**Developer Guide for Splunk Cloud Platform and Splunk Enterprise**

This Developer Guide contains documentation for developers who want to build solutions for Splunk Cloud Platform and Splunk Enterprise. Use this documentation to learn how to create Splunk apps, add-ons, and integrations specific to your business needs.

**What is a Splunk app?**

A Splunk app is a packaged solution that runs in a Splunk Cloud Platform or Splunk Enterprise deployment. A Splunk app contains a collection of knowledge objects and extensions for a specific technology or use case. Developers can create Splunk apps to build solutions on top of the Splunk platform or to extend the Splunk platform so that your organization or your customers can more easily get value from the data in a Splunk platform deployment.

A Splunk app typically contains one or more dashboards with data visualizations, along with saved configurations and knowledge objects such as reports, saved searches, lookups, data inputs, a KV store, alerts, and more. If the standard Splunk platform configurations and knowledge objects don't address your specific needs, you can develop custom solutions by extending the Splunk platform using the following tools:

* Custom data inputs: Define scripted or modular data inputs to work with unique data sources or formats.
* Custom alert actions: Create alert actions that are integrated with your app's workflow.
* Custom visualizations and UI: Create your own data visualizations using third-party libraries.
* Custom search commands: Define your own Splunk Search Processing Language (SPL) commands to perform additional data analysis.
* Custom REST endpoints: Create a custom REST handler to make a custom REST API to automate your own activities or to interact with third-party services.

When used together in a Splunk app, these knowledge objects provide specialized insight:

* An app can handle getting data into the Splunk platform in different ways, such as by using a scripted input or modular input, and from different data sources.
* An app can include a custom UI with dashboards, reports, custom search commands, field extraction definitions, data lookups, a navigation menu, and custom alert actions.
* An app often targets a specific type of role, restricting read/write access by role. Therefore, different users of the same Splunk Cloud Platform or Splunk Enterprise instance can see only the data that is relevant to them.
* An app can be developed for a specific company to use internally or as a public solution available through Splunkbase.

You can also integrate external products or applications with the Splunk platform by building non-app solutions using the Splunk Enterprise SDKs or the Splunk Enterprise REST API. For more information, see [Developer tools](https://dev.splunk.com/enterprise/docs/devtools).

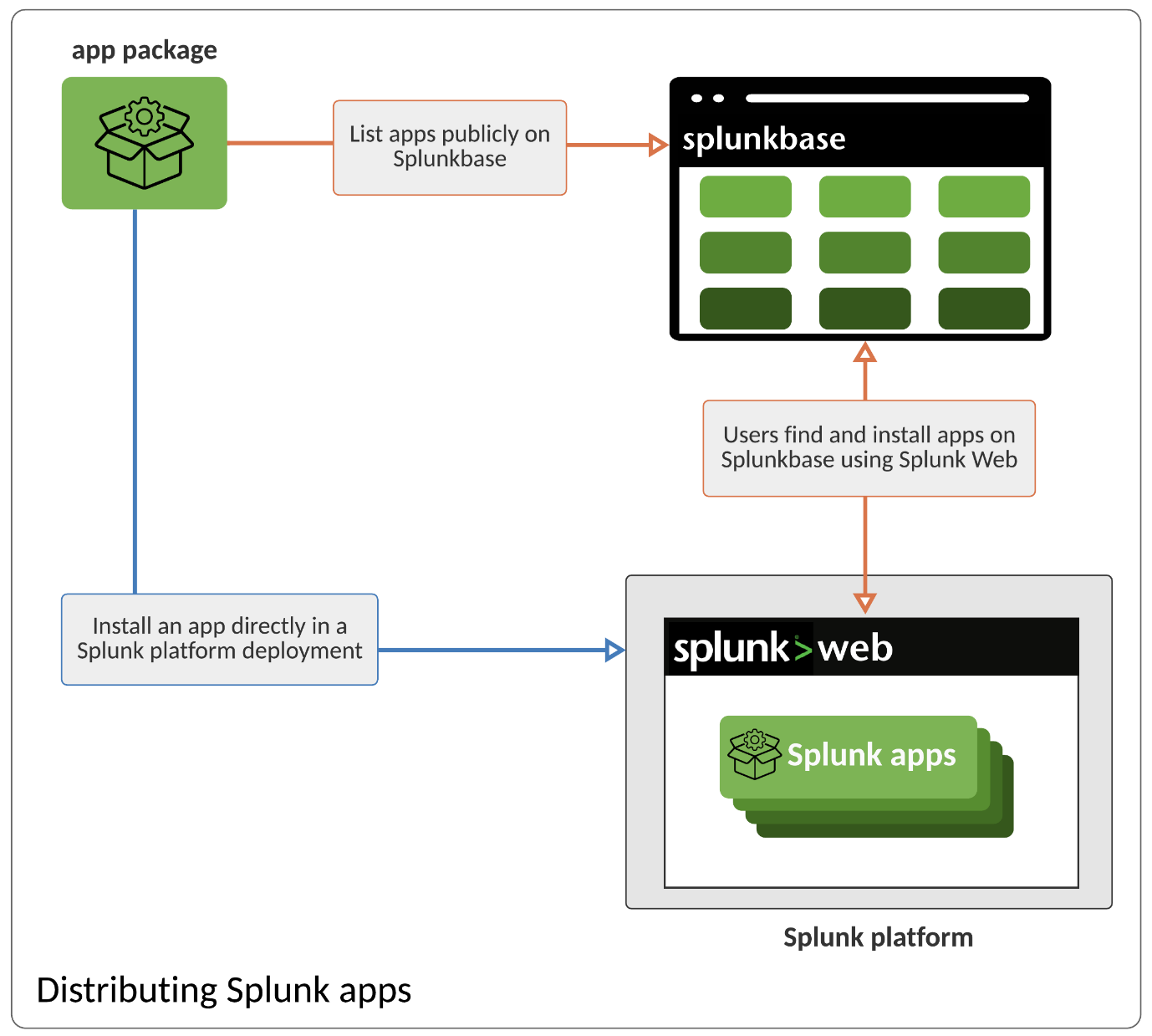
**What is a Splunk add-on?**

Technically, a Splunk add-on is a Splunk app. An add-on is usually an app that does not contain a full UI but provides some custom configurations or data inputs to support other Splunk apps. For example, an add-on might provide back-end functionality for another app, or data-parsing configurations that conform to the Splunk Common Information Model (CIM). A single add-on can be implemented as a reusable component in multiple apps, suites, or solutions.

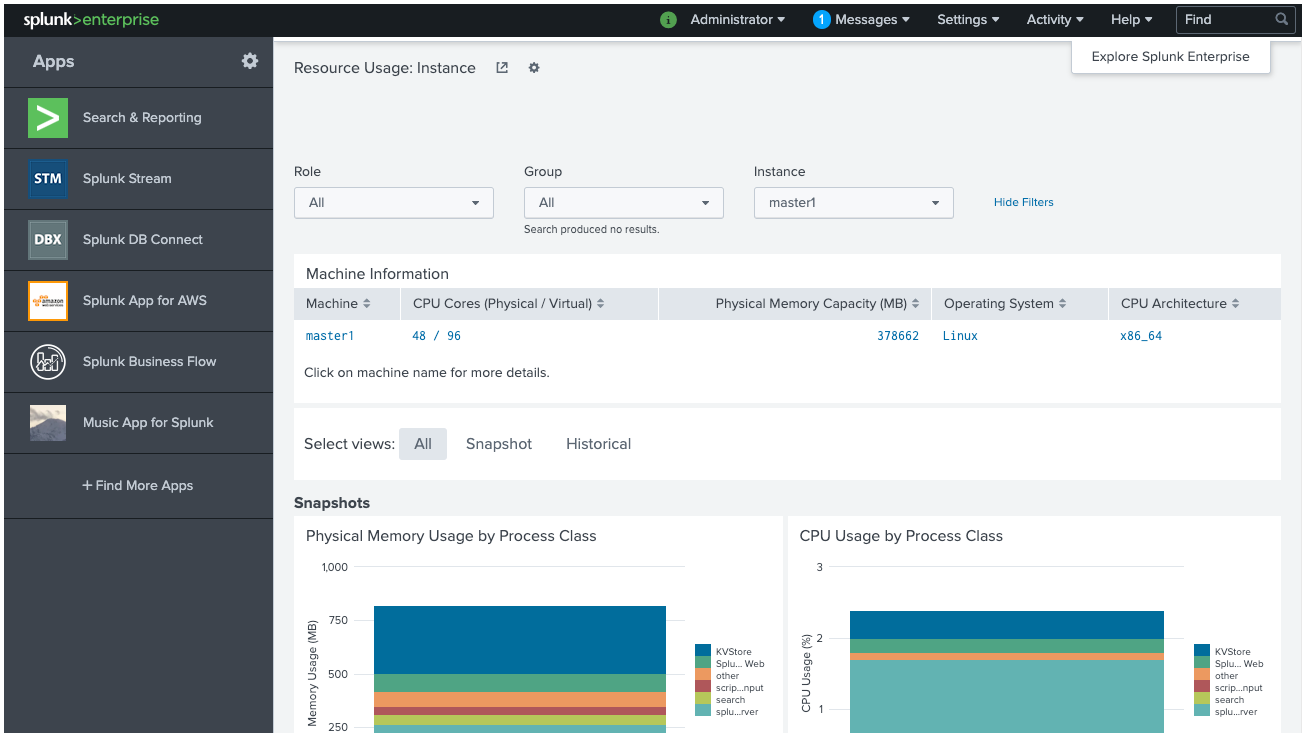
But because apps and add-ons are technically the same, the rest of this documentation refers to apps and add-ons collectively as apps.

**How do users find and install Splunk apps?**

[Splunkbase](https://splunkbase.splunk.com/) is the site for the Splunk community to find apps. You can upload public app packages to Splunkbase for Splunk customers to download and install to Splunk Cloud Platform or Splunk Enterprise deployments. Or, you can distribute app packages privately to your customers to run on a Splunk Cloud Platform or Splunk Enterprise deployment. The following diagram shows how to distribute Splunk apps:



Splunk apps are hosted in Splunk Cloud Platform or Splunk Enterprise in the Splunk Web interface. Users access Splunk apps from the home page or Apps menu as shown in the following image:



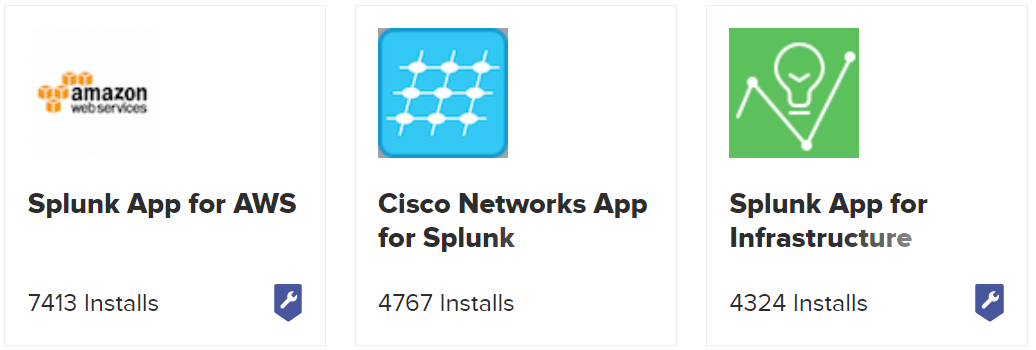
**Why make a Splunk app?**

While users of Splunk Cloud Platform or Splunk Enterprise can search data and generate reports directly using Splunk Web, a Splunk app typically provides a customized experience that targets a specific set of data for a specific purpose. For example, as a developer in an organization, you can create Splunk apps for the following purposes:

* Integrate with your company's product, using that product's company logo, styles, and branding.
* Showcase business analytics using custom dashboards and visualizations.
* Provide custom data inputs allowing all Splunk users to work with your specific type of data. Include data parsing and field normalization using the Common Information Model (CIM).
* Notify users of your technology that something has happened using custom alert actions. Integrate with third-party products using webhooks to respond to alerts.
* Share knowledge objects, such as saved searches, lookups, reports, and data models, to provide a task workflow for different roles in your organization. Use permissions to apply role-based access.

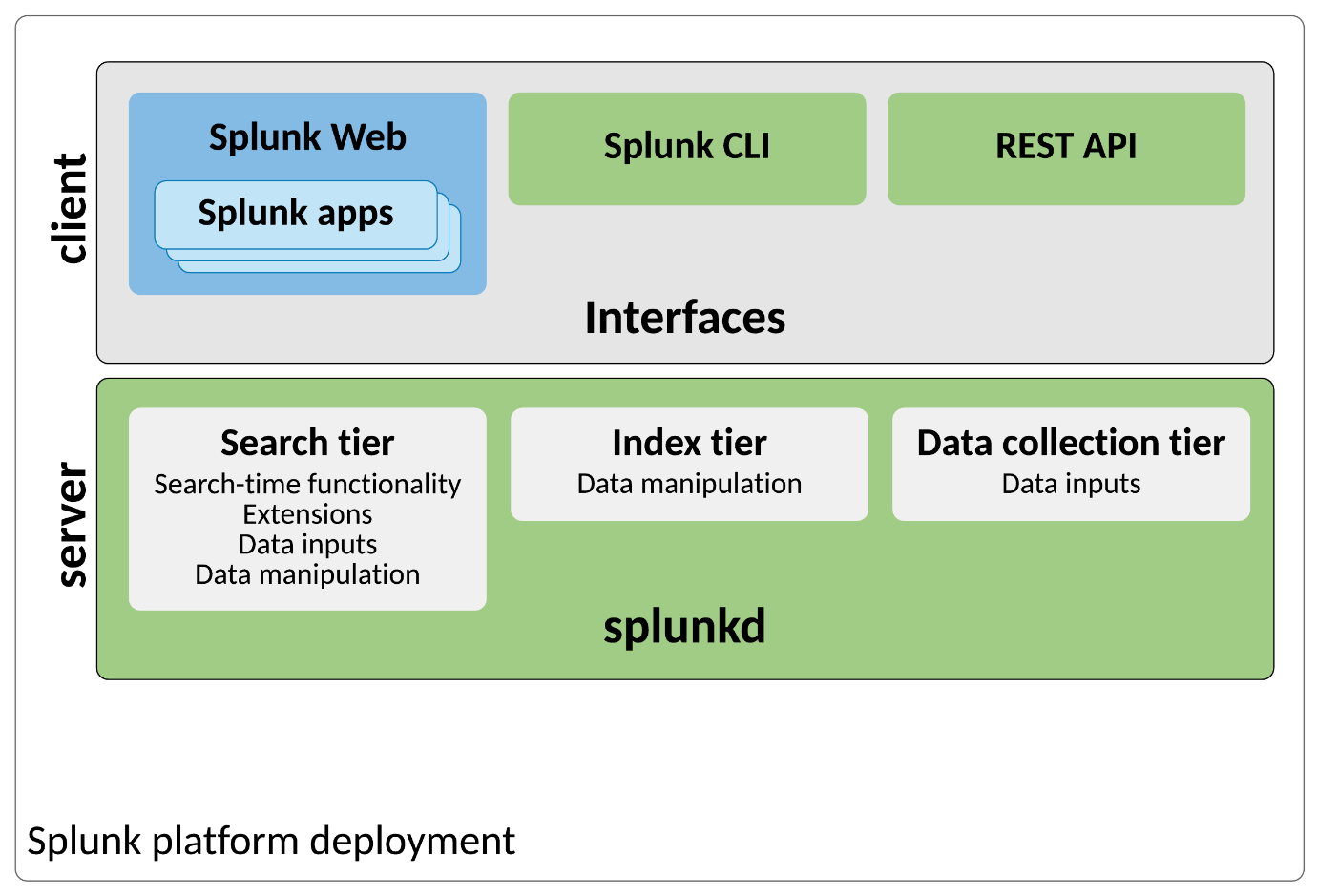
As a developer, you can make Splunk apps for your own organization, for clients of other organizations, or for everyone who uses Splunk software as a way to promote your products.

For ideas and examples, browse the variety of apps that are available on Splunkbase. Apps that are published to Splunkbase undergo quality controls. Before being published on Splunkbase, any apps for Splunk Cloud Platform must complete a vetting process to ensure they meet security requirements and work correctly. If an app is built by Splunk, the app gets a blue wrench badge. For example, here are some apps as they appear on Splunkbase:



**How do Splunk apps run?**

Apps are hosted in Splunk Web, which is a front-end web interface that supports HTML, CSS, and JavaScript. Splunk Web is built on the Pythonic web framework CherryPy and uses the Python 3.7 runtime. Users access Splunk Web with a URL in the format https://<server>:<port>. Apps with front-end components use the SplunkJS Stack libraries to interact with the Splunk platform server engine, splunkd, for all Splunk platform operations. Apps that do not have front-end UIs, such as technology add-ons, are still uploaded and managed in Splunk Web. The following diagram shows the client-server architecture of Splunk platform deployments:



You can also host Splunk-integrated applications outside of Splunk Web, and these applications might or might not use SplunkJS Stack libraries or the Splunk Enterprise SDKs to communicate with splunkd. For more information, see [Integrate with the Splunk platform](https://dev.splunk.com/enterprise/docs/devtools#Integrate-with-the-Splunk-platform).

**Where do Splunk apps run?**

Where Splunk apps run in Splunk Cloud Platform or Splunk Enterprise depends on the individual deployment. The functions of a Splunk platform deployment are categorized by the following tiers:

| **Tier** | **Component** | **Description** |
| --- | --- | --- |
| Search | Search head | Handles search-time functionality, and includes dashboards, searches, macros, tags, lookups, data models, and other knowledge objects. |
| Index | Indexer, Search peer | Handles data-manipulation functionality including parsing, indexing, and search. Indexes data, transforming raw data into events and placing results into an index. |
| Data collection | Inputs Data Manager, Universal forwarder, Heavy forwarder | Handles data inputs such as modular or scripted inputs, network inputs, and HTTP event collector. Forwards data to indexers. |

Here's a basic summary of the types of Splunk platform deployments:

* A single-instance deployment includes all three tiers of functionality.
* A distributed deployment separates these tiers across different components.
* A clustered deployment is similar to a distributed deployment, but includes additional cluster-management components and spreads load across multiple instances.

A Splunk app contains the configuration files and knowledge objects that perform different functions for the app. When apps are installed in a single-instance deployment, a single app package is adequate. However, for distributed and clustered deployments, you need to partition app packages into sub-packages for the tiers with the corresponding functionality that the app provides. For example, if an app contains a lookup and a data input, the lookup needs to be part of a sub-package that is installed on the search tier, and the data input needs to be part of a sub-package that is installed on the data collection tier.

You must design an app so that its features can be installed on the correct tier. When an app is published as a single package, IT admins might repackage the app into different components when deploying the app to avoid conflicts. For example, you don't want to install a data input on an indexer, nor do you want to install a dashboard on a forwarder. As an app developer, you can create these packages manually or use the Splunk Packaging Toolkit to partition apps into the correct packages.

**Troubleshoot**:

To troubleshoot a Splunk issue, systematically examine log files, check Splunk processes and network connectivity, use Splunk's built-in tools like the Search Job Inspector, and consult the Splunk documentation and community resources.

Here's a more detailed breakdown of troubleshooting steps:

1. Identify the Problem:

* **What's happening?** Pinpoint the specific issue: is data not being ingested, searches are slow, or is Splunk down?
* **When did it start?** Note the time the problem started to help narrow down potential causes.
* **Where is the problem?** Is it a specific index, forwarder, or the Splunk instance itself?

2. Check Splunk Logs:

* **Location:** Splunk logs are typically found in $SPLUNK\_HOME/var/log/splunk.
* **Key logs:**
  + splunkd.log: General Splunk server logs.
  + splunk-web.log: Logs related to the Splunk web interface.
  + audit.log: Logs user actions and system events.
  + metrics.log: Logs Splunk metrics and performance data.
  + splunk\_stderr.log: Logs errors from the Splunk daemon.
  + crash-\*: Logs related to Splunk crashes.
* **Use Splunk search:** You can search Splunk's internal index (\_internal) for specific error messages or keywords.

3. Check Splunk Processes and Connectivity:

* **Verify Splunk is running:**

Use the command splunk status (from $SPLUNK\_HOME/bin) to check the status of the Splunk daemon.

* **Check process status:**

On Linux, use ps -ef | grep splunkd to see if the Splunk process is running.

* **Network connectivity:**
  + **Ping:** Ensure that forwarders can ping the indexer.
  + **Telnet:** Use telnet to check if the forwarder can connect to the indexer on port 9997.
  + **Firewall:** Verify that firewalls aren't blocking communication between forwarders and indexers.

4. Use Splunk's Built-in Tools:

* **Search Job Inspector:**

Use this tool to analyze the performance of searches and identify bottlenecks.

* **btool:**

Use splunk btool to inspect configuration files and identify potential issues.

* **Input Status:**

Check the status of data inputs using the Splunk web interface (Settings > Data Inputs).

5. Consult Splunk Documentation and Community:

* **Splunk Documentation:**

The Splunk documentation is a valuable resource for troubleshooting and understanding Splunk's architecture and configuration.

* **Splunk Answers:**

The Splunk Answers community forum is a great place to find answers to common problems and ask for help.

* **Splunk Community:**

The Splunk community is a valuable resource for sharing knowledge and getting help from other Splunk users.

6. Common Issues and Solutions:

* **Data not being ingested:**
  + **Forwarder issues:** Check if the forwarder is running and configured correctly.
  + **Network problems:** Ensure network connectivity between the forwarder and indexer.
  + **Indexer issues:** Check if the indexer is receiving data and if the relevant indexes are configured correctly.
* **Slow searches:**
  + **Search complexity:** Simplify your searches to improve performance.
  + **Index health:** Ensure that your indexes are healthy and properly configured.
  + **Resource constraints:** Check if the Splunk instance has sufficient resources (CPU, memory, disk space).