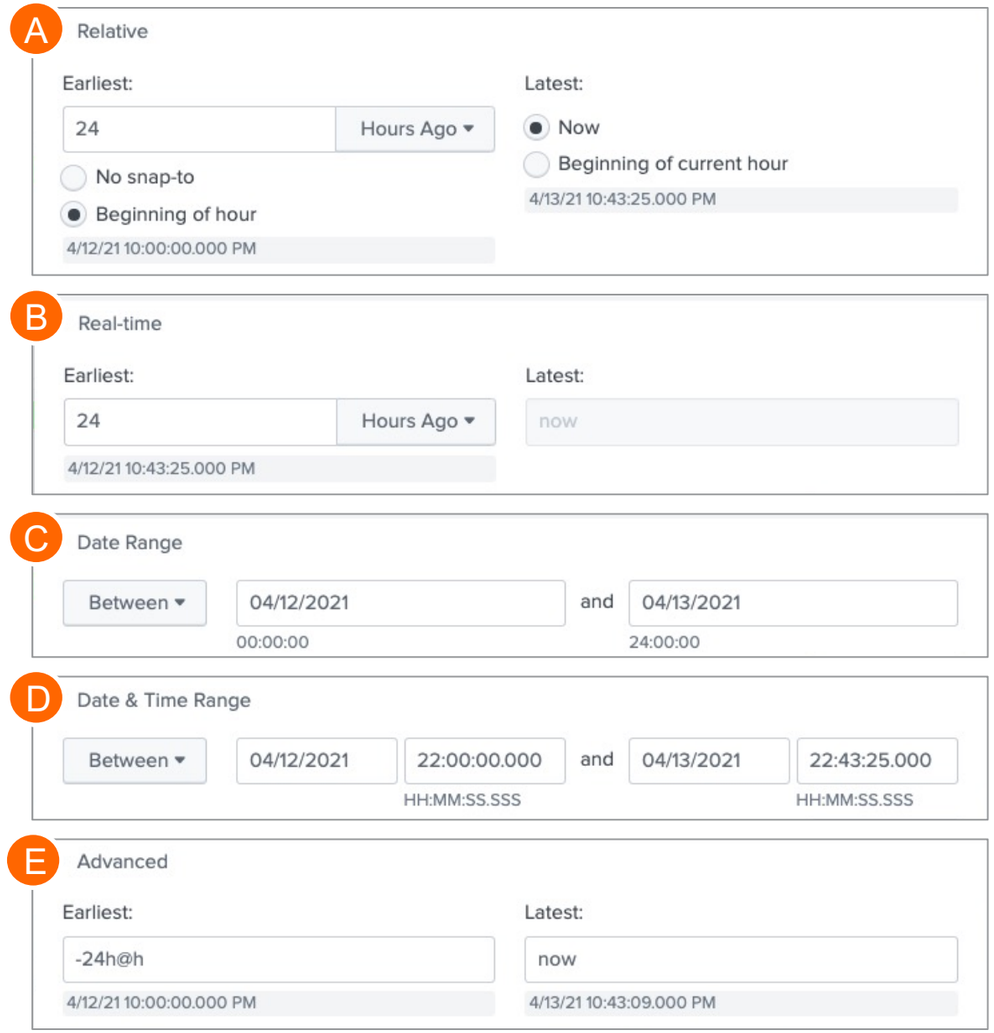
- Always specify a time range before running a search by:
  - Using the Time Range Picker
  - Including time modifiers in the basic search





Using the Time Range Picker





`earliest` and `latest` Time Modifiers

```
... earliest=[+|-]<timeInt><timeUnit>@<timeUnit>
```

```
... latest=[+|-]<timeInt><timeUnit>@<timeUnit>
```

- Include in basic search to override the Time Range Picker
- **`<timeInt><timeUnit>`** is the time amount expressed as an integer and a unit, e.g. **`3h`**
- **`@timeUnit`** "snaps" to the specified time unit
  - Always rounds down, i.e. go backwards through time
  - Can be used to snap to a certain day of the week: **`@w0`** for Sunday, **`@w1`** for Monday, etc.

Note

`earliest` and **`latest`** are rarely used by themselves. If only **`earliest`** is specified, **`latest`** defaults to **`now()`**. If only **`latest`** is specified, all events up to **`latest`** are retrieved.

Using @<timeUnit>

@ symbol "snaps" to the time unit you specify and will always round down to the nearest specified unit

> **Note**
>
> These searches assume the current time is exactly 9:45am on April 1st, 2021.

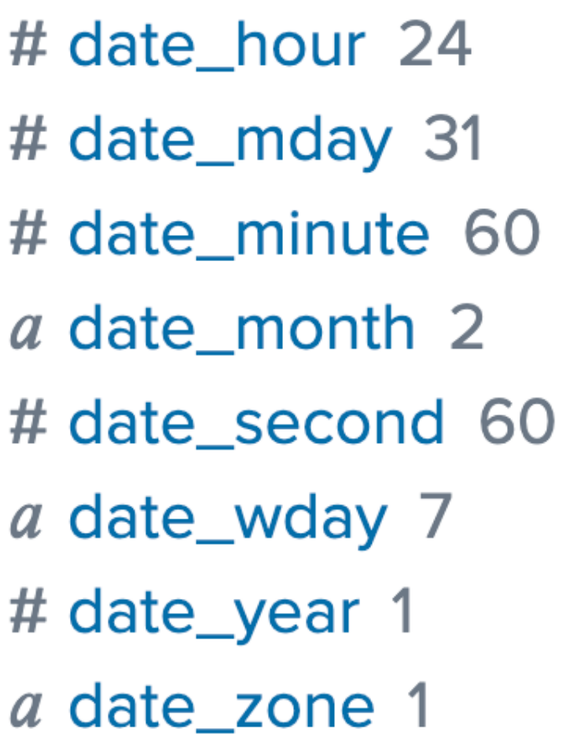
Relative Time Modifiers	Search
`-30m@h`	Looks back to 09:00:00 on April 1st 2021
`earliest=-h@h`	Rounds down to 08:00:00 on April 1st 2021
`earliest=-mon@mon latest=@mon`	Looks for events from 00:00:00 on March 1st 2021 to 00:00:00 on April 1st 2021
`earliest=-7d@d`	Looks for events from 00:00:00 on March 25th (7 days before April 1st) to 09:45:00 on April 1st 2021
`earliest=@d+3h`	Looks for events from 03:00:00 to 09:45:00 on April 1st 2021

<timeUnit> Values List

Relative Time Modifiers	<timeUnit>
Current date & time	now
Second	s, sec, secs, second, seconds
Minute	m, min, minute, minutes
Hour	h, hr, hrs, hour, hours
Day	d, day, days
Week	w, week, weeks
Days of the week	w1 (Monday)…w6 (Saturday), w7 or w0 (Sunday)
Month	mon, month, months
Quarter	q, qtr, qtrs, quarter, quarters
Year	y, yr, yrs, year, years

Default Time Fields

- Events with timestamp information have `date_*` fields
- These fields are generated for events that include date/timestamp in the raw data
- Provides extra information for searching
- These fields do not change based on a user's time zone



Time Modifiers and Time Fields Example

Scenario

A new campaign aimed at early morning sales is ongoing. Display early morning retail sales for 2-5 am for the previous two days.

```
index=sales sourcetype=vendor_sales                    B
A earliest=-2d@d latest=@d date_hour>=2 AND date_hour<5
  | bin span=1h _time
  | stats sum(price) as "Hourly Sales" by _time
C | eval Hour = strftime(_time, "%b %d, %I %p")
  | table Hour, "Hourly Sales"
```

Hour	Hourly Sales
Apr 11, 02 AM	702.64
Apr 11, 03 AM	817.62
Apr 11, 04 AM	817.62
Apr 12, 02 AM	609.67
Apr 12, 03 AM	881.58
Apr 12, 04 AM	770.64

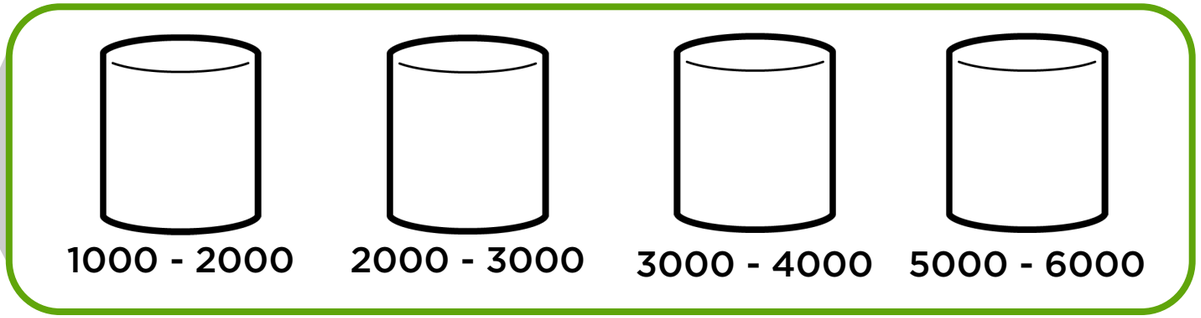
- A The search looks for events from the last 2 days, excluding today
- B Splunk retrieves events from 2am to 5am with no timezone adjustment
- C `strftime` function and time zones are discussed in the following topics

`bin` Command

```
...| bin <field> [span=<int>[<timescale>]] [bins=<int>] [as <newfield>]
```

- Puts numerical values into discrete sets, or bins
- Adjusts values so all events in a bin share the same **`<field>`** value
- Set the size for each bin with the **`span`** option
- Specify a maximum number of bins with the **`bins`** option

```
...| bin <field> span=1000
```



`bin` Command Example 1

Scenario

An analyst in BizOps wants a list of products grouped by revenue range over the last 24 hours.

```
index=web sourcetype=access_combined
| stats sum(price) as totalSales by product_name
| bin totalSales span=100
| stats list(product_name) as product_name by totalSales
| sort totalSales
| eval totalSales = "$".totalSales
```

totalSales	product_name
$2000-2100	Fire Resistance Suit of Provolone
$2400-2500	Puppies vs. Zombies
$3100-3200	Holy Blade of Gouda
$3700-3800	World of Cheese Tee
$5800-5900	Manganiello Bros. Tee
$5900-6000	Curling 2014

`bin` Command Example 1 (cont.)

The **`bins`** option grants more flexibility because it adapts the results to different time ranges

```
index=web sourcetype=access_combined
| stats sum(price) as totalSales by product_name
| bin totalSales bins=10
| stats list(product_name) as product_name by totalSales
| sort totalSales
| eval totalSales = "$".totalSales
```

Last 24 hours

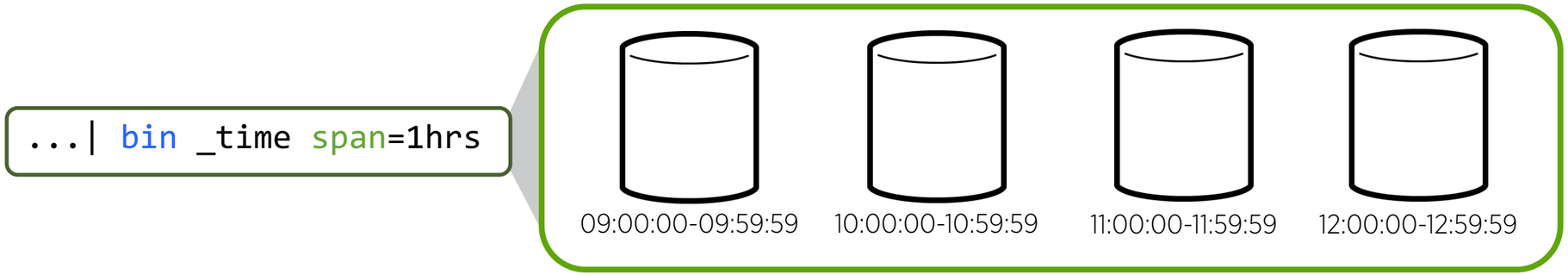
totalSales	product_name
$0-10000	Curling 2014 Fire Resistance Suit of Provolone Holy Blade of Gouda Manganiello Bros. Tee Puppies vs. Zombies World of Cheese Tee
$10000-20000	Benign Space Debris Final Sequel Mediocre Kingdoms SIM Cubicle World of Cheese
$20000-30000	Dream Crusher Manganiello Bros. Orvil the Wolverine

Last 60 minutes

product_name
Benign Space Debris Curling 2014 Dream Crusher Final Sequel Fire Resistance Suit of Provolone Holy Blade of Gouda Manganiello Bros. Manganiello Bros. Tee Mediocre Kingdoms Puppies vs. Zombies SIM Cubicle World of Cheese World of Cheese Tee
Orvil the Wolverine

`bin` Command with the `_time` Field

- When used with **`_time`**, the **`bin`** command can sort result values into bins based on time
- Time values are adjusted so that all items in a bin share the same time value



`bin` Command: `<timescale>` Values

Time Scale	Syntax
<sec>	s | sec | secs | second | seconds
<min>	m | min | mins | minute | minutes
<hr>	h | hr | hrs | hour | hours
<day>	d | day | days
<month>	mon | month | months
<subseconds>	us | ms | cd | ds

bin Command Example 2

Scenario

List the number of times an action was seen on the Sim Cubicle Beta server over the last 10 minutes. Group actions into 2-minute chunks.

```
1 index=games sourcetype=SimCubeBeta Action=* earliest=-10m@m
2 | bin span=2m _time
  | stats count, list(Action) as Action by _time
```

i	Time	Event
1 >	4/20/21 11:40:55.000 PM	20/Apr/2021:23:40:55 , 130.253.37.97 , v2.002B , Use dothis' Action:'Missed Meeting' CurrentStanding:'Man host = sim_cube_server source = /opt/log/SIMlog/simga

i	Time	Event
2 >	4/20/21 11:40:00.000 PM	20/Apr/2021:23:40:55 , 130.253.37.97 , v2.002B , Use dothis' Action:'Missed Meeting' CurrentStanding:'Man host = sim_cube_server source = /opt/log/SIMlog/simga

_time	count	Action
2021-04-20 23:26:00	2	Forgot Password Made Out In Copy Room
2021-04-20 23:28:00	2	Cleaned Desk Slept Under Desk
2021-04-20 23:30:00	2	Made Personal Phone Call Made Photocopy Of Body Part
2021-04-20 23:32:00	3	Missed Meeting Got A Case Of The Mondays Forgot Sandwich In Desk Over Long Weekend
2021-04-20 23:34:00	1	Got A Case Of The Mondays
2021-04-20 23:36:00	1	Got Caught Making Fun Of Boss

Preview: The Next Topics

Formatting Time	Using Time Commands	Working with Time Zones

Time functions:
`now`
`time`
`relative_time`
`strftime`
`strptime`
	`timechart`
	`timewrap`
		Time zones and your data
		
		Using `strftime` to convert timestamps to local timezone

Searching With Time Lab Exercise

Time: 20 minutes

Tasks:

- Test your knowledge of the **`earliest`** and **`latest`** time modifiers
- Use the **`bin`** command to group badge reader events

Formatting Time

Topic Objectives

- Using `eval` date and time functions to format time:
  - `now`
  - `time`
  - `relative_time`
  - `strftime`
  - `strptime`

`eval` Command

```
...| eval <field1>=<expression1>[, <field2>=<expression2>]
```

- Calculates an expression and puts the resulting value into a new or existing field which can be reused in the search pipeline
- Extremely powerful and useful command that supports a vast assortment of functions
- Supports a vast assortment of functions
- Can exist as an expression

`eval` Date and Time Functions

`now()`: returns the time a search was started

```
...| eval field1 = now()
```

`time()`: returns the time an event was processed by **`eval`** command

```
...| eval field1 = time()
```

Date and Time Functions: relative_time

```
...| eval field1 = relative_time(X,Y)
```

- Returns an epoch timestamp relative to a supplied time
- X is a number, representing desired time in epoch seconds
- Y is a relative time specifier
- Relative time specifiers use time unit abbreviations such as:

```
s = seconds    m = minutes    h = hours    d = days    w = week    mon = months    y = year
```

```
...| eval yesterday = relative_time(now(),"-1d@h")
```

_time	yesterday
2021-04-21 16:43:32	1618934400.000000

Date and Time Functions: Format Variables

Format variables are used by the **`strftime`** and **`strptime`** functions discussed in the following slides

Time		
%H	24 hour	00 - 23
%T	24 hour	HMS
%I	12 hour	01 - 12
%M	minute	00 - 59
%p		AM or PM

Days		
%d	Day of month	01 to 31
%w	Weekday	0 to 6
%a	Abbreviated weekday	Sun
%A	Weekday	Sunday
%F	year-month-day	%Y-%m-%d

Months & Years		
%b	Abbreviated month name	Jan
%B	Month name	January
%m	Month number	01 - 12
%Y	Year	2020

```
...| eval yesterday = relative_time(now(),"-1d@h")
   | eval yesterdayString = strftime(yesterday,"%F %H:%M")
```

Previous example converted to a string format

_time	yesterday	yesterdayString
2021-04-21 16:46:21	1618934400.000000	2021-04-20 16:00

Date and Time Functions: `strftime`

```
...| eval field1 = strftime(X,Y)
```

Renders a UNIX timestamp (**X**) as a string based on the format specified by **Y**

```
index=sales sourcetype=vendor_sales
| timechart span=1h sum(price) as h_sales
```

_time	h_sales
2020-10-06 00:00	1839.15
2020-10-06 01:00	2084.15
2020-10-06 02:00	1519.26
2020-10-06 03:00	1624.13
2020-10-06 04:00	1678.24
2020-10-	595.17

Before formatting

```
index=sales sourcetype=vendor_sales
| timechart span=1h sum(price) as h_sales
| eval _time = strftime(_time,"%b %d, %I %p")
```

_time	h_sales
Oct 06, 12 AM	1839.15
Oct 06, 01 AM	2084.15
Oct 06, 02 AM	1519.26
Oct 06, 03 AM	1624.13
Oct 06, 04 AM	1678.24
Oct 06,	595.17

After formatting

 | 25 January 2023

Date and Time Functions: `strptime`

```
...| eval field1 = strptime(X,Y)
```

Converts a time represented by a string (**X**) to a UNIX timestamp based on formatting determined by **Y**

```
index=systems sourcetype=system_info asctime=*
| eval NewAsctime = strptime(asctime, "%Y-%m-%d %H:%M:%S,%N")
| table asctime, NewAsctime
```

asctime	NewAsctime
2020-03-05 21:42:55,814	1583444575.814000
2020-03-05 21:42:41,796	1583444561.796000
2020-03-05 21:41:59,745	1583444519.745000
2020-03-05 21:41:45,728	1583444505.728000
2020-03-05 21:37:33,420	1583444253.420000

Using Time Commands

Topic Objectives

- Use the **`timechart`** command
- Use the **`timewrap`** command

`timechart` Command

```
...| timechart <stats-func>(<field>) by <field>
     [span=<int><timescale>] [limit=<int>]
```

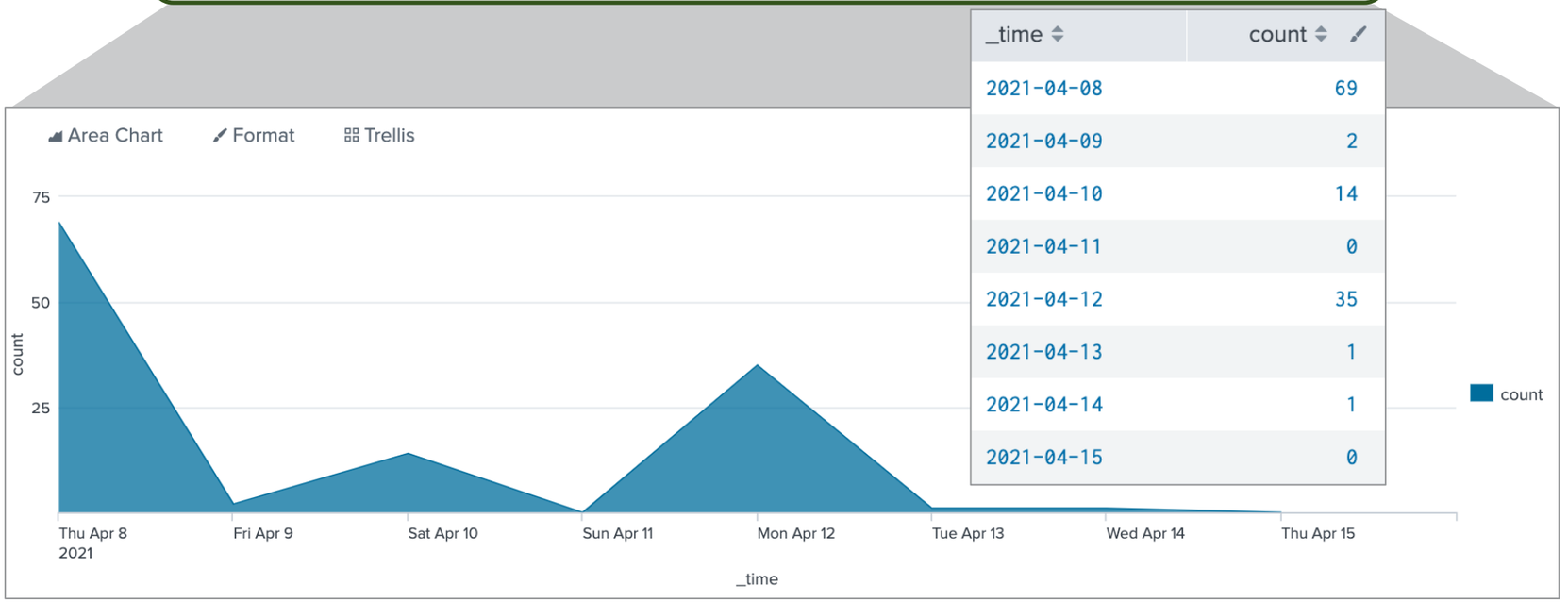
- Performs statistical aggregations against time
- Can utilize various statistical aggregate functions
  - Commonly used with **`count`** and **`sum`** functions
  - Not all functions are discussed in this module
- Plots and trends data over time where **`_time`** is always the x-axis
- Results can be split by another **`<field>`** using a **`by`** clause
- The **`span`** and **`limit`** options control additional aspects of timechart output and are discussed in succeeding slides

`timechart` Command Example

Scenario

How many usage violations have occurred during the last 7 days?

```
index=network sourcetype=cisco_wsa_squid usage=Violation
| timechart count
```



_time	count
2021-04-08	69
2021-04-09	2
2021-04-10	14
2021-04-11	0
2021-04-12	35
2021-04-13	1
2021-04-14	1
2021-04-15	0

Note

The **`count`** function returns a count of all events or for a specific field.

`timechart` Command: With `by` Clause

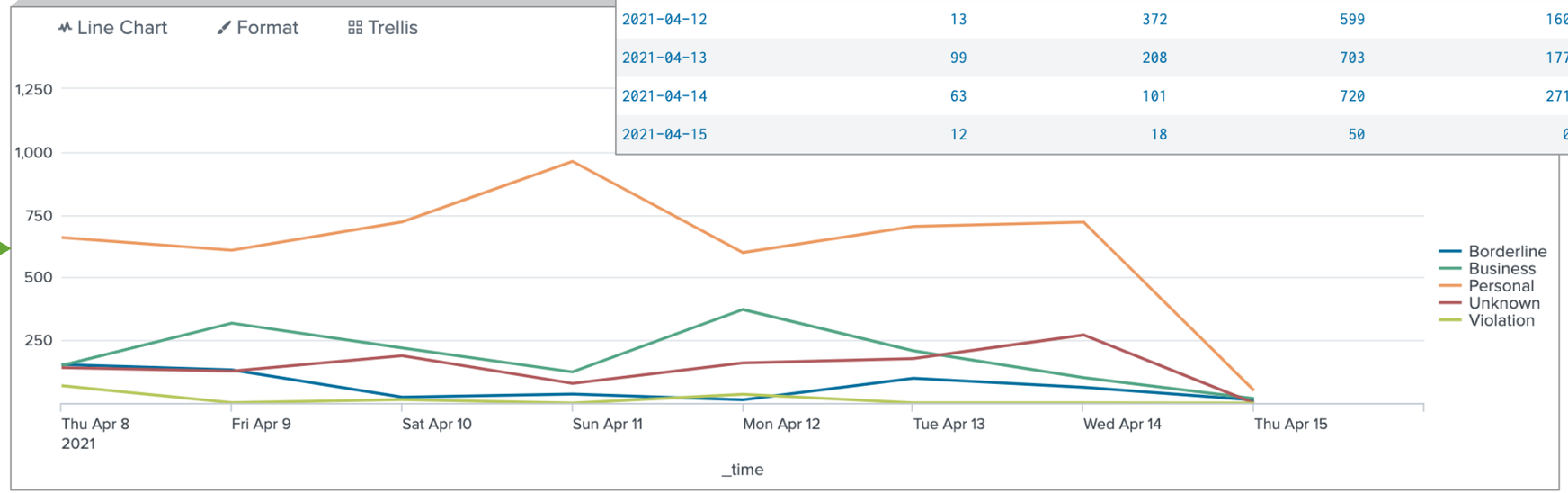
Scenario

What is the overall usage trend for the last 24 hours?

```
index=network sourcetype=cisco_wsa_squid earliest=-24h
| timechart count by usage
```

Each column represents a line in the line chart

_time	Borderline	Business	Personal	Unknown	Violation
2021-04-08	153	149	659	141	69
2021-04-09	132	318	608	127	2
2021-04-10	24	219	721	188	14
2021-04-11	36	124	962	78	0
2021-04-12	13	372	599	160	35
2021-04-13	99	208	703	177	1
2021-04-14	63	101	720	271	1
2021-04-15	12	18	50	0	0



Note

Using **`timechart`**, you can only split by one field because **`_time`** is the implied first by field.

`timechart` Command: `span` Option

- The **`timechart`** command "buckets" the values of the **`_time`** field based on time range if no **`span`** argument is specified
- Examples:
  - A **Last 60 minutes** uses **`span=1m`**
  - B **Last 24 hours** uses **`span=30m`**

```
index=security sourcetype=linux_secure vendor_action=*
| timechart count by vendor_action
```

Last 60 minutes

_time	Accepted	Failed	session opened
2021-04-15 01:43:00	0	14	0
2021-04-15 01:44:00	0	0	1
2021-04-15 01:45:00	0	0	1
2021-04-15 01:46:00	1	11	0

Last 24 hours

_time	Accepted	Failed	session opened
2021-04-14 02:00:00	13	133	15
2021-04-14 02:30:00	10	250	13
2021-04-14 03:00:00	10	154	16
2021-04-14 03:30:00	8	227	22

`timechart` Command: `span` Option (cont.)

Manually adjust the interval using the **`span`** option

```
index=security sourcetype=linux_secure vendor_action=*
| timechart span=15m count by vendor_action
```

_time	Accepted	Failed	session opened
2021-04-15 01:45:00	3	80	1
2021-04-15 02:00:00	5	113	7
2021-04-15 02:15:00	2	62	8
2021-04-15 02:30:00	3	67	14
2021-04-15 02:45:00	2	173	10

`timechart` Command: `limit` Option

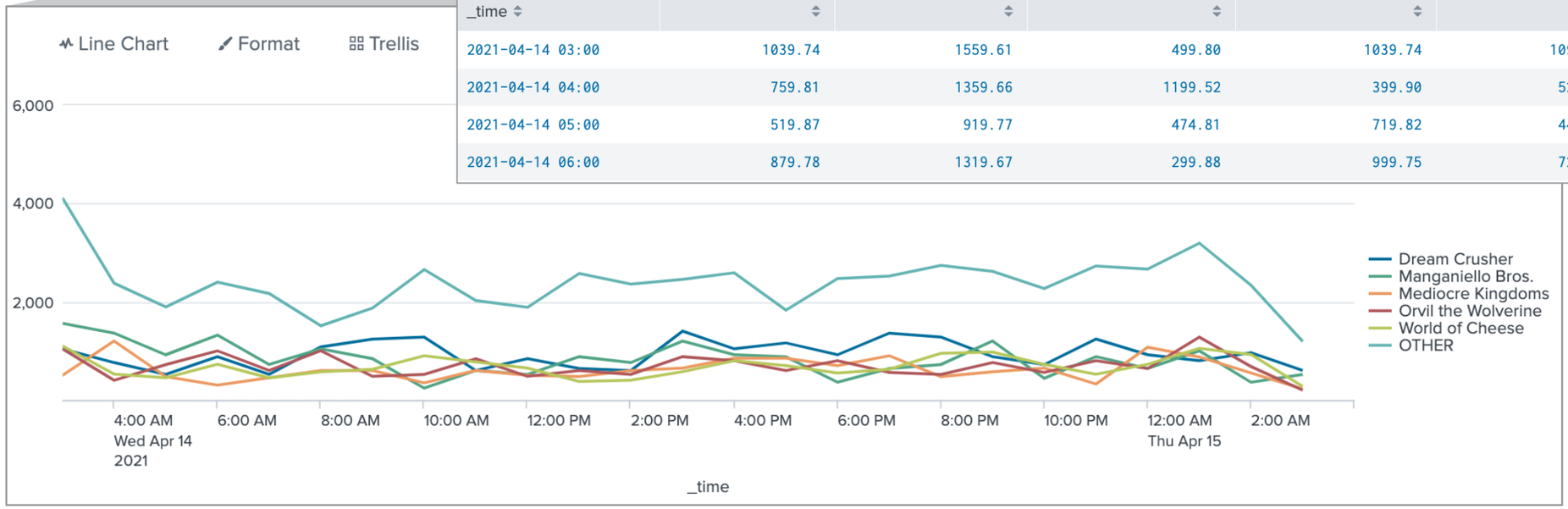
The `limit` option controls the number of distinct values returned by the `by` clause field

Scenario

What was the total web sales for our top products during the last 24 hours?

```
index=web sourcetype=access_combined
| timechart span=1h sum(price) by product_name limit=5
```

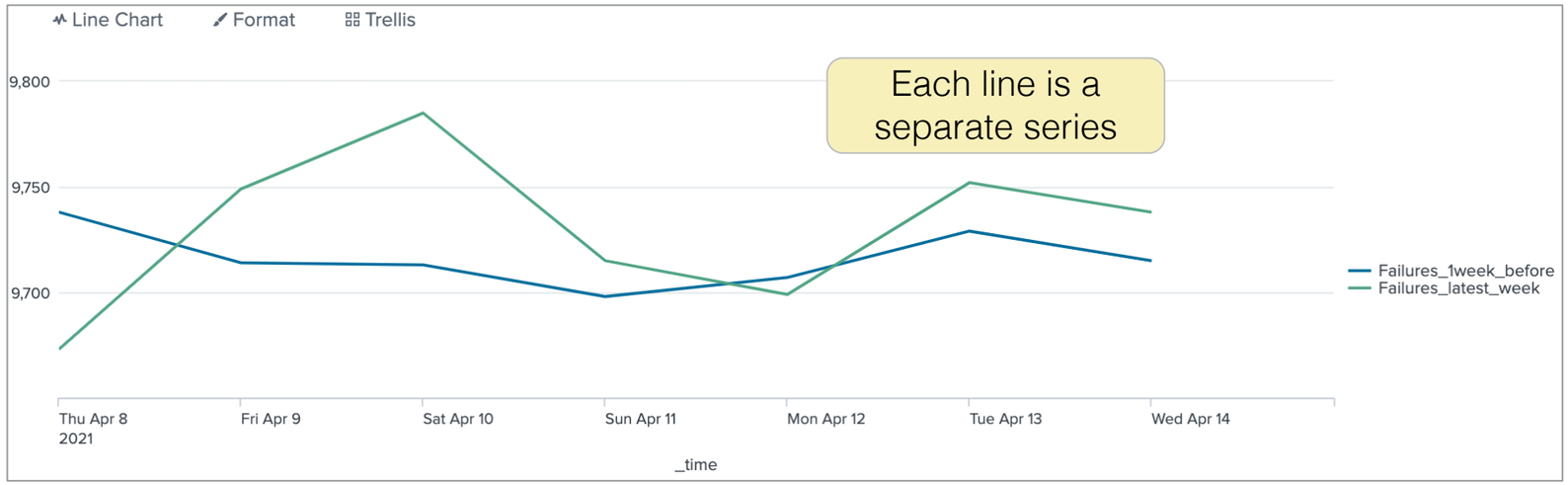
_time	Dream Crusher	Manganiello Bros.	Mediocre Kingdoms	Orvil the Wolverine	World of Cheese	OTHER
2021-04-14 03:00	1039.74	1559.61	499.80	1039.74	1099.56	4111.23
2021-04-14 04:00	759.81	1359.66	1199.52	399.90	524.79	2376.17
2021-04-14 05:00	519.87	919.77	474.81	719.82	449.82	1894.52
2021-04-14 06:00	879.78	1319.67	299.88	999.75	724.71	2398.30



`timewrap` Command

```
...| timewrap [<int>]<timescale>
```

- Displays the output of the **`timechart`** command, so that each time period is a separate series
- Can compare data over a specific time period, such as day-over-day or month-over-month

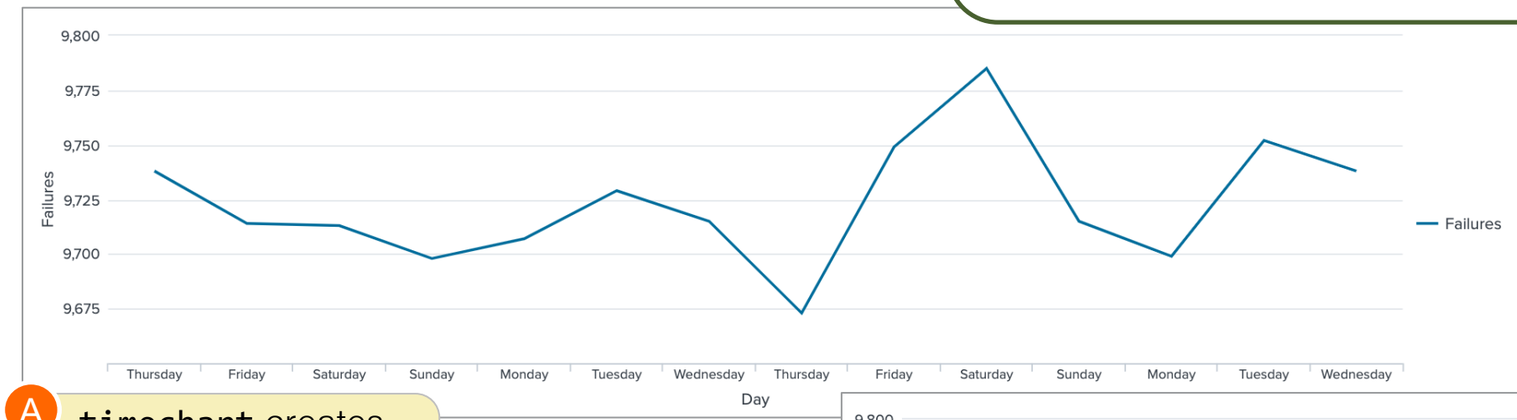


`timewrap` Command Example

Scenario

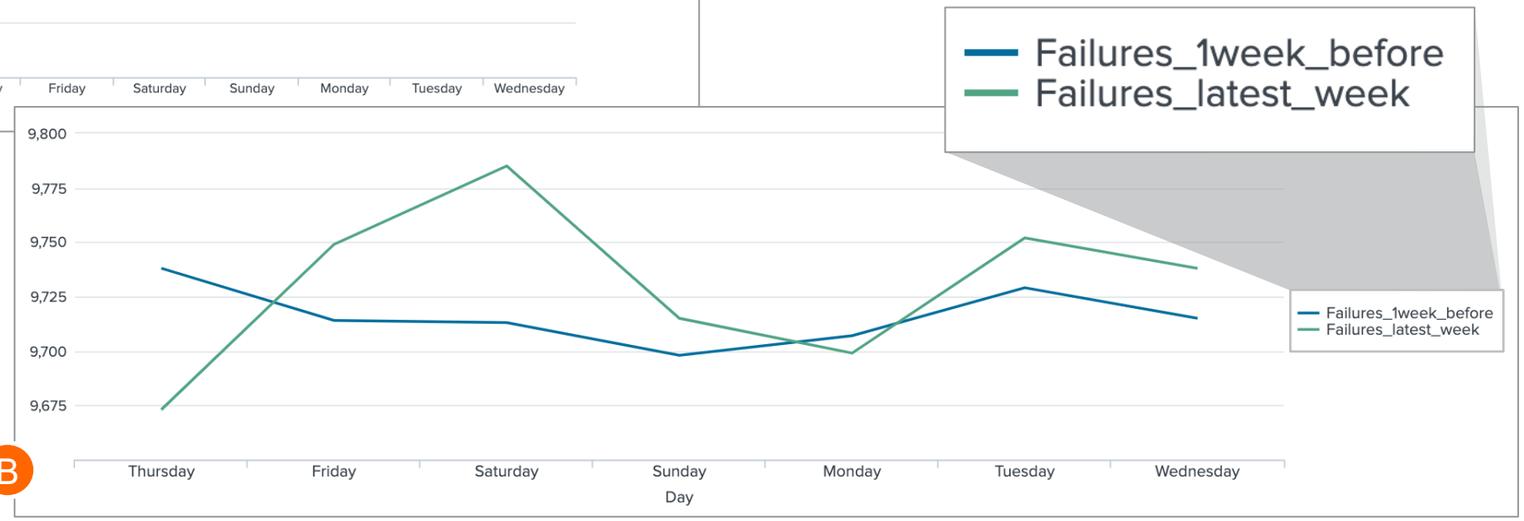
Compare the number of password failures over the last week to password failures over the previous week.

```
index=security "failed password" earliest=-14d@d latest=@d
A | timechart span=1d count as Failures
B | timewrap 1w
  | rename _time as Day
  | eval Day = strftime(Day, "%A")
```



A `timechart` creates one data series spanning 14 days

Data series are automatically renamed based on timescale

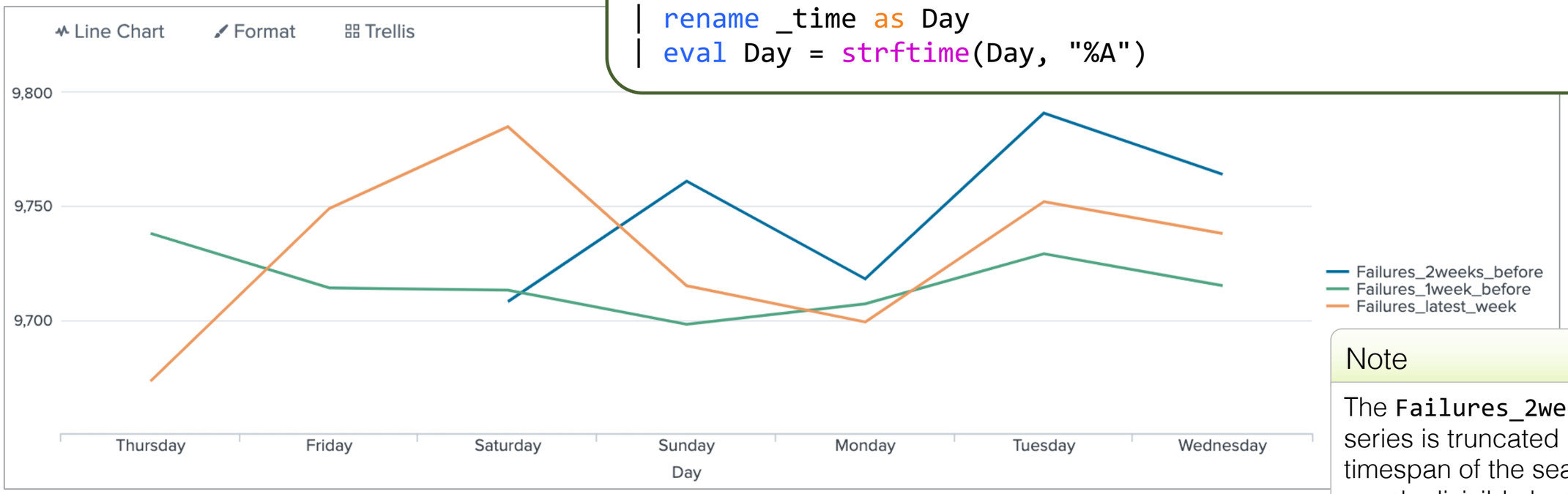


B `timewrap` splits data into two series, each spanning a week and sharing the same weekdays

`timewrap` Command Example (cont.)

Expanding the time range adds more data series

```
index=security "failed password" earliest=-19d@d latest=@d
| timechart span=1d count as Failures
| timewrap 1w
| rename _time as Day
| eval Day = strftime(Day, "%A")
```



Note

The `Failures_2weeks_before` series is truncated because the timespan of the search is not evenly divisible by the duration specified by `timewrap`.

Formatting Time & Using Time Commands Lab Exercise

Time: 30 minutes

Tasks:

- Use the **`timechart`** command and time formatting functions to fulfill three different scenario requests
  - Find and visualize non-business network activity from the previous business week
  - Compare network server errors from last week to the daily average over the last month
  - Generate a detailed sales report from last week's online sales data

Working with Time Zones

Topic Objectives

- Understand how time and timezones are represented in your data
- Determine the time zone of your server
- Use `strftime` to correct timezones in results

Checking Your Data

Scenario

A new campaign aimed at early morning sales is ongoing. Display early morning retail sales for 2-5 am for the previous two days.

```
index=sales sourcetype=vendor_sales
  earliest=-2d@d latest=@d date_hour>=2 AND date_hour<5
| bin span=1h _time
| stats sum(price) as "Hourly Sales" by _time
| eval Hour=strftime(_time, "%b %d, %I %p")
| table Hour, "Hourly Sales"
```

What you see

Hour	Hourly Sales
Feb 12, 09 PM	1173.44
Feb 12, 10 PM	818.58
Feb 12, 11 PM	986.51
Feb 13, 09 PM	778.61
Feb 13, 10 PM	842.57
Feb 13, 11 PM	793.58

What you expected to see

Hour	Hourly Sales
Feb 12, 02 AM	698.57
Feb 12, 03 AM	721.60
Feb 12, 04 AM	805.57
Feb 13, 02 AM	1173.44
Feb 13, 03 AM	818.58
Feb 13, 04 AM	986.51

Don't Forget Time Zones!

Scenario

A new campaign aimed at early morning sales is ongoing. Display early morning retail sales for 2-5 am for the previous two days.

```
index=sales sourcetype=vendor_sales
  earliest=-2d@d latest=@d date_hour>=2 AND date_hour<5
| bin span=1h _time
| stats sum(price) as "Hourly Sales" by _time
| eval Hour=strftime(_time, "%b %d, %I %p")
| table Hour, "Hourly Sales"
```

- Remember, **`date_*`** fields do not reflect your local time, but are the values of time/date directly from the raw events
- To determine your time zone:
  1. In **Preferences**, set **Time Zone** to **Default System Timezone**
  2. Run a search over the last 15 minutes
  3. Read the event timestamps and compare with your local time

Using `strftime` with Time Zones: `%H`

- Many organizations that span multiple time zones normalize their data to UTC (Universal Time Coordinated)
- Use the `%H` argument with the `strftime` function to display data with user's time zone preference

```
index=sales sourcetype=vendor_sales
 earliest=-d@d latest=@d
| eval my_hour = strftime(_time,"%H")
| table my_hour, date_hour
```

my_hour	date_hour
23	4

Using `strftime` with Time Zones Example

Scenario

A new campaign aimed at early morning sales is ongoing. Display early morning retail sales for 2-5 am for the previous two days.

```
index=sales sourcetype=vendor_sales earliest=-2d@d latest=@d
| eval my_hour = strftime(_time,"%H")
| search my_hour>=2 AND my_hour<5
| bin span=1h _time
| stats sum(price) as "Hourly Sales" by _time
| eval Hour=strftime(_time, "%b %d, %I %p")
| table Hour, "Hourly Sales"
```

This is the same search shown earlier, modified to work in any time zone using **%H**

Wrap-up Slides

Wrap-up

- You should now be able to:
  - Use time modifiers, **`date_*`** fields, and time range picker to control search behavior
  - Group events using time with the **`bin`** command
  - Format time with the **`eval`** command
  - Use the **`timewrap`** command with **`timechart`**
  - Use **`strftime`** to correct timezones in results

Community

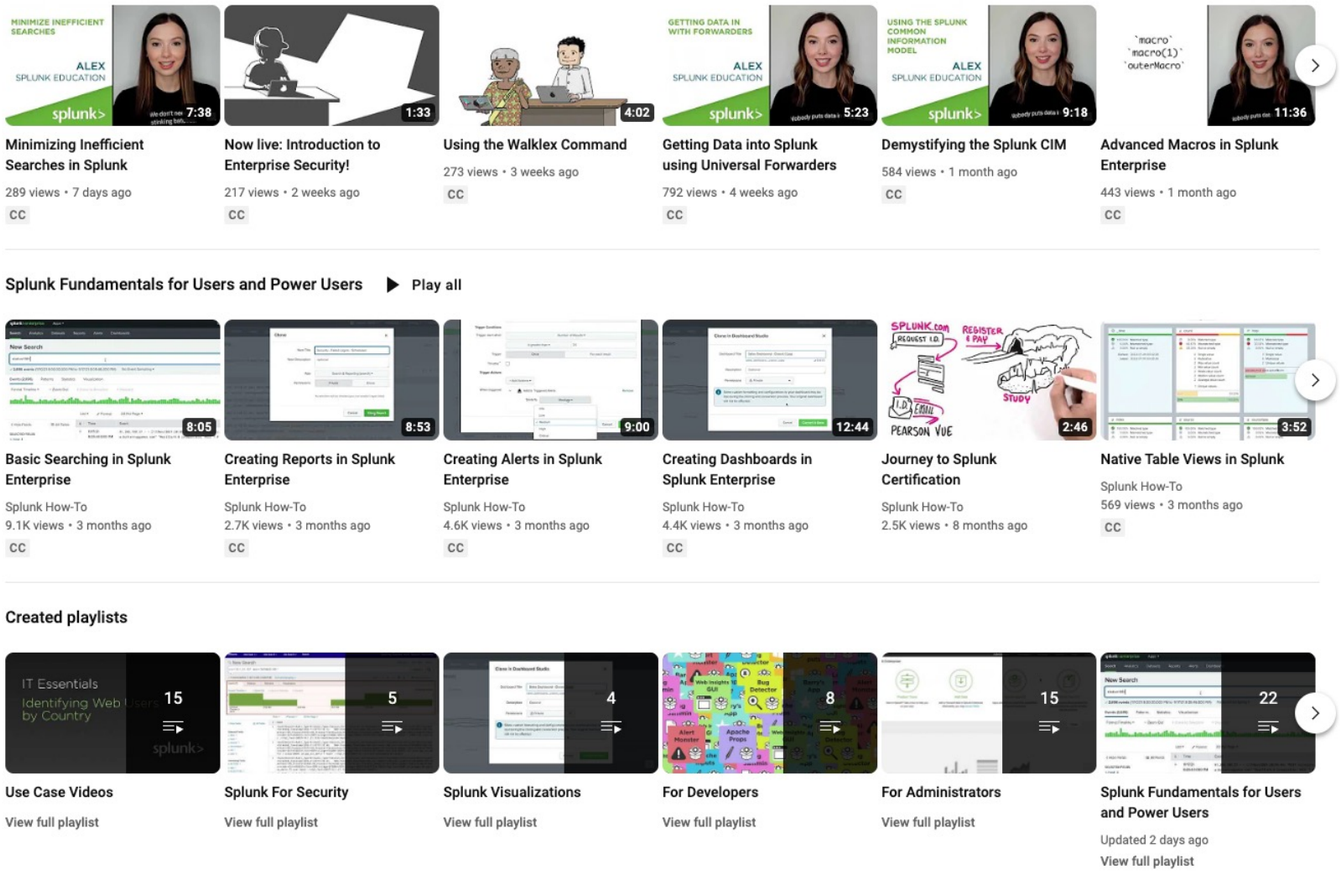
- Splunk Community Portal – community.splunk.com
  - Answers
  - Discussions
  - Splunk Trust
  - User Groups
  - Ideas
- Splunk Blogs – splunk.com/blog/
- Splunk Base – splunkbase.com
  - Apps
  - Curated Collections
- Splunk Docs on Twitter – twitter.com/splunkdocs
- Splunk Dev on Twitter – twitter.com/splunkdev
- Splunk on Slack – splk.it/slack
- .conf – conf.splunk.com

Community

- Knowledge Base – Search knowledge base, answers, and docs to troubleshoot your issue
- splunk>dev – Documentation for developers
- Splunk Docs – Product, best practices, and tools documentation for all Splunk products
- Splunk Lantern – Actionable guidance by experts
- Create a case – Support for critical issues
- Contact Us – Find region-specific support
  - (855) SPLUNK.S or (855) 775.8657
  - Not in the US? Find your local office
- System Status – Cloud Services, Observability Cloud, Splunk On-Call, Synthetic Monitoring
- Splunk Product Security – Critical Security Alerts, Quarterly Security Patches, and 3rd Party Bulletins

Splunk How-To Channel

Free, short videos on a variety of Splunk topics: splk.it/How-To



Learning Paths

Search Expert – Recommended Courses

Free eLearning courses are highlighted in blue and courses with an * are present in both learning paths.

- What is Splunk *
- Introduction to Splunk *
- Using Fields *
- Scheduling Reports and Alerts
- Visualizations
- Statistical Processing
- Working with Time
- Comparing Values
- Result Modification
- Leveraging Lookups and Subsearches
- Correlation Analysis
- Search Under the Hood
- Multivalue Fields
- Search Optimization *

Learning Paths

Knowledge Manager – Recommended Courses

Free eLearning courses are highlighted in blue and courses with an * are present in both learning paths.

- What is Splunk *
- Introduction to Splunk *
- Using Fields *
- Introduction to Knowledge Objects
- Creating Knowledge Objects
- Creating Field Extractions
- Enriching Data with Lookups
- Data Models
- Introduction to Dashboards
- Dynamic Dashboards
- Using Choropleth
- Search Optimization *

Splunk Mobile

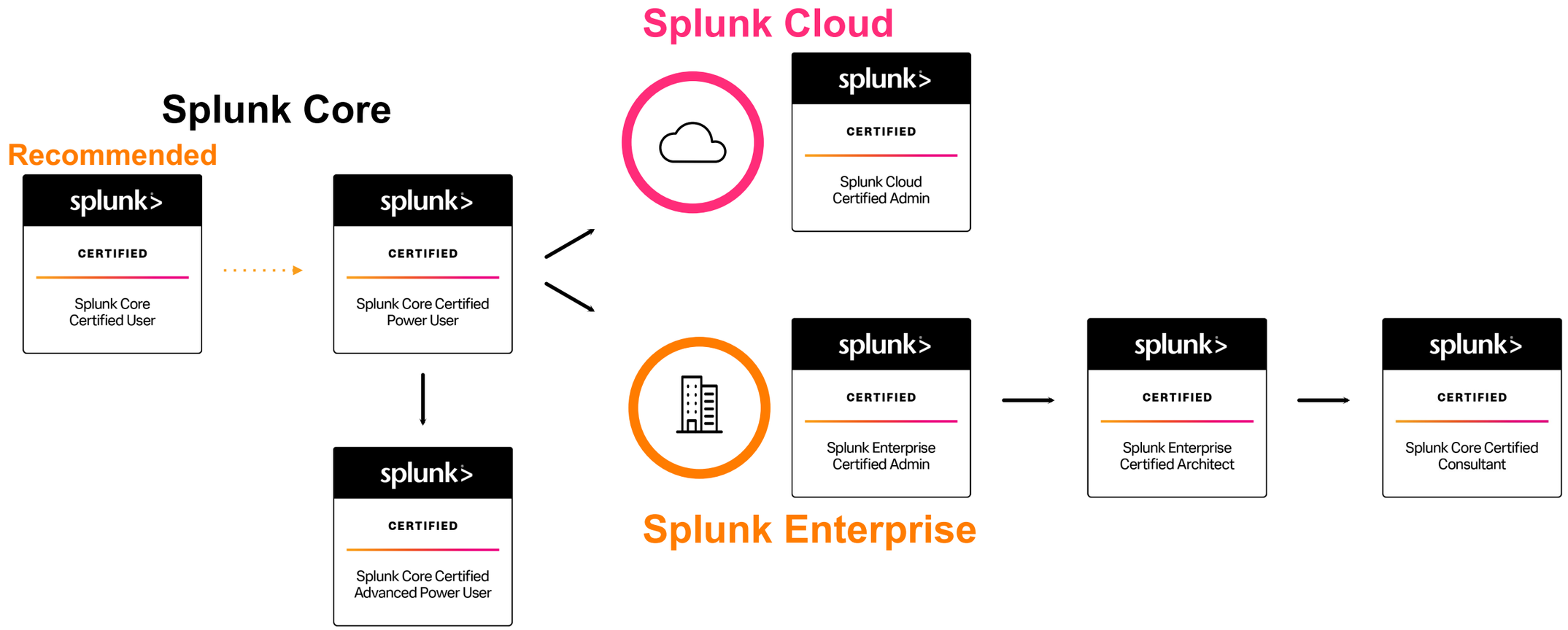
- Free app available to all Splunk Cloud and Splunk Enterprise customers
- Analyze data and receive actionable alerts on-the-go with mobile-friendly dashboards
- iOS and Android
- See the Product Brief
- Download for iOS splk.it/ios

Splunk Certification

Offerings & Requirements

Splunk Core and Beyond

Regardless of which Splunk product you use, it all starts with Splunk Core



App-Specific Offerings

For Splunk Add-Ons

App Developer

ES Administration

ITSI Administration

SOAR Automation Developer

Splunk Core Certified User

This entry-level certification demonstrates an individual's basic ability to navigate and use Splunk software

Prerequisite Certification(s):

- None

Prerequisite Course(s):

- None

Splunk Core Certified User Exam

Time to study! We suggest candidates looking to prepare for this exam complete Fundamentals 1 **or** the following courses:

- What is Splunk?
- Intro to Splunk
- Using Fields
- Scheduling Reports and Alerts
- Visualizations
- Statistical Processing
- Working with Time
- Leveraging Lookups and Subsearches
- Search Optimization
- Enriching Data with Lookups
- Data Models

See here for registration assistance.

Congratulations! You are a...

Recommended Next Step

- Splunk Core Certified Power User

Splunk Core Certified Power User

This entry-level certification demonstrates an individual's foundational competence of Splunk's core software

Prerequisite Certification(s):

- None

Prerequisite Course(s):

- None

Splunk Core Certified Power User Exam

Time to study! We suggest candidates looking to prepare for this exam complete Fundamentals 2 **or** the following courses:

- Visualizations
- Statistical Processing
- Working with Time
- Comparing Values
- Result Modification
- Correlation Analysis
- Search Under the Hood
- Introduction to Knowledge Objects
- Creating Knowledge Objects
- Creating Field Extractions
- Data Models
- Using Choropleth

See here for registration assistance.

Congratulations! You are a...

Recommended Next Steps

- Splunk Core Certified Advanced Power User
- Splunk Enterprise Certified Admin
- Splunk Cloud Certified Admin

Splunk Core Certified Advanced Power User

This certification demonstrates an individual's ability to generate complex searches, reports, and dashboards with Splunk's core software to get the most out of their data

Prerequisite Certification(s):

- Splunk Core Certified Power User

Prerequisite Course(s):

- None

Splunk Core Certified Advanced Power User Exam

Time to study! We suggest candidates looking to prepare for this exam complete Fundamentals 3, Creating Dashboards, and Advanced Searching & Reporting **or** the following courses:

- Using Fields
- Working with Time
- Comparing Values
- Result Modification
- Leveraging Lookups and Subsearches
- Correlation Analysis
- Search Under the Hood
- Multivalue Fields
- Search Optimization
- Creating Field Extractions
- Enriching Data with Lookups
- Data Models
- Using Choropleth
- Introduction to Dashboards
- Dynamic Dashboards

See here for registration assistance.

Congratulations! You are a...

Recommended Next Steps

- Splunk Enterprise Certified Admin
- Splunk Cloud Certified Admin

 | 25 January 2023

Splunk Cloud Certified Admin

This certification demonstrates an individual's ability to support the day-to-day administration and health of a Splunk Cloud environment

Prerequisite Certification(s):

- Splunk Core Certified Power User

Prerequisite Course(s):

- None

Splunk Cloud Certified Admin Exam

Time to study! We suggest candidates looking to prepare for this exam complete **either** the Splunk Cloud Administration **or** the Transitioning to Splunk Cloud course.

Both courses will equally prepare candidates for the exam, but are tailored to meet the needs of the individual based on prior Splunk experience.

Splunk Cloud Administration is designed for net-new administrators working in a Splunk Cloud environment. **Transitioning to Splunk Cloud** is for experienced Enterprise administrators looking to maximize their success in migrating to a Cloud environment.

See here for registration assistance.

Congratulations! You are a...

Recommended Next Steps

- Splunk Certified Developer

Splunk Enterprise Certified Admin

This certification demonstrates an individual's ability to support the day-to-day administration and health of a Splunk Enterprise environment

Prerequisite Certification(s):

- Splunk Core Certified Power User

Prerequisite Course(s):

- None

Splunk Enterprise Certified Admin Exam

Time to study! We suggest candidates looking to prepare for this exam complete the following courses:

- Splunk System Administration
- Splunk Data Administration

See here for registration assistance.

Congratulations! You are a...

Recommended Next Steps

- Splunk Enterprise Certified Architect
- Splunk Certified Developer

Splunk Certified Architect

This certification demonstrates an individual's ability to deploy, manage, and troubleshoot complex Splunk Enterprise environments

Prerequisite Certification(s):

- Splunk Core Certified Power User
- Splunk Enterprise Certified Admin

Prerequisite Course(s):

- Architecting Splunk Enterprise Deployments
- Troubleshooting Splunk Enterprise
- Splunk Cluster Administration
- Splunk Deployment Practical Lab

Splunk Enterprise Certified Architect Exam

Time to study! We **require** candidates looking to register for this exam to complete the following prerequisite courses:

- Architecting Splunk Enterprise Deployments
- Troubleshooting Splunk Enterprise
- Splunk Cluster Administration
- Splunk Deployment Practical Lab

Candidates who are **Splunk Enterprise Certified Admin** and have completed all of the above courses will automatically receive an exam authorization for the Splunk Enterprise Certified Architect exam within 5-7 business days of receiving their passing lab results.

See here for registration assistance.

Congratulations! You are a...

Recommended Next Steps

- Splunk Core Certified Consultant

Splunk Core Certified Consultant

This certification demonstrates an individual's ability to properly size, install, and implement Splunk environments and to advise others on how to utilize the product and maximize its value for their needs

Prerequisite Certification(s):

- Splunk Core Certified Power User
- Splunk Enterprise Certified Admin
- Splunk Enterprise Certified Architect

Prerequisite Course(s):

- *Advanced Power User courses* ***or*** *digital badge**
- Core Consultant Labs
  - Indexer Cluster Implementation
  - Distributed Search Migration
  - Implementation Fundamentals
  - Architect Implementation 1-3
- Services Core Implementation

Splunk Core Certified Consultant Exam

Time to study! We **require** candidates looking to register for this exam to complete the following prerequisite courses:

- *Fundamentals 3*, *Creating Dashboards*, *Advanced Searching & Reporting**
- Core Consultant Labs
- Services Core Implementation

Candidates who are **Splunk Enterprise Certified Architects** and have completed all of the above courses must contact certification@splunk.com to request their Core Consultant exam authorization.

See here for registration assistance.

These Advanced Power User courses can be replaced with a Splunk Certified Advanced Power User badge* ***or* *completion of the following courses:*

- Using Fields
- Creating Field Extractions
- Enriching Data with Lookups
- Data Models
- Search Optimization
- Working with Time
- Leveraging Lookups and Subsearches
- Comparing Values
- Correlation Analysis
- Result Modification
- Multivalue Fields
- Search Under the Hood
- Introduction to Dashboards
- Dynamic Dashboards
- Using Choropleth

Congratulations! You are a...

Recommended Next Steps

- None

 | 25 January 2023

Splunk Certified Developer

This certification demonstrates an individual's expertise in drilldowns, advanced behaviors and visualizations, planning, creating, and packaging apps, and REST endpoints

Prerequisite Certification(s):

- Splunk Core Certified Power User

AND

- Splunk Enterprise Certified Admin

OR

- Splunk Cloud Certified Admin

Prerequisite Course(s):

- None

Splunk Certified Developer Exam

Time to study! We suggest candidates looking to prepare for this exam complete the following courses:

- Creating Dashboards with Splunk*
- Advanced Dashboards & Visualizations
- Building Splunk Apps
- Developing with Splunk's REST API

This course may also be substituted with the following newly-launched courses:

- Introduction to Dashboards
- Dynamic Dashboards
- Using Choropleth

See here for registration assistance.

Congratulations! You are a...

Recommended Next Steps

- None

Splunk Enterprise Security Certified Admin

This certification demonstrates an individual's ability to install, configure, and manage a Splunk Enterprise Security deployment

Prerequisite Certification(s):

- None

Prerequisite Course(s):

- None

Splunk Enterprise Security Certified Admin Exam

Time to study! We suggest candidates looking to prepare for this exam complete the following course:

- Administering Splunk Enterprise Security

Please note: all candidates are expected to have working knowledge and experience as either Splunk Cloud or Splunk Enterprise Administrators.

See here for registration assistance.

Congratulations! You are a...

Recommended Next Steps

- Splunk Phantom Certified Admin

Splunk IT Service Intelligence Certified Admin

This certification demonstrates an individual's ability to deploy, manage, and utilize Splunk ITSI to monitor mission-critical services

Prerequisite Certification(s):

- None

Prerequisite Course(s):

- None

Splunk IT Service Intelligence Certified Admin Exam

Time to study! We suggest candidates looking to prepare for this exam complete the following course:

- Implementing Splunk IT Service Intelligence

Please note: all candidates are expected to have working knowledge and experience as either Splunk Cloud or Splunk Enterprise Administrators.

See here for registration assistance.

Congratulations! You are a...

Recommended Next Steps

- Courses on Observability

Splunk SOAR Certified Automation Developer

This certification demonstrates an individual's ability to install and configure a SOAR server, integrate it with Splunk, and plan, design, create, and debug playbooks

Prerequisite Certification(s):

- None

Prerequisite Course(s):

- None

Splunk SOAR Certified Automation Developer Exam

Time to study! We suggest candidates looking to prepare for this exam complete the following courses:

- Administering SOAR (Phantom)
- Developing SOAR (Phantom) Playbooks
- Advanced SOAR (Phantom) Implementation

Please note: all candidates are expected to have working knowledge and experience as either Splunk Cloud or Splunk Enterprise Administrators.

See here for registration assistance.

Congratulations! You are a...

Recommended Next Steps

- None

Thank You

splunk> turn data into doing™