Platform & Data

Splunk Core Certified User

KNOWLEDGE CHEATSHEET

FROM SOURCE TO SEARCHABLE DATA

Forwarder Collect → **Parsing** Form events → **Indexer** Store → **Search head** Query

01 Platform and data ingestion

Components and deployment

- **Splunk** searches, analyzes and visualizes machine data, including structured records and unstructured logs from applications, servers, network devices and databases.
- **Forwarders** collect and send data. **Indexers** process and store searchable events. **Search heads** coordinate searches and present results; one Enterprise instance can perform several roles.
- **Universal forwarders** are lightweight collectors. **Heavy forwarders** use a full Splunk instance and can parse, filter, mask and route events. General event parsing occurs on a heavy forwarder or indexer.
- **Splunk Enterprise** can be self-managed. **Splunk Cloud Platform** provides a managed service. **Splunk Web** commonly listens on port **8000**; **8089** is the management port.

Apps, roles and preferences

- **Apps** package configuration, inputs, knowledge objects and interfaces for a use case. **Add-ons** commonly supply collection and parsing support. **Splunkbase** distributes apps and add-ons.
- **Search & Reporting** provides search, statistics, visualizations, reports, dashboards and alerts. Specialized solutions include **Enterprise Security**, **IT Service Intelligence** and **User Behavior Analytics**.
- **User**, **Power** and **Admin** are common built-in roles. Roles combine capabilities and index access; custom roles can change effective permissions. Power commonly adds sharing and alert-authoring capabilities.
- Use the signed-in user menu for **preferences**, including full name, time zone and default app. The configured time zone affects displayed event times and calendar-based time boundaries.

Inputs and event processing

- **Upload** ingests a local file once. **Monitor** configures ongoing inputs such as files, directories, network ports or scripts. **Forward** supports collection from remote machines through forwarders.
- Inputs can be configured through **Splunk Web**, the **CLI**, apps or add-ons, and **inputs.conf**. **HTTP Event Collector (HEC)** accepts authenticated HTTP submissions.
- **Input** reads the source; **parsing** breaks data into events, determines timestamps and assigns metadata; **indexing** builds searchable structures and stores events. Ingest licensing measures applicable data before storage compression.
- **Host** identifies the originating machine or assigned host value; **source** identifies the input origin; **sourcetype** describes the data format; **index** selects the event store.
- Review **sourcetype detection** and timestamp recognition before ingesting. Set host, index and app context deliberately; administrators can create indexes and change input settings with the required permissions.

Search & Time

Splunk Core Certified User

KNOWLEDGE CHEATSHEET

BUILD A SELECTIVE SEARCH

Time Bound the range → **Index** Choose data → **Terms** Filter events → **Pipe** Process results

02 Search syntax and time

SPL pipeline and Boolean logic

- **SPL** combines search terms, commands, functions, arguments and clauses. A **pipe (|)** passes one command's results to the next: `index=web status=200 | stats count by host`.
- **field=value** targets a field: `host=www3 status=503` requires both conditions. Without an explicit index, Splunk searches the user's permitted **default indexes**, not every accessible index.
- Write Boolean operators as **AND**, **OR** and **NOT**. Adjacent search terms imply AND. Use parentheses to make grouping explicit: `(index=netfw failure) OR (index=netops (warn OR critical))`.
- In the base **search** command, OR binds before AND; **eval** and **where** use AND before OR. Parentheses prevent ambiguity when mixing operators.
- **Quotation marks** preserve phrases or values containing spaces: `"failed password"`. **Field names are case sensitive**; ordinary search terms and search field-value matching are generally case insensitive. Eval and where string comparisons differ.

Wildcards, missing fields and efficiency

- **Asterisk (*)** matches multiple characters: `host=www*`. Prefer a known prefix over leading or embedded wildcards; a broad wildcard can increase the data scanned.
- **status!=100** requires status to exist with a different value. **NOT status=100** also includes events without status. **status=*** tests for a value; **NOT status=*** finds events without one.
- Narrow **time** and **index** first, then use specific host, sourcetype and other selective terms. Prefer precise inclusion filters over broad exclusions when they express the same requirement.
- Filter early and discard unneeded fields before expensive processing. Keep presentation commands near the end. Select multiple indexes with `(index=web OR index=sales)`, rather than requiring one event to belong to both.

Relative time and snapping

- The **time range picker** offers presets, relative ranges, calendar dates and advanced expressions. **earliest=** sets the start; **latest=** sets the end. Explicit search modifiers override the picker for that search.
- **earliest=-2h latest=now** searches the preceding two hours when the job runs. A historical relative search has a fixed execution window; a **real-time** search continually updates its selected window.

Search & Time / Fields & Results

Splunk Core Certified User

KNOWLEDGE CHEATSHEET

UNDERSTAND THE RETURNED FIELDS

- **Metadata** — Event identity
- **Extraction** — Search-time fields
- **Sidebar** — Explore values
- **Commands** — Shape output

- Common units are **s** seconds, **m** minutes, **h** hours, **d** days, **w** weeks, **mon** months and **y** years. **@** snaps backward to the beginning of the specified unit, after applying the offset.
- **earliest=-1d@d latest=@d** selects yesterday from midnight to midnight. **Last 24 hours** is a rolling duration relative to execution; it is different from the previous calendar day.
- **earliest=-72h@h latest=@d** starts 72 hours ago rounded down to the hour and ends at today's midnight. At 03:35, **-30m@h** offsets to 03:05, then snaps to 03:00.

Timeline and search assistance

- The **timeline** shows event counts in time buckets. Peaks suggest bursts; valleys may indicate quiet periods or collection gaps. Hover for a bucket's time and count.
- **Click or drag** on the timeline selects a period within current results. **Zoom to selection** narrows the search range and reruns it; **Zoom out** expands and reruns it. Deselect clears the selection.
- **Search Assistant** offers completions and syntax help. Highlighting and matching parentheses help edit SPL; use command syntax and grouping, rather than color alone, to interpret a search.

03 Fields and result refinement

Fields, metadata and search modes

- **Fields** are searchable name-value pairs. Splunk assigns core metadata during ingestion and extracts many other fields at **search time**, based on sourcetype rules and patterns such as key-value pairs.
- **_raw** holds the original event text; **_time** holds the event timestamp; **_indextime** records indexing time. Internal fields commonly begin with an underscore; host, source and sourcetype do not.
- **Fast mode** limits field discovery to fields needed by the search. **Verbose mode** performs fuller discovery and returns event detail. **Smart mode** adjusts behavior according to whether the search transforms events.
- **Data Summary** lists hosts, sources and sourcetypes visible to the user. Use it to explore available data, then constrain a search rather than scanning everything.

Fields sidebar and event views

- **Selected Fields** appear beneath events; host, source and sourcetype are standard selections. **Interesting Fields** normally occur in at least **20% of returned events**. **All Fields** exposes additional discovered fields.
- The sidebar uses **a** for text values and **#** for numeric values. The number beside a field describes its distinct values in the results; click the field for value distributions and quick reports.

Fields & Results / Statistics & Charts

Splunk Core Certified User

KNOWLEDGE CHEATSHEET

UNDERSTAND THE RETURNED FIELDS

- **Metadata** — Event identity
- **Extraction** — Search-time fields
- **Sidebar** — Explore values
- **Commands** — Shape output

- Promote a field into **Selected Fields** through its field dialog or All Fields. Selecting a field changes its display beneath events; it does not itself filter the results.
- Event views include **List**, **Table** and **Raw**. Events normally appear newest first. Click a field value or highlighted term to add it to the search, exclude it or open another search.

Keep, reshape and order results

- **fields + host status** keeps specified fields, with applicable internal fields retained. **fields - unwanted** removes a field. This changes the result fields, not the indexed source data.
- **table action src dest** presents only those columns, in that order. It does not aggregate events. **rename action AS "Customer Action"** changes the output field name; use the original field's exact case.
- **sort status, -bytes** sorts by status ascending, then bytes descending. A minus sign requests descending order. **dedup host** keeps one result per host according to the incoming result order.
- After a **transforming command**, only the generated result fields remain. Keep needed grouping fields or aggregate them explicitly; a later table or rename cannot recover discarded event fields.

04 Statistics and visualizations

Aggregates and split-by fields

- **stats** transforms matching events into aggregate rows. **stats count** counts events; **stats count(vendor_action)** counts events with a value in that field.
- **sum(bytes)** adds numeric values; **avg(bytes)** finds their mean; **min** and **max** return extrema. **AS** labels an output: `stats count AS "Event Count"`.
- **BY** groups the results: `stats count sum(bytes) AS total_bytes by host`. Without BY, the aggregate covers the whole result set; multiple BY fields create groups for their combinations.
- **dc(user)** counts distinct users. **values(user)** returns distinct values. **list(user)** retains repeated values in event order, subject to command limits. A distinct count is a number, not a list.

Most and least common values

- **top status** returns the most frequent status values, normally ten, with **count** and **percent** columns. **rare status** returns the least frequent values; frequency is different from numeric magnitude.
- **top limit=5 status** requests five common values. **rare limit=15 dest_ip** requests fifteen uncommon values. **top 5 status** is also valid shorthand.

Statistics & Charts / Saved Content

Splunk Core Certified User

KNOWLEDGE CHEATSHEET

FROM EVENTS TO A CHART

Events Filter → **Aggregate** Calculate → **Group** Split by fields → **Visualize** Compare

- **top 3 status by user** returns up to three common status values **for each user**. The BY clause changes the grouping rather than requesting three overall rows.
- **showperc=f** hides the percentage column; **showcount=f** hides counts. **countfield=total** renames the count column; **percentfield=share** renames the percentage column.

Tables, charts and exports

- Use a **transforming search** to produce a statistical table for standard charts. The **Statistics** tab shows aggregate rows; **Visualization** displays the selected chart.
- **timechart count by host** builds time buckets and a series per host. **chart count over status by host** compares categories. **stats** produces general aggregates without automatically making time buckets.
- **Line charts** show trends for one or more numeric series, usually with time on the horizontal axis. **Bar and column charts** compare categories; **pie charts** show a limited set of parts of a whole.
- Configure chart type, labels, axes and legends before saving a report. **Drilldown** can use a table cell or chart value to open related events or another view; behavior depends on panel configuration.
- Search results can be exported as **CSV**, **XML** or **JSON**; **raw events** apply to event output. **PDF** is a presentation export where supported, rather than a structured statistics-table format.

05 Jobs, reports and dashboards

Search jobs and retained results

- Every search execution creates a **job** with its own results and lifetime. Use job controls to pause, resume, stop or inspect a running search, and **Activity > Jobs** to revisit retained jobs.
- Opening an existing **job** displays its original results. Rerunning its SPL creates a new execution, potentially with a new relative time window and newly indexed events.
- Completed **ad hoc jobs** normally have a **10-minute** lifetime. **Edit Job Settings** offers a seven-day extension; viewing an active retained job refreshes its lifetime. Scheduled jobs have separate retention settings.
- A **report** saves the search definition for reuse; it does not permanently archive one job's results. Export data for a durable copy, and use a schedule when new results must be generated regularly.

Saving, scheduling and sharing reports

- **Save As > Report** can save an event search or a transforming search, with its chosen presentation. A report does not require a chart. Use descriptive, consistent names that identify group, object and purpose.
- After saving, edit the report's **schedule**, time range, permissions and supported acceleration settings. The **schedule** controls when it runs; the **time range** controls which events each run considers.

Saved Content

Splunk Core Certified User

KNOWLEDGE CHEATSHEET

CHOOSE WHAT TO RETAIN

- **Job** — Execution results
- **Report** — Saved search
- **Schedule** — Repeated runs
- **Dashboard** — Related panels

- **Scheduled reports run as their owner**, so the owner's search permissions determine the accessible data. Unscheduled shared reports can use owner or user execution permissions where configured.
- **Sharing scope** and **role permissions** govern who can read or edit a knowledge object. Choose private, app or global sharing deliberately; sharing an object is separate from granting direct index access.
- Scheduled reports automate recurring results and can support configured delivery actions. **Alerts** add explicit trigger conditions and actions; a report schedule alone is not a condition-based notification.

Inline searches and report-backed panels

- **Dashboards** combine panels for a common purpose. An **inline panel** stores its search with the dashboard. A **report-backed panel** references a reusable saved report.
- Reuse a **saved report** across dashboards when the same search logic should stay consistent. Changes to that report propagate to dependent panels; change the underlying report to edit its saved search.
- Dashboard editing can move panels and change supported visualization settings. Report-backed panels can retain panel-specific presentation settings; search ownership remains with the referenced report.
- Dashboard inputs and time pickers control searches that are configured to use them. A panel based on a **scheduled report** can reuse scheduled results; its stored report range may differ from the dashboard picker.

Lookups & Alerts

Splunk Core Certified User

KNOWLEDGE CHEATSHEET

ENRICH AND ACT

- **Lookup** — Reference data
- **Match** — Add fields
- **Condition** — Evaluate results
- **Action** — Notify

06 Lookups and alerts

Lookup tables and field enrichment

- **Lookups** enrich events by matching existing field values to a reference dataset. Types include **CSV file**, **KV Store**, **external** and **geospatial** lookups. Ordinary enrichment occurs at search time.
- For a managed CSV lookup, upload a header-based **lookup table file**, then create a **lookup definition** that names and configures it. Match field names and check sharing permissions.
- **| inputlookup products.csv** displays the lookup rows as search results. It helps inspect headers and contents; it does not enrich a preceding event stream.
- **lookup product_catalog product_id OUTPUT product_name** matches the event's product_id against the named definition and adds product_name. Use **AS** when the event and lookup key names differ.
- Configure an **automatic lookup** after the table and definition exist, then associate it with the relevant data and fields. Matching searches gain the enrichment without an explicit lookup command.
- **outputlookup** writes results to a lookup destination. **OUTPUT** can overwrite existing output fields; **OUTPUTNEW** preserves fields already present. Reading, enriching and writing are distinct operations.

Trigger conditions, actions and noise

- **Alerts are saved searches** evaluated on a schedule or in real time. An alert triggers when its results meet a defined condition, such as a result count threshold.
- **Once** triggers one action set for a matching result set; **For each result** can trigger an action set per row. Choose Once when a batch should produce one notification.
- **Throttling** suppresses repeated notifications for a configured interval, optionally using selected fields. Tune conditions, schedule and time range together to avoid duplicate or missed detections.
- Actions can include **email**, **webhooks**, logging or installed custom actions. Legacy scripted alerts locate scripts in **$SPLUNK_HOME/bin/scripts**; script availability and permissions are deployment-dependent.
- Use alert and **Triggered Alerts** views to review definitions and recorded firings. App, owner, severity and alert type help organize them; a triggered-alert record requires the corresponding action or tracking configuration.
- **Capabilities**, ownership and object permissions control alert creation and editing. Scheduled alerts execute with the owner's access, so review who receives results as well as who can edit the definition.