Splunk 4 Rookies – Lab Guide

Overview

This lab guide contains the hands-on exercises for the Splunk 4 Rookies workshop. Before proceeding with these exercises, please ensure that you have a copy of the Splunk 4 Rookies slide deck, which will help to put into context the tasks you are carrying out.

Get the Splunk 4 Rookies slide deck: https://splk.it/S4R-Attendee

Prerequisites

In order to complete these exercises, you will need your own Splunk instance. The presenter delivering your Splunk 4 Rookies workshop will give you a unique URL for your event, which will allow you to register and create your own Splunk instance.

Please follow the steps provided by the presenter to get access to Splunk!

Table of Contents

Exercise 1 – Register and Create Your Environment

Description

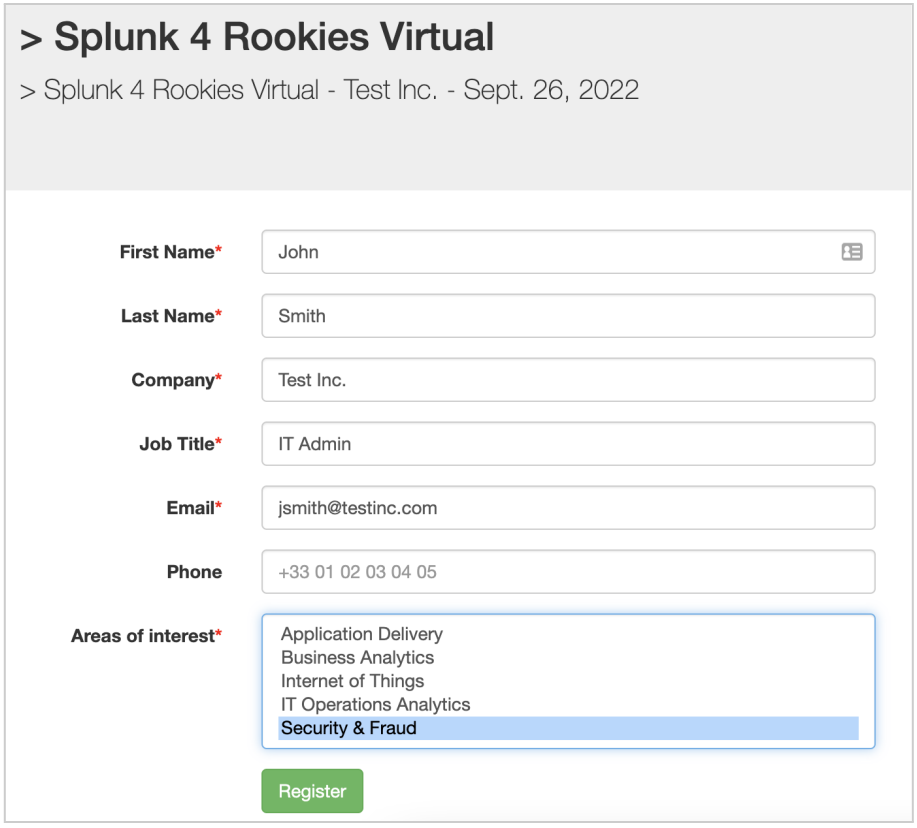
You'll need a Splunk instance to do these hands-on exercises – time to get one!

In this exercise, you will create your own Splunk Enterprise instance using our Splunk 4 Rookies registration portal.

Please note that the instance you create will automatically be terminated after 24 hours from the time you register (see steps below).

Steps

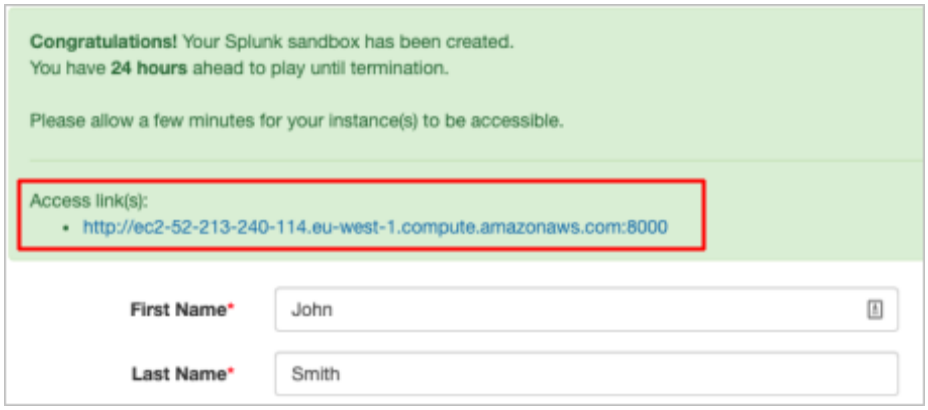
1. Go to the unique URL given to you by the Splunk representative delivering your Splunk 4 Rookies workshop (it will be something like **http://splunk4rookies.com/xxxx/self_register**, where 'xxxx' is a unique code for your session.)



2. Fill out the registration page, ensuring that you select at least one of the '**Areas of Interest**'.

3. Click on **Register** to submit the form.

4. Congratulations - your very own Splunk instance is now being deployed and configured! You will be presented with a unique URL for your Splunk instance – **this will take a few minutes to deploy, so the link won't work immediately**. Please be patient before trying it!



Note: Your lab environment will be live for **24 hours** from the time you get your link so feel free to continue to play around after today's workshop!

Exercise 2 – Create an App and Add Data to Splunk

Description

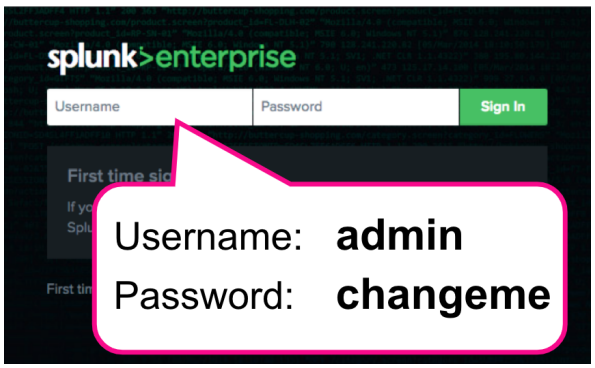
Splunk apps and add-ons provide customisable content and capabilities for a variety of technologies and use cases, accelerating the time it takes to get value from your data. They're also a great way to organise and share your content - such as reports and dashboards - to Splunk users. Anyone can build apps and add-ons, and today we're going to create our own app that contains a dashboard.

Since Splunk is a data platform, we'll also need to load some data in before we can do anything!

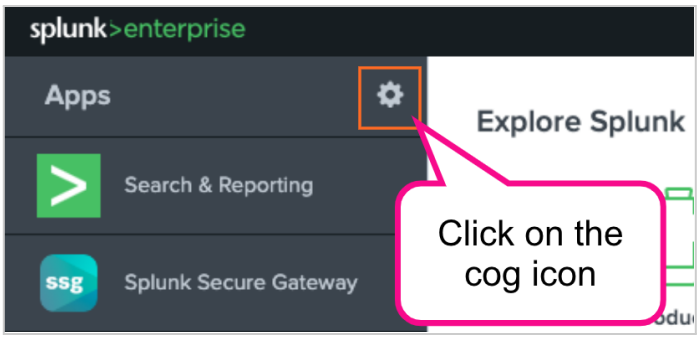
In this exercise, you will create a new app and then add some data to your Splunk Enterprise instance. We will configure Splunk to monitor some sample web server logs, which are currently being generated on the same server that Splunk is running on.

Steps

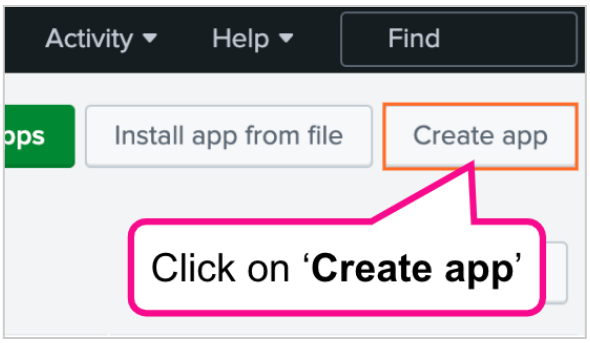
1. Browse to the Splunk login page using the unique URL generated by the Splunk 4 Rookies registration portal (see task 1).
2. Log in using the following credentials:
   Username: **admin**
   Password: **changeme** (note: you do not have to change it ☺)



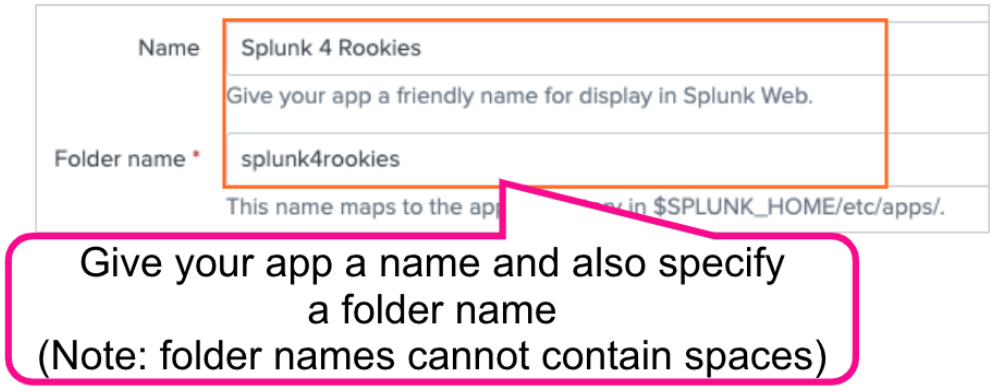
3. On the left side of the page, under the **Apps** section, click on the cog (or wheel) icon.



4. On the top right corner of the screen, click on **Create app**.

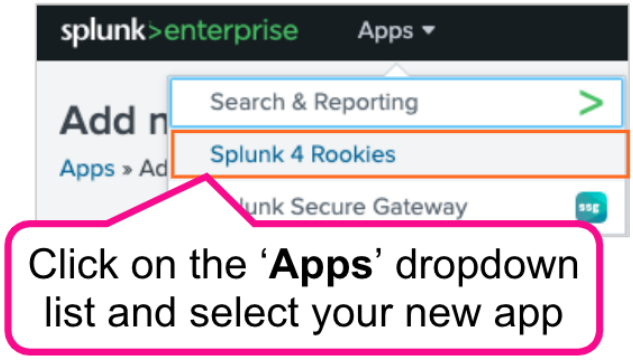


5. Give your app a name and enter a folder name. Leave all other values as they are and click on **Save**.



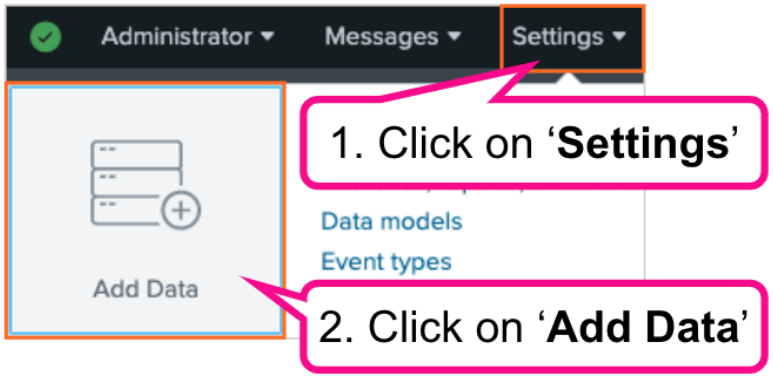
6. Now that our blank app has been created, we need to select the app so that everything we do from now on will be created and saved within the new app.

   To select your app, click on the **Apps** dropdown list at the top left of the page and select your app.

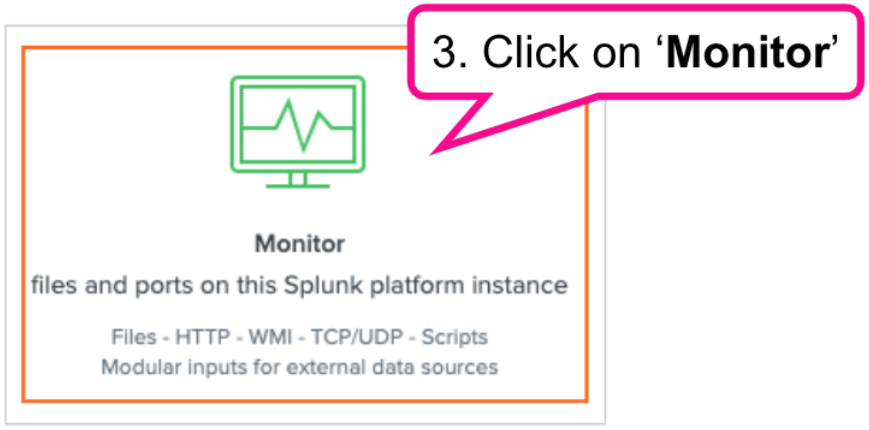


   Now let's add some data!

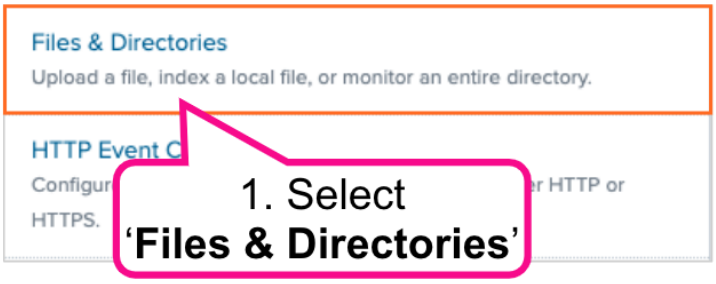
7. With our new app still selected from the dropdown list, go to **Settings** > **Add Data**.



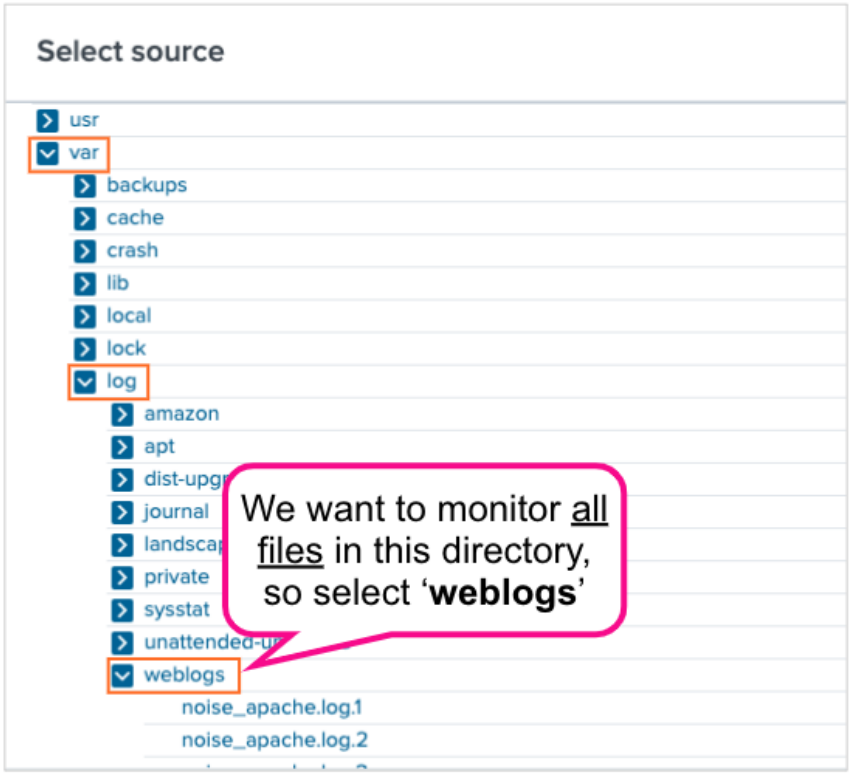
8. For this exercise we will monitor a directory, as this will allow us to pick up new data as it is generated by the web server. To do this, click on '**Monitor**'.



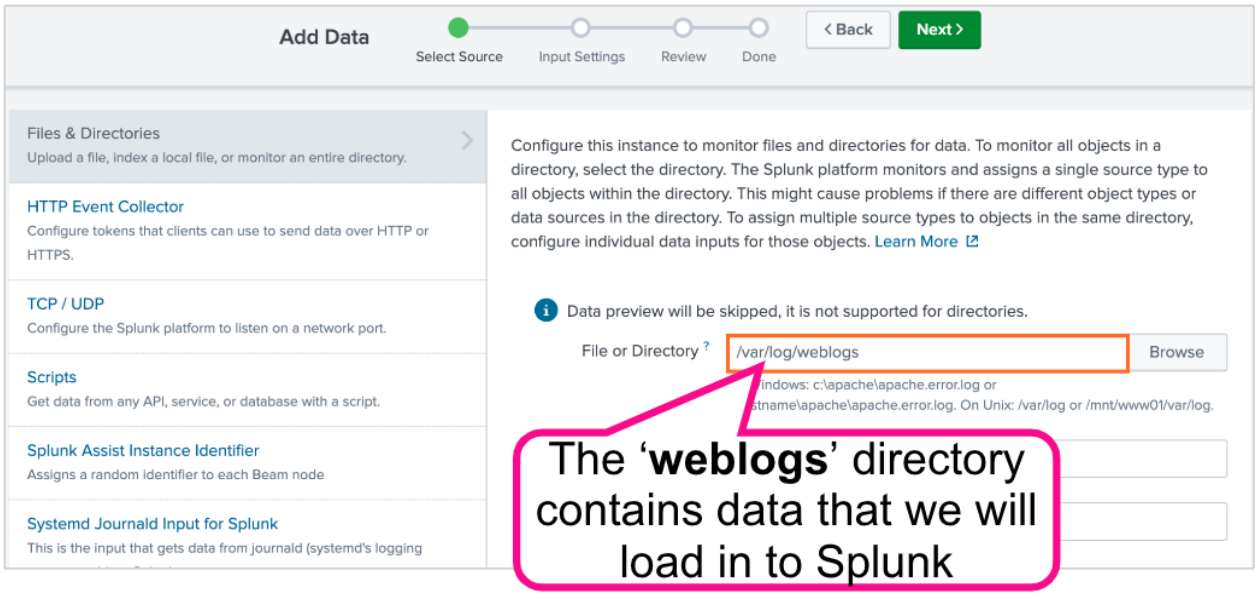
9. Select '**Files & Directories**' and then click '**Browse**'.



10. Browse to **/var/log** and select the **weblogs** directory. Click on **Select** to choose this directory.

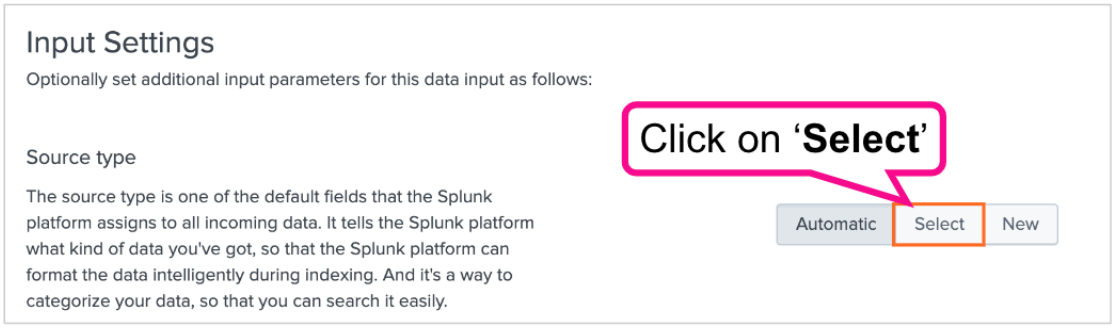


11. Check that the directory path is correct (**/var/log/weblogs**) and click on **Next**.

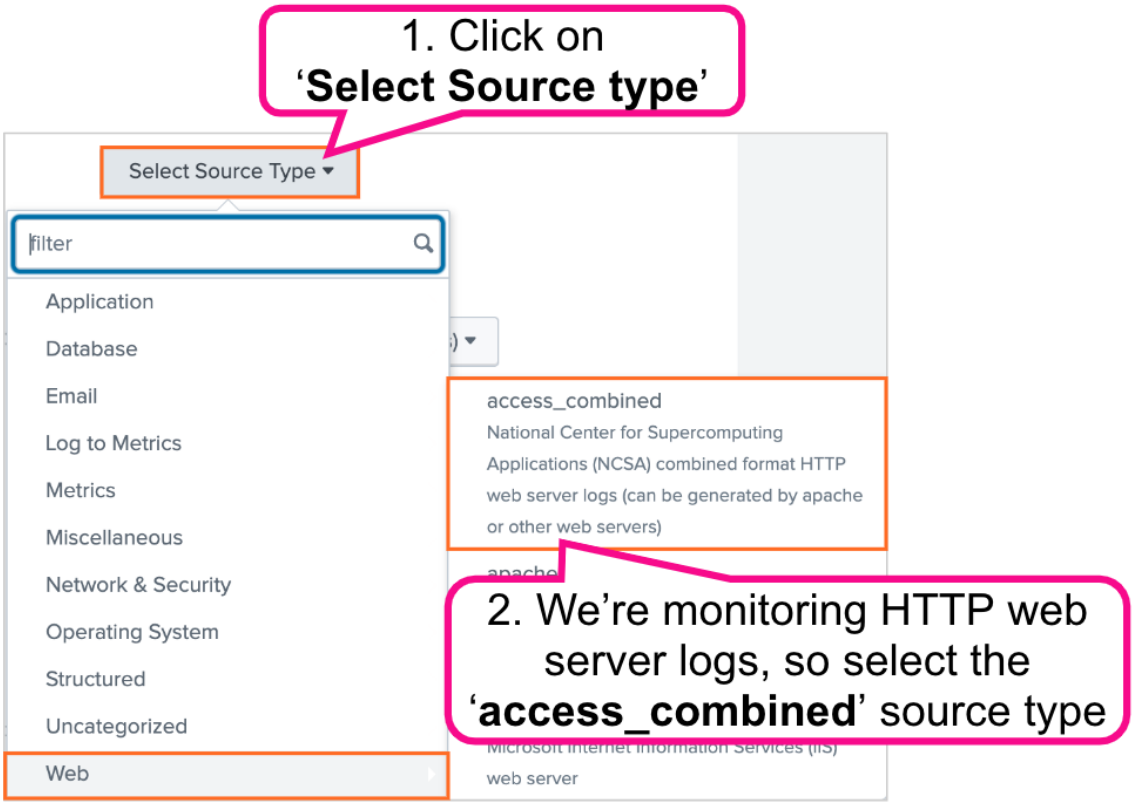


12. Now we need to select a source type for this data. A source type determines how Splunk formats the data during the indexing process. Splunk comes with a large set of predefined source types and can often detect the source type automatically. However, for this exercise you will specify the source type.

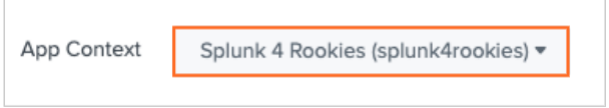
    On the **Input Settings** screen, to the right of the **Source type** section, click on **Select**.



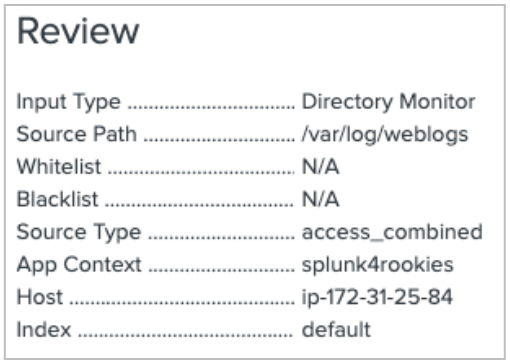
13. Click on the **Select Source Type** dropdown list and browse to **Web > access_combined**. Alternatively, you can start typing 'access' in the **filter** field and the 'access_combined' source type should appear.



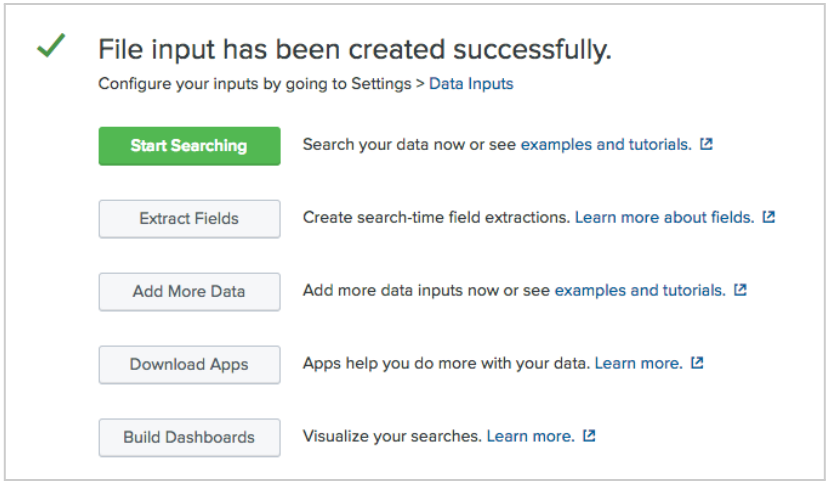
14. For the **App Context**, ensure that your new app is selected from the list.



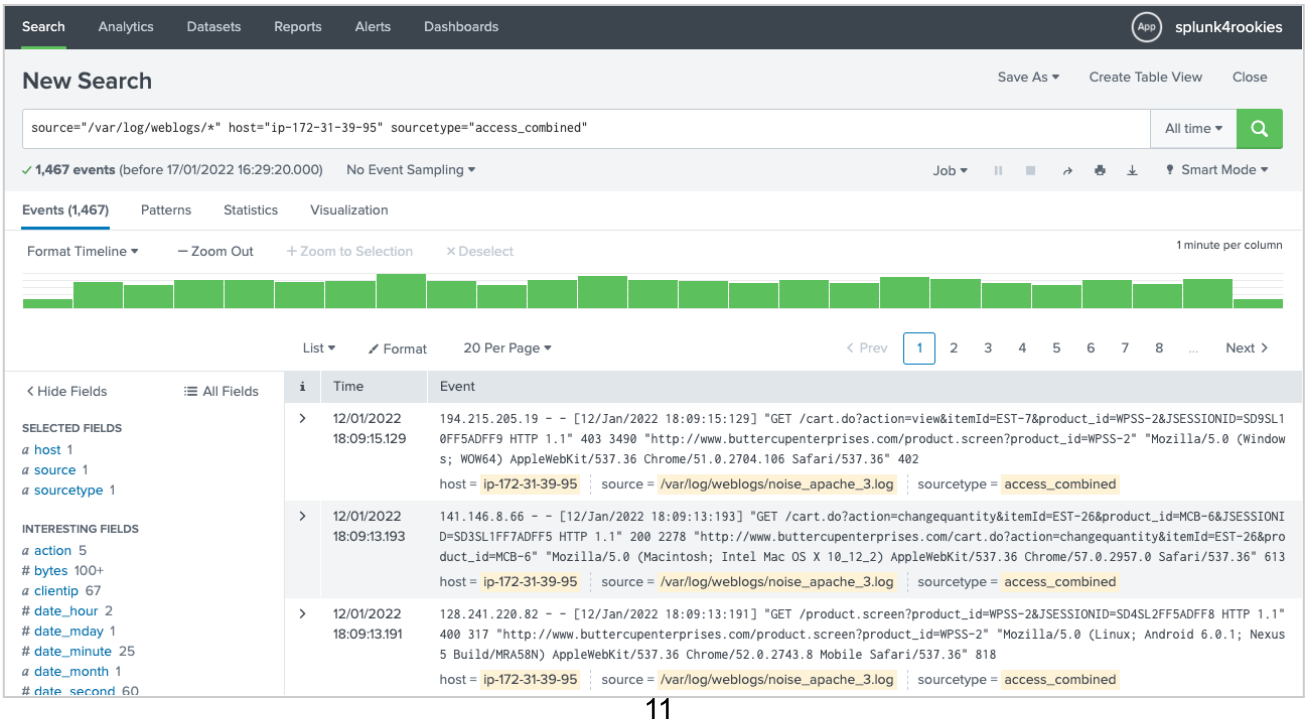
15. Leave all other values as default and click on **Review**.
16. Review your settings and click on **Submit**.



17. You should now receive a message stating that your '**File input has been created successfully**'. Click on **Start Searching** to search the data you have just added to Splunk.



You should now see the raw events being shown in Splunk.



Start Exploring Your Data

Description

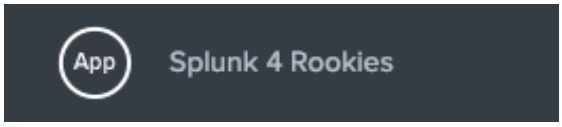
In this exercise, you will try some basic Splunk searches using the Search section of your new app.

Steps

1. Click on the Splunk logo in the top left corner of the screen to take you back to the default home screen.

2. Under the **Apps** section on the left of the page, click on the new app that you created in task 2 (Note: the name will be whatever you entered when you created it.)

3. To search, just type any word or phrase into the search bar and Splunk will search for all events that contain those words.

   So enough talking – let's try some searches!

   Firstly, set the time picker (to the right of the search bar) to **Last 60 minutes**. Your environment has an event generator running in the background, which is constantly creating sample data for you to use. This data started being generated from the moment you registered for your Splunk environment, so let's stick to the last 60 minutes of data…

   Try the following search:

   ```
   503 purchase
   ```

   This will return all events from Splunk that contain the number '**503**' and the word '**purchase**'.

Note: In Splunk, a space between two words is an implied Boolean 'AND', meaning that Splunk will automatically search for events containing both words – you don't need to specify it.

---

4. That's great, but what if there are events with the word '*purchased*', '*purchasing*', or '*purchaser*', for example? Well, we can use a wildcard asterisk (*****) to search for any events containing '**503**' and any word beginning with '**pur**':

   ```
   503 pur*
   ```

   A wildcard is useful if we want to be a bit more flexible with what we're searching for.

5. Remember the '`AND`' operator we mentioned in step 3? Well you can also use the other Boolean operators as well: `OR` and `NOT`. Note that these must be in UPPERCASE.

   Let's try using one of these operators in a search:

   ```
   503 (purchase OR addtocart)
   ```

   This search will return all events containing the number '**503**' and either the word '**purchase**' or the word '**addtocart**'.

6. So far, we've just been searching for text – those numbers could appear anywhere in our data, so how do we know that we're searching the right values? Depending on our data '**503**' could be a HTTP status code, or it could be part of a session ID or a phone number.

   Well, we know we're looking at web logs, so let's include field/value pairs in our search to be more specific with what we're looking for:

   ```
   status=503 action=purchase
   ```

   This will ensure that our results only return web server **purchase** events where the HTTP status code is '**503**'. Always specify field names where possible, to ensure that your results are as accurate as possible!

Challenge Tasks

Q1. How can we find events with a status of 200 that are not purchase events?

Q2. How can we find events where someone had an error when trying to either add an item or remove an item from their cart? (Hint: A HTTP status code of 200 means the transaction was successful. A code of 400 or higher usually means that a failure occurred.)

Note: The challenge task solutions are at the end of this document.

Exercise 3 – IT Operations team: Investigate successful vs unsuccessful web server requests over time

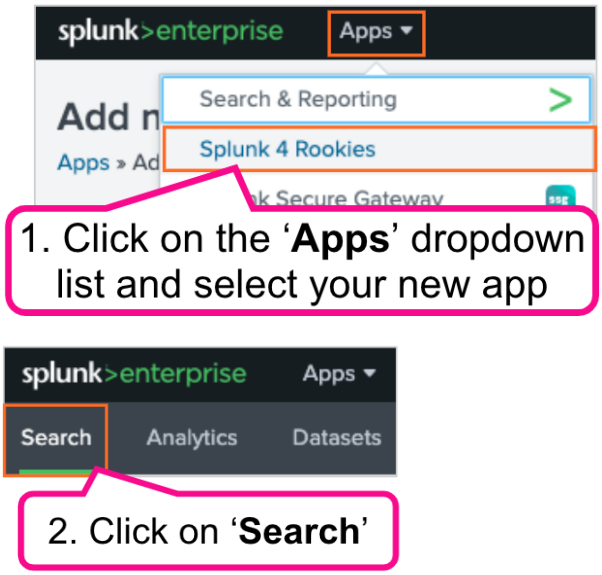
Description

The IT Operations team currently has no visibility of failures on the Buttercup Enterprises website.

In this exercise, you will produce a dashboard panel for the IT Operations team, showing website successes vs failures over time.

Steps

1. To start a new search, first make sure your app is selected from the Apps dropdown list and then click Search on the app menu bar.



2. Search for all web server events over the **Last 60 minutes**:

```
sourcetype=access_combined
```

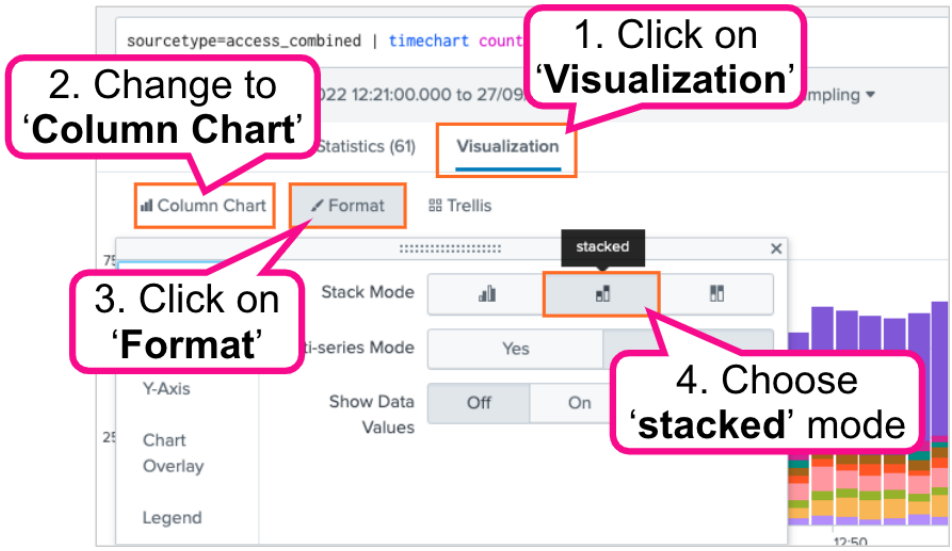
3. Scroll down the page and find the **status** field. Click on the field name to display the field window and select **Top values by time**.

Splunk will automatically populate your search as follows:

```
sourcetype=access_combined | timechart count by status limit=10
```

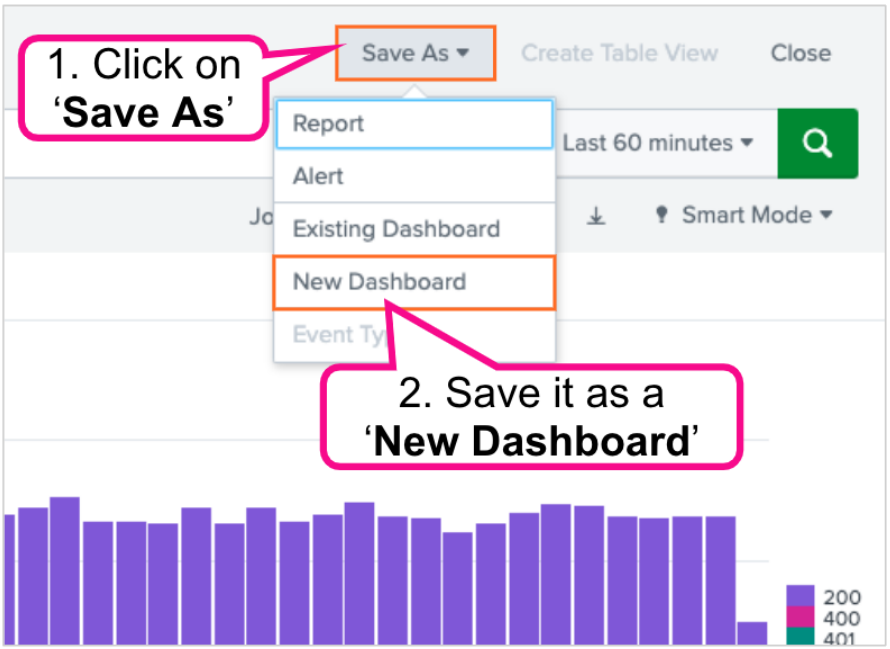
4. A chart will display on the **Visualization** tab. Change the visualization to a **Column Chart**.

Click on **Format** and then on the **General** tab to change the **Stack Mode** to '**stacked**'. Feel free to play around with the formatting until you're happy with the visualization.



5. Now that we have a nice chart visualization, let's add it to a new dashboard so we can share this information with the business.

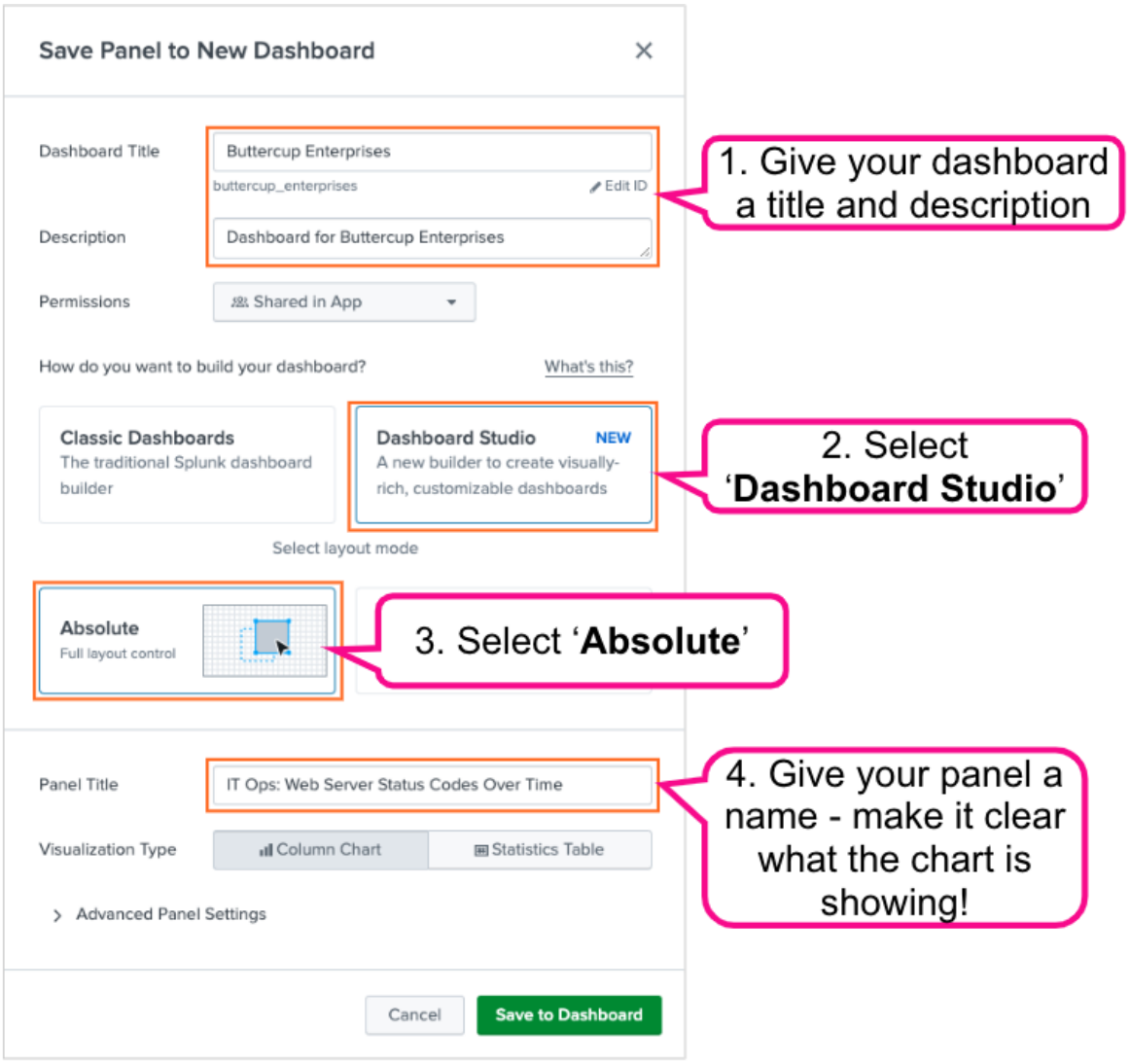
In the top right corner of the screen, go to **Save As > New Dashboard**.



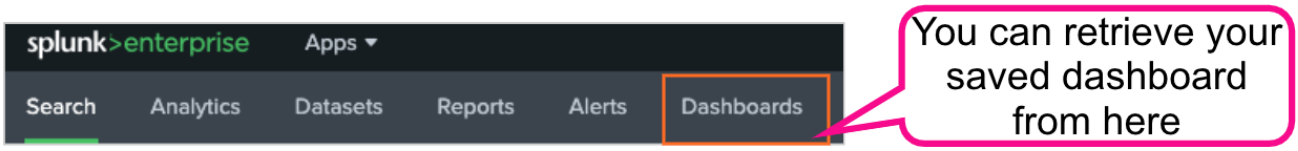
6. On the **Save Panel to New Dashboard** screen, give your dashboard a suitable title and optionally a description too. If you can't think of a name for your dashboard, call it '**Buttercup Enterprises**, or something else meaningful to you.

   Choose how you want to build your dashboard. For today's workshop we will use **Dashboard Studio**. For your layout mode, select **Absolute**.

   Give your panel a title – something that describes what this chart is showing, such as '**IT Ops: Web Server Status Codes Over Time**'.



Congratulations - you've just created a Splunk dashboard with your first panel! Anytime you want to access a dashboard, click on **Dashboards** in the menu bar and select the dashboard you wish to display. Go ahead – give it a try!



Exercise 4 – DevOps team: Show the most common customer operating systems and which web browsers are experiencing the most failures

Description

In this exercise, you will need to extract a new field from your events in order to create the report we need. To accomplish this, we will use Splunk's field extractor wizard.

Custom field extractions are useful in a variety of scenarios, such as:

- When you have custom data and Splunk did not recognise/extract a particular field that you need
- When you need to extract a particular part of an event in order to be able to search/report on that value

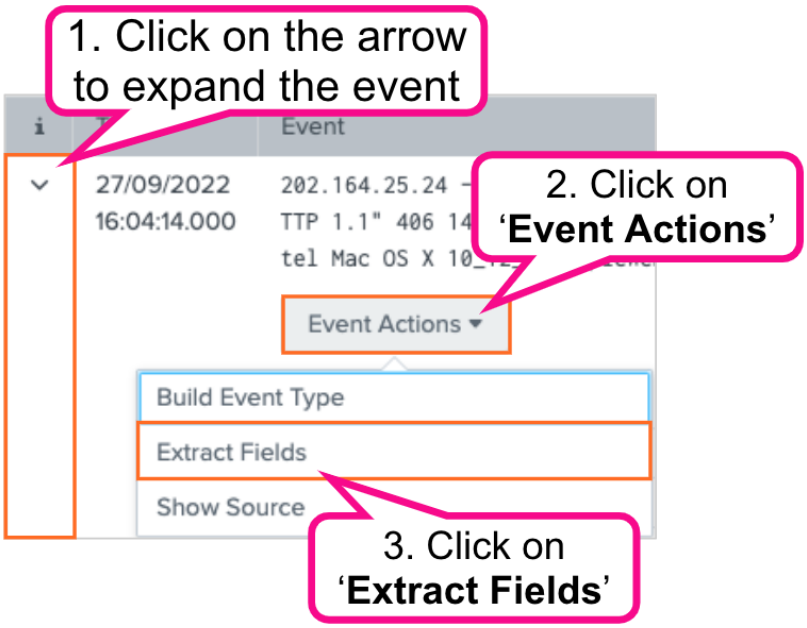
Steps

Extract a New Field

1. Click **Search** if you don't see the search bar displayed. Search for all web server events over the **Last 60 minutes**:

```
sourcetype=access_combined
```

2. Expand out one of the events by clicking on the arrow (**>**) to the left of the event timestamp. Click on the **Event Actions** dropdown list and select **Extract Fields**:



3. We have two options for extracting fields: Regular Expression or Delimiters. For this exercise, we will choose Regular Expression. Click on **Regular Expression** and then click on **Next**.

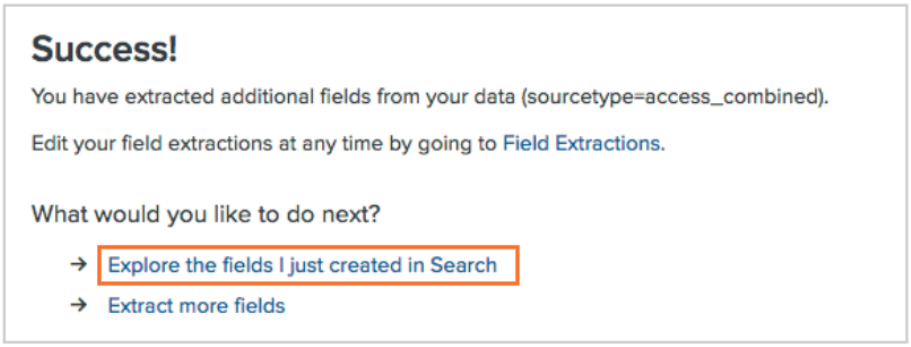
4. You will now be presented with a sample event from which to extract your field. For this exercise, we will need to extract the platform (operating system) information from each event so we can report on it. Look for the platform/operating system information in your event (e.g. Linux, Macintosh, Windows, etc.) contained in the useragent string towards the end of the event and highlight it.

   Give the new field the following name: **platform** (field names are case sensitive, so be sure to use all lowercase letters for this to make your life easier!)

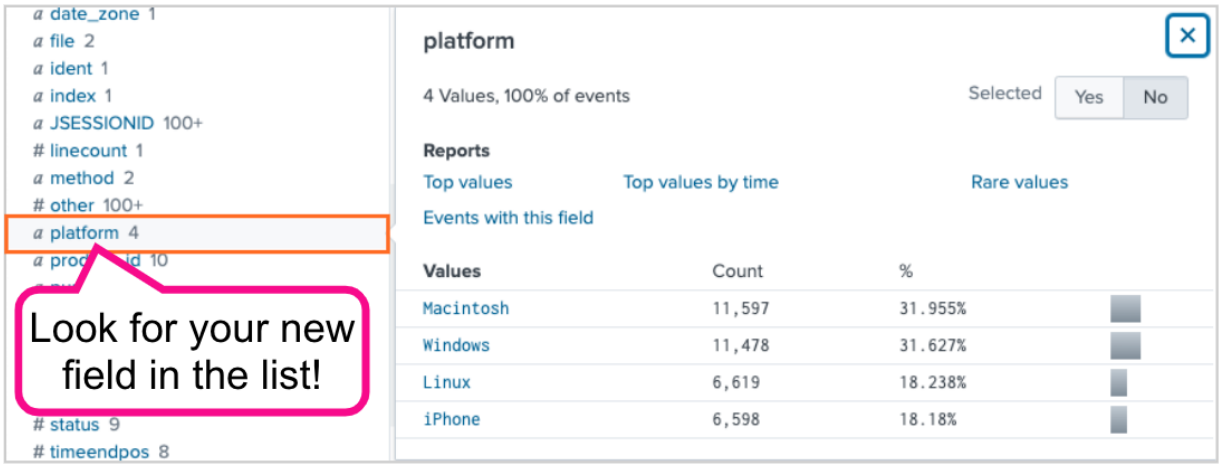
5. Click on **Add Extraction** and then click on **Next**.

6. Click on **Next** again to reach the **Save** screen. On the Save screen, click on **Finish** to save your new field extraction.

7. You should now see a **Success!** page. Click on **Explore the fields I just created in Search**.



8. Splunk will show you search results for all of your web server data over the last 24 hours. Scroll down the page and look for your new field listed on the left – you can now use it in your searches!



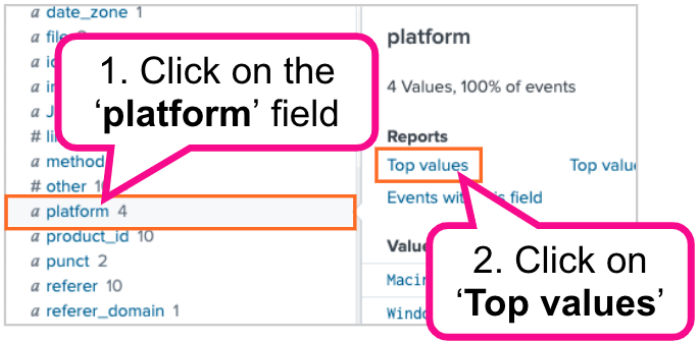
Show the most common customer operating systems

Now that we have our new field, we can use it to report for the DevOps team!

1. Search for all web server events over the **Last 60 minutes**:

```
sourcetype=access_combined
```

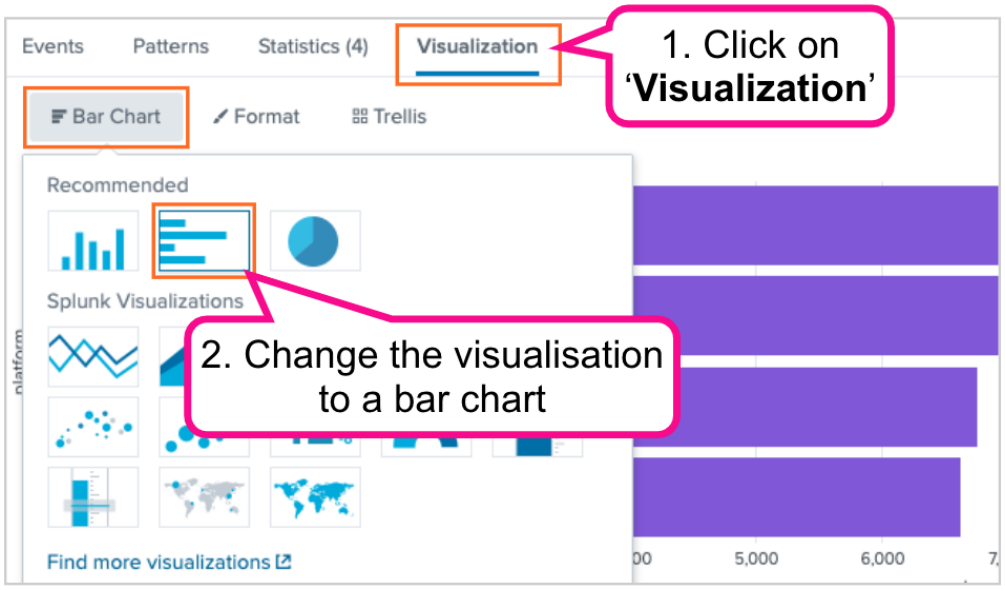
2. Scroll down the page and find the **platform** field that you just extracted. Click on the field name to display the field window, and then select **Top values**.



Splunk will automatically populate your search as follows:

```
sourcetype=access_combined | top limit=20 platform
```

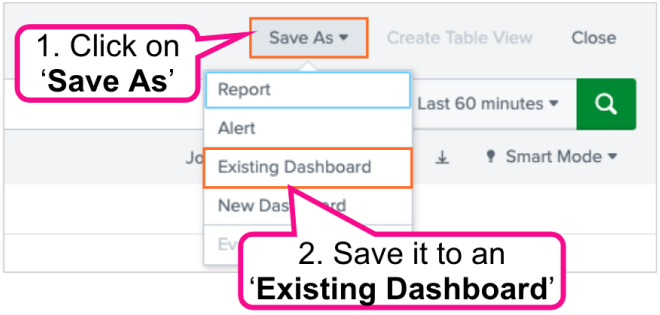
3. Select the **Visualization** tab if not already displayed and change the visualization to a **Bar Chart**.

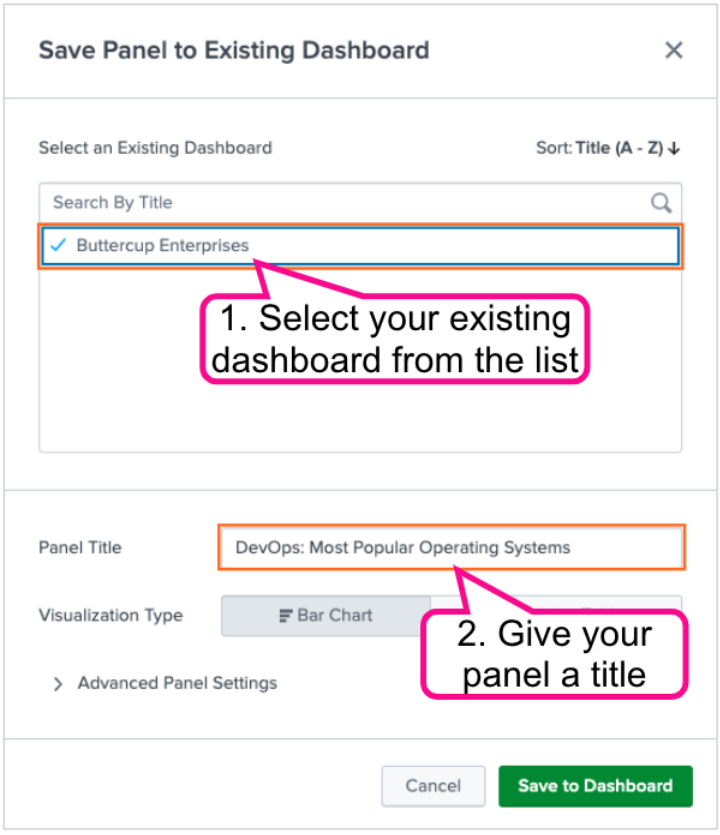


Tip: You can optionally add `showperc=f` to the `top` command to remove the 'percent' column from our table of statistics. This will help to keep the chart nice and clean when we view it on our dashboard later.

```
sourcetype=access_combined | top limit=20 platform showperc=f
```

4. When you're happy with your chart, save it to an '**Existing Dashboard**' and select the dashboard you previously created from the list. Finally, give the dashboard panel a suitable title, such as '**DevOps: Most Popular Operating Systems**' and click on **Save to Dashboard**.





Show which web browsers are experiencing the most failures

One DevOps use case down, one more to go! We now need to report on failures by web browser.

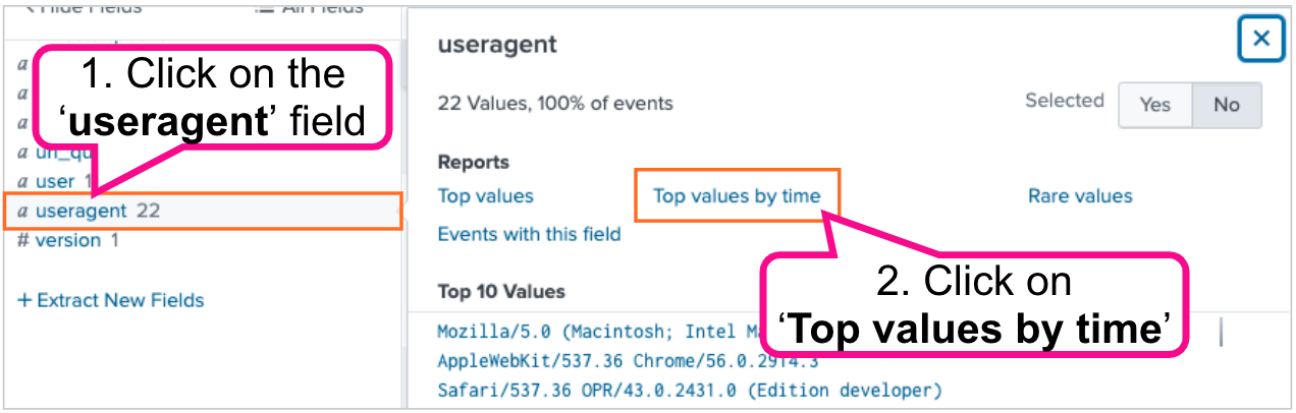
1. Search for all web server events over the **Last 60 minutes**:

```
sourcetype=access_combined
```

2. Add a search filter to return only events with a status code of 400 or higher (an event with a status value of 400 or higher is considered a failure of some kind.)

```
sourcetype=access_combined status>=400
```

3. Scroll down the page and find the **useragent** field (Note: 'useragent' is a field containing information about the web browsers that are interacting with our website.) Click on the field name to display the field window and then select **Top values by time**.



Splunk will automatically populate your search as follows:

```
sourcetype=access_combined status>=400
| timechart count by useragent limit=10
```

4. Select the **Visualization** tab if not already displayed and change the visualization to an **Area Chart**.

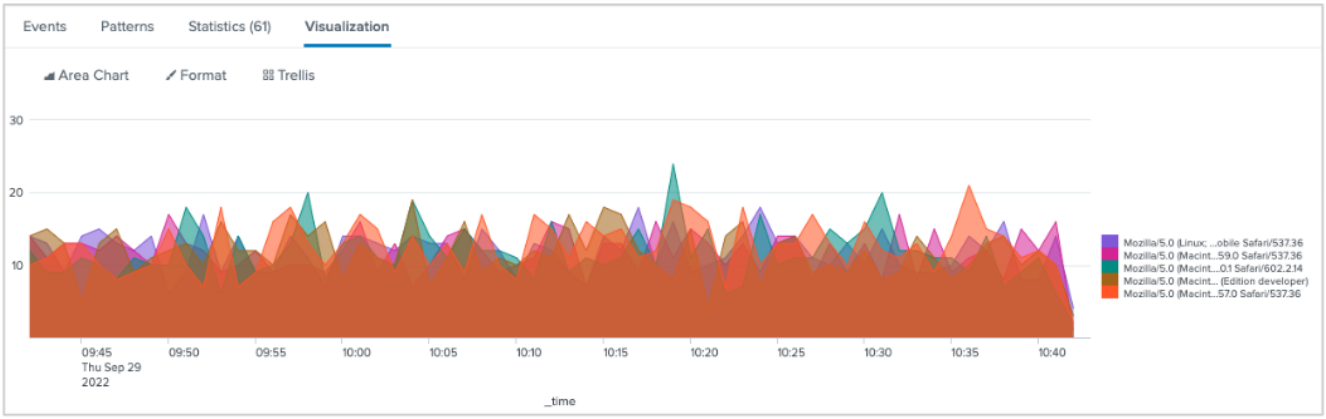
To make your chart cleaner, limit your output to the top 5 useragents by changing the "limit" to 5 in your search.

```
sourcetype=access_combined status>=400
| timechart count by useragent limit=5
```

Tip: You can optionally add `useother=f` to the `timechart` command to remove the 'OTHER' value from your chart.

```
sourcetype=access_combined status>=400
| timechart count by useragent limit=5 useother=f
```

When you're happy with your chart, add it to your dashboard and give the panel a title such as '**DevOps: Web Browsers With Most Failures**'.



Exercise 5 – Sales/Business Analytics teams: Show lost revenue from the website

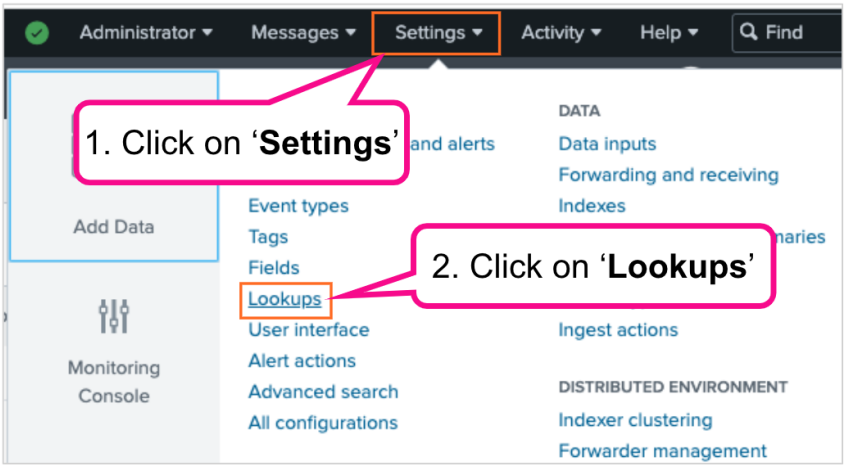
Description

Buttercup Enterprises does not have a way of seeing lost revenue from the website in real-time and the senior managers would like to track lost revenue trends throughout the day via a dashboard.

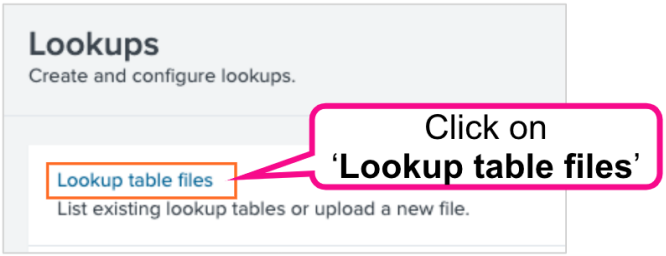
In this exercise, we will create a Single Value visualization that shows lost revenue from the company website and add this to our dashboard.

Steps

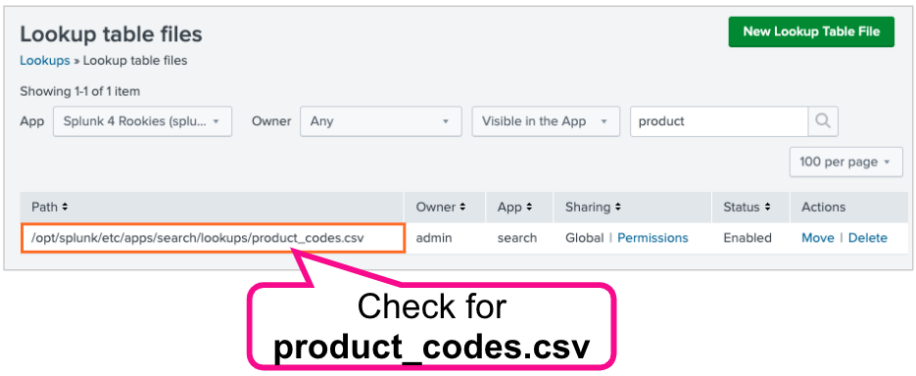
1. Go to **Settings > Lookups**.



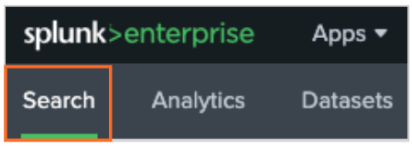
2. Click on **Lookup table files**.



Check that the '**product_codes.csv**' file exists in your environment.



3. Return to your app and make sure you are on the **Search** view.



You may want to view the contents of the lookup file to familiarise yourself with the fields and values that it contains. To do this, use the `inputlookup` command along with the name of your lookup file:

```
| inputlookup product_codes.csv
```

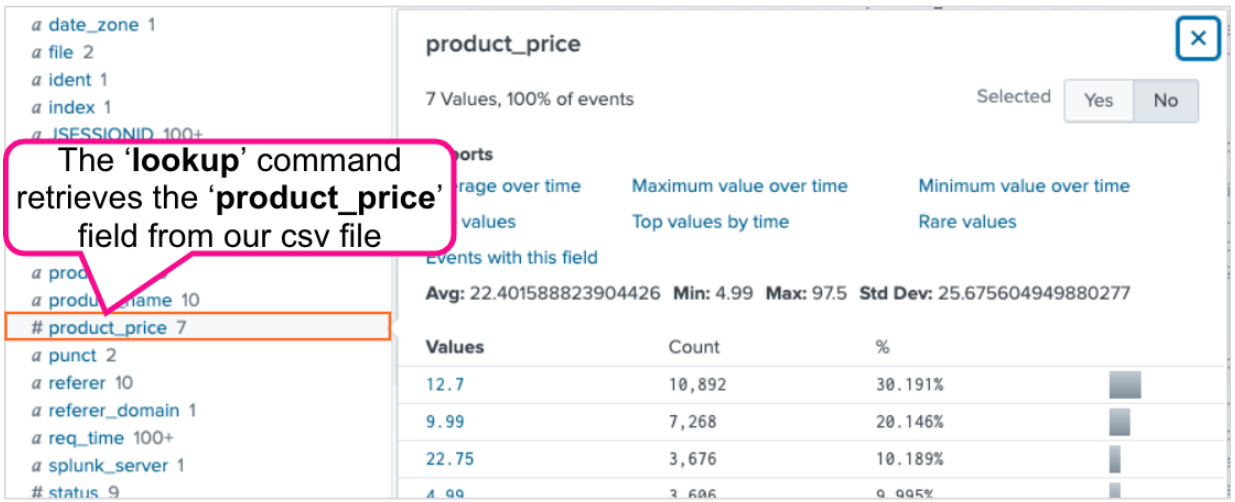
The resulting table should look like this:

category	product_id	product_name	product_price
Clothing	BS-2	Batguy Slippers	25.7
Books	MCB-5	Mad Comics- Batguy	12.7
Books	MCB-6	Mad Comics- Bronze Man	12.7
Books	MCF-3	Mad Comics- Flyman	12.7
Books	ZSG-2	Zombie Survival Guide	15.21
Clothing	CM-1	Costume- ManHawk	97.5
Gifts	DFS-2	Double Fudge Sundae	22.75
Gifts	PP-5	Pony Potpourri	9.99
Clothing	BW-3	Batguy Watch	9.99
Gifts	WPSS-2	Waterproof Scratch and Sniff	4.99

4. Now that you've checked the lookup file, you can use the `lookup` command to extract the **product_price** field from the csv file and add it to the web server purchase events by running the following search over the **Last 60 minutes**:

```
sourcetype=access_combined action=purchase | lookup product_codes.csv product_id
```

You will notice that a **product_price** field now appears under the extracted fields on the left side of the page, along with a couple of other new fields: 'category' and 'product_name'.



Splunk is pulling this new data from the csv file we specified using the **product_id** field, which exists in our data. You can now use these additional fields in your searches!

5. We now need to customise our search to focus on **failed purchase events**, since this is what we need to measure in order to calculate lost revenue. To do this, add a search filter to find events where the status is **400** or greater (i.e. an error of some kind has occurred.)

```
sourcetype=access_combined action=purchase status>=400
| lookup product_codes.csv product_id
```

6. Finally, we need to calculate the total of the **product_price** field for all of these failed purchase events **over time**. To do this we will use the `timechart` command along with the `sum` function.

   The `sum` function returns the sum of the values of a field, so we need to tell it the field we want it to work with. We will specify **product_price** as knowing the sum of this field will tell us how much total revenue we have lost.

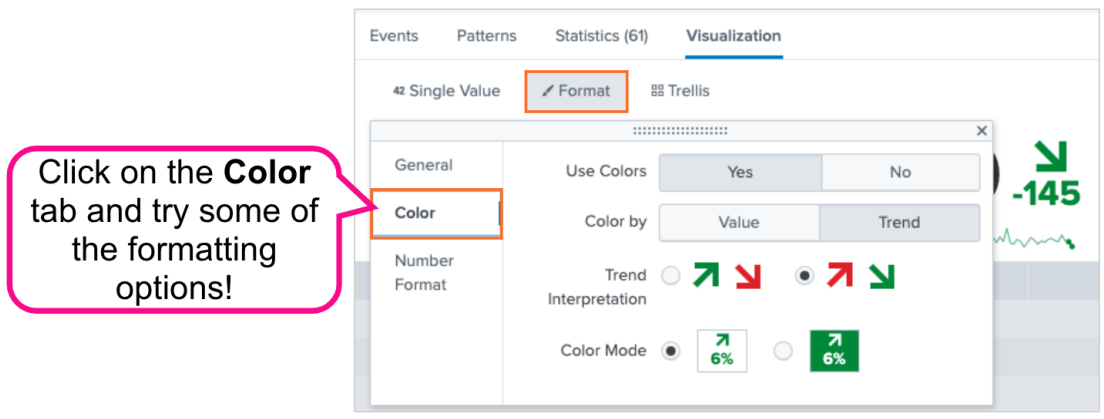
```
sourcetype=access_combined action=purchase status>=400
| lookup product_codes.csv product_id
| timechart sum(product_price)
```

Tip: Check out Splunk's Search Reference documentation page for a catalog of all the search commands and functions, along with complete syntax, descriptions, and examples of how to use them!

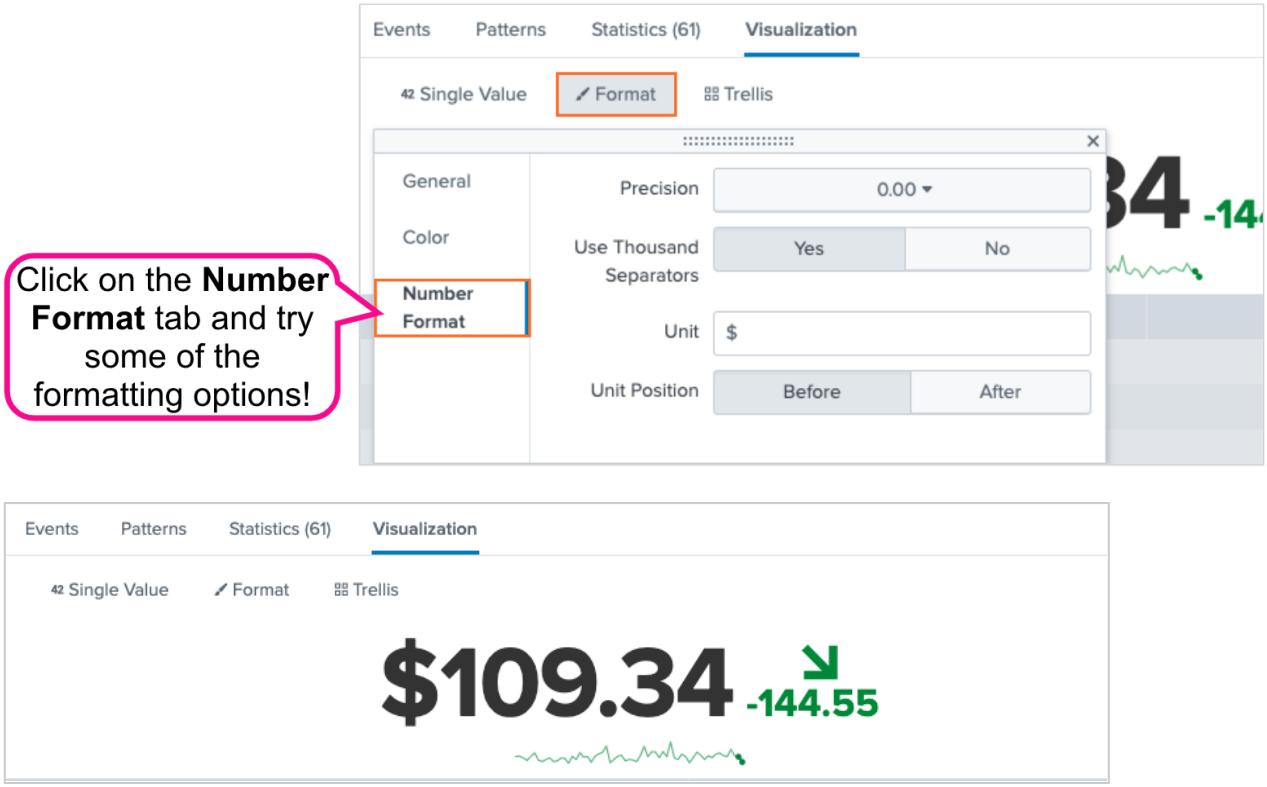
---

7. Select the **Visualization** tab if not already displayed and change the visualization to a **Single Value** visualization.

   Click on **Format** and use the side tabs to change the formatting options. Try adding some color!



8. Click on **Number Format** and add a currency unit symbol (£, $ or €) to make it clear that it's a monetary value.



Once you're happy with the visualization, add it to your dashboard and give the panel a title such as '**Business Analytics: Lost Revenue**'.

Exercise 6 – Security/Fraud teams: Show website activity by geographic location

Description

Buttercup Enterprises is based in the United States, and there is a concern that there could be many potentially fraudulent transactions coming from other countries. However, they don't currently have any visibility of where website traffic is originating from.

In this exercise, we will create a **Cluster Map** visualization that shows the geographic location of anyone connecting to the company website.

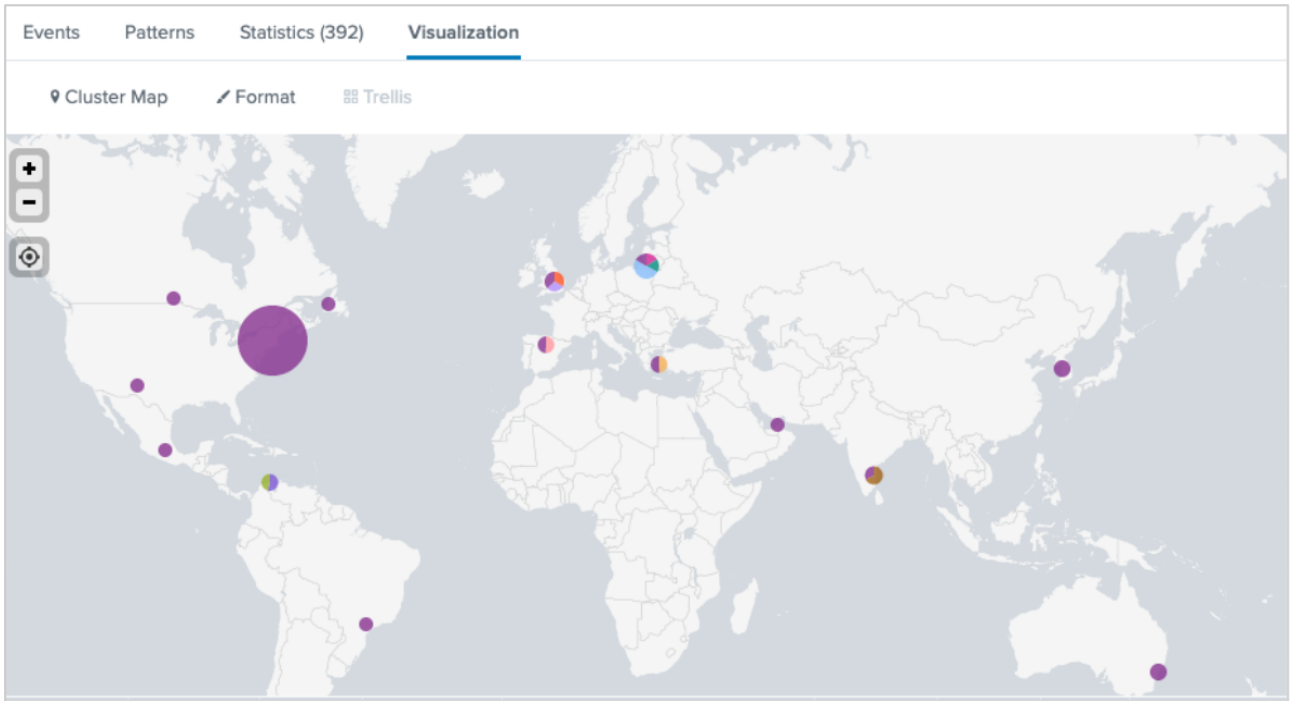
Steps

1. First, search for all web server events and use the `iplocation` and `geostats` commands to count the events by **City** (**Note:** 'City' is one of the fields in our data that's created when we use the `iplocation` command):

   ```
   sourcetype=access_combined | iplocation clientip | geostats count by City
   ```

   If it isn't selected already, click on the **Visualization** tab. For your visualization type, choose **Cluster Map**.

   You should now have a map showing the location of clients (i.e. customers) connecting to the company website.



Don't forget to add the resulting map to your dashboard and give your panel a name such as '**Security/Fraud: Customer Locations**'.

Challenge Tasks

The map we've generated shows customers from all countries, but since Buttercup Enterprises is a US-based company, the Security team may only be interested in seeing customers who are NOT located in the US.

Q1. How would you update your search to remove events coming from "**United States**" from your map?

Hints:

- The first part of every Splunk search includes an implicit `search` command, so we don't need to use a `search` command at the start of our searches. However, in Splunk if we want to apply a search filter after a pipe (" | ") has been used – such as to filter out certain results - then we will need to specify the `search` command somewhere in our search query (i.e. | `search <search terms>`)
- **Note:** Remember that when searching, if we want to use a field to filter our results, we need to make sure the field exists at that point in our search – as we've seen today, some commands will add or remove fields as Splunk steps through our search query! Look at the commands you're using and remember which fields each command may be adding or removing from your data.

Note: The challenge task solutions are at the end of this document.

Exercise 7 – Customize Your Dashboard

Description

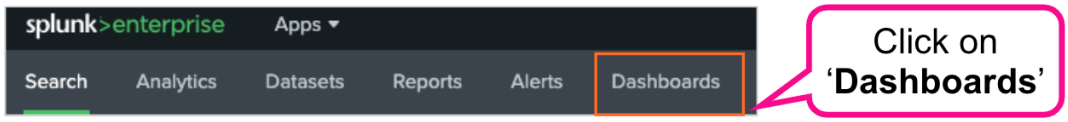
Having a dashboard with multiple panels is powerful, but the layout of your dashboard is also important to ensure that the information presented is clear and easy for users to consume.

The Buttercup Enterprises Marketing team has seen what we've built so far and have provided us with a custom background image that they would like us to use on our new dashboard. In this exercise we will upload the custom background image and rearrange our panels to work with the new background. Finally, we will configure each of our dashboard panels to use the global time picker so it's all ready to share with the business!

Steps

Add a Custom Background Image to Your Dashboard

1. First, open your dashboard. To do that, click on **Dashboards** in the top menu bar.

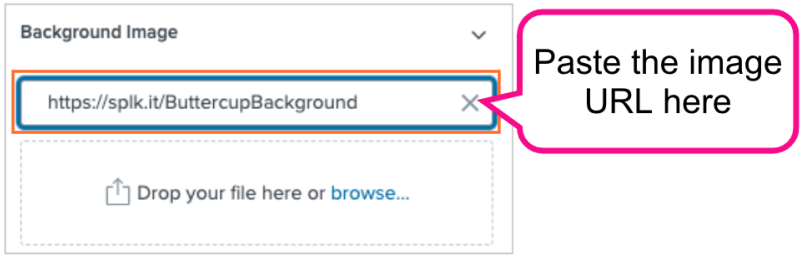


2. Click on the name of your dashboard to open it.

3. Click on the **Edit** button to put your dashboard into edit mode.

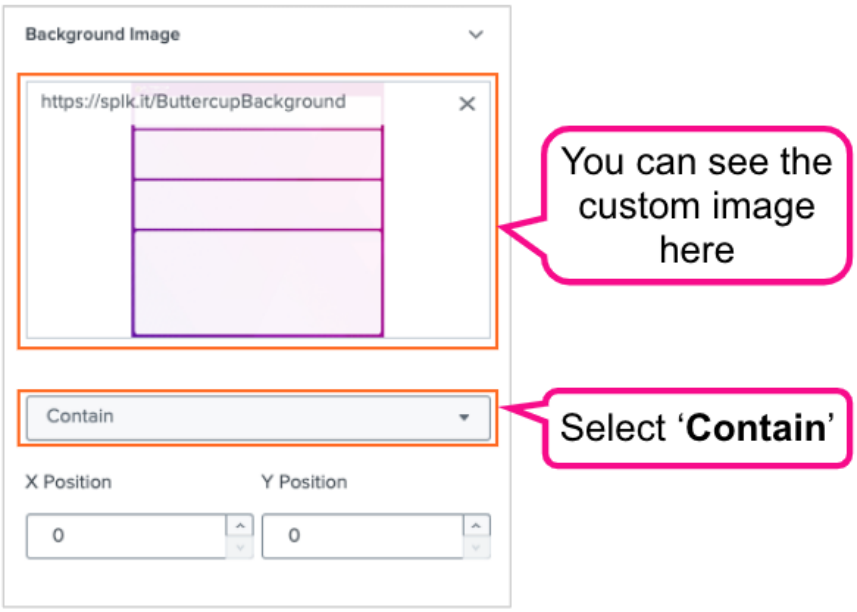
4. Locate the **Background Image** section and copy/paste the following image URL into the '**Enter URL**' box:

```
https://splk.it/ButtercupBackground
```

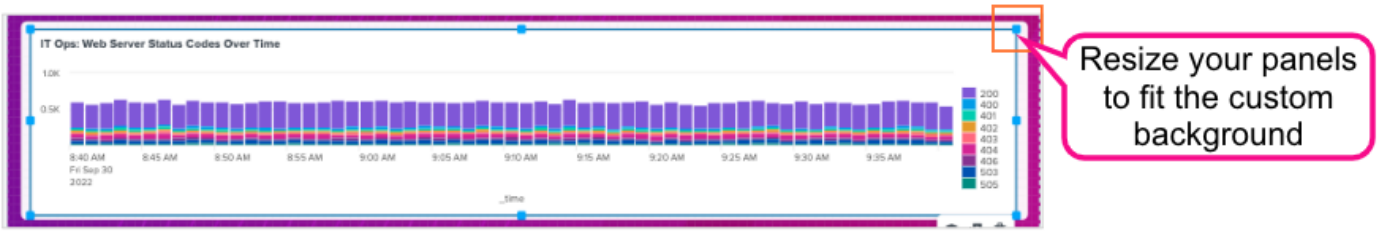


To upload the image, either hit the Enter key on your keyboard or click anywhere on your dashboard.

To ensure that our custom image is contained within the dimensions of our dashboard, click on the dropdown list beneath the image preview and select **‘Contain**’.

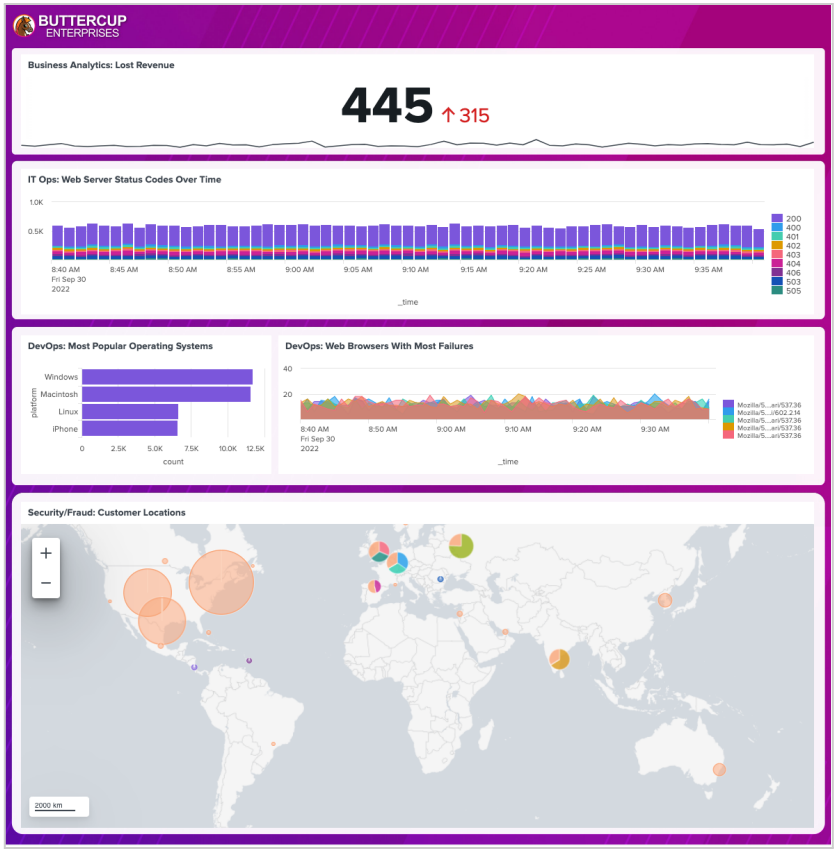


5. Now click on each dashboard panel and drag the blue squares that appear around the edges of the panels to resize them to fit within the areas on your custom background image.



Be sure to click on **Save** when you’ve finished rearranging everything!

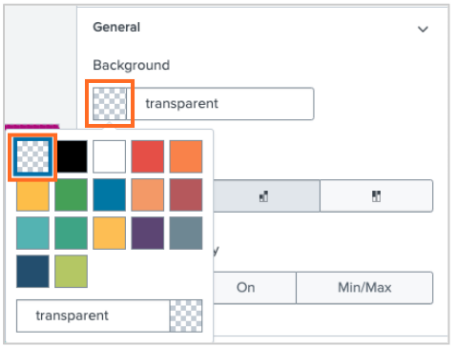
When you're finished, your dashboard should now look something like this:



6. Finally, since we have a nice colored background to show off we can set each panel to be transparent to help the color to shine through! To do this, click on a dashboard panel and in the Configuration panel on the right find the **Coloring** section.

   Find the '**Background**' or '**Static Background**' option for your panel (the name will vary from visualization to visualization) and change the background color to be transparent. Repeat this step for each dashboard panel. Note that the Cluster Map visualization has no background color option so you can ignore this panel.

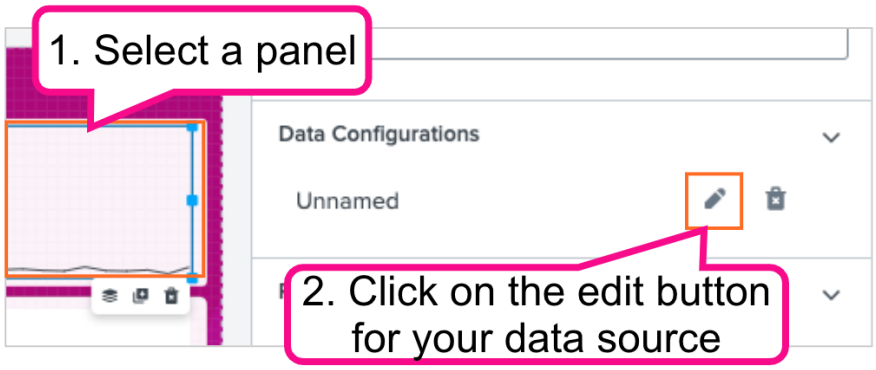
Note: Some visualizations may have a slightly different name for the background color setting, for example:



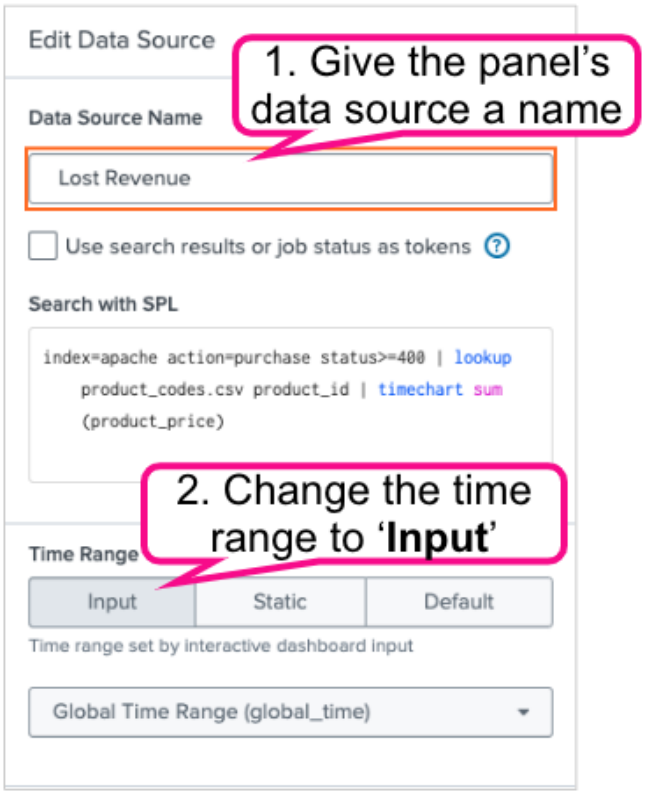
Link Your Dashboard Panels to the Global Time Picker

The global time picker is included in all new dashboards by default and allows you to control the time range of all dashboard panels from a single place. Since each of our panels was using a static time range (i.e. **Last 60 minutes**) when we added them to our dashboard we just need to switch each panel to use the global time picker instead.

1. With your dashboard in edit mode, click a dashboard panel and in the Configuration panel on the right find the **Data Configurations** section. Click on the pencil icon to edit the '**Unnamed**' data source.

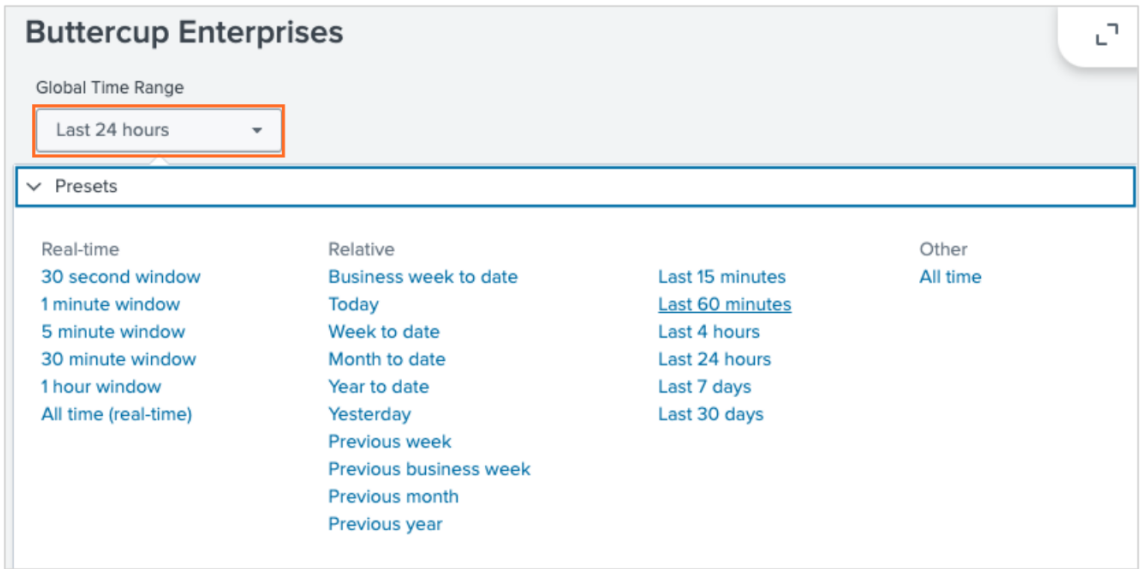


2. To make changes to the data source we will need to give each data source a name. For simplicity, use the name of the dashboard panel. For example, if you're working with our Single Value visualization panel, use 'Lost Revenue' as the data source name.



3. Click on **Apply & Close** to save your panel changes. Repeat this step for each dashboard panel and save your dashboard.

   Now that you've linked all your panels to the global time picker, click on **Save** and then click on **View** to view your updated dashboard. Try changing the search time range for your dashboard by choosing different time ranges from the dropdown list. All of your panels should update to reflect the time setting.



Challenge Task Solutions

Below are suggested solutions to the challenge tasks contained in this lab guide. Don't worry if you used a slightly different method – there are often multiple ways of reaching the same result!

Start Searching in Splunk

- Q1. How can we find events with a status of **200** that are not purchase events?

  Solution:

  ```
  status=200 action!=purchase
  ```

  Note: `status=200 NOT action=purchase` will also work for this exercise but this is not a good way of performing this query due to the way that the `NOT` operator works (see https://docs.splunk.com/Documentation/Splunk/latest/Search/NOTexpressions for a full explanation of the differences between these two methods.)

- Q2. How can we find events where someone had an error when trying to either add an item or remove an item from their cart?

  Solution:

  ```
  sourcetype=access_combined status>=400 (action=addtocart OR action=remove)
  ```

Exercise 6 – Security/Fraud teams: Show any activity on the website coming from outside the United States

- Q1. How would you remove events coming from "**United States**" from your map?

  Solution:

  ```
  sourcetype=access_combined | iplocation clientip
  | search Country!="United States" | geostats count by City
  ```