

# Forward- looking statements

This presentation may contain forward-looking statements that are subject to the safe harbors created under the Securities Act of 1933, as amended, and the Securities Exchange Act of 1934, as amended. All statements other than statements of historical facts are statements that could be deemed forward-looking statements. These statements are based on current expectations, estimates, forecasts, and projections about the industries in which we operate and the beliefs and assumptions of our management based on the information currently available to us. Words such as “expects,” “anticipates,” “targets,” “goals,” “projects,” “intends,” “plans,” “believes,” “momentum,” “seeks,” “estimates,” “continues,” “endeavors,” “strives,” “may,” variations of such words, and similar expressions are intended to identify such forward-looking statements. In addition, any statements that refer to (1) our goals, commitments, and programs; (2) our business plans, initiatives, and objectives; and (3) our assumptions and expectations, including our expectations regarding our financial performance, products, technology, strategy, customers, markets, acquisitions and investments are forward-looking statements. These forward-looking statements are not guarantees of future performance and involve significant risks, uncertainties and other factors that may cause our actual results, performance or achievements to be materially different from results, performance or achievements expressed or implied by the forward-looking statements contained in this presentation. Readers are cautioned that these forward-looking statements are only predictions and are subject to risks, uncertainties, and assumptions that are difficult to predict, including those identified in the “Risk Factors” section of Cisco’s most recent report on Form 10-Q filed on May 21, 2024 and its most recent report on Form 10-K filed on September 7, 2023. The forward-looking statements made in this presentation are made as of the time and date of this presentation. If reviewed after the initial presentation, even if made available by Cisco or Splunk, on Cisco or Splunk’s website or otherwise, it may not contain current or accurate information. Cisco and Splunk undertake no obligation to revise or update any forward-looking statements for any reason, except as required by law.

In addition, any information about new products, features, functionality or our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only and shall not be incorporated into any contract or other commitment or be relied upon in making a purchasing decision. We undertake no commitment, promise or obligation either to develop the features or functionalities described, in beta or in preview (used interchangeably), or to include any such feature or functionality in a future release. The development, release, and timing of any features or functionality described for our products remains at our sole discretion.

---

Splunk and Splunk> are trademarks and registered trademarks of Splunk Inc. in the United States and other countries. All other brand names, product names, or trademarks belong to their respective owners. © 2024 Splunk Inc. All rights reserved.

# Splunk App Building 101: Learn, Create and Navigate the Ecosystem

DEV1885B



**Bring on  
the future.**



- Dad | Husband | Geek
- Innovative Product Leadership, Stanford GSB
- Masters in Cybersecurity, Saint Peter's University
- Photographer | help people see the world through my lens! 📷
- Cooking + Meal Prep Hacks | Designated vegetable chopper 🍳
- Occasionally helping build homes | Habitat for Humanity 🏠
- Fine tuning music on [7.2.4](#) 🎵

*“Love being in the intersection of customer-led product innovation and growing people/organization ”*



#MillionDataPoints



**Mayur Pipaliya**  
Sr. Manager, GSS FDSE

Years at Splunk: 6 years 2 months

Located: San Francisco Bay Area

Patents filed: 4+

mayur@splunk.com | #team-fdse

Last 17 years: Entrepreneur → Startup Advisor / Building Products → Splunk Partner Hackathons → SOC Leader → Splunk

```
<link href ="  
minding.my.own.  
business.com">
```



**COMARCH**



**Brown Brothers  
Harriman**



**#MillionDataPoints**



**Rafał Piekarz**  
Leader, GSS FDSE

Years at Splunk: 1 year 4 months

Located: Kraków. Poland

Solution designer, problem solver, ITSM  
trainer and practitioner, overthinker,  
astrophysics and general science fan 📡

[rpiekarz@splunk.com](mailto:rpiekarz@splunk.com) | [#team-fdse](https://twitter.com/team-fdse)

My path: App Developer → System Integrator → Project Manager → Service Manager → Tools and Automation Manager → Engineering Manager at Splunk

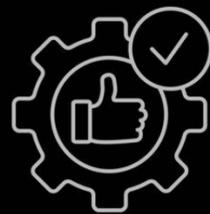
# Splunk App Building 101

Learn, Create and Navigate the Ecosystem

# Session objectives.



Understand App  
Development  
Ecosystem.



Learn best  
practices around  
Splunk App  
development.



Open you Apps  
for contributions.



Think of your  
next app!

# Milestones

- Developer Landscape : Splunk Products
- Hands-on ( GDI ) : Building Splunk Add-on
- Hands-on ( Visualize ) : Building Splunk Dashboard App
- Best Practices : Building Apps
- Giveaways
- Q&A



# Splunk Developer Landscape

**Forest View:** Splunk Core | SOAR | O11y



# Why? Because you can!

Build Custom Apps, Add-ons, Connector, and Collector.

## → Full Stack Development

Gain insights and value from data

Get data into

Create new way to visualize data

Manipulate data when searching

Integrate and extend to other software

Take action/alert on data

Dispatch Alerts to worker nodes

SOAR Connectors and Playbooks

- Knowledge Objects

- Modular Inputs, HEC, OTEL Collector

- Modular Visualizations, Splunk UI Toolkit

- Custom Search Commands

- REST Endpoints

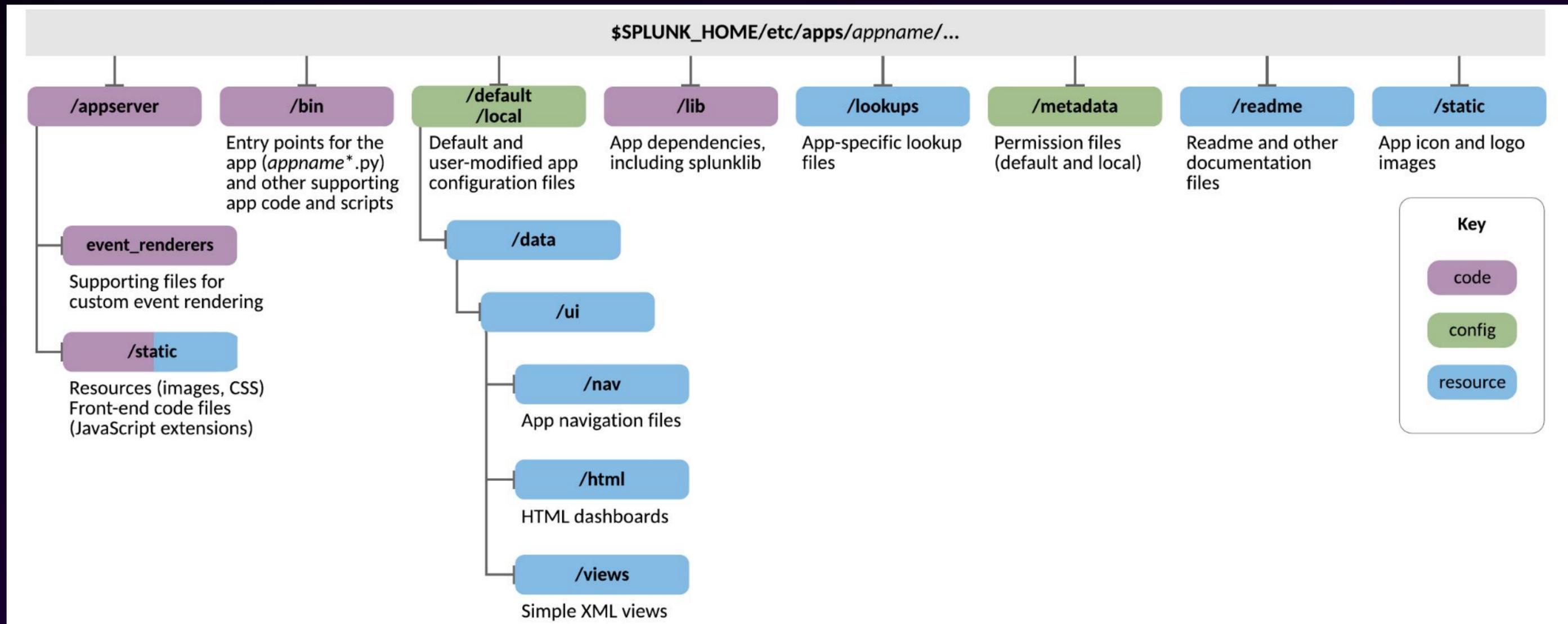
- Modular Alerts

- Adaptive Response Relay

- Epitome of Automation

# Structure of an App

## Right Files in the Right Locations



# Apps vs Add-ons

(a.k.a. TAs)

## Splunk Apps

- Designed to visualize and analyze data by users
- Operated primarily from search heads
- Usually include:
  - dashboards
  - reports
  - alerts
  - access management
  - navigation
  - custom search commands
  - alert actions

## Splunk Add-ons

- Designed to provide specific capabilities around getting data in (GDI), data transformation, normalization and enrichment.
- Often deployed on universal/heavy forwarders, indexers
- Usually include:
  - input configuration
  - parsing and transformation
  - lookup files for data enrichment
  - scripts for modular inputs

# Hands-on Ahead!



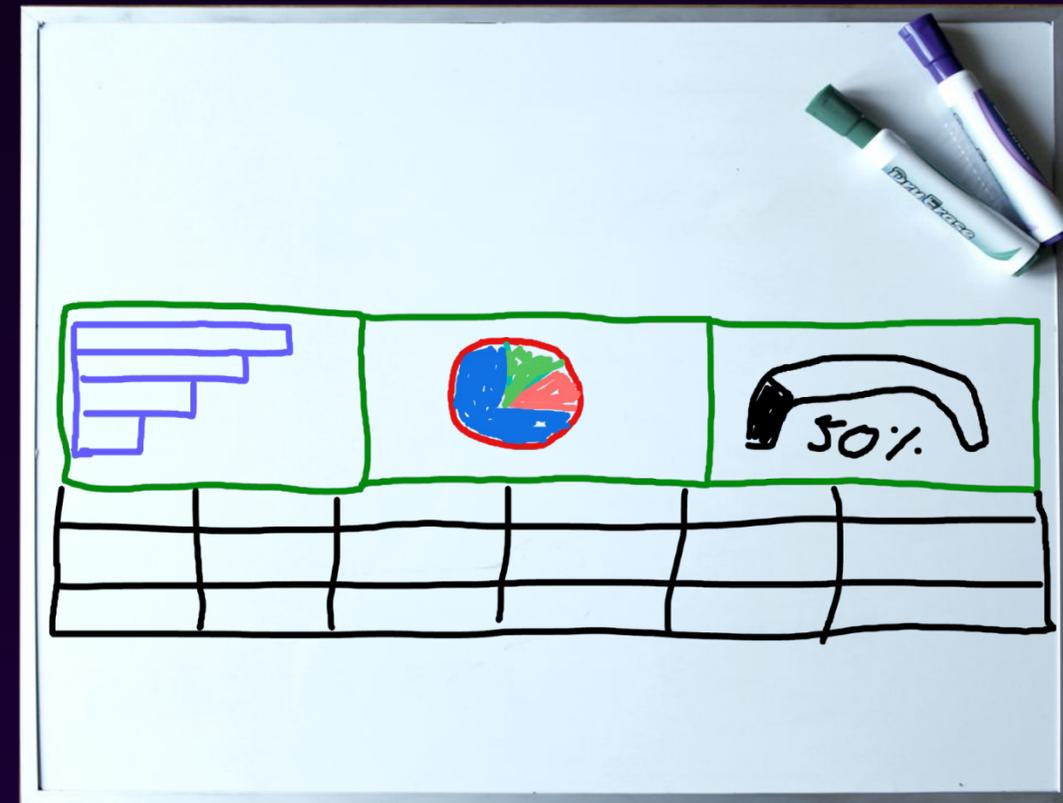
**.conf24**  
splunk>

**Bring on  
the future.**

# Scenario - activities for bored people

```
{  
  "accessibility": "0.1",  
  "activity": "Study a foreign language",  
  "key": 12312123,  
  "link": "",  
  "participants": 1,  
  "price": 0,  
  "type": "education"  
}
```

Get data in - API endpoint



Visualize data for the analysis

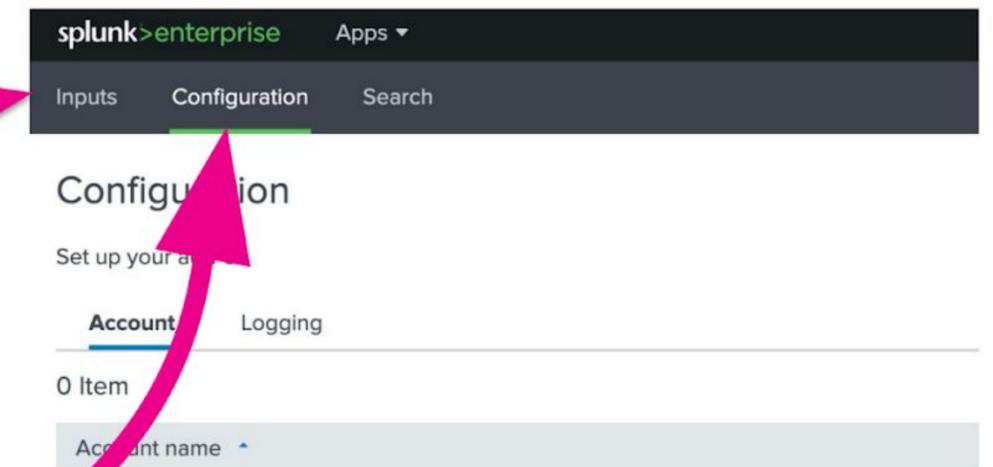
# Universal Configuration Console

(a.k.a. UCC)

- A framework for add-on generation
- UI framework based on React
- Powered by Splunk libraries: **solnlib** and **splunktaucclib**

## globalConfig.json

```
{
  "meta": {
    "name": "TA-conf21",
    "displayName": "Splunk .conf21 Example App",
    "version": "1.0.0",
    "apiVersion": "3.2.0",
    "restRoot": "TA_conf21",
    "schemaVersion": "0.0.2"
  },
  "pages": {
    "inputs": {
      "title": "Inputs",
      "description": "Manage your data inputs",
      "table": { ...
    },
    "services": [ ...
  ],
  "configuration": {
    "title": "Configuration",
    "description": "Set up your add-on",
    "tabs": [ ...
  ]
}
```



← globalConfig.json  
schema

# Add-on Building Checklist

To make your building experience easier, use **Splunk Web UI** and **Online IDE!**

## Web-based development (recommended)

- Web browser
  - Splunk Web UI
  - code-server

## Local development

- Web browser
  - Splunk Web UI (SHOW)
- Python 3.8+
- Python venv
- Ability to install software on your PC
- IDE (e.g. VSCode) or text file editor
- (optional) Splunk Extension for VSCode

# Lab Guide and Supplementary Materials



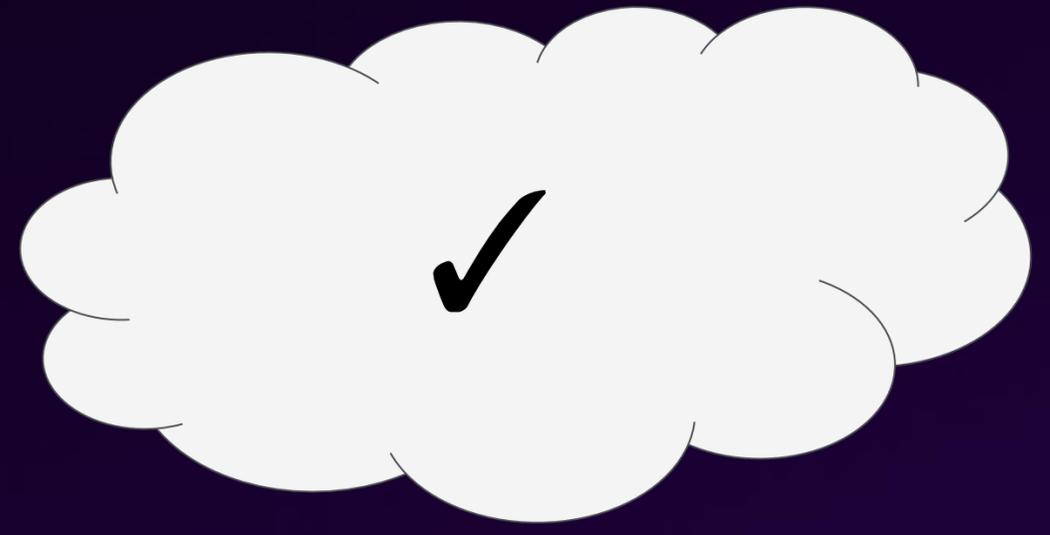
<https://splk.it/dev1885b-resources>

# Exercise

# 1+2



# Check your environment



# Initialize and Build Your Add-on

```
# create and activate venv for your app development
```

```
python3 -m venv .venv  
source .venv/bin/activate
```

```
# install UCC Framework on you venv
```

```
pip install splunk-add-on-ucc-framework=="5.43.0"
```

```
# initialize your add-on
```

```
ucc-gen init --addon-name "demo_addon_for_activities" --addon-display-name  
"Demo Add-on for Activities" --addon-input-name activities
```

```
# initial build
```

```
cd demo_addon_for_activities  
ucc-gen build --ta-version "0.0.1"
```

```
# package the add-on to be able to install it
```

```
ucc-gen package --path ./output/demo_addon_for_activities
```

# Install Add-on

Code Server IDE > Terminal

```
# extract add-on package to Splunk apps folder
```

```
tar -xvzf demo_addon_for_activities-0.0.1.tar.gz -C /opt/splunk/etc/apps/
```

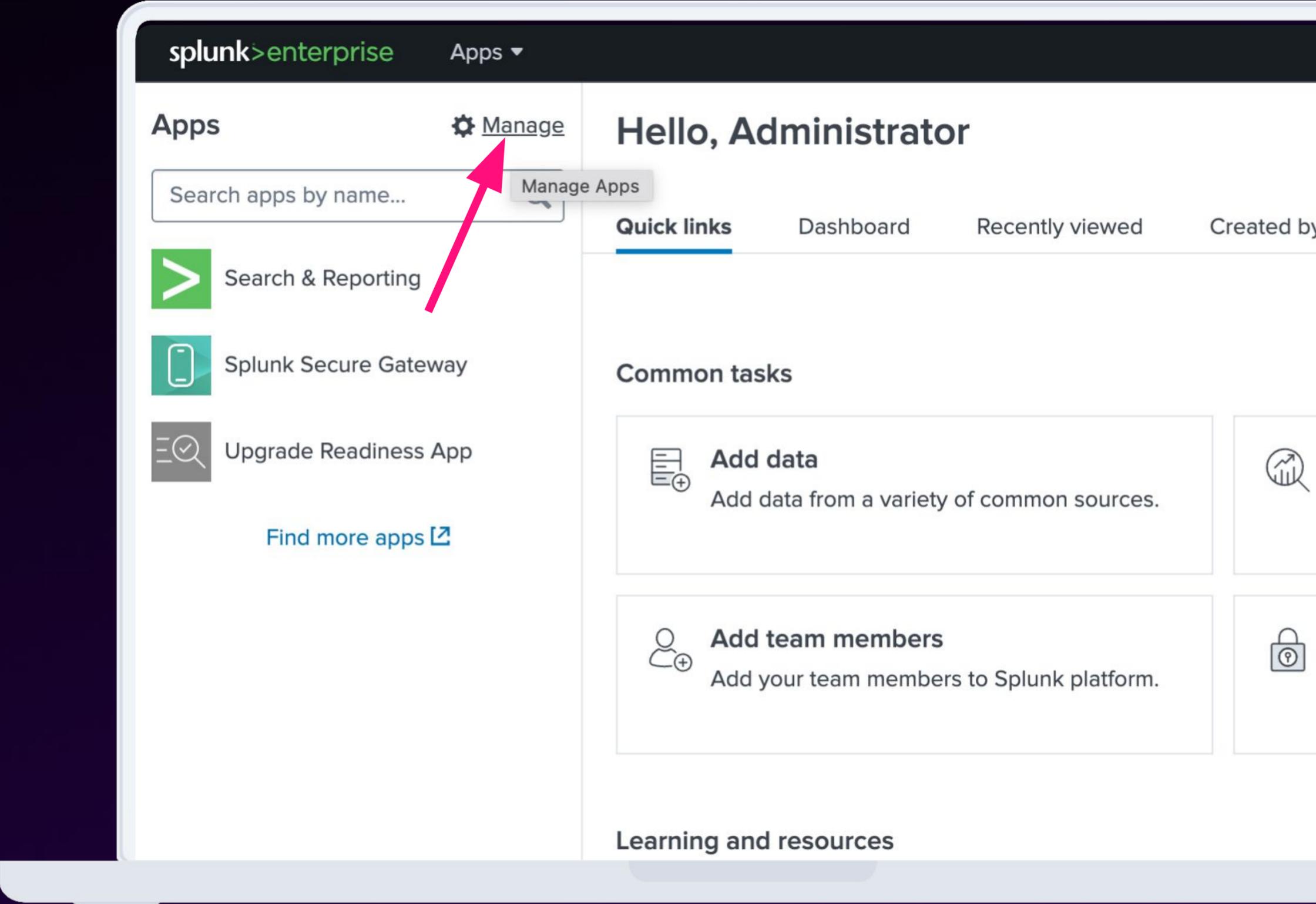
```
# restart Splunk
```

```
sudo /opt/splunk/bin/splunk restart
```

# Install Add-on

Splunk UI - Step 1

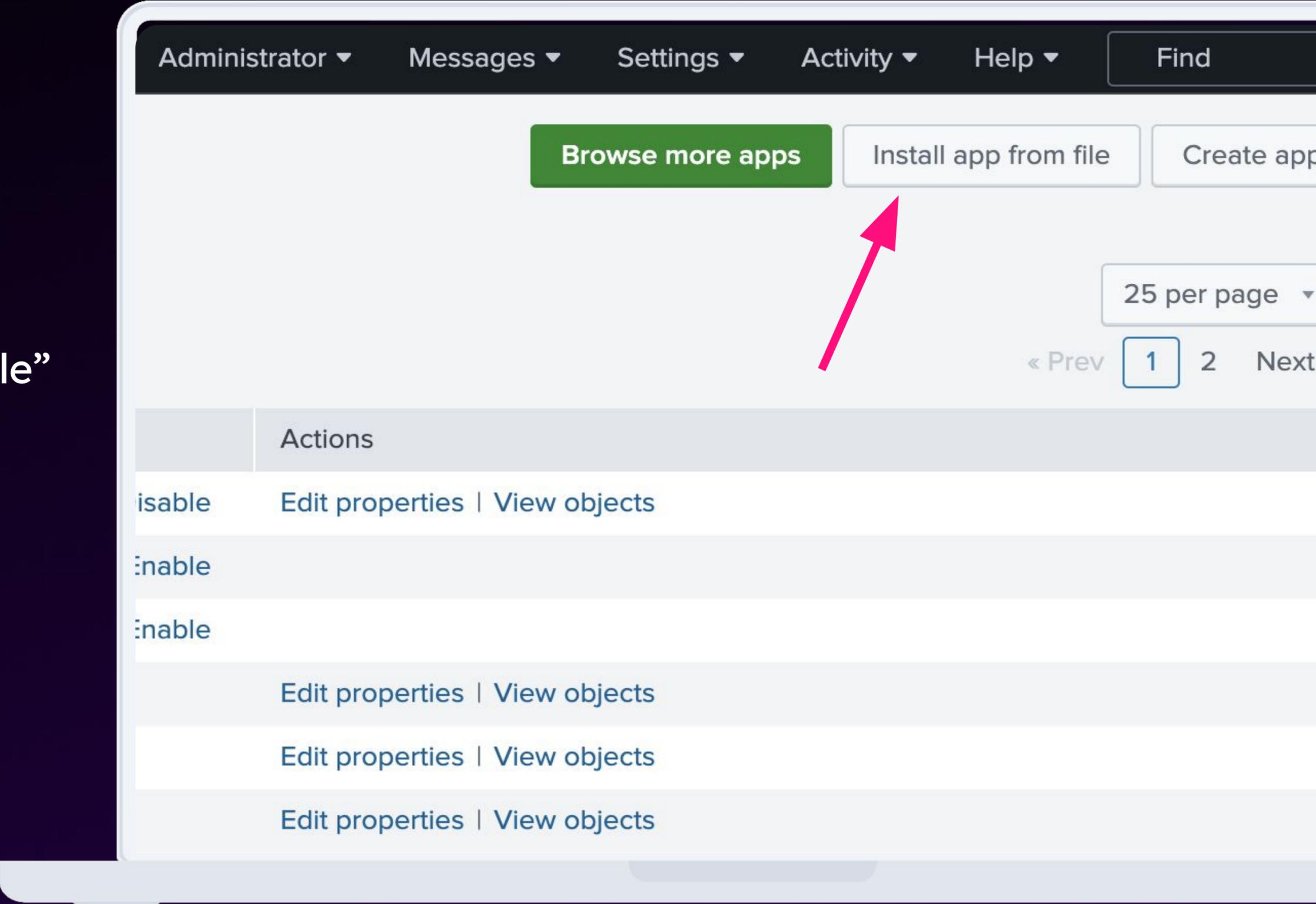
Let's click on "Manage".



# Install Add-on

Splunk UI - Step 2

Click on “Install app from file” button.

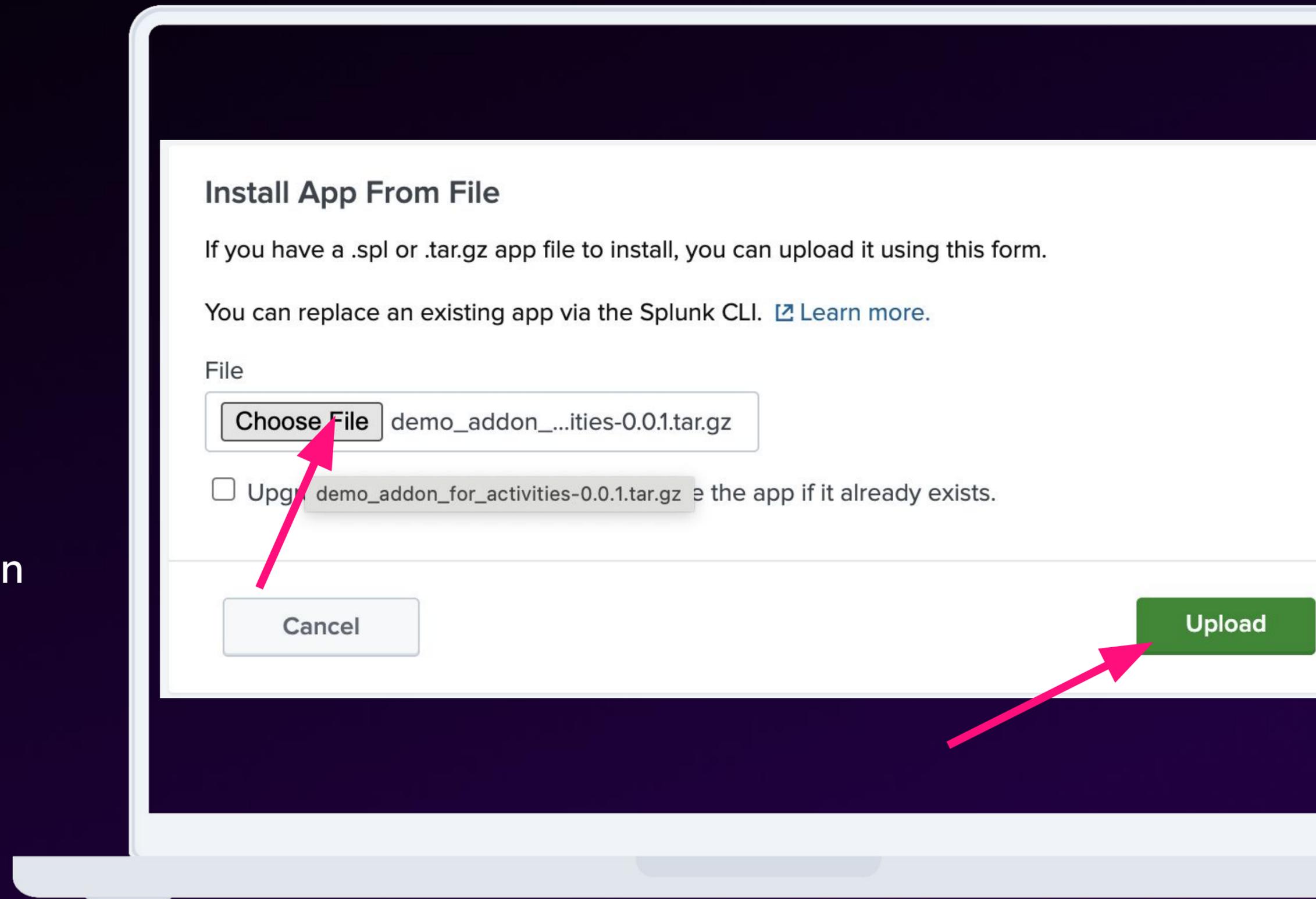


# Install Add-on

## Splunk UI - Step 3

Click on “Choose File” to browse TAR.GZ file.

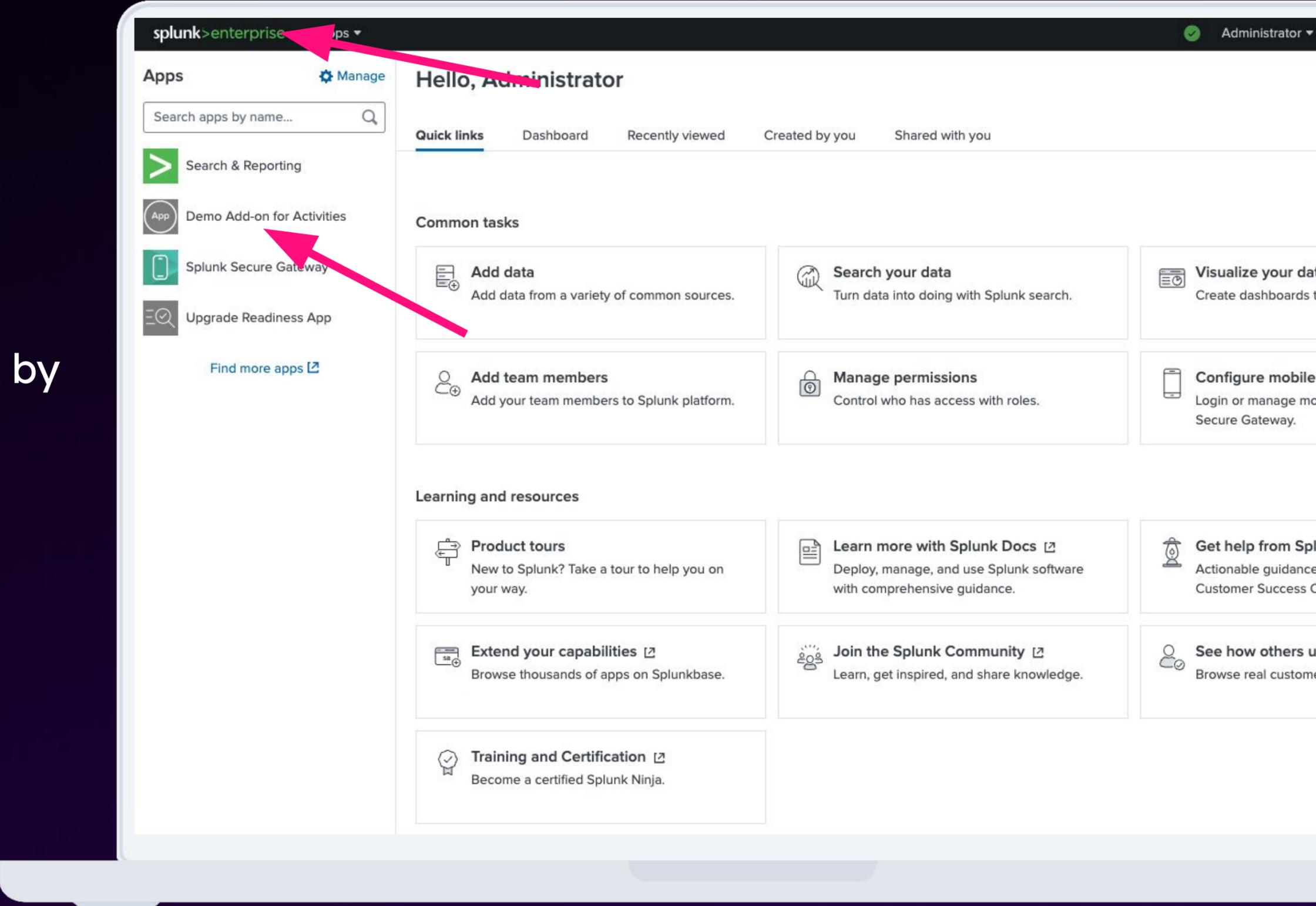
Once app is chosen, click on “**Upload**” button.



# Let's Check the Progress!

Let's Go back to main page by clicking on **top-left Splunk logo**.

Add-on should be installed!



# Add-on TODO

## → Configuration

- We want to:
  - Set up account for an endpoint
- We need:
  - URL field (with validation)

## Inputs

- We want to:
  - Get the data from activities endpoint
  - Ingest to a specified index
  - sourcetype = activities
- We need:
  - Account field (list of configured accounts)
  - Index field (selection from available indexes)
  - Sourcetype field (free text with validation)

## Python

- We want to:
  - Collect data from an endpoint
  - Ingest data to Splunk
- We need:
  - read configuration data from account and inputs
  - handle request to the endpoint
  - handle writing events to Splunk

# Exercise

# 3



# Edit

## globalConfig.json

3 main elements of globalConfig schema.

```
{ } globalConfig.json 9+ X
{ } globalConfig.json > ...
1  {
2    "pages": {
3  >   "configuration": { ...
58  },
59  >   "inputs": { ...
163  },
164 >   "dashboard": { ...
170  }
171 },
172 "meta": {
173   "name": "demo_addon_for_activities",
174   "restRoot": "demo_addon_for_activities",
175   "version": "0.0.1",
176   "displayName": "Demo Add-on for Activities",
177   "schemaVersion": "0.0.6",
178   "_uccVersion": "5.43.0"
179 }
180 }
```

# Edit

globalConfig.json

Observe “pages” > “configuration”  
> “tabs” > “accounts”

(no change!)

```
{ } globalConfig.json > { } pages > { } configuration
1  {
2    "pages": {
3      "configuration": {
4        "tabs": [
5          {
6            "name": "account",
7            "table": {
8              "actions": [
9                "edit",
10               "delete",
11               "clone"
12             ],
13             "header": [
14               {
15                 "label": "Name",
16                 "field": "name"
17               }
18             ]
19           },

```

# Edit

## globalConfig.json

### Configure entities

- Account Name field

(no change!)

```
{ } globalConfig.json > { } pages > { } configuration > [ ] tabs
  2   "pages": {
  3     "configuration": {
  4       "tabs": [
  5         {
20         "entity": [
21           {
22             "type": "text",
23             "label": "Name",
24             "validators": [
25               {
26                 "type": "regex",
27                 "errorMsg": "Account Name must begin with a letter and
28                 "pattern": "^[a-zA-Z]\\w*$"
29               },
30               {
31                 "type": "string",
32                 "errorMsg": "Length of input name should be between 1 a
33                 "minLength": 1,
34                 "maxLength": 100
35               }
36             ],
37             "field": "name",
38             "help": "A unique name for the account.",
39             "required": true
40           },
```

# Edit

## globalConfig.json

### Configure entities

- Endpoint URL field  
(replacing API Key field)

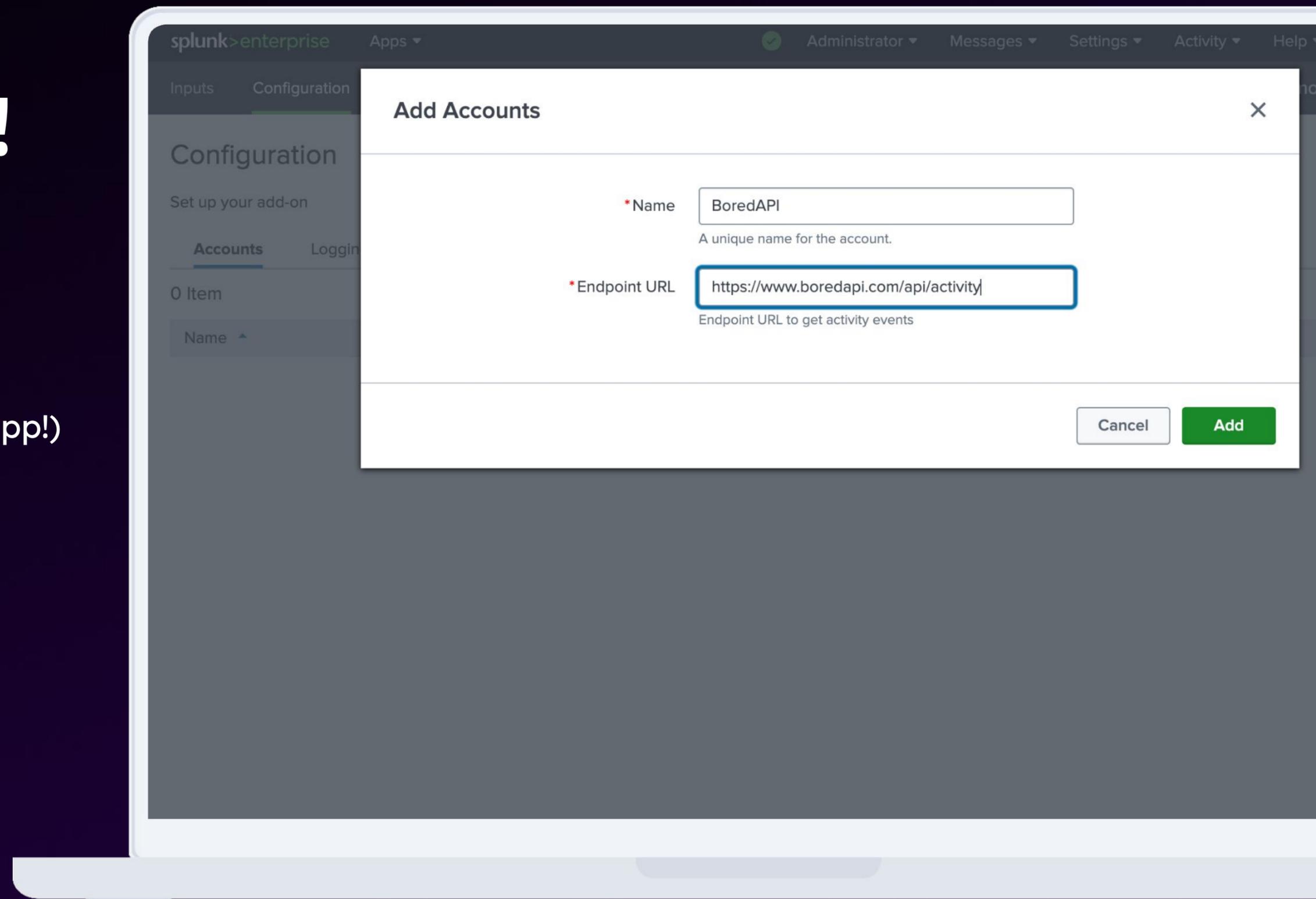
```
{ } globalConfig.json > { } pages > { } configuration > [ ] tabs > { } 0 > [ ] entity
    {
      "type": "text",
      "label": "Endpoint URL",
      "field": "endpoint_url",
      "help": "Endpoint URL to get activity events",
      "required": true,
      "validators": [
        {
          "type": "url",
          "errorMsg": "Input must an URL address."
        }
      ]
    }
  }
```

# Let's Check the Progress!

Build new version 0.0.2

Package and reinstall (Upgrade app!)

Create an account ->



# Add-on TODO



## Configuration

- We want to:
  - Set up account for an endpoint
- We need:
  - URL field (with validation)

## Inputs

- We want to:
  - Get the data from activities endpoint
  - Ingest to a specified index
  - sourcetype = activities
- We need:
  - Account field (list of configured accounts)
  - Index field (selection from available indexes)
  - Sourcetype field (free text with validation)

## Python

- We want to:
  - Collect data from an endpoint
  - Ingest data to Splunk
- We need:
  - read configuration data from account and inputs
  - handle request to the endpoint
  - handle writing events to Splunk

# Exercise

# 4



# Observe

## globalConfig.json

Configure inputs for activities

- Input Name field

(no changes!)

```
{ } globalConfig.json > { } pages > { } inputs > { } table > [ ] header > { } 1
2      "pages": {
64      "inputs": {
65          "services": [
66              {
67                  "name": "activities",
68                  "entity": [
69                      {
70                          "type": "text",
71                          "label": "Name",
72                          "validators": [
73                              {
74                                  "type": "regex",
75                                  "errorMsg": "Input Name must begin with a letter and consist of",
76                                  "pattern": "^[a-zA-Z]\\w*$"
77                              },
78                              {
79                                  "type": "string",
80                                  "errorMsg": "Length of input name should be between 1 and 100",
81                                  "minLength": 1,
82                                  "maxLength": 100
83                              }
84                          ],
85                          "field": "name",
86                          "help": "A unique name for the data input.",
87                          "required": true
88                      }
89                  ]
90              }
91          ]
92      }
93  }
```

# Observe

## globalConfig.json

Configure inputs for activities

- Interval field
- Account to use field

(no changes!)

```
{ } globalConfig.json > { } pages > { } inputs > [ ] services > { } 0 > [ ] entity > { } 1
  2     "pages": {
  64         "inputs": {
  65             "services": [
  66                 {
  68                     "entity": [
  89                         {
  90                             "type": "text",
  91                             "label": "Interval",
  92                             "validators": [
  93                                 {
  94                                     "type": "regex",
  95                                     "errorMsg": "Interval must be an integer.",
  96                                     "pattern": "^\\-[1-9]\\d*$|^\\d*$"
  97                                 }
  98                             ],
  99                             "defaultValue": "300",
 100                             "field": "interval",
 101                             "help": "Time interval of the data input, in seconds.",
 102                             "required": true
 103                         },
 104                         {
 105                             "type": "singleSelect",
 106                             "label": "Account to use",
 107                             "options": {
 108                                 "referenceName": "account"
 109                             },
 110                             "help": "Account to use for this input.",
 111                             "field": "account",
 112                             "required": true
 113                         }
 114                     ]
 115                 }
 116             ]
 117         }
 118     }
 119 }
```

# Edit

## globalConfig.json

Configure inputs for activities

- Index field

(singleSelect element)

```
{ } globalConfig.json > { } pages > { } inputs > [ ] services > { } 0 > [ ] entity

    {
      "type": "singleSelect",
      "label": "Index",
      "validators": [
        {
          "type": "string",
          "errorMsg": "Length of index name should be between 1 to 80 characters.",
          "minLength": 1,
          "maxLength": 80
        }
      ],
      "defaultValue": "default",
      "options": {
        "endpointUrl": "data/indexes",
        "denyList": "^_.*$",
        "createSearchChoice": true
      },
      "field": "index",
      "required": true
    },
  ],
}
```

# Edit

## globalConfig.json

Configure inputs for activities

- Sourcetype field

(text element with validators)

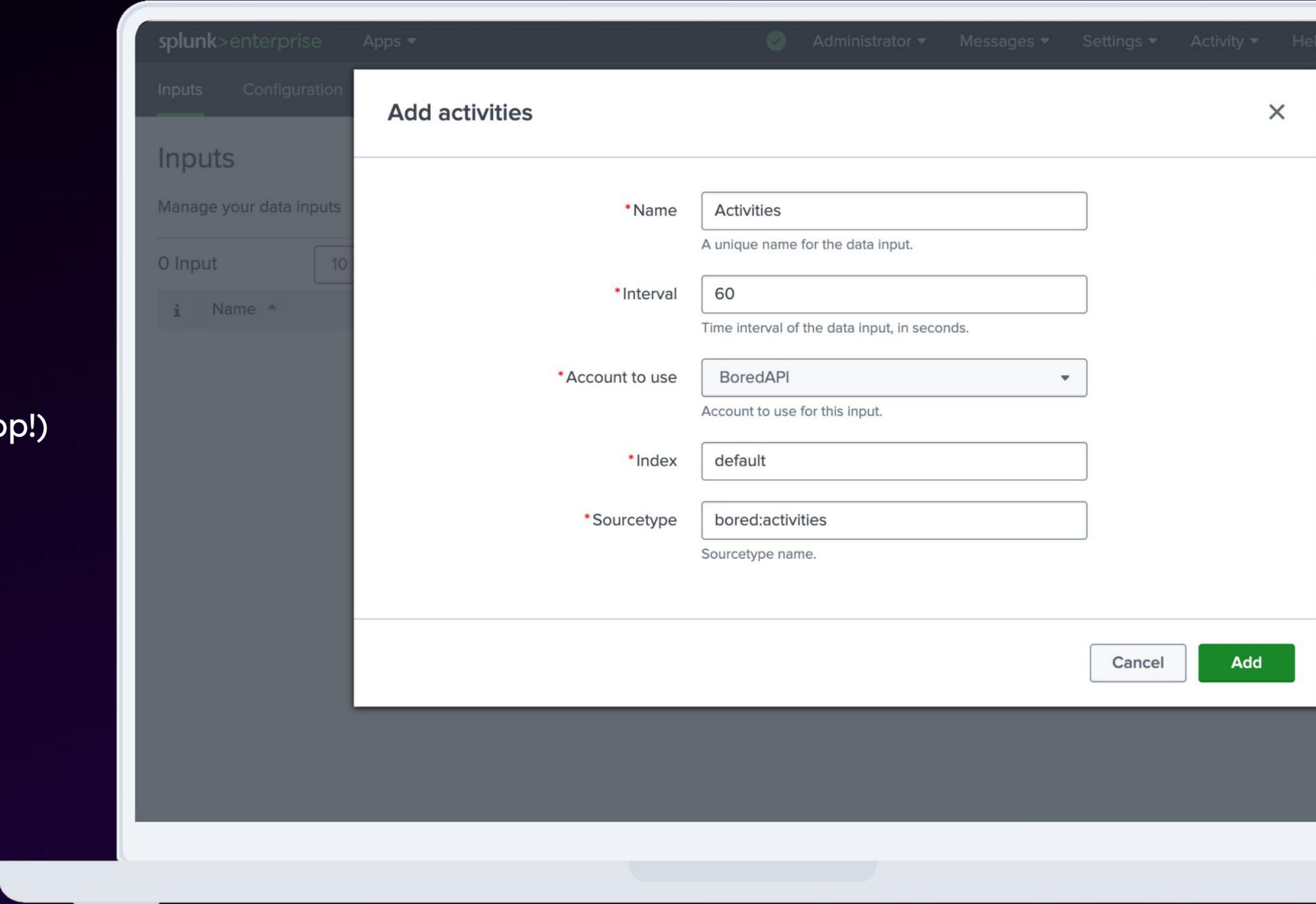
```
{ } globalConfig.json > { } pages > { } inputs > [ ] services > { } 0 > [ ] entity
    {
      "type": "text",
      "label": "Sourcetype",
      "validators": [
        {
          "type": "regex",
          "errorMsg": "Sourcetype name must start with a letter, contain
alphanumeric characters, underscores, dashes, colons only and must end with an alphanumeric character.",
          "pattern": "^[a-zA-Z][\\w:~]*[a-zA-Z0-9]$"
        },
        {
          "type": "string",
          "errorMsg": "Length of sourcetype name should be between 1 and 100",
          "minLength": 1,
          "maxLength": 100
        }
      ],
      "field": "sourcetype",
      "help": "Sourcetype name.",
      "required": true
    }
  }
```

# Let's Check the Progress!

Build new version 0.0.3

Package and reinstall (Upgrade app!)

Create an input ->



# Add-on TODO

## ✓ Configuration

- We want to:
  - Set up account for an endpoint
- We need:
  - URL field (with validation)

## ✓ Inputs

- We want to:
  - Get the data from activities endpoint
  - Ingest to a specified index
  - sourcetype = activities
- We need:
  - Account field (list of configured accounts)
  - Index field (selection from available indexes)
  - Sourcetype field (free text with validation)

## → Python

- We want to:
  - Collect data from an endpoint
  - Ingest data to Splunk
- We need:
  - read configuration data from account and inputs
  - handle request to the endpoint
  - handle writing events to Splunk

# Exercise

# 5



# Edit

package/bin/activities.py

```
import requests
```

Create get\_account\_property function

Replace get\_data\_from\_api function

```
def get_account_property(session_key: str, account_name: str, property_name: str):
    cfm = conf_manager.ConfManager(
        session_key,
        ADDON_NAME,
        realm=f"__REST_CREDENTIAL__#{ADDON_NAME}#configs/conf-demo_addon_for_activities_account",
    )
    account_conf_file = cfm.get_conf("demo_addon_for_activities_account")
    return account_conf_file.get(account_name).get(property_name)

def get_data_from_api(logger: logging.Logger, url: str, api_key: str):
    logger.info("Getting data from an external API")
    if url == None:
        raise Exception("Empty URL. Check configuration.")
    data_from_api = requests.get(url=url).json()
    return data_from_api
```

# Edit

package/bin/activities.py

Modify `stream_events` function

- get `api_url` property
- get data from url
- get `sourcetype` property
- write event to index

```
try:
    session_key = self._input_definition.metadata["session_key"]
    log_level = conf_manager.get_log_level(
        logger=logger,
        session_key=session_key,
        app_name=ADDON_NAME,
        conf_name=f"{ADDON_NAME}_settings",
    )
    logger.setLevel(log_level)
    log.modular_input_start(logger, normalized_input_name)
    api_key = get_account_property(session_key, input_item.get("account"), "api_key")
    api_url = get_account_property(session_key, input_item.get("account"), "url")
    data = get_data_from_api(logger, api_url, api_key)
    sourcetype = input_item.get("sourcetype")
    event_writer.write_event(
        smi.Event(
            data=json.dumps(data, ensure_ascii=False, default=str),
            index=input_item.get("index"),
            sourcetype=sourcetype,
        )
    )
```

# Checking Progress... Again!

Build new version 0.0.4

Package and reinstall (Upgrade app!)

Search for data in **index=main**

The screenshot shows the Splunk search interface for the index 'index=main'. It displays 23 events from 07/05/2024 17:00:00.000 to 08/05/2024 17:40:17.000. The interface includes tabs for Events (23), Patterns, Statistics, and Visualization. Below the tabs are controls for Format Timeline, Zoom Out, Zoom to Selection, and Deselect. A table view shows two event entries with their respective timestamps and JSON payloads. The left sidebar contains field lists and a '+ Extract New Fields' button.

i	Time	Event
>	08/05/2024 17:40:05.000	{ [-] accessibility: 0.1 activity: Shop at support your local farmers market key: 8979931 link: participants: 1 price: 0.2 type: relaxation } Show as raw text host = show-demo-i-030f99c77e17bc1f6   source = activities://Activities   sourcetype = bored:activities
>	08/05/2024 17:39:05.000	{ [-] accessibility: 0.1 activity: Study a foreign language key: 9765530 link: participants: 1 price: 0 type: education } Show as raw text host = show-demo-i-030f99c77e17bc1f6   source = activities://Activities   sourcetype = bored:activities

# Add-on TODO

## ✓ Configuration

- We want to:
  - Set up account for an endpoint
- We need:
  - URL field (with validation)

## ✓ Inputs

- We want to:
  - Get the data from activities endpoint
  - Ingest to a specified index
  - sourcetype = activities
- We need:
  - Account field (list of configured accounts)
  - Index field (selection from available indexes)
  - Sourcetype field (free text with validation)

## ✓ Python

- We want to:
  - Collect data from an endpoint
  - Ingest data to Splunk
- We need:
  - read configuration data from account and inputs
  - handle request to the endpoint
  - handle writing events to Splunk

# Splunk App development - summary

- Use UCC Framework to speed-up the development and for easy code management
- Bring any data from a web endpoint or other sources to Splunk
- Customize configuration elements to adapt to your specific needs

# Create an App

Splunk Core | Splunk® SOAR | O11y



**Bring on  
the future.**





“If you torture the data  
long enough, it will confess  
to anything...”

Ronald H. Coase,  
Essays on Economics and Economists, 1994

# Dashboard App TODO

## → Create an Empty App

- We want to:
  - Build a dedicated app for activities
- We need:
  - Splunk app to manage our knowledge objects

## Build a Dashboard

- We want to:
  - Create a dashboard to visualize data on activities
- We need:
  - Splunk searches to query data (SPL)
  - Splunk dashboard with diagrams and tables to present data

## Customize Navigation

- We want to:
  - Show dashboard on a default view in this app
- We need to:
  - Change default view for the app

# Exercise

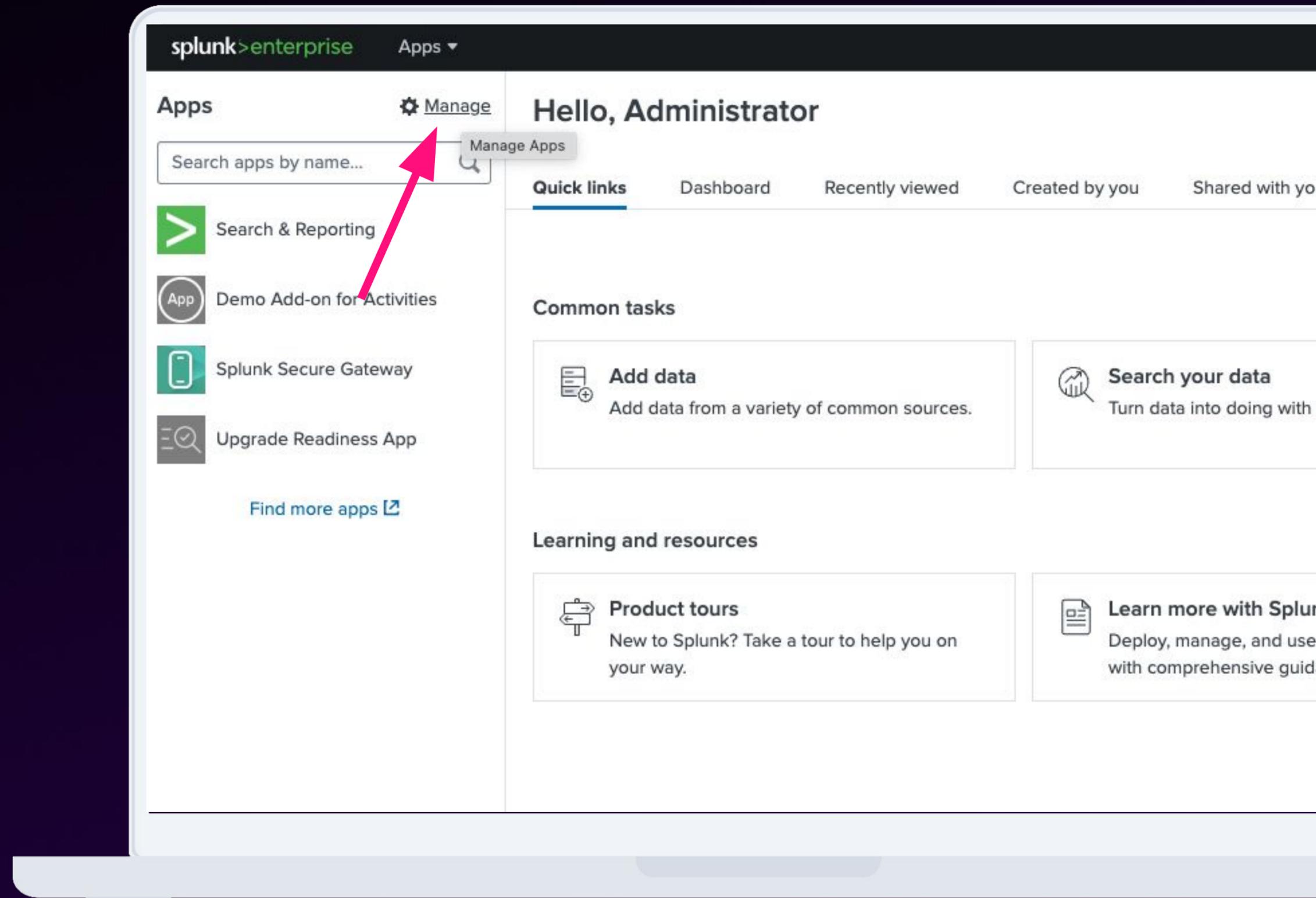
## 6+7



# Configure

Create new app

Click on “Manage: button



# Configure

Create new app

Click on “Create app”

The screenshot shows the Splunk Apps management page. At the top right, there are buttons for 'Browse more apps', 'Install app from file', and 'Create app'. A red arrow points to the 'Create app' button. Below the buttons is a table listing various apps with columns for Name, Folder name, Version, Update checking, Visible, Sharing, Status, and Actions.

Name	Folder name	Version	Update checking	Visible	Sharing	Status	Actions
SplunkDeploymentServerConfig	SplunkDeploymentServerConfig		Yes	No	App   Permissions	Enabled   Disable	Edit properties   View objects
SplunkForwarder	SplunkForwarder		Yes	No	App   Permissions	Disabled   Enable	
SplunkLightForwarder	SplunkLightForwarder		Yes	No	App   Permissions	Disabled   Enable	
Log Event Alert Action	alert_logevent	9.2.1	Yes	No	App   Permissions	Enabled	Edit properties   View objects
Webhook Alert Action	alert_webhook	9.2.1	Yes	No	App   Permissions	Enabled	Edit properties   View objects
Apps Browser	appsbrowser	9.2.1	Yes	No	App   Permissions	Enabled	Edit properties   View objects
Demo Add-on for Activities	demo_addon_for_activities	0.0.4	Yes	Yes	Global   Permissions	Enabled	Launch app   Edit properties   View objects   View details on Splunkbase
introspection_generator_addon	introspection_generator_addon	9.2.1	Yes	No	App   Permissions	Enabled   Disable	Edit properties   View objects
journald_input	journald_input		Yes	No	App   Permissions	Enabled   Disable	Edit properties   View objects
Home	launcher		Yes	Yes	App   Permissions	Enabled	Launch app   Edit properties   View objects
learned	learned		Yes	No	App   Permissions	Enabled   Disable	Edit properties   View objects
legacy	legacy		Yes	No	App   Permissions	Disabled   Enable	
Upgrade Readiness App	python_upgrade_readiness_app	4.3.0	Yes	Yes	App   Permissions	Enabled   Disable	Launch app   Edit properties   View objects   View details on Splunkbase
sample data	sample_app		Yes	No	App   Permissions	Disabled   Enable	
Search & Reporting	search	9.2.1	Yes	Yes	App   Permissions	Enabled	Launch app   Edit properties   View objects
Splunk Dashboard Studio	splunk-dashboard-studio	1.13.3	Yes	Yes	App   Permissions	Enabled   Disable	Launch app   Edit properties   View objects
Splunk Rolling Upgrade	splunk-rolling-upgrade	1.0.0	Yes	No	App   Permissions	Enabled   Disable	Edit properties   View objects
Splunk Archiver App	splunk_archiver	1.0	Yes	No	App   Permissions	Enabled   Disable	Edit properties   View objects   View details on Splunkbase
Splunk Assist	splunk_assist	1.0.59	No	No	App   Permissions	Enabled	Edit properties   View objects   View details on Splunkbase
Splunk Get Data In	splunk_gdi	1.0.6	Yes	No	App   Permissions	Enabled	Edit properties   View objects
splunk_httpinput	splunk_httpinput		Yes	No	App   Permissions	Enabled   Disable	Edit properties   View objects
Instrumentation	splunk_instrumentation	6.0.10	Yes	Yes	App   Permissions	Enabled	Launch app   Edit properties   View objects
Clones Internal Metrics into Metrics Index	splunk_internal_metrics		Yes	No	App   Permissions	Enabled   Disable	Edit properties   View objects
Splunk Analytics Workspace	splunk_metrics_workspace	2.47.0	Yes	Yes	App   Permissions	Enabled   Disable	Launch app   Edit properties   View objects
Monitoring Console	splunk_monitoring_console	10.0.0	Yes	Yes	App   Permissions	Enabled   Disable	Launch app   Edit properties   View objects

# Configure

## Create new app

- Name: **Demo App for Activities**
- Folder name: **demo\_app\_for\_activities**
- Version: **1.0.0**
- Template: **barebones**
- Click on **“Save”**

Name   
Give your app a friendly name for display in Splunk Web.

Folder name \*   
This name maps to the app's directory in \$SPLUNK\_HOME/etc/apps/.

Version   
App version.

Visible  No  Yes  
Only apps with views should be made visible.

Author   
Name of the app's owner.

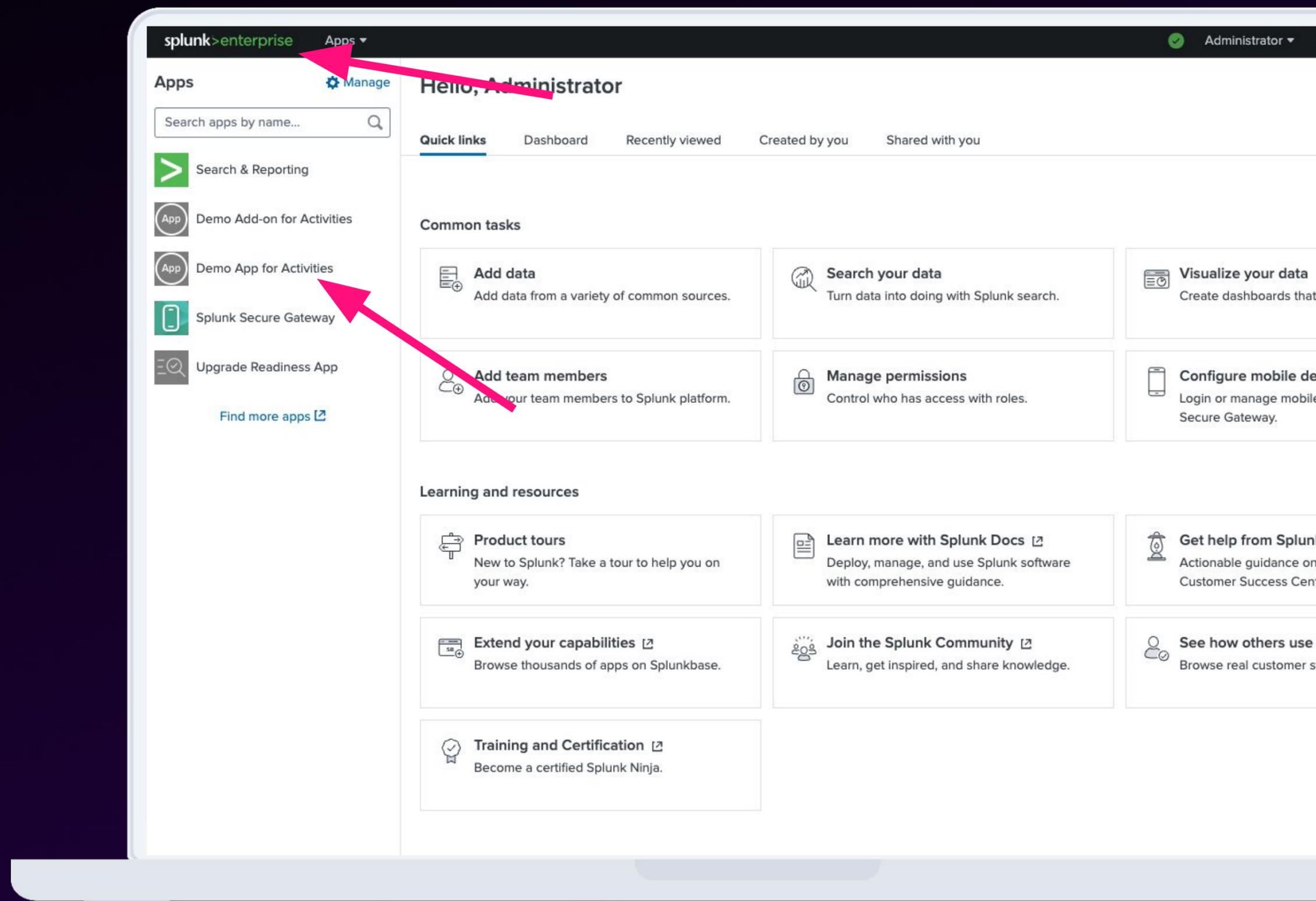
Description   
Enter a description for your app.

Template   
These templates contain example views and searches.

Upload asset  No file chosen  
Can be any html, js, or other file to add to your app.

# Progress Check!

Go back to main page  
Your app is on the list!



# Dashboard App TODO

## ✓ Create an Empty App

- We want to:
  - Build a dedicated app for activities
- We need:
  - Splunk app to manage our knowledge objects

## → Build a Dashboard

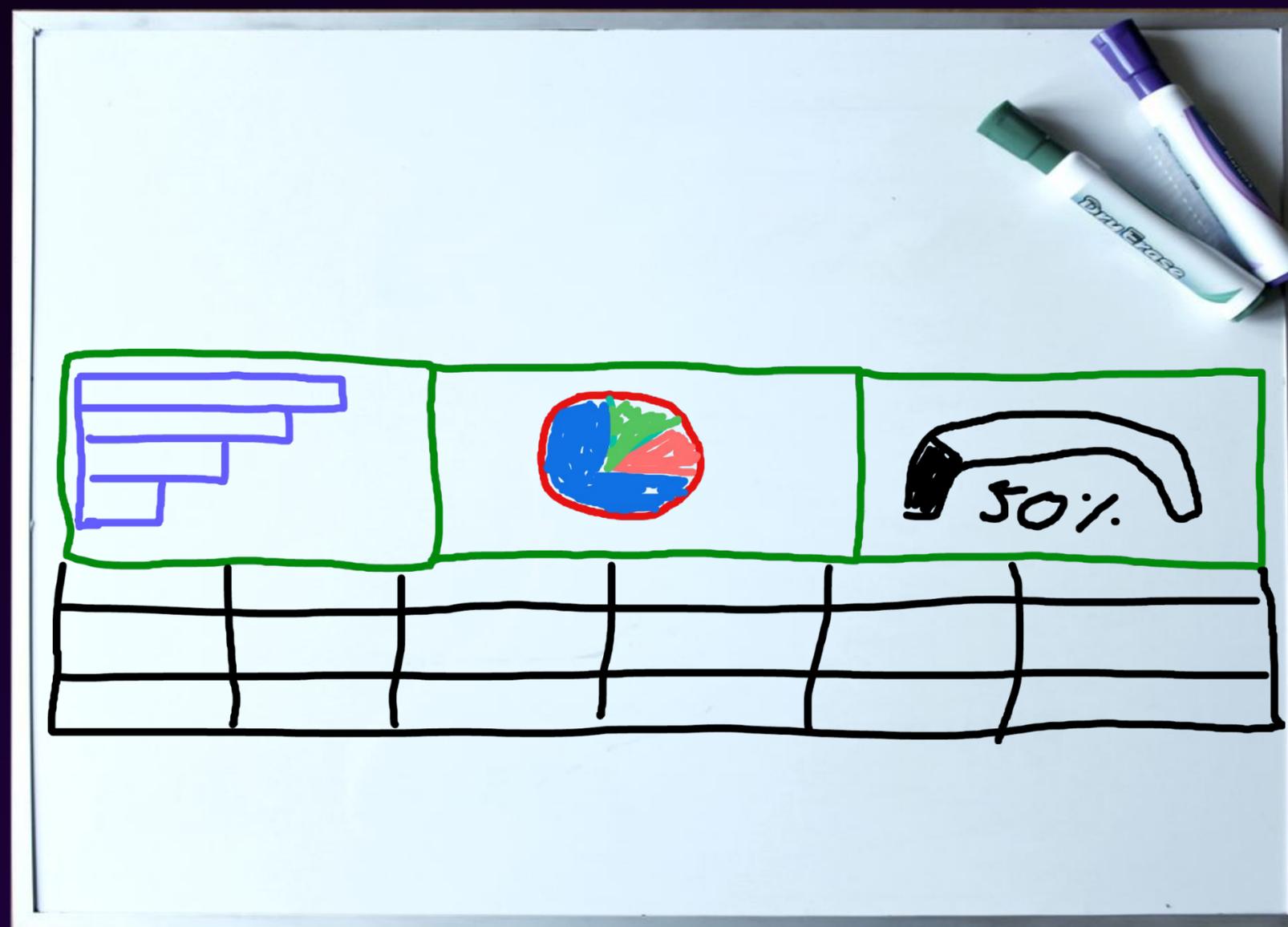
- We want to:
  - Create a dashboard to visualize data on activities
- We need:
  - Splunk searches to query data (SPL)
  - Splunk dashboard with diagrams and tables to present data

## Customize Navigation

- We want to:
  - Show dashboard on a default view in this app
- We need to:
  - Change default view for the app

# Dashboard Design

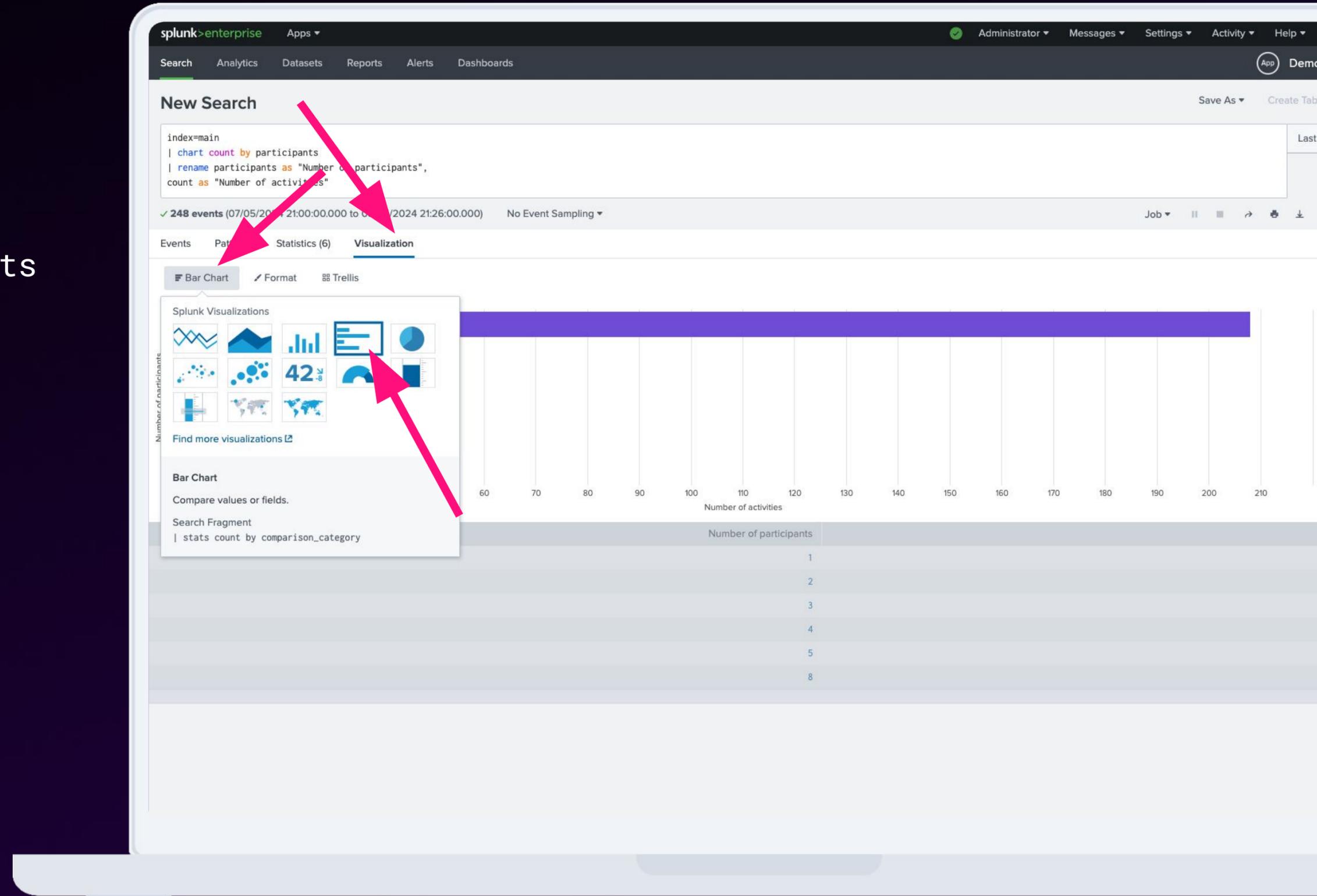
- **Panel 1:** Number of activities by number of participants
- **Panel 2:** Activities by type
- **Panel 3:** Free of charge activities
- **Panel 4:** Latest 10 activities



# Search

```
index=main  
| chart count by participants  
| rename participants as "Number of participants",  
count as "Number of activities"
```

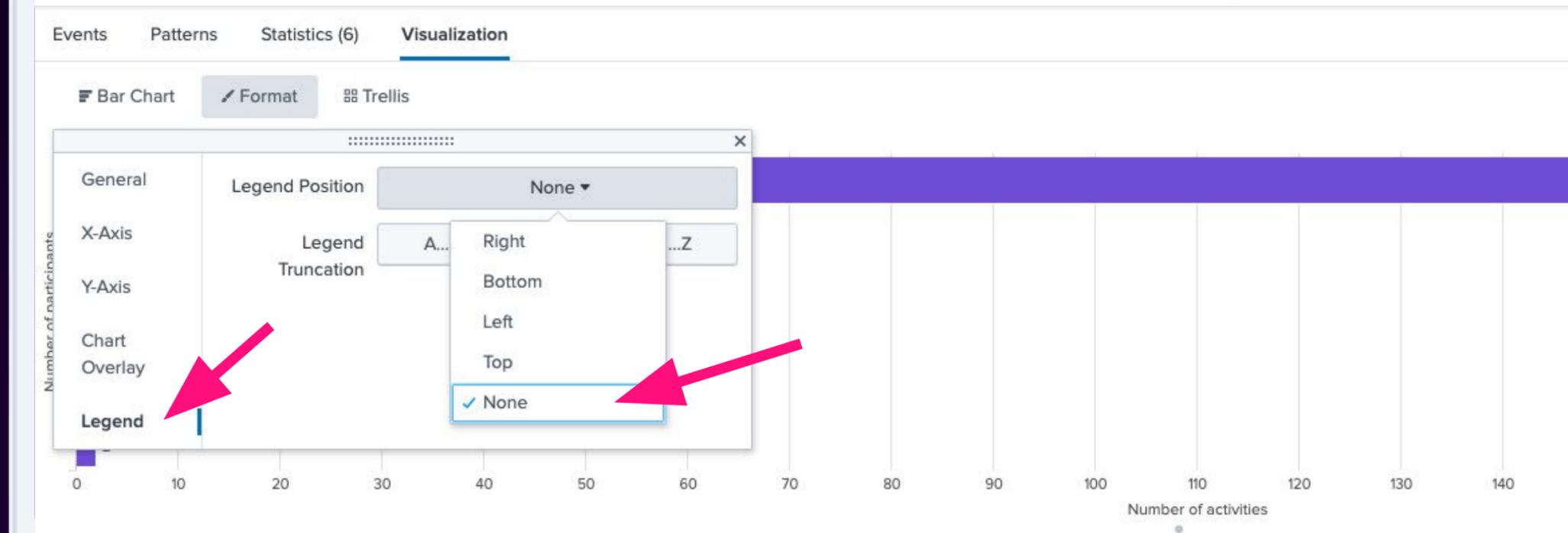
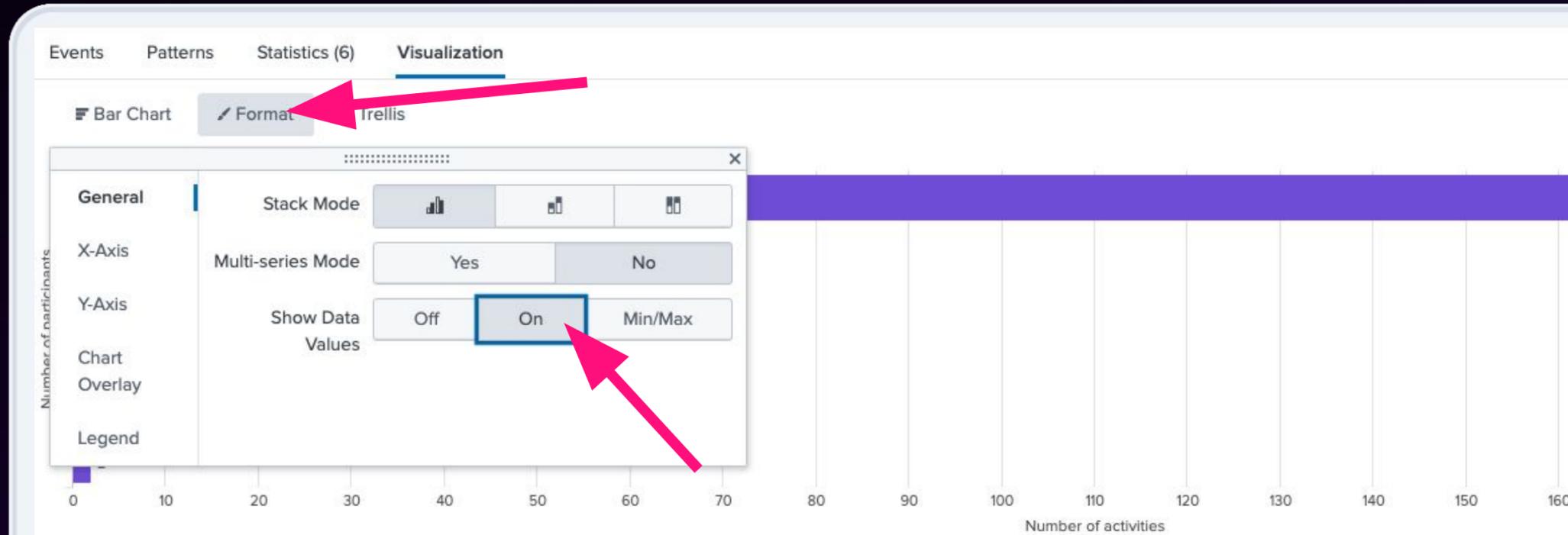
Go to Visualization  
Change visualization  
to Bar Chart



# Configure

Go to “Format”

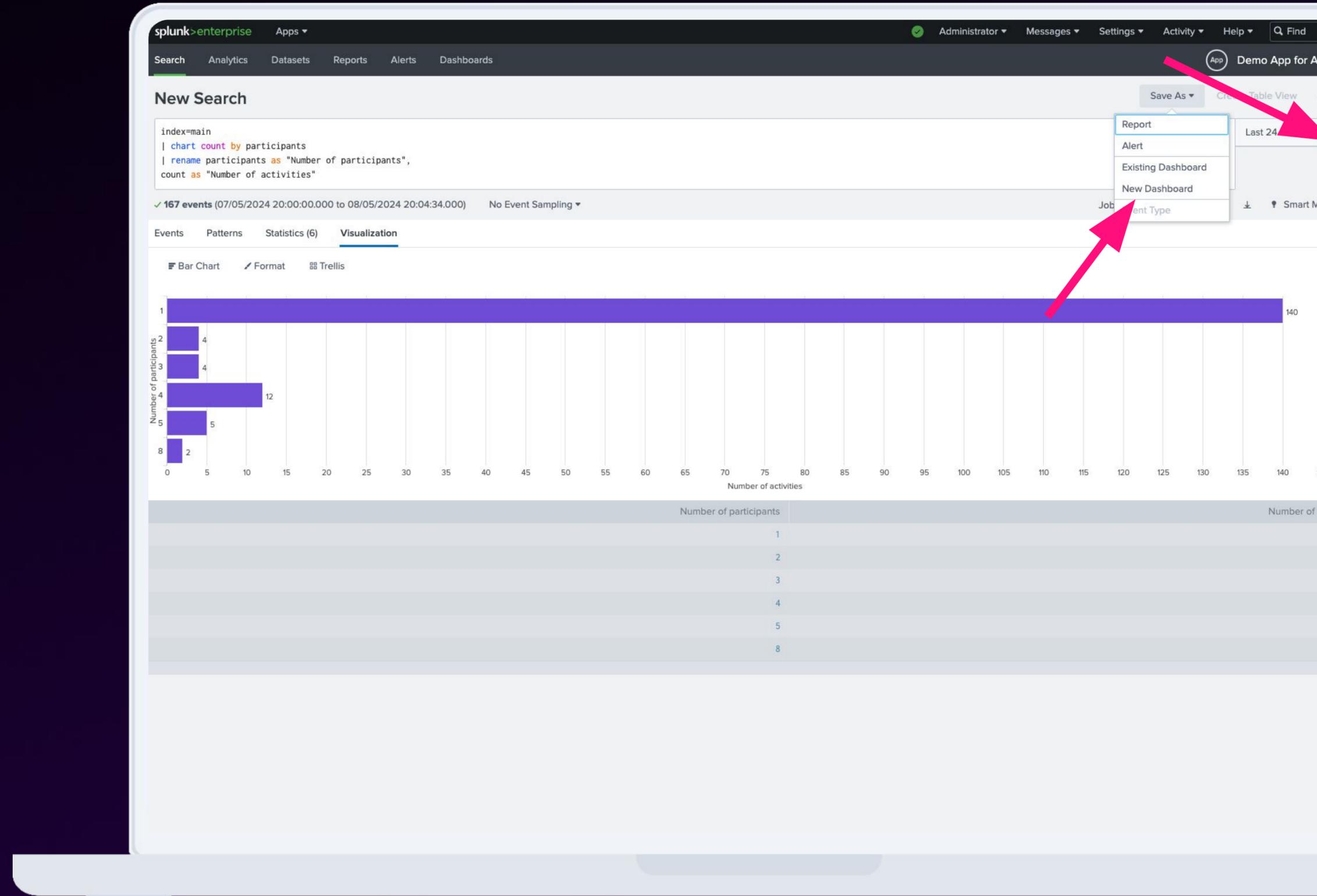
- General / Show Data Values: **On**
- Legend / Legend Position: **None**



# Create Dashboard

Click on “**Save As**” on top-right.

Click on “**New Dashboard**” from drop down.



# Configure

Dashboard Title: **Activities Dashboard**

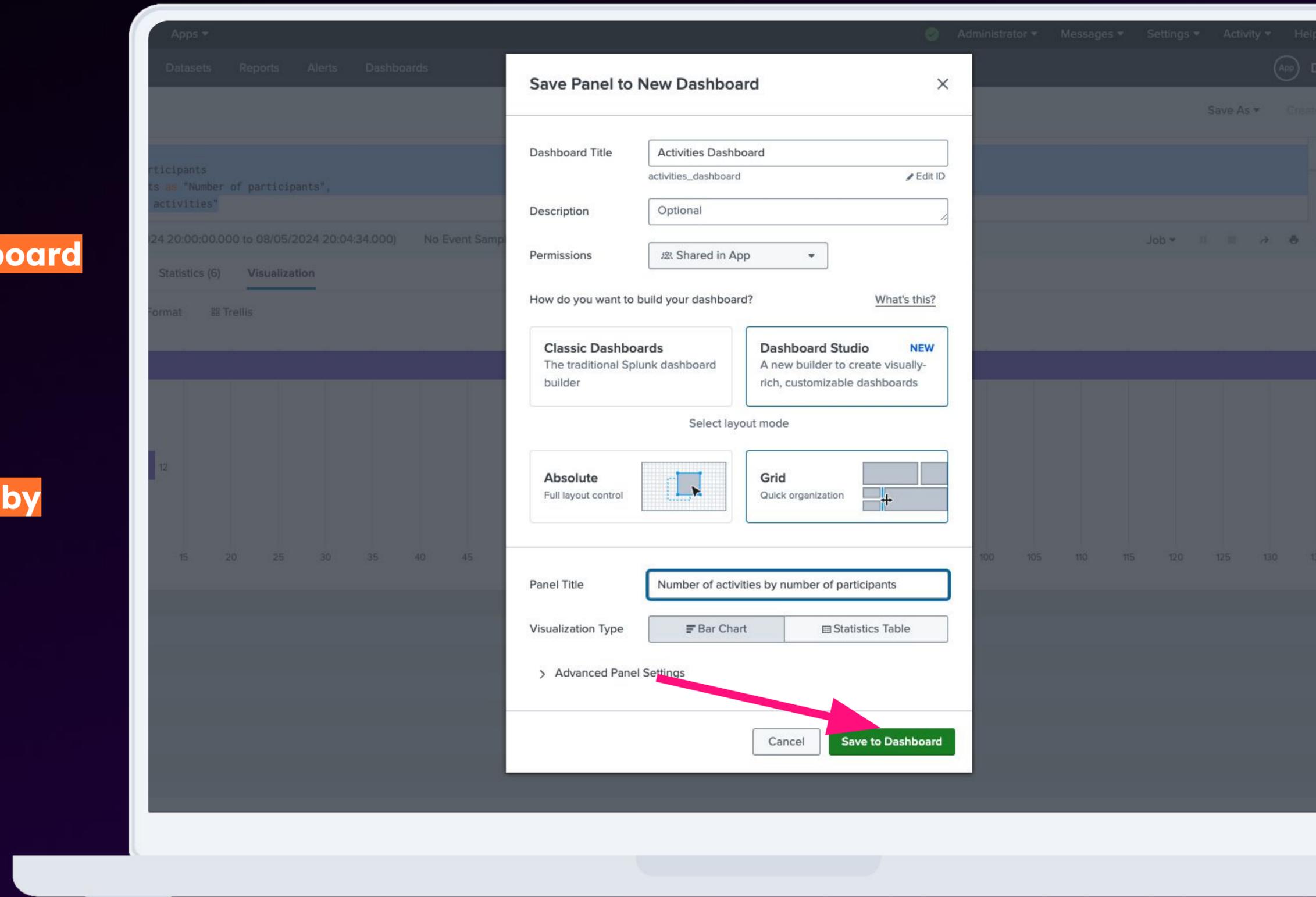
Permissions: **Shared in App**

Type: **Dashboard Studio**

Layout mode: **Grid**

Panel Title: **Number of activities by number of participants**

Click on **“Save to Dashboard”**



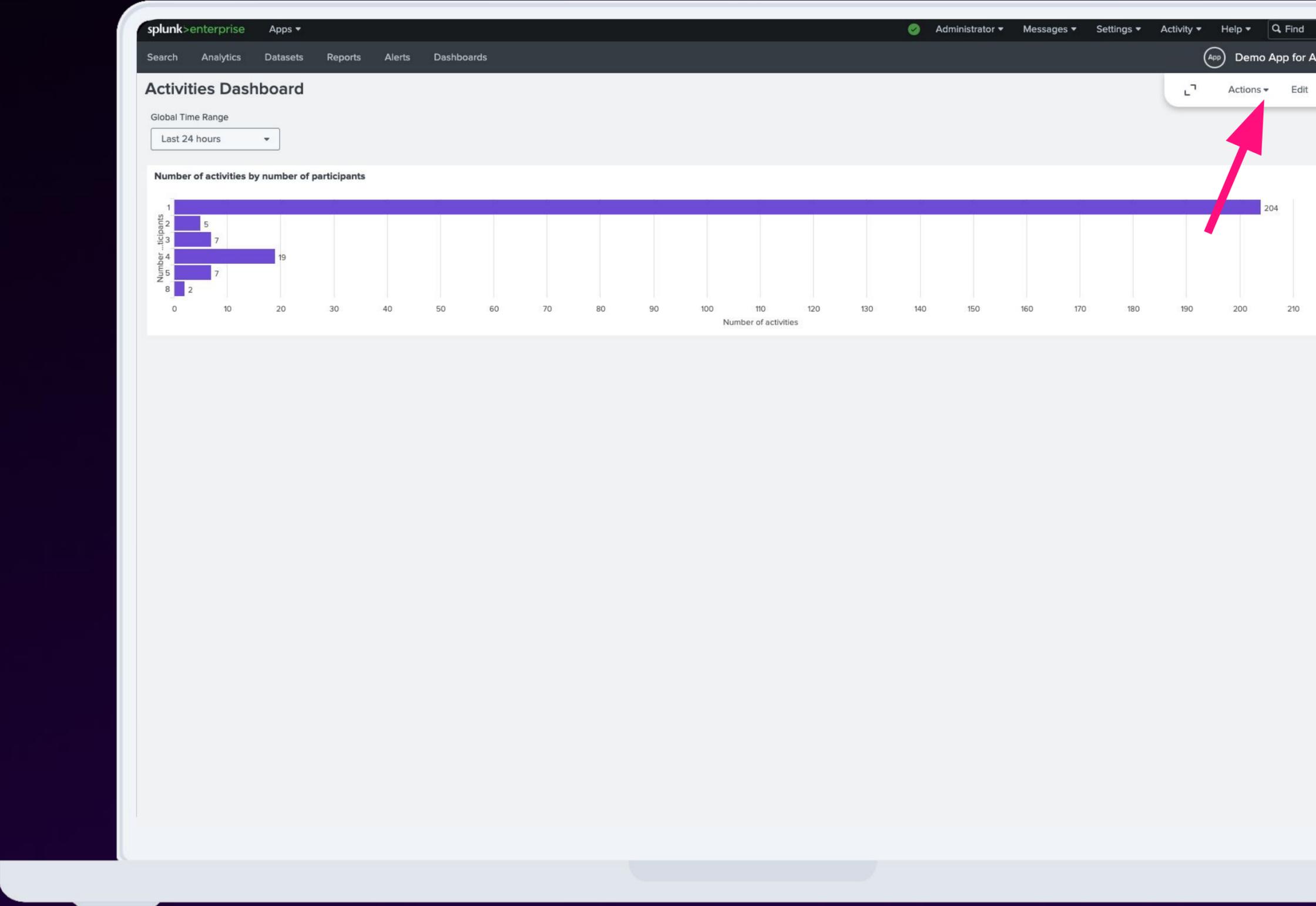
# Exercise

# 8+9+10

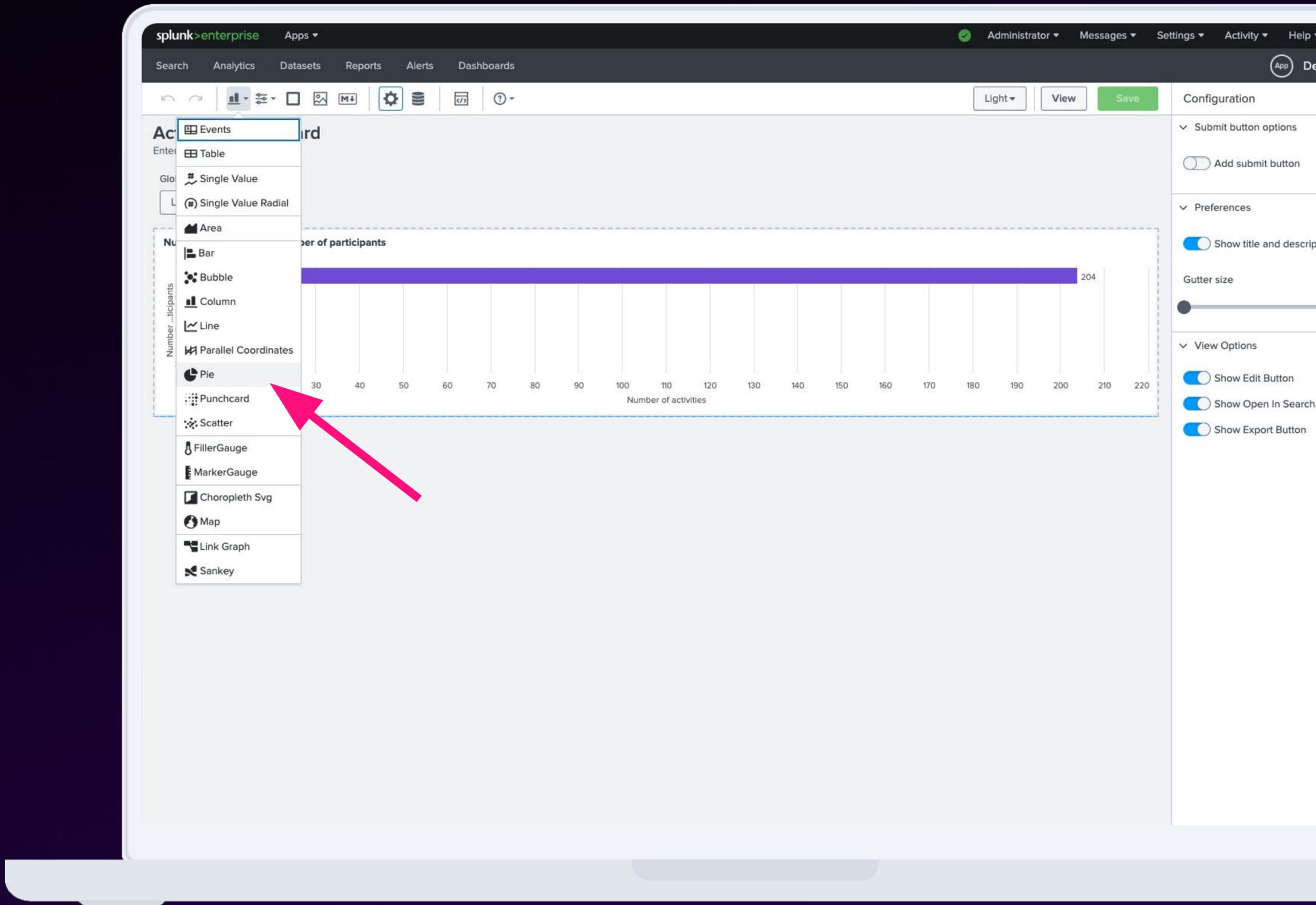


# Add Another Panel

Click on “Edit” button on your top right!

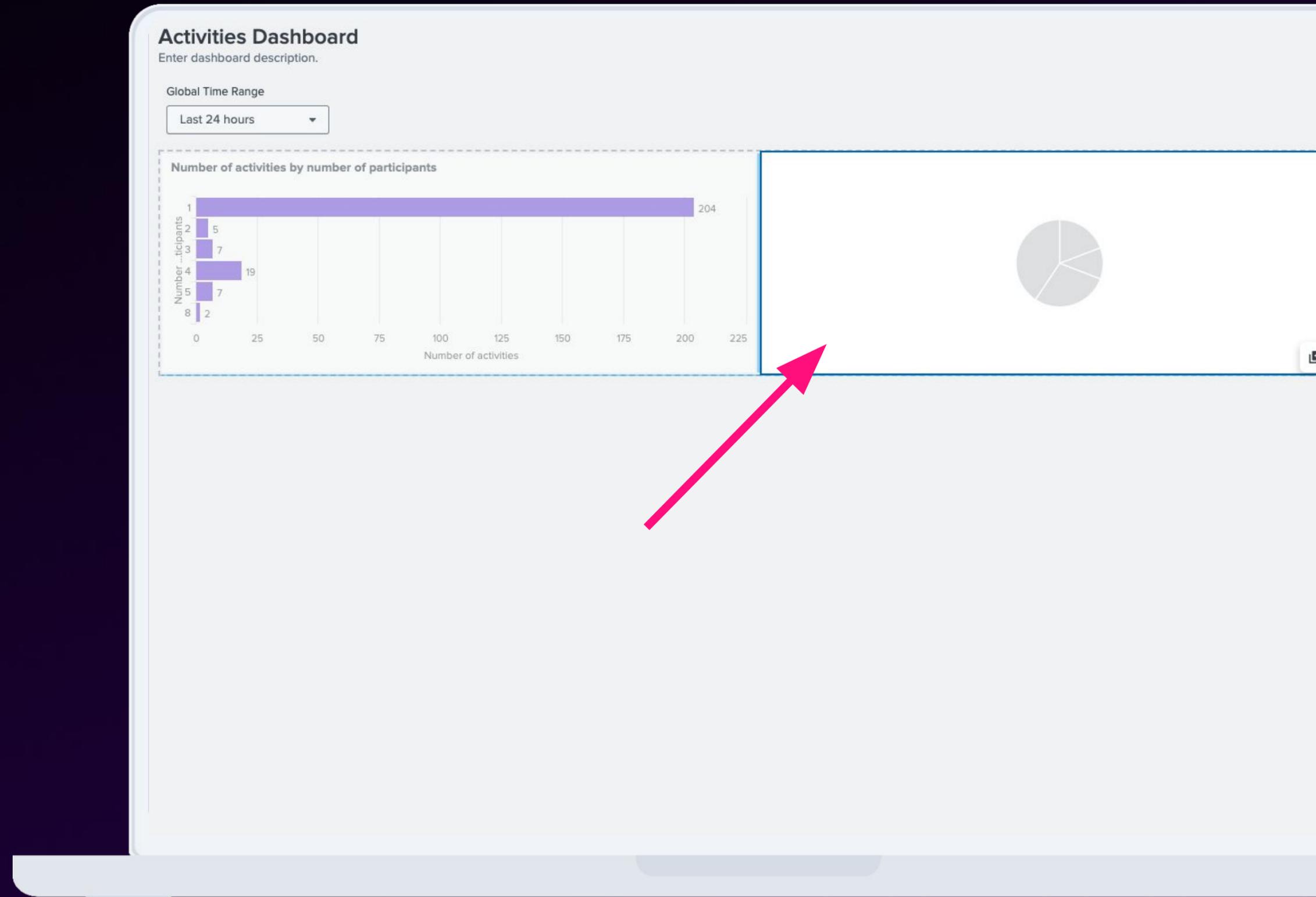


# Choose Pie



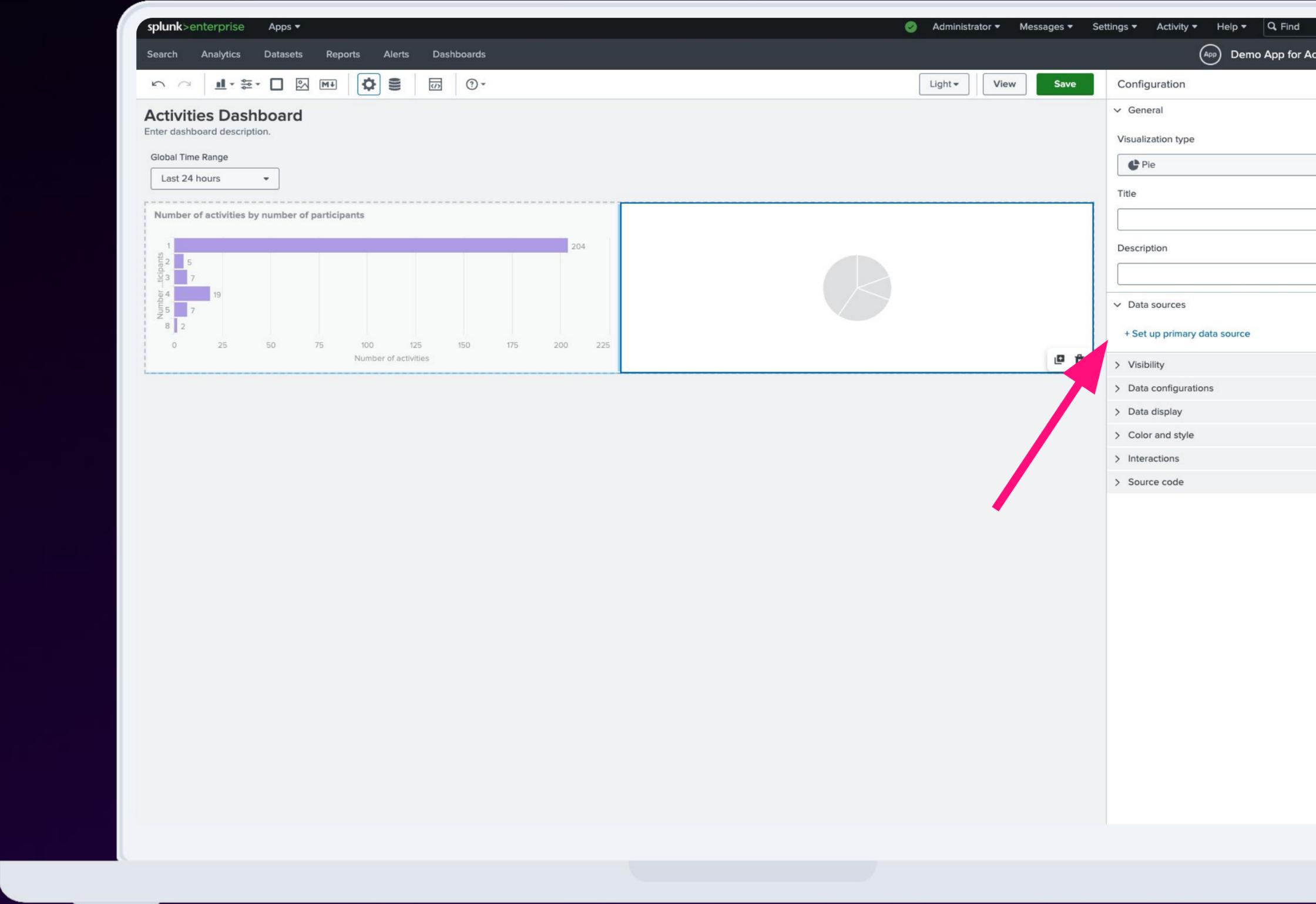
# Move Panel to the 1st Row

Drag it to this position



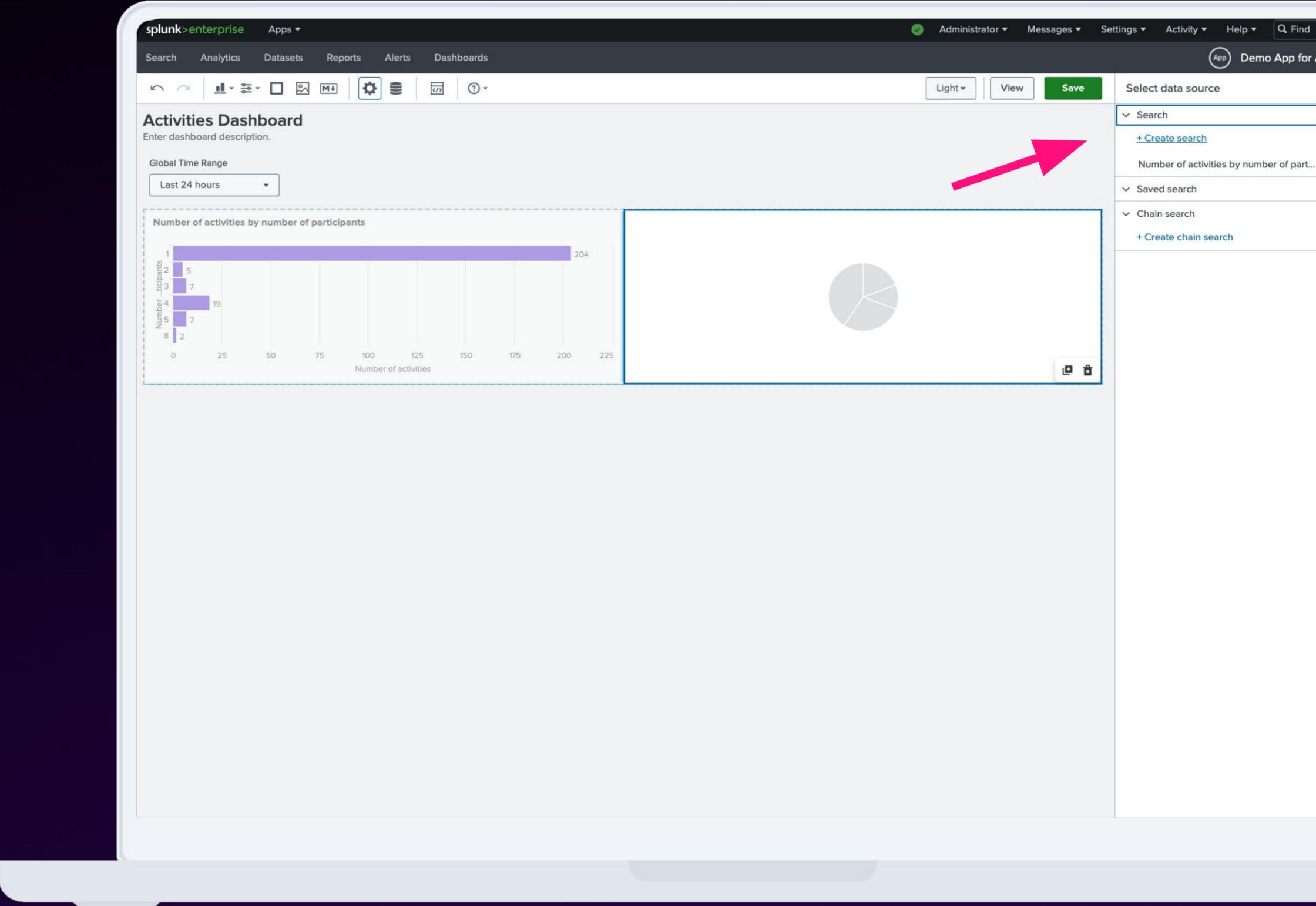
# Set Up Data Source

Click on “Set up primary data source”



# Set Up Data Source

Click on “Create search”

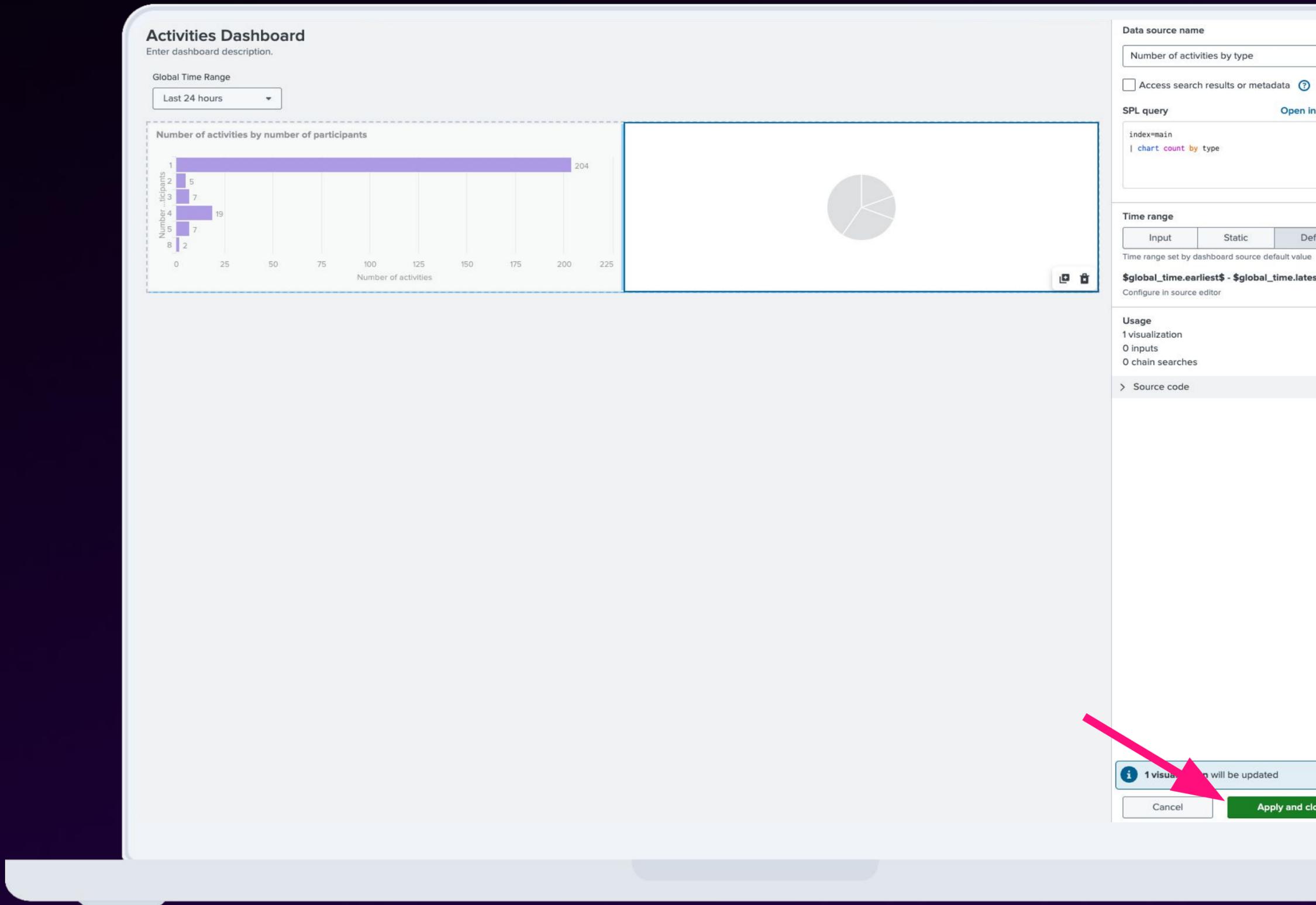


# Configure Data Source

Data source name: **Number of activities by type**

SPL query: `index=main | chart count by type`

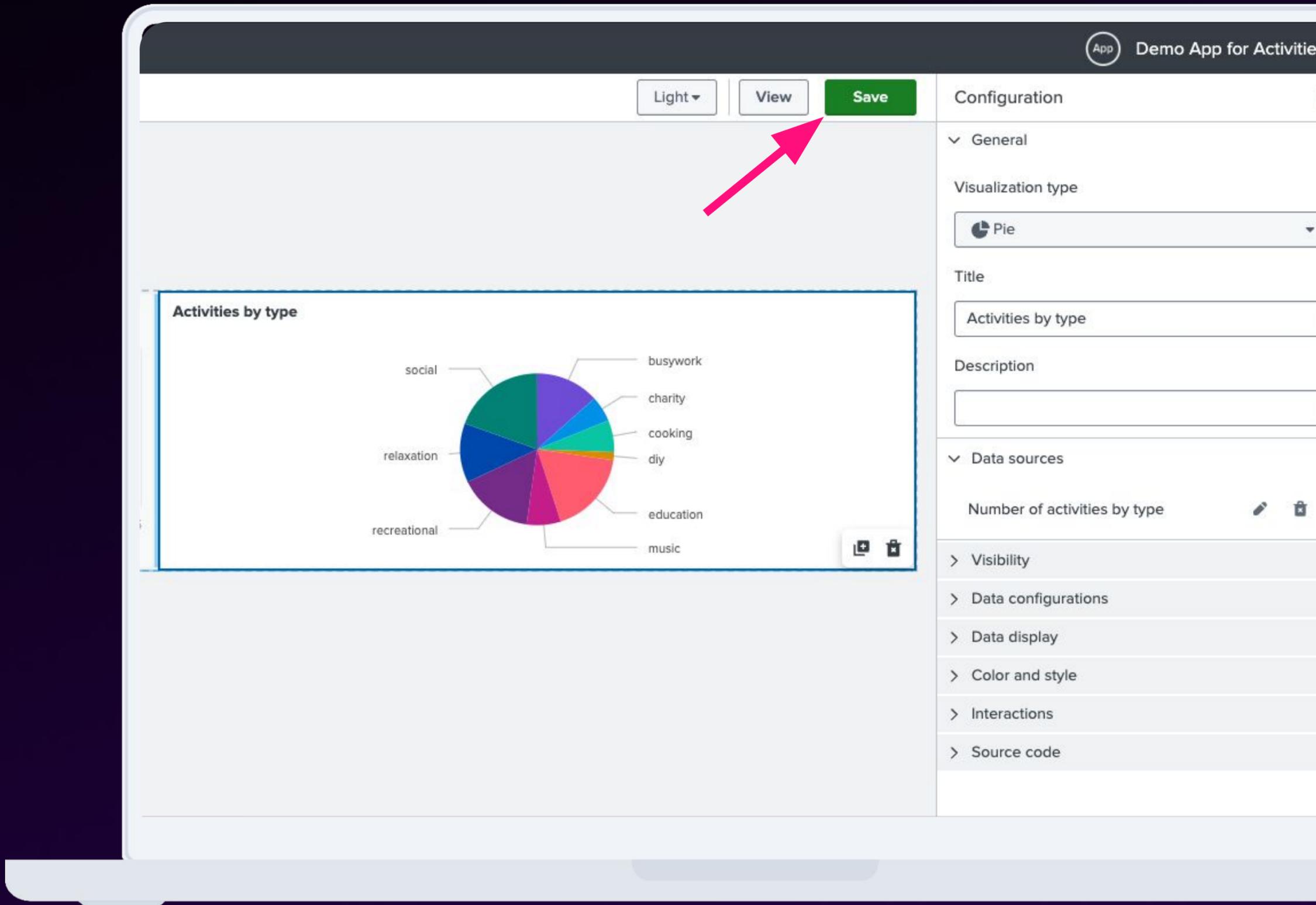
Click **Apply and close**



# Configure Graph

Title: **Activities by type**

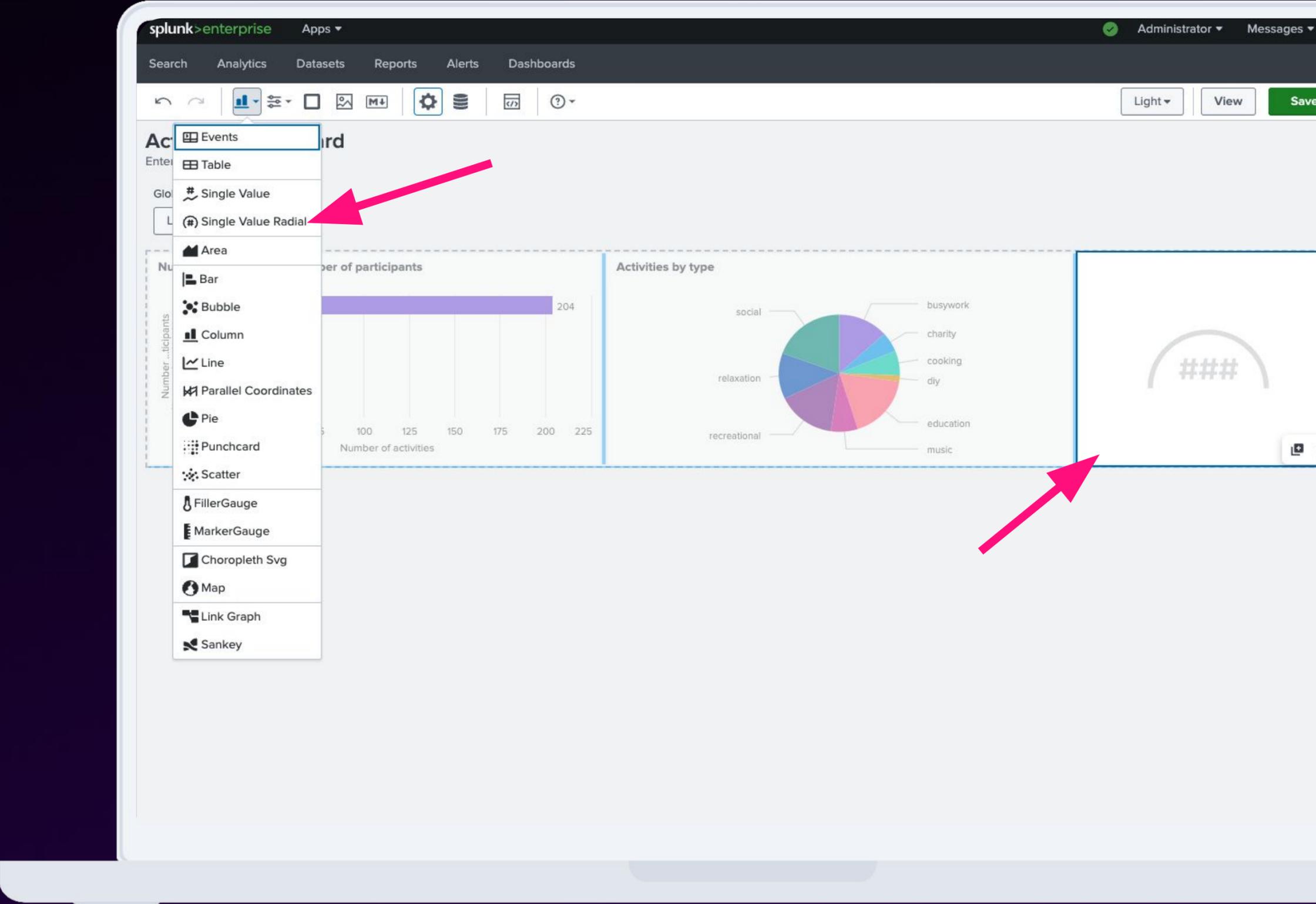
Click on **Save**



# Add Next Panel

Choose **Single Value Radial**

Move the panel to the 1st row



# Configure Data Source

Data source name: **Percentage of free of charge activities**

SPL query: index=main

| timechart count(eval(price=0)) as free, count as total

| eval free\_pct = round(free/total,2)\*100

| fields free\_pct

Click on **Apply and close**

**Activities Dashboard**  
Enter dashboard description.

Global Time Range  
Last 24 hours

**Number of activities by number of participants**

Number of participants	Number of activities
1	204
2	5
3	7
4	19
5	7
8	2

**Activities by type**

- social
- relaxation
- recreational
- busywork
- charity
- cooking
- diy
- education
- music

**Data source name**  
Percentage of free of charge activities

Access search results or metadata

**SPL query**  
index=main  
| timechart count(eval(price=0)) as free, count as total  
| eval free\_pct = round(free/total,2)

**Time range**  
Input Static Default  
Time range set by dashboard source default value  
\$global\_time.earliest\$ - \$global\_time.latest\$  
Configure in source editor

**Usage**  
0 visualizations  
0 inputs  
0 chain searches

> Source code

Cancel **Apply and close**

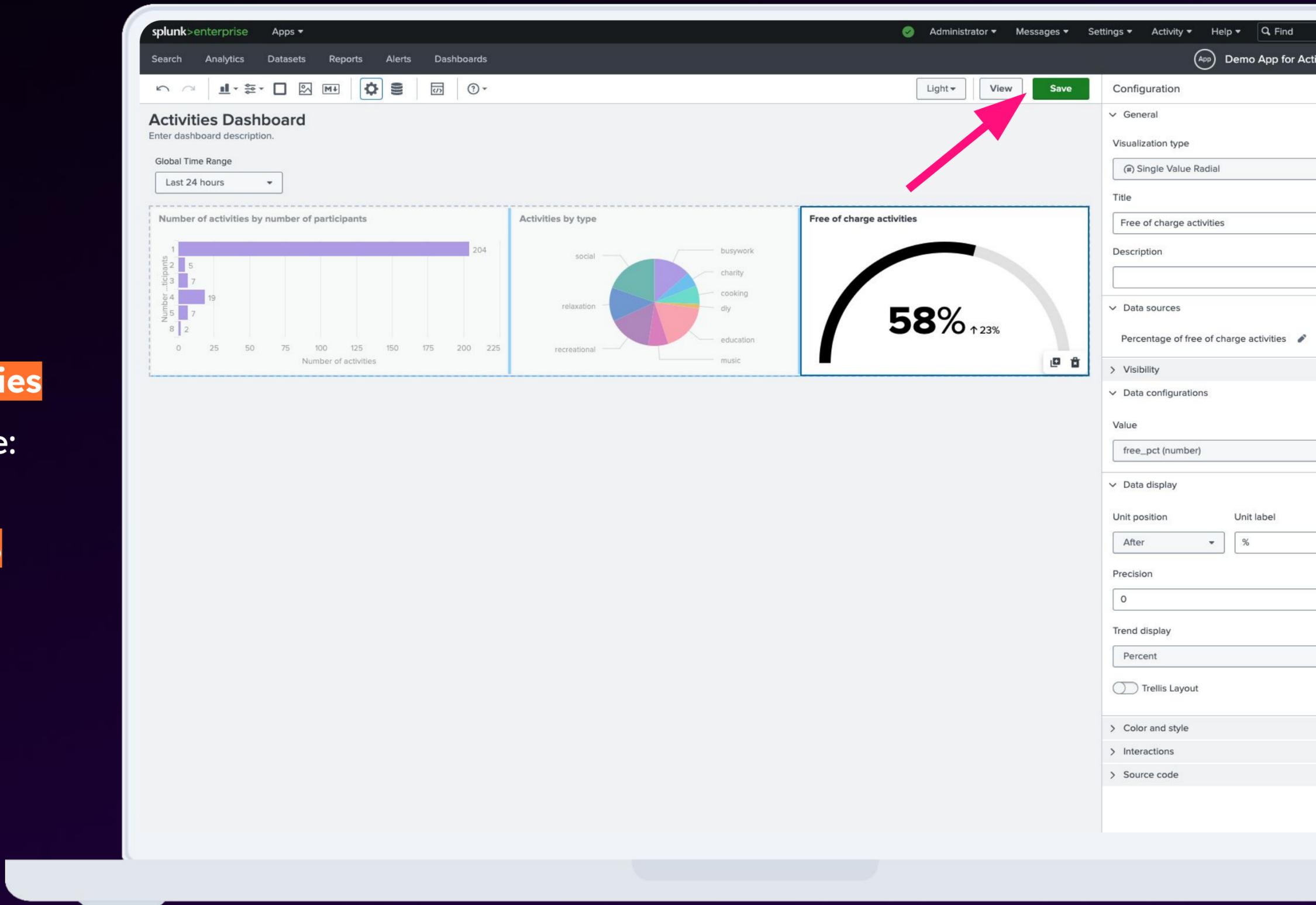
# Configure Graph

On the left **Configuration** panel, enter Title: **Free of charge activities**

Under Data configurations > Value: select **free\_pct(number)**

Under Data display > Unit label: **%**

Click on **Save** button!



# Add Last Panel and Configure Data Source

Choose "Table"

Data source name: **Latest events**

Search:

```
index=main  
| head 10  
| table key, activity, type,  
price, participants,  
accessibility, link
```

Apply and close

The screenshot shows the Splunk dashboard configuration interface. A pink arrow points to the 'Table' option in the visualization menu. Another pink arrow points to the 'Apply and close' button at the bottom right. The dashboard displays several panels: 'Number of participants' (bar chart), 'Activities by type' (pie chart), and 'Free of charge activities' (gauge chart). The gauge chart shows 58% with a 23% increase. The configuration panel on the right shows the data source name 'Latest events', the SPL query, and the time range settings.

Search Analytics Datasets Reports Alerts Dashboards

Light View Save

New data source

Data source name  
Latest events

Access search results or metadata

SPL query [Open in search](#)

```
index=main  
| head 10  
| table key, activity, type, price, participants,  
accessibility, link
```

Time range  
Input Static Default

Time range set by dashboard source default value

$\$global\_time.earliest\$ - \$global\_time.latest\$$   
Configure in source editor

Usage  
1 visualization  
0 inputs  
0 chain searches

> Source code

1 visualization will be updated

Cancel Apply and close

# Configure Table

Title: Latest activities

Click on “Save” and “View”

**Activities Dashboard**  
Enter dashboard description.

Global Time Range  
Last 24 hours

**Number of activities by number of participants**

Number of participants	Number of activities
1	204
2	5
3	7
4	19
5	7
8	2

**Activities by type**

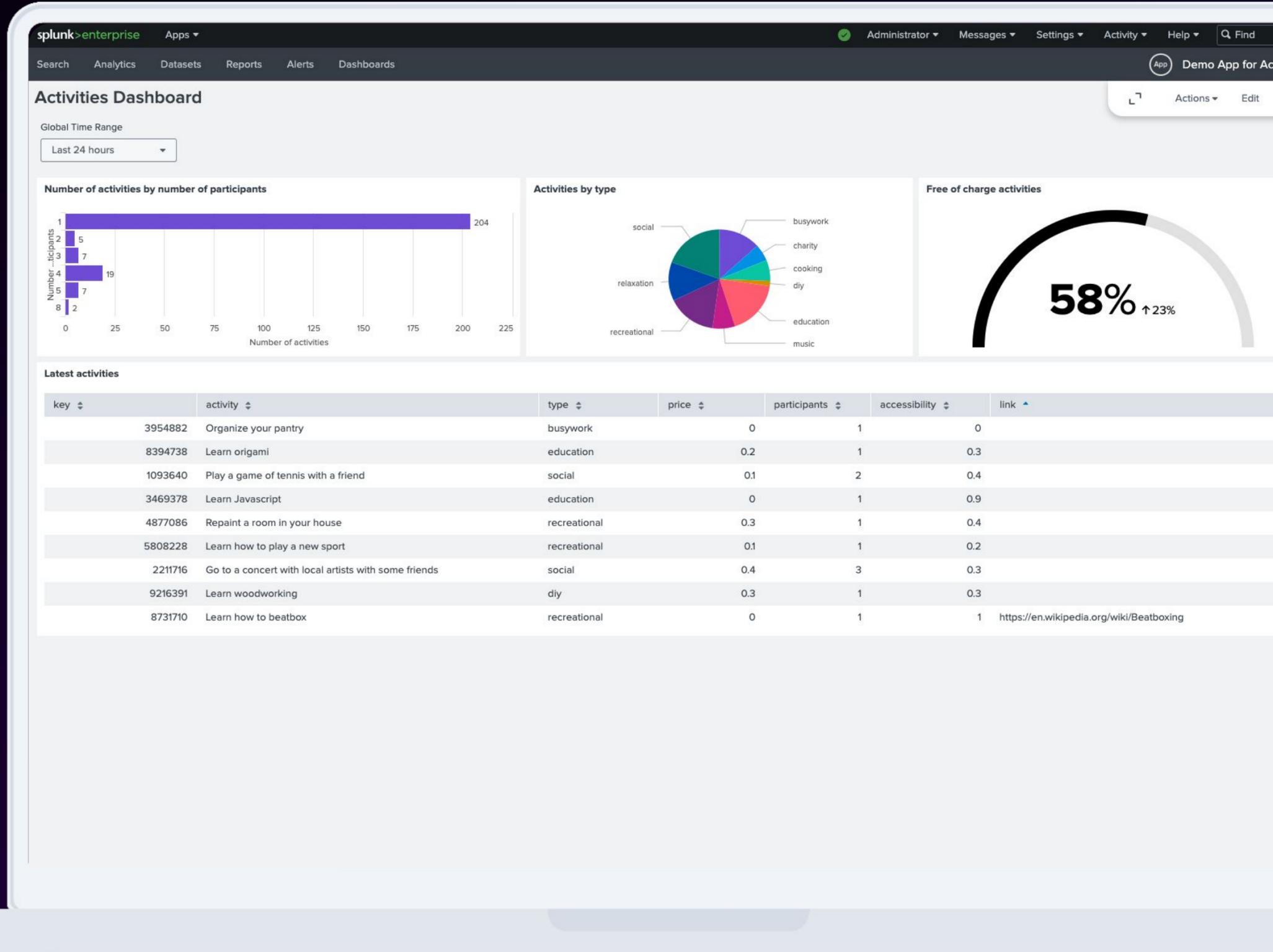
social, busywork, charity, cooking, diy, education, music, recreational, relaxation

**Free of charge activities**  
58% ↑ 23%

key	activity	type	price	participa...	accessibility	link
3954882	Organize your pantry	busywork	0	1	0	
8394738	Learn origami	education	0.2	1	0.3	
1093640	Play a game of tennis with a friend	social	0.1	2	0.4	
3469378	Learn Javascript	education	0	1	0.9	
4877086	Repaint a room in your house	recreational	0.3	1	0.4	
5808228	Learn how to play a new sport	recreational	0.1	1	0.2	
2211716	Go to a concert with local artists with some friends	social	0.4	3	0.3	
9216391	Learn woodworking	diy	0.3	1	0.3	
8731710	Learn how to beatbox	recreational	0	1	1	<a href="https://en.wikipedia.org/wiki/Beatboxing">https://en.wikipedia.org/wiki/Beatboxing</a>
3950821	Learn Kotlin	education	0	1	0.8	<a href="https://kotlinlang.org/">https://kotlinlang.org/</a>

# Let's Check Our Work!

The App is ready!



# Dashboard App TODO

## ✓ Create an Empty App

- We want to:
  - Build a dedicated app for activities
- We need:
  - Splunk app to manage our knowledge objects

## ✓ Build a Dashboard

- We want to:
  - Create a dashboard to visualize data on activities
- We need:
  - Splunk searches to query data (SPL)
  - Splunk dashboard with diagrams and tables to present data

## → Customize Navigation

- We want to:
  - Show dashboard on a default view in this app
- We need to:
  - Change default view for the app

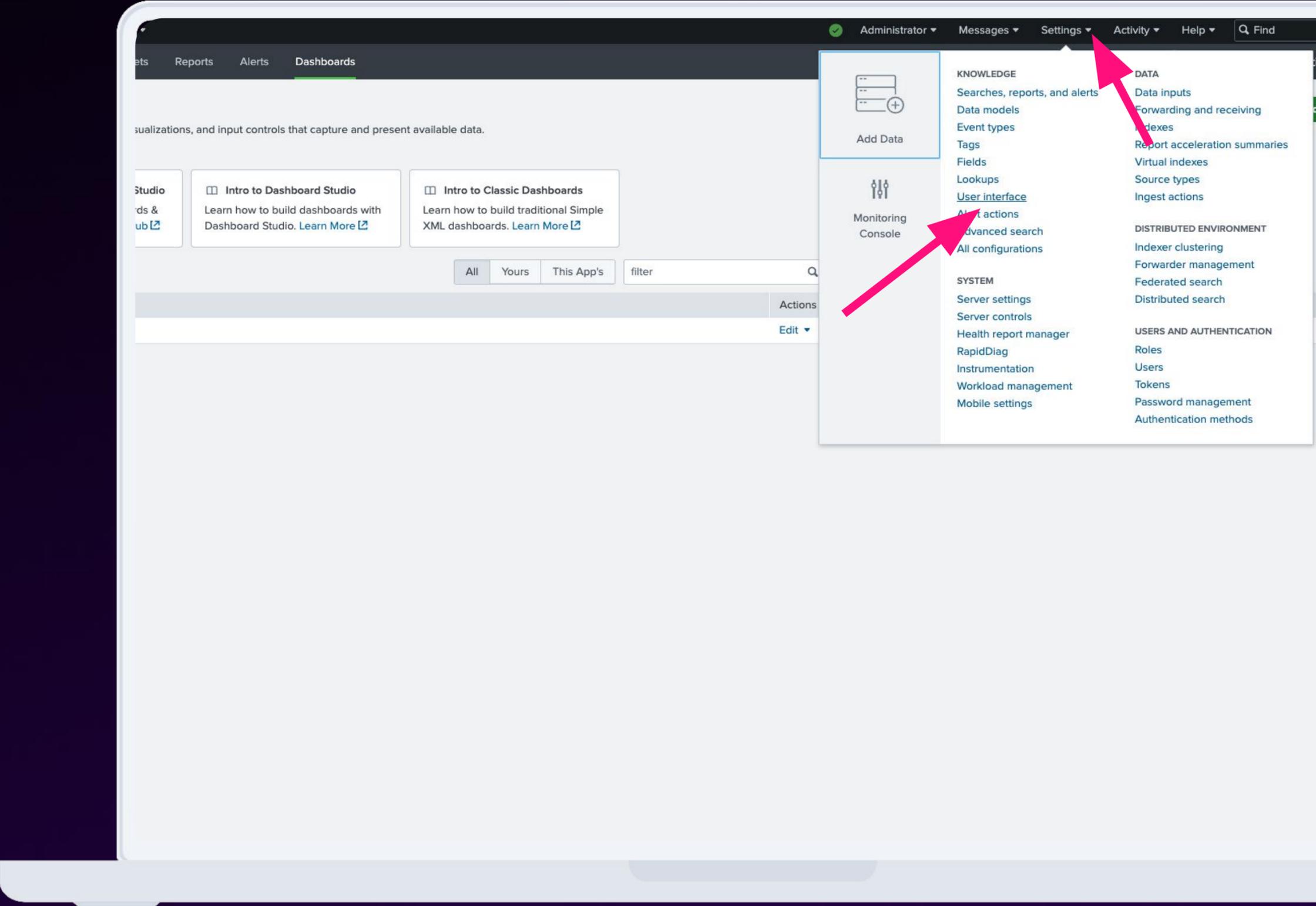
# Exercise

# 11



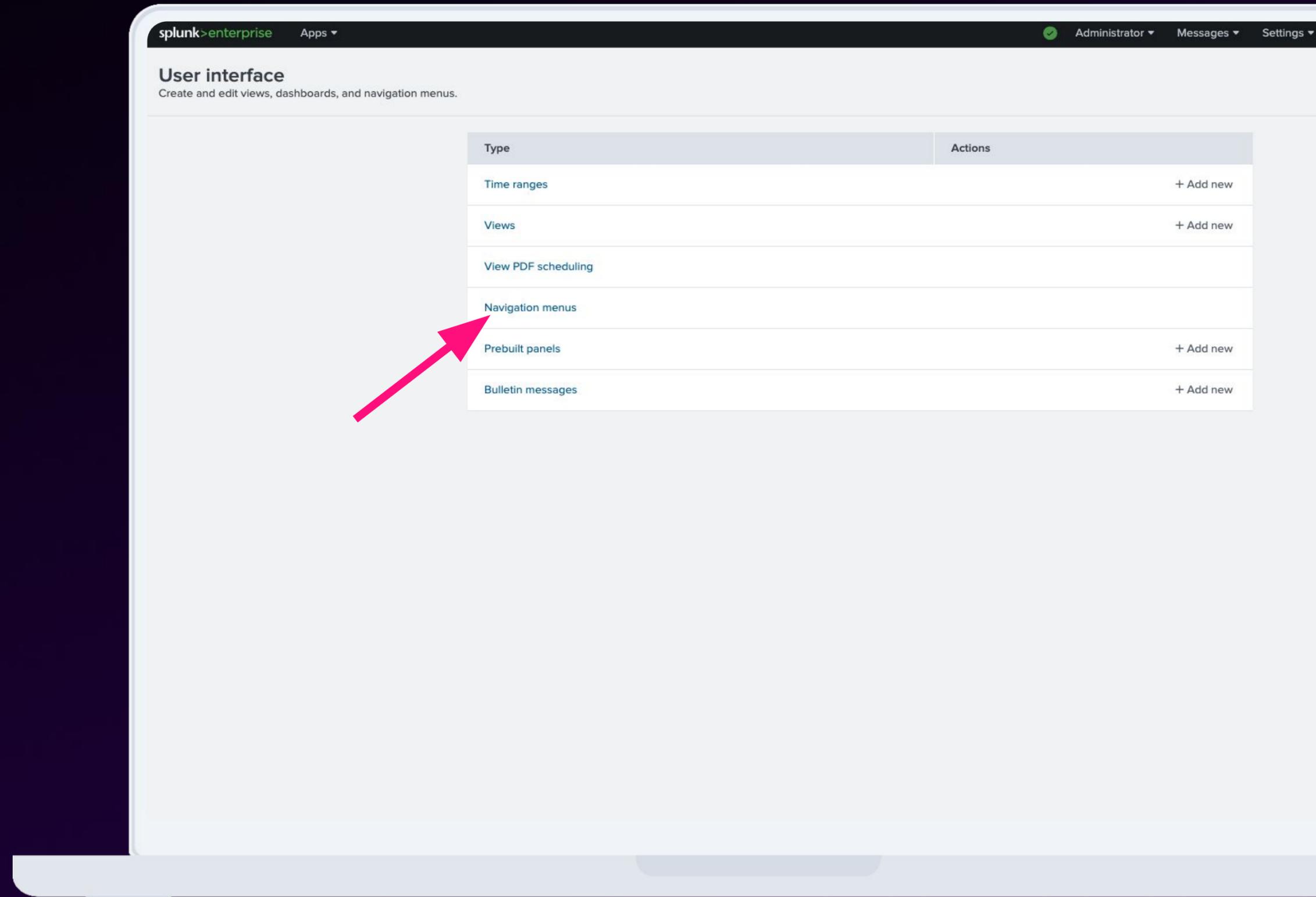
# Go to Settings

Settings  
User interface



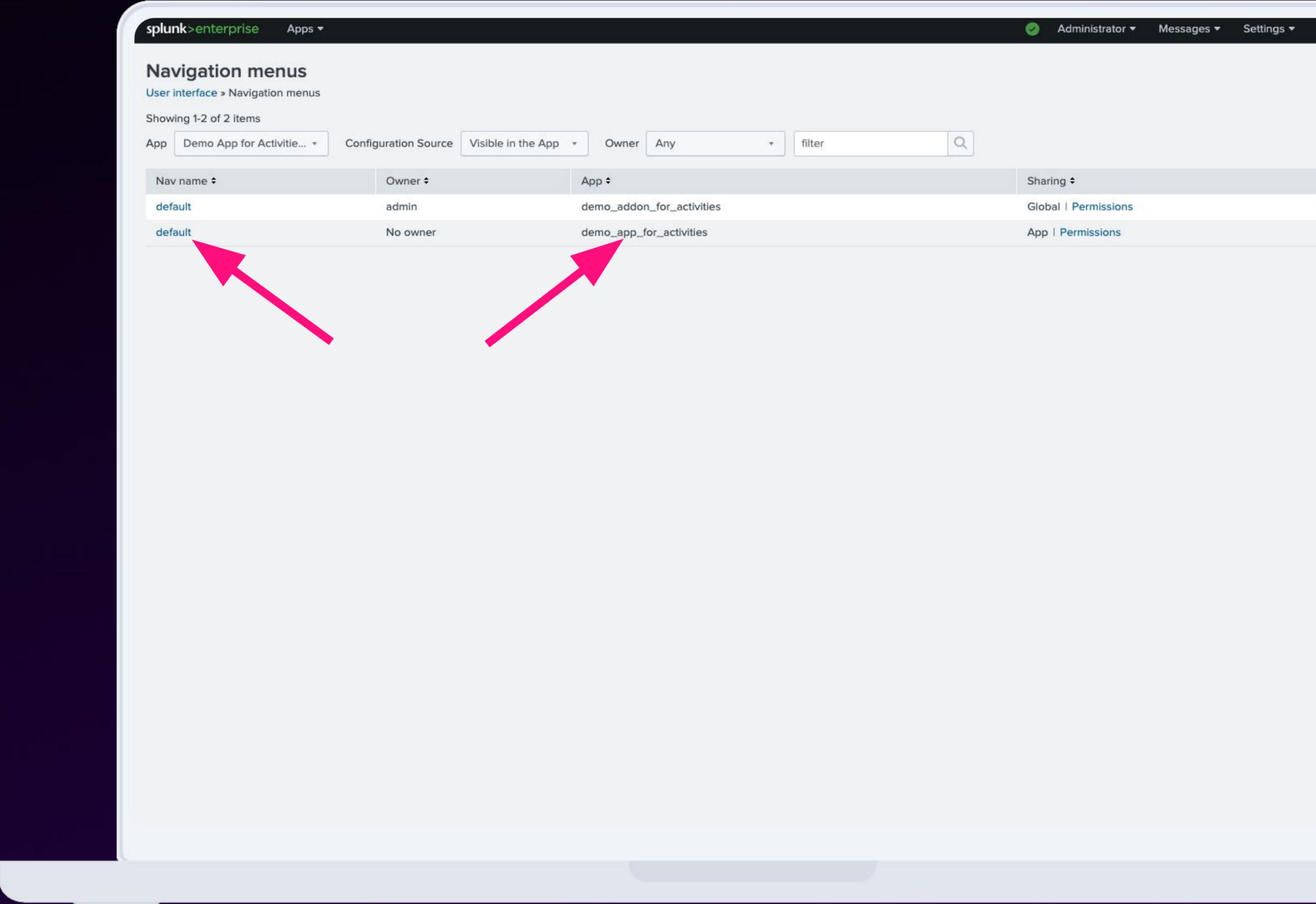
# Go to Settings

Navigation menus



# Go to Settings

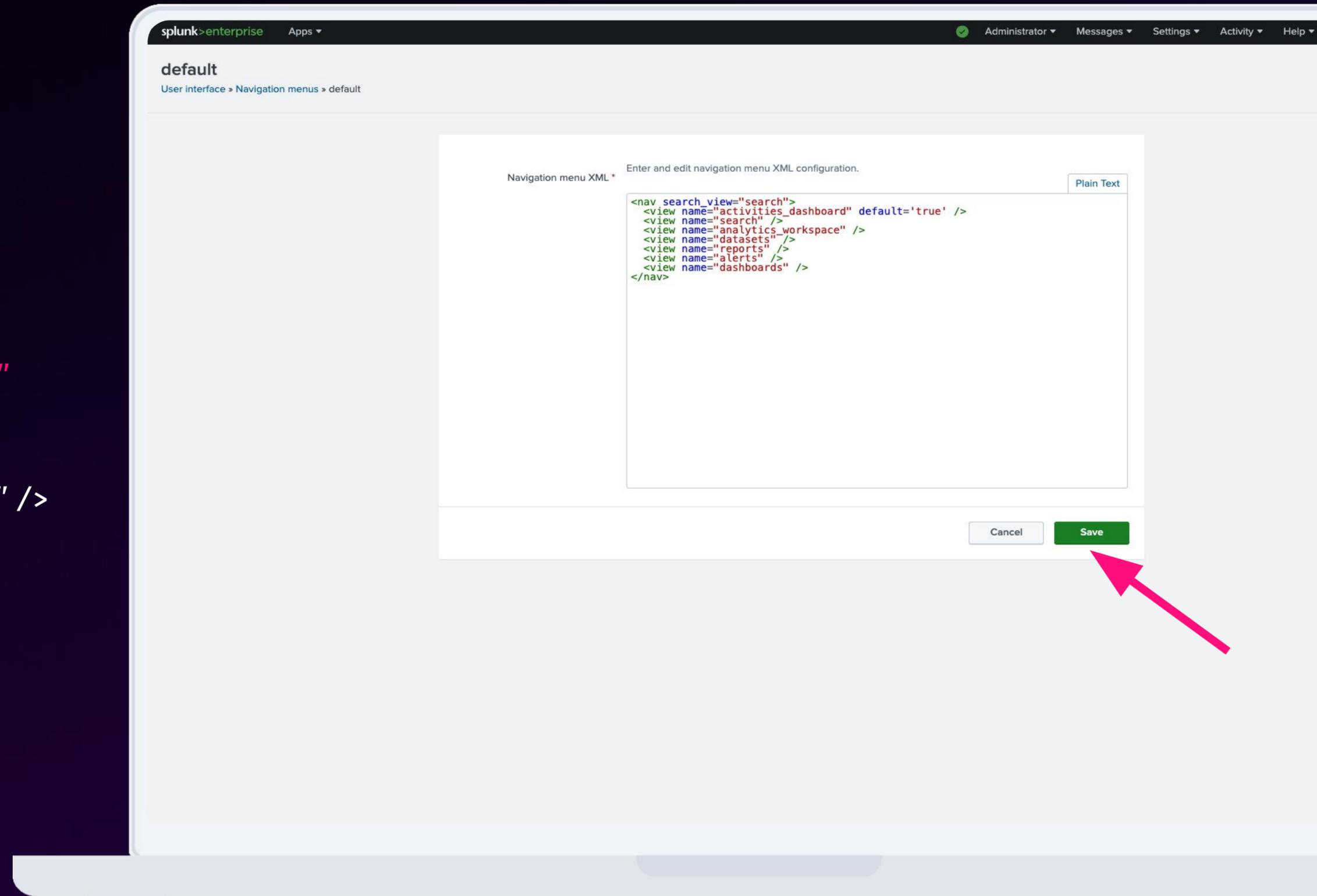
Click on “default” for demo\_app\_for\_activities



# Go to Settings

```
<nav search_view="search">  
  <view name="activities_dashboard" default='true' />  
  <view name="search" />  
  <view name="analytics_workspace" />  
  <view name="datasets" />  
  <view name="reports" />  
  <view name="alerts" />  
  <view name="dashboards" />  
</nav>
```

Click on **Save**



The screenshot shows the Splunk Enterprise web interface. The breadcrumb trail is "User interface > Navigation menus > default". The main content area is a "Navigation menu XML" editor. The editor title is "Navigation menu XML" and it contains the text "Enter and edit navigation menu XML configuration." and a "Plain Text" button. The XML code in the editor is:

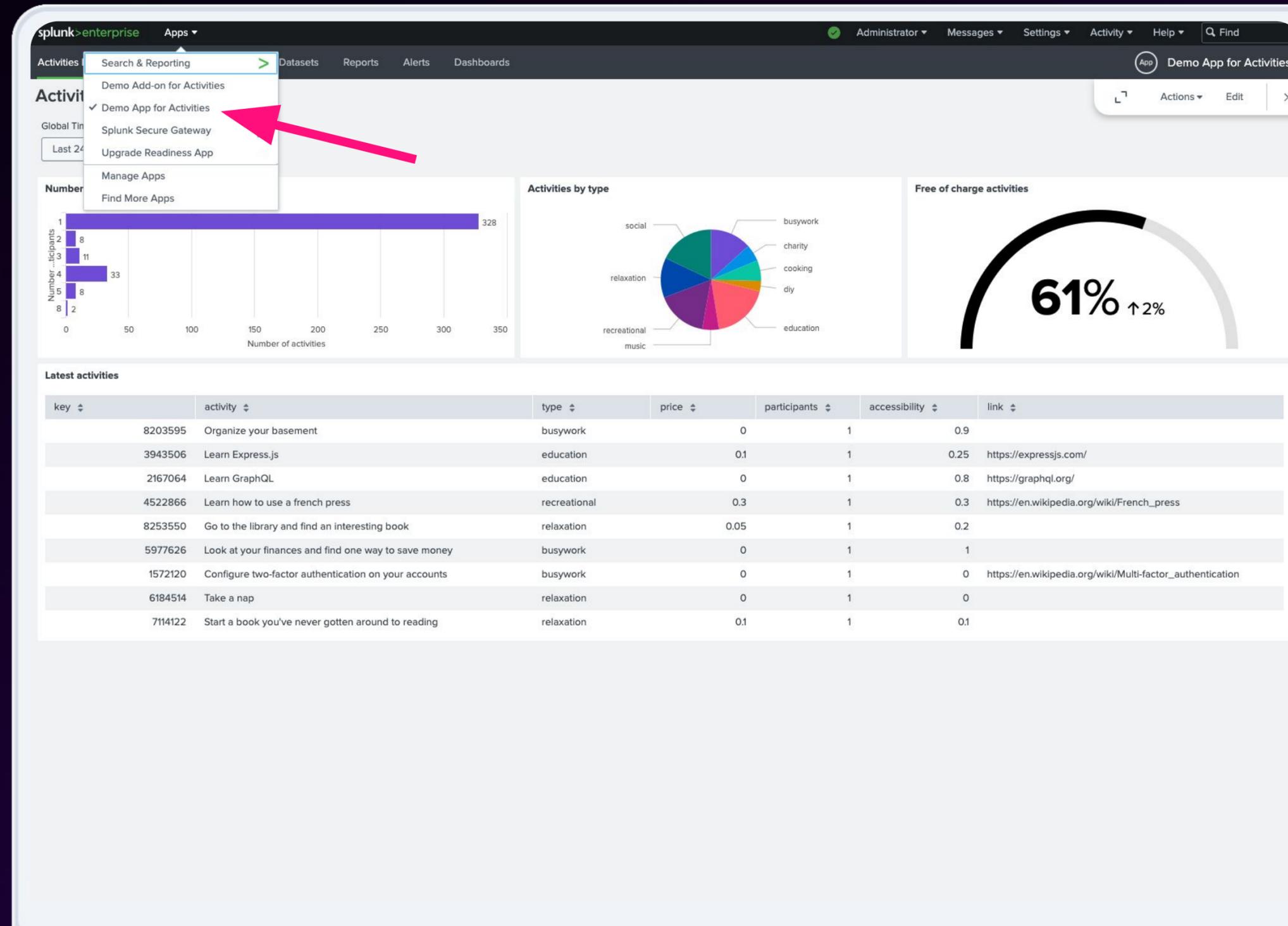
```
<nav search_view="search">  
  <view name="activities_dashboard" default='true' />  
  <view name="search" />  
  <view name="analytics_workspace" />  
  <view name="datasets" />  
  <view name="reports" />  
  <view name="alerts" />  
  <view name="dashboards" />  
</nav>
```

At the bottom right of the editor, there are two buttons: "Cancel" and "Save". A red arrow points to the "Save" button.

# Let's Check Our Work!

Select Demo App for Activities from Apps menu

The app displays our new dashboard!



# Dashboard App TODO

## ✓ Create an Empty App

- We want to:
  - Build a dedicated app for activities
- We need:
  - Splunk app to manage our knowledge objects

## ✓ Build a Dashboard

- We want to:
  - Create a dashboard to visualize data on activities
- We need:
  - Splunk searches to query data (SPL)
  - Splunk dashboard with diagrams and tables to present data

## ✓ Customize Navigation

- We want to:
  - Show dashboard on a default view in this app
- We need to:
  - Change default view for the app

# Package App

## SHOW CLI

Package your app for  
installation on other  
instances or for staging

```
# create package with app folder  
sudo tar -cvzf demo_app_for_activities_100.tgz -C  
/opt/splunk/etc/apps demo_app_for_activities
```

# Splunk App development - summary

- Visualize any types of data
- Customize navigation and other elements of the app
- Package app for distribution or installation on other environments

# Splunk Apps Best Practices



**Bring on  
the future.**

# Before You Start

- Review product documentation and product release notes for the latest features that might resolve your problem
- Always check [Splunkbase](#) and [Splunk Github](#) for existing solutions
- Use knowledge articles published by Splunk experts:
  - [Splunk Blogs](#)
  - [Splunk Lantern](#)
- Use Splunk Community resources to see if other Splunk users tried to solve a similar problem:
  - [Splunk Answers](#)
  - [splunk-usergroups](#) Slack channels
- Request new features for officially supported apps and add-ons via [Splunk Ideas](#) portal

# Apps and Add-ons Best Practices

# Splunk Add-ons

- Use [UCC Framework](#) for a quick start and code shareability
- Leverage SDKs (e.g. [Splunk SDK for Python](#)) to abstract code for basic operations
- Leverage [logging best practices](#) to manage exceptions
- Use json format, when feasible, to write data into index
- Follow Python best practices and coding standards
- Mock input data for development:
  - API - e.g.: Postman, Mockon
  - Event generation - e.g.: [Eventgen](#)

# Apps and Add-ons Best Practices

# Splunk Apps

Check [Splunk Dashboard Examples](#) for design inspirations

Use [Dashboard Studio](#) to build future proof dashboards

Learn [Splunk Processing Language](#) well! Optimize your SPLs.

Customize [app navigation](#) for your users

Secure data with [access control](#) configuration

Expand user experience with [Splunk UI Toolkit](#)

# Apps and Add-ons Best Practices

# When You're Done

-  **PACKAGE** your apps and add-ons for distribution
-  **PUBLISH** your work on [Splunkbase](https://splunkbase.com) to help others solve their problems
-  **SHARE** your code via Github to allow others contribute to your work
-  **CONTRIBUTE** to others' work to practice your amazing skills!

# Apps and Add-ons Best Practices

# Giveaways

Splunk Product Ecosystem

Apps & Integration



**Bring on  
the future.**

# Giveaways

Github links

## [Splunk Product Landscape](#)



## [Cisco + Splunk Integrations](#)



# DEVELOPER activations at .conf24

Meet us at the **Builder Bar** in source=\*Pavillion

## What's the Builder Bar?

- Interactive space for anyone who builds custom SPLs, dashboards, apps and integrations to extend Splunk capabilities.

## Key Sessions

- **DEV Breakout Sessions:** 11 sessions and workshops to dive deep into development topics.
- **Builder Bar Theater Sessions:** 14 sessions showcasing app building and best practices.
- **DEV Roundtable Sessions:** 8 Sessions focused on feedback, new features and more (there might be a surprise in store!)

## Special Features

- 'What's on your mind?': Share your ideas and feedback on note cards!
- Collaboration stations: Network and engage with experts on SPL2, GDI and all things app dev.
- Fun swag: Builder pins, Jenga tower and LEGO mug build!



Q/A



.conf24  
splunk>

Bring on  
the future.

# Thank you

