null
# Table of Contents

**Overview**..................................................................................................................1
- Welcome to the Splunk Success Framework..............................................................1
- Release notes for the SSF Handbook........................................................................4

**Fundamental best practices**..........................................................................................6
- About the SSF fundamental best practices...............................................................6
- Purpose and scope best practices for a Splunk deployment.......................................6
- Executive sponsor engagement best practices for a Splunk deployment..................8
- Success measurements best practices for a Splunk deployment.............................9
- Operating framework best practices for a Splunk deployment...............................10

**Functional areas overview**..........................................................................................13
- About the SSF functional areas................................................................................13
- Platform best practices overview............................................................................13
- Program best practices overview.............................................................................15
- Data best practices overview..................................................................................19
- People best practices overview...............................................................................22

**Success Framework best practices**.............................................................................24
- Change management best practices for a Splunk deployment.................................24
- Communication best practices for a Splunk deployment.........................................32
- Community portal best practices for a Splunk deployment.....................................34
- Back up and restore best practices for a Splunk deployment....................................36
- Data onboarding best practices for a Splunk deployment........................................38
- User enablement best practices for a Splunk deployment........................................41
- Lab environment best practices for a Splunk deployment........................................43
- Logging best practices for a Splunk deployment.....................................................45
- Naming convention best practices for a Splunk deployment.....................................47
- Newsletter best practices for a Splunk deployment..................................................52
- Quarterly business review best practices for a Splunk deployment.........................54
- Responsibility assignment best practices for a Splunk deployment.........................56
- Roles best practices for a Splunk deployment............................................................59
- Role-based data management best practices for a Splunk deployment.......................63
- Sandbox best practices for a Splunk deployment.....................................................68
- Service-level best practices for a Splunk deployment...............................................72
- Showback best practices for a Splunk deployment...................................................77
- Utility tier disaster recovery best practices for a Splunk deployment.........................79
- Staffing best practices for a Splunk deployment.......................................................82
- Staffing recommendations.........................................................................................85
Table of Contents

Success Framework best practices
  Stakeholder best practices for a Splunk deployment..........................85
  Unix profile best practices for a Splunk deployment..........................88
  Welcome page best practices for a Splunk deployment........................92
  Workspace best practices for a Splunk deployment............................95
Overview

Welcome to the Splunk Success Framework

The Splunk Success Framework (SSF) is a flexible collection of best practices that accelerate and increase the value you derive from your data using Splunk software.

The best practices in the Splunk Success Framework include everything your organization needs to implement and maintain a thriving Splunk environment that focuses on what you want to do with your data. Whether you use Splunk Cloud, host Splunk Enterprise on premises, or have a hybrid of both, the Success Framework fits flexibly into any business model to support everything from ad-hoc searches to enterprise-wide solutions for organizations of all sizes.

The goal of the Success Framework’s best practices is to achieve a successful Splunk deployment and help people in your organization think differently about their data and its potential to enlighten. The Success Framework’s best practices are designed to increase time to value, drive adoption across your organization, and enable your Splunk environment to scale flexibly as you grow.

The best practices in the Success Framework are modular. You can apply any practice any time according to your needs and priorities.

About the Success Framework Handbook

The Success Framework Handbook provides reference materials, templates, and expert guidance for every aspect of your Splunk implementation, from data onboarding and platform management to suggestions for user education.

The Success Framework Handbook starts with best practices that establish strong fundamentals, then offers implementation best practices organized into four functional areas that support basic, intermediate, and advanced goals.

Most of the best practices in the Success Framework apply to both Splunk Cloud and on-premises Splunk Enterprise deployments. Where needed, topics specify whether something applies to only one or the other.
**Fundamental best practices**

Fundamental best practices are decisions, agreements, and success criteria that establish the purpose, goals, and ownership of your Splunk implementation. These tactical decisions provide clarity and accountability that are essential elements of a successful deployment. The Success Framework Handbook lays out four fundamental best practices:

Determine the purpose and scope of your Splunk deployment  
A purpose sets out the objectives of your Splunk implementation.

Identify an executive sponsor  
An executive sponsor is the leader accountable for the success of your Splunk implementation.

Establish success measurements  
Metrics set benchmarks so you can measure success as your Splunk implementation matures.

Establish an operations framework  
An operations framework defines how to set up your Splunk environment depending on your goals, and best practices for setting up a successful Splunk implementation team.

The fundamental best practices set expectations with stakeholders and ensure that your Splunk implementation stays on track and can grow and expand as your needs grow and expand. For more, see About the SSF fundamental best practices.

**Success Framework functional areas**

Best practices for implementing Splunk are organized into four functional areas:

**Platform**  
Platform management best practices support the availability, scalability, and maintainability of your Splunk deployment

**Program**  
Program management best practices support how you conduct your Splunk implementation to drive adoption and realize maximum value from your Splunk deployment

**Data**
Data lifecycle best practices support efficient data management practices and generate effective use cases that are tightly aligned to data.

People
User management best practices enable users and teams using learning incentives and role-based access to features and data.

Success Framework maturity levels

The Success Framework defines three maturity levels that apply to the functional best practices: standard, intermediate, and advanced. The maturity levels classify the best practices according to the level of effort needed to meet your priorities, needs and goals for each activity.

**Standard**
Best practices that establish the basis for an optimally performing Splunk environment.

**Intermediate**
Best practices that offer more control for results you can tailor to how you organize your Splunk implementation.

**Advanced**
Best practices that suggest configurations and optimizations to grow and expand your Splunk implementation.

Maturity levels can grow with you.

For example, when you start out, you may have a moderate sized team and a few core use cases on established systems. You might benefit from putting some standard and intermediate-level best practices in place. If it's a priority, you could also apply a few advanced best practices in areas where you already have strong practices in place.

For details, see About the Splunk Success Framework functional areas.

Success Framework terminology

The Splunk Success Framework uses the following terms:

**Splunk deployment**
A Splunk deployment refers to Splunk software that has been installed and configured on a system and is accessible to at least one user and
data source.

Splunk environment
A Splunk environment refers to the equipment that hosts your Splunk software. For on-prem Splunk Enterprise deployments, this is the hardware, virtual machines, and operating systems upon which your Splunk software is deployed. For Splunk Cloud deployments, this is the service hosted by Splunk.

Splunk implementation
A Splunk implementation refers to your Splunk deployment and Splunk environment (platform), the team of people that use and support Splunk software (people), the data and use cases you use Splunk software and solutions to address (data), and the processes your community of users follow to deploy, use, maintain, and grow an organization’s use of Splunk software and solutions (program).

Where is the Splunk Center of Excellence (CoE)?

Splunk Success Framework is the new name for the Splunk Center of Excellence (CoE)! The Splunk Success Framework emphasizes best practices you can apply flexibly at any stage of implementation to realize the full benefits of Splunk software. The SSF best practices support an existing CoE model if you have one, but do not require that you build a CoE to experience the benefits of best practices.

Release notes for the SSF Handbook

Updated August 20, 2019

What's new with this update

Overhauled change management best practices
Change management best practices for a Splunk deployment contains an overview of the change management (CM) process that governs changes to your Splunk environments. It contains guidelines and examples to reference as you define the change management scope, process, pathways, and identify the Responsibility assignment matrix (RACI) for your organization. It also includes a template to download and use for planning.
Earlier updates

April 10, 2019

Renamed service areas to 'functional areas'
Not all Splunk deployments are set up as services. The word functional means 'designed to be practical and useful,' which better describes the different categories of best practices.

Renamed foundation best practices to 'fundamental best practices'
Fundamental best practices address how you conduct your Splunk implementation. The word fundamental means 'forming a necessary base or core,' which better describes the value of these basic operational best practices.

Renamed the good, better, best maturity levels to 'standard, intermediate, advanced'
Maturity level is based on business objectives. Standard, intermediate, advanced keeps the focus on the level of effort to achieve the results best suited for the situation.

March 26, 2019

• Renamed Center of Excellence (CoE) to Splunk Success Framework
• Simplified the service area names
  ♦ Platform management and support is now platform
  ♦ Program management and value realization is now program
  ♦ Use case and data lifecycle is now data
  ♦ User and team lifecycle is now people
• Changed charter to purpose
• Removed legacy graphics
• Renamed in-depth articles "best practices"
Fundamental best practices

About the SSF fundamental best practices

The fundamental best practices of the Splunk Success Framework are decisions, agreements, and operating infrastructure that establish the purpose, goals, and ownership of your Splunk environment. Strong fundamental best practices ensure a better track record of success, and buy-in at all levels.

<table>
<thead>
<tr>
<th>Fundamental best practice</th>
<th>Purpose</th>
<th>Go to topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purpose and scope</td>
<td>Determine what you want to accomplish using Splunk software. Decide whether to use Splunk as a solution, service, or strategy.</td>
<td>Purpose and scope best practices</td>
</tr>
<tr>
<td>Executive sponsor engagement</td>
<td>Engage an executive or senior leader who will provide resources and endorse the success of your Splunk implementation at all levels.</td>
<td>Executive sponsor engagement best practices</td>
</tr>
<tr>
<td>Success measurements</td>
<td>Develop ways to measure the success of your Splunk implementation as it matures.</td>
<td>Success measurements best practices</td>
</tr>
<tr>
<td>Operations framework</td>
<td>Choose a centralized, federated, or hybrid operating model. Identify a program manager.</td>
<td>Operations framework best practices</td>
</tr>
</tbody>
</table>

Purpose and scope best practices for a Splunk deployment

Define a purpose and scope for your Splunk implementation to focus on what you want to use Splunk software and solutions to do. Who are the stakeholders? How do you want to use Splunk software and solutions to leverage your machine data? Defining a purpose and scope sets expectations among stakeholders, and helps prioritize the decisions you make and the actions you take.
Guidelines for defining a purpose

Use these guidelines to establish a purpose for your Splunk implementation.

Define clear objectives you want to useSplunk software and solutions to achieve.
Knowing these basic goals helps focus your efforts. Clear objectives also communicate clearly to executive sponsors and stakeholders the return on investment the organization will receive from Splunk software and solutions.

Investigate any constraints on your Splunk implementation.
Constraints can include things like access to data, adequate staffing, or user education. Knowing these constraints up front can help set expectations about how to resolve them.

Identify your stakeholders.
Stakeholders are any group who will benefit from the answers and insights Splunk software and solutions provide from your machine data. For guidelines about how to build a strong, comprehensive list of stakeholders, see Stakeholder best practices.

List the benefits of implementing Splunk software and solutions on your machine data.
Beyond meeting initial objectives, implementing Splunk software and solutions brings other benefits, such as optimizing your data infrastructure, getting smarter about your machine data, and the possibility of learning things you didn't even know to ask. Listing benefits beyond the initial objectives helps motivate your implementation team and inspire your stakeholders about the benefits of Splunk software and solutions.

Guidelines for defining a scope

Scope defines how you want to implement Splunk software and solutions in your data infrastructure.

Splunk as a solution.
Use Splunk software to address use cases for a single team, group, or purpose.

Splunk as a service.
Use Splunk software to provide Splunk-related services for multiple teams, groups, and purposes.
Splunk as a strategy
Use Splunk software to provide mature services that position Splunk as a competitive differentiator for your business.

Executive sponsor engagement best practices for a Splunk deployment
An executive sponsor makes sure the goals for your Splunk implementation are aligned with overall company strategy. They foster support (and overcomes resistance) from other stakeholders, and provide a backstop for accountability to ensure you have the resources you need for your Splunk implementation to be successful.

Best practices for ensuring effective executive sponsorship
Here are some best practices for ensuring that your Splunk implementation receives the attention it needs from your executive sponsor to be successful.

Identify an executive sponsor who is invested in the value of data
Ideally, your sponsor should be fully invested in the value of machine data and the potential it has to transform your business, such as informing strategic and competitive decisions, optimizing IT infrastructure, and securing your data infrastructure against cyber threats and malware.

Identify a sponsor whose objectives depend on the success of your Splunk implementation
An executive sponsor will be a more effective champion if their own objectives depend on the value Splunk software and solutions provide.

Understand what matters to your executive sponsor and key stakeholders
Research what information your executive sponsor and key stakeholders are likely to be most interested in. This will help the innovators on your Splunk implementation team know how to focus their experiments, measure results, validate the importance of their approach, and iterate as they go. By focusing efforts on what matters to stakeholders, you can clearly demonstrate the importance of your Splunk implementation when it comes time to scale, which validates your executive sponsor and helps garner additional support.

Involve your executive sponsor in Splunk implementation planning
Ensure that the executive sponsor is involved in project planning to set, clarify, and get aligned on objectives and expectations. Clarify the guidance, senior-level support, and resources you will need for a successful Splunk implementation. Ask your executive sponsor to participate in analyzing and sharpening the objectives of your Splunk implementation, establishing key milestones, and setting up a communication schedule.

Set expectations on incremental milestones to show how small wins lead to big ones

A fully optimized, well-tailored Splunk implementation is continually evolving as you perfect what works and iterate on what doesn't. Make sure that project milestones are focused on realistic, measurable goals that clearly factor into a longer-term overall strategy. Make these milestones visible, and communicate clearly and often when you achieve them.

Use Splunk dashboards, reports, and data-driven visuals to show visible progress

To help keep your executive sponsor motivated and involved, and to give them visible return on investment to share with other executive stakeholders, use Splunk dashboards and other visuals to demonstrate early wins. Focus on new insights, efficiencies, and progress on stated goals. This helps encourage a strong emotional commitment from your executive sponsor to do what is necessary for your Splunk implementation to succeed.

Establish accountability

The executive sponsor helps establish accountability for completing project milestones. You can also foster accountability from your executive sponsor by setting up regular meetings, reviews, and checkpoints.

**Success measurements best practices for a Splunk deployment**

Measurements provide statistics about key performance indicators (KPIs), and help you measure whether the best practices you adopt are having an effect on measurable goals. Good metrics provide transparency to you and your stakeholders.
Guidelines for measuring success

Develop a metrics dashboard
A metrics dashboard provides a way to show progress as you implement more best practices and optimize your Splunk implementation. A metrics dashboard enables you to quickly and proactively discover whether the best practices you put in place are having a measurable effect on the goals you set so you can adjust and pivot as needed.

Enable telemetry
Leverage the expertise of the Splunk Customer Success organization by enabling telemetry features on your Splunk deployment.

Operating framework best practices for a Splunk deployment

An operating framework provides structure for how you set up and manage your Splunk implementation.

Choose an operating model

You can organize your Splunk implementation a number of ways, depending on how you want to manage your resources. This topic outlines three possible approaches: centralized, federated, and a hybrid of both.

Federated model

With a federated model, teams operate their own independent Splunk deployments and projects. Program management provides best practice guidance and a forum for meetings to keep them coordinated. Each team can manage its own deployment architecture and operations.

You can store event data on separate indexers.

<table>
<thead>
<tr>
<th>Federated model advantages</th>
<th>Challenges</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Teams can manage their infrastructure independently under a common set of standards</td>
<td>• Requires more coordination from the program manager</td>
</tr>
</tbody>
</table>
### Federated model advantages
- Scaling is more modular, and easier to plan and execute
- Good for large organizations
- Isolates "noisy neighbors"
- Changes by one team have less risk of impacting other teams
- Flexible infrastructure deployment options

### Challenges
- More complex to set up and manage
- Managing deployment-wide search concurrency is more complex

---

**Centralized model**

A centralized model concentrates Splunk engineering (hardware and people) into a central team with a single Splunk deployment. A majority or the entirety of event data is stored on a common set of indexers, and users access a common search head or search head cluster.

<table>
<thead>
<tr>
<th>Centralized model advantages</th>
<th>Challenges</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Good for small deployments</td>
<td>• Requires effort to scale as more groups adopt Splunk and the number of use cases/number of users grows</td>
</tr>
<tr>
<td>• Data is easily accessible and shared</td>
<td>• &quot;Noisy neighbors&quot; can slow the system down</td>
</tr>
<tr>
<td>• Faster to get started and simple to set up</td>
<td></td>
</tr>
<tr>
<td>• Allows for quick growth</td>
<td></td>
</tr>
<tr>
<td>• Easier to manage deployment-wide search concurrency</td>
<td></td>
</tr>
<tr>
<td>• Requires the least hardware</td>
<td></td>
</tr>
</tbody>
</table>

**Hybrid model**

A hybrid model is a mix of both centralized and federated, where a critical mass of the Splunk activity is within a central team. Satellite deployments can exist outside of the central team. You can set up dedicated indexers and search head(s) for a use case or department, and the search heads may have the ability to search other deployments.

<table>
<thead>
<tr>
<th>Hybrid model advantages</th>
<th>Challenges</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Less complex to manage</td>
<td></td>
</tr>
<tr>
<td>Hybrid model advantages</td>
<td>Challenges</td>
</tr>
<tr>
<td>--------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>• A centralized team can still manage smaller groups or business units</td>
<td>• Requires coordination for federated resources</td>
</tr>
<tr>
<td>• A centralized operations team can provide Splunk as a Service</td>
<td>• The Operations team must be prepared to operate a large Splunk deployment</td>
</tr>
<tr>
<td>• Federated customer teams can meet scale demands</td>
<td>• Requires the most hardware</td>
</tr>
</tbody>
</table>

**Identify the program manager**

It is important to identify someone who is responsible to fulfill the program manager role, someone who has clear authority to manage operations for your entire Splunk implementation. The program manager role performs one of the most crucial functions on your team.

Program managers fulfill the following responsibilities:

- Drive decision-making
- Manage inter-dependencies between Success Framework pillars
- Ensure the Splunk implementation plan aligns with business objectives
- Oversee Splunk success measurements
- Is accountable for return on investment
- Promotes and facilitates program-wide communication
- Supports initiatives for knowledge sharing and collaboration
- Ensures executive alignment

For more information, see [Roles best practices](#).

**Post a service catalog**

If you are providing Splunk as a service, you can post a catalog of Splunk-related services and processes for your user community. A service catalog communicates to your community the services you offer, and indicates how they can engage with your team. Post your service catalog in a publicly accessible space, such as your team wiki, community, or internal web site.

**Define service level objectives and agreements**

Service level definitions include service-level objectives (SLOs), service-level agreements (SLAs), and case priorities. For more about creating service-level agreements, see [Service-level best practices](#).
Functional areas overview

About the SSF functional areas

The best practices in the Splunk Success Framework are organized into four functional areas:

<table>
<thead>
<tr>
<th>Functional area</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Platform</td>
<td>Platform management best practices support the availability, scalability, and maintainability of your Splunk deployment</td>
</tr>
<tr>
<td>Program</td>
<td>Program management best practices drive engagement that enables you to realize maximum value from your Splunk deployment</td>
</tr>
<tr>
<td>Data</td>
<td>Data lifecycle best practices generate well-designed and effective use cases that are tightly aligned to data</td>
</tr>
<tr>
<td>People</td>
<td>User management best practices empower users and teams using learning incentives and role-based access controls</td>
</tr>
</tbody>
</table>

The best practices in each functional area offers three implementation options, standard, intermediate, and advanced, to match your priorities, needs, and goals.

Platform best practices overview

The best practices in the platform functional area support the availability, scalability, and maintainability of your Splunk deployment. They help establish an optimized Splunk platform architecture and systems for continuity planning, capacity planning, and incident management.

Follow these best practices according to the standard, intermediate, or advanced goals you have set.

<table>
<thead>
<tr>
<th>Activities</th>
<th>Standard</th>
<th>Intermediate</th>
<th>Advanced</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDUCATION</td>
<td>Self education</td>
<td>Everything outlined in standard</td>
<td>Everything outlined in intermediate</td>
</tr>
<tr>
<td>How the Splunk engineering team</td>
<td>Leverage Splunk Documentation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Activities</td>
<td>Standard</td>
<td>Intermediate</td>
<td>Advanced</td>
</tr>
<tr>
<td>------------</td>
<td>----------</td>
<td>--------------</td>
<td>----------</td>
</tr>
<tr>
<td>(not users) stays current on how to administer Splunk software.</td>
<td>Leverage Splunk Answers</td>
<td>Splunk education paths by role</td>
<td>Defined attendance policy for Splunk events</td>
</tr>
<tr>
<td><strong>ARCHITECTURE</strong></td>
<td>Deploy software using recommended system requirements</td>
<td>Everything outlined in standard</td>
<td>Everything outlined in intermediate</td>
</tr>
<tr>
<td>Optimizations to your platform architecture that support performance and scale.</td>
<td>Set up a Splunk lab (see Lab environment best practices)</td>
<td>Make use of the Splunk Validated Architectures (see Splunk Validated Architectures)</td>
<td>Deploy a Universal Forwarder as part of the standard OS build (see The Universal Forwarder in the Splunk Forwarder Manual)</td>
</tr>
<tr>
<td><strong>CONTINUITY PLANNING</strong></td>
<td>Set up data replication (see Data replication in the Splunk Managing Indexers and Clusters of Indexers manual)</td>
<td>Everything outlined in standard</td>
<td>Everything outlined in intermediate</td>
</tr>
<tr>
<td>Product features or other solutions that facilitate high availability or disaster recovery scenarios.</td>
<td>Set up a backup policy</td>
<td>Set up search head clustering (see About search head clustering in the Splunk Distributed Search manual.)</td>
<td>Set up automated failover of the utility tier (see Disaster recovery best practices)</td>
</tr>
<tr>
<td></td>
<td>Set up system snapshots (see System snapshots) OR virtual</td>
<td>Set up multi-site data replication (see Multisite)</td>
<td>Implement source control for configuration and user knowledge</td>
</tr>
<tr>
<td>Activities</td>
<td>Standard</td>
<td>Intermediate</td>
<td>Advanced</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>--------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>migrations</td>
<td>indexer cluster deployment in the <em>Managing Indexers and Clusters of Indexers</em> manual</td>
<td>objects (see <a href="#">Back up and restore best practices</a>)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generate backups of configuration and user knowledge objects (see <a href="#">Back up and restore best practices</a>)</td>
<td></td>
</tr>
<tr>
<td>SUPPORT AND INCIDENT MANAGEMENT</td>
<td>Email or vocal request</td>
<td>Implement a ticketing system (see the Request workflow for Splunk app on Splunkbase)</td>
<td>Everything outlined in intermediate</td>
</tr>
<tr>
<td>Procedures to track and mitigate issues with the Splunk deployment.</td>
<td></td>
<td>24/7 live help desk</td>
<td>Splunk runbook</td>
</tr>
<tr>
<td>CAPACITY MANAGEMENT</td>
<td>Use the Splunk monitoring console (see Monitoring console in the <em>Monitoring Splunk Enterprise</em> manual)</td>
<td>Everything outlined in standard</td>
<td>Everything outlined in intermediate</td>
</tr>
<tr>
<td>Practices for staying informed about resource usage and staying ahead of demand on the Splunk platform.</td>
<td></td>
<td>Discuss anticipated needs with stakeholders (see <a href="#">Stakeholder best practices</a>)</td>
<td></td>
</tr>
</tbody>
</table>

**Program best practices overview**

The best practices in the program functional area include business alignment, operations, collaboration, use cases, and staffing, which enable you to realize maximum value from your Splunk deployment. A program manager generally drives these activities and manages interdependencies among the stakeholders.
Follow these best practices according to the standard, intermediate, or advanced goals you have set.

<table>
<thead>
<tr>
<th>Activities</th>
<th>Standard</th>
<th>Intermediate</th>
<th>Advanced</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHANGE CONTROL, MAINTENANCE, UPGRADES</td>
<td>Establish a regular schedule for releasing updated Splunk content and software updates</td>
<td>Everything outlined in standard</td>
<td>Implement a source control system (see Change management best practices)</td>
</tr>
<tr>
<td></td>
<td>Establish a communication plan (see Communication best practices)</td>
<td>Create a change control board</td>
<td>Establish an executive review board</td>
</tr>
<tr>
<td>COMMUNITY MANAGEMENT</td>
<td>Establish standards and policies</td>
<td>Everything outlined in intermediate</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Establish naming conventions for indexes, apps, and source types (see Naming convention best practices)</td>
<td>Enforce standards and policies</td>
<td>Use Splunk software to monitor for configurations and knowledge objects that do not meet standards</td>
</tr>
<tr>
<td></td>
<td>Use roles to separate access to data from capabilities (see Role-based data management best practices)</td>
<td>Publish standards and policies</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Establish a program that fosters user development (see User enablement best practices)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>COLLABORATION</td>
<td>Establish a program that fosters user development (see User enablement best practices)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Activities</td>
<td>Standard</td>
<td>Intermediate</td>
<td>Advanced</td>
</tr>
<tr>
<td>---------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Practices to share successes that inspire new ideas across different user communities and expose day-to-day wins to organization management.</strong></td>
<td>Establish an informal or ad hoc community forum</td>
<td>Everything outlined in standard</td>
<td>Everything outlined in intermediate</td>
</tr>
<tr>
<td></td>
<td>Participate in local Splunk user groups (see user groups)</td>
<td>Establish a central network location where users can save and access Splunk content</td>
<td>Establish a Splunk community portal (see Community portal best practices)</td>
</tr>
<tr>
<td></td>
<td>Attend local Splunk Live events (see Splunk Live)</td>
<td>Establish a stakeholder register (see Stakeholder best practices)</td>
<td>Attend the annual Splunk user conference (see .conf) and take education courses from the pre-conference training sessions, Splunk University (see Splunk University)</td>
</tr>
<tr>
<td></td>
<td>Follow other Splunk users on Splunk Answers (see answers.splunk.com)</td>
<td>Establish office hours for Splunk subject matter experts to consult with your Splunk user community</td>
<td>Publish a regular Splunk newsletter (see Newsletter best practices)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Host internal workshops</td>
</tr>
<tr>
<td><strong>STAFFING</strong></td>
<td>Establish and maintain a staff list with contact information</td>
<td>Everything outlined in standard</td>
<td>Everything outlined in intermediate</td>
</tr>
<tr>
<td><strong>Practices to maintain staff skills and responsibilities</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
and communicate regularly about responsibilities and assignments.

Establish Splunk roles and responsibilities on your team (see Roles best practices)

Establish a staffing model (see Staffing best practices)

Establish a RACI (responsible, accountable, consulted and informed) matrix as part of your staffing model (see Responsibility best practices)

**BUSINESS ALIGNMENT**

*Practices to ensure that your Splunk implementation stays on track with business goals*

**Standard**

*Align with stakeholders*  
Establish a stakeholder register (see Stakeholder best practices)

**Intermediate**

*Align with departments*  
Everything outlined in standard  
Hold regular stakeholder coordination meetings

**Advanced**

*Align with business units*  
Everything outlined in intermediate  
Hold a regular quarterly business review (see Quarterly business review best practices)

**VALUE REALIZATION**

*Practices to unlock, discover, and demonstrate the value of your Splunk investment.*

**Standard**

*Demonstrate value to users*  
Establish a use case registry

**Intermediate**

*Demonstrate value to departments*  
Everything outlined in standard  
Establish metrics (see Develop success measurements for your Splunk implementation)

**Advanced**

*Demonstrate value to business units*  
Everything outlined in intermediate  
Assess and optimize the input from data sources  
Establish dashboards that
Data best practices overview

The data lifecycle is the center point of a Splunk implementation. Before you even engage Splunk software, best practices can help you manage and structure your data efficiently to optimize its searchability and value.

Best practices in the data functional area also help you design effective use cases that are tightly aligned to data, so Splunk can answer questions you know about and reveal answers to questions you didn't know you had.

Follow these best practices according to the standard, intermediate, or advanced goals you have set.

<table>
<thead>
<tr>
<th>Activities</th>
<th>Standard</th>
<th>Intermediate</th>
<th>Advanced</th>
</tr>
</thead>
<tbody>
<tr>
<td>REQUEST DATA</td>
<td></td>
<td>Accept ad-hoc requests (for example, email, chat, voice).</td>
<td>Establish cost chargeback estimates for budget owner (see Showback best practices for Splunk implementations)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Utilize the Request Workflow for Splunk app (see Ticketing/workflow system on Splunkbase)</td>
<td></td>
</tr>
</tbody>
</table>

Publish Splunk use case success stories
Establish a log that measures the positive effect of Splunk use cases over time
Establish a showback plan (Showback best practices)
<table>
<thead>
<tr>
<th>Activities</th>
<th>Standard</th>
<th>Intermediate</th>
<th>Advanced</th>
</tr>
</thead>
<tbody>
<tr>
<td>requests.</td>
<td></td>
<td>Establish a process for defining technical data that includes the following:</td>
<td>Everything outlined in intermediate</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Defined source types (discover an existing add-on or create a new one)</td>
<td>Establish a process for defining value-oriented data</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Target index(es)</td>
<td>• Normalize fields with a common information model</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Data sensitivity searches, including personally identifiable information</td>
<td>• Define tags</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• End-user needs and outcomes</td>
<td>• Develop a corporate information model</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Field and value extractions</td>
<td>• Consider license and storage impact</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Generated knowledge objects, dashboards, and alerts</td>
<td>• Identify business priority</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Reuse data and knowledge objects for other use cases</td>
</tr>
<tr>
<td>DEFINING THE DATA</td>
<td></td>
<td>Establish a process for defining baseline data that uses learned source</td>
<td>Everything outlined in intermediate</td>
</tr>
<tr>
<td>Guidelines to determine where to</td>
<td></td>
<td>types and little or no source data optimization</td>
<td>Establish initial use case requirements</td>
</tr>
<tr>
<td>place line breaks and timestamps</td>
<td></td>
<td></td>
<td>Everything outlined in intermediate</td>
</tr>
<tr>
<td>on incoming data and to identify</td>
<td></td>
<td></td>
<td>Create a lab environment for developing system and test</td>
</tr>
<tr>
<td>the intended use case or value of</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>the data.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IMPLEMENT THE USE CASE</td>
<td></td>
<td>Deploy technical add-ons that support getting data in</td>
<td></td>
</tr>
<tr>
<td>Processes to carry out the Splunk</td>
<td></td>
<td>Everything outlined in standard</td>
<td></td>
</tr>
<tr>
<td>configuration and to engage the</td>
<td></td>
<td>Establish initial use case requirements</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Create a deployment server class that includes search, index, and forwarding and data collection tiers. Apply naming conventions to knowledge objects (see Naming convention best practices). Utilize the app builder for custom components.

**VALIDATE THE USE CASE**

*Processes to verify that the use case meets the requester's expectations and needs, and to enable the requester to communicate feedback.*

- **Validate reactively:** requester validates the work after it is completed.
- **Validate proactively:** requester validates the work at regular intervals during development.
- **Validate demonstratively:** requester validates the work in real-time from a hands-on demonstration of the use case (knowledge objects, data, and so on).

**STEP 5: COMMUNICATE USE CASE CHANGES**

*Structures to inform the requester that the related work is completed and to enable others to learn about it.*

- Communicate with individuals that the work is completed.
- Everything outlined in standard.
- Announce to the community that the work is completed.
- Everything outlined in intermediate.
- Share final showback calculations with the requester.
- Track and communicate the business value of use cases to executive stakeholders.

**MAINTAIN AND RETIRE USE CASES**

- Monitor the ongoing need of use cases.
- Everything outlined in standard.
- Everything outlined in intermediate.
For more about data onboarding best practices, see Data onboarding best practices.

People best practices overview

The best practices in the people functional area focus on learning incentives and role-based access to features and data to empower users to get the most out of Splunk software. The people functional area ensures that everyone who uses Splunk has an education plan, and that they earn additional access and capabilities when they advance their knowledge and experience.

People best practices also ensure that each team has a safe workspace where they can experiment with new ideas and collaborate.

Follow these best practices according to the standard, intermediate, or advanced goals you have set.
<table>
<thead>
<tr>
<th>AUTHORIZATION REQUEST</th>
<th>Accept ad-hoc requests (email, chat, voice)</th>
<th>Utilize the Request Workflow for Splunk app (see Ticketing/workflow system on Splunkbase)</th>
<th>Establish self-service automation that includes universal access</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFINE ROLES AND CAPABILITIES</td>
<td>Utilize default roles</td>
<td>Develop custom roles that inherit capabilities hierarchically</td>
<td>Everything outlined in standard</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Develop a team workspace app as the default app (see Workspace best practices)</td>
<td>Use roles to separate access to data from capabilities (see Role-based data management best practices)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Develop a welcome page to help users get started (see Welcome page best practices)</td>
<td></td>
</tr>
<tr>
<td>AUTHORIZATION IMPLEMENTATION</td>
<td>Utilize native Splunk Enterprise authentication</td>
<td>Leverage an external directory system such as LDAP or SSO</td>
<td>Use only an external directory system</td>
</tr>
<tr>
<td>AUTHORIZATION VALIDATION</td>
<td>Requester validates completed work</td>
<td>Requester validates work in progress</td>
<td>Requester validates work in real-time</td>
</tr>
<tr>
<td>USER AND TEAM MAINTENANCE AND</td>
<td>Utilize an IT-defined</td>
<td>Everything outlined in standard</td>
<td>Same as intermediate</td>
</tr>
</tbody>
</table>
Change management best practices for a Splunk deployment

Change management (CM) is the end-to-end process that governs the life-cycle of a change from the initial request for change to the final deployment and communication of the change. As organizations consider which changes in Splunk warrant CM oversight, CM guidelines help organizations define the appropriate process steps to include, activities to complete, and define the participants in the CM process and their responsibilities. Since every organization is unique, how they decide to implement CM can vary. CM guidelines give organizations a best practices framework and tools to help them define and develop the CM process(es) that fit their unique needs and requirements. Using CM to govern changes in your Splunk environment has many benefits such as:

- Establishes a documented, defined process to authorize and deploy changes in your Splunk environment
- Provides awareness and a common and visible method to address changes
- Significantly improves the ability to respond to issues by providing a record of change to support platform health and troubleshooting

Audience

- Executive sponsor
- Program manager
- Project manager

Differentiate between change management and change control

CM encompasses the entire change life cycle. Change control is the execution of a specific subset of process steps with a core focus of ensuring releases or deployments do not conflict with other production components. Critical and time sensitive platform changes, such as a security patch, are often subjects that path directly to change control. As such, change control has a level of risk that is typically higher and often acceptable.
Change management framework

The CM framework is a set of components to help design and implement CM. This framework design provides both structure and flexibility with the goal to allow you to build a new CM process or to incorporate into an existing CM process. However, with flexibility comes the opportunity for an endless cycle of variations. Consider a moderate or limited approach that can quickly evolve through usage and experience. The CM framework includes guidelines to the define:

- CM scope
- CM process
- Responsibility assignment matrix (RACI)
- CM pathways
- Change guidance examples

Change management scope

Since a CM process deals with the management of changes, you must have a common understanding of which changes are in the scope of the process. This section describes a sample scope and includes areas which are both within and outside of the CM process scope. This sample is provided to spur thought so that customers can engage in conversations to define what scope is best for your organization.

In scope

The scope of the CM process covers all production systems and platforms of an organization. The primary functional components covered in the CM process may be similar to the following examples:

- **Software development life cycle**: Changes handled through the formal software development life cycle
- **Hardware**: Installation, modification, removal or relocation of computing equipment.
- **Software**: Installation, patching, upgrade or removal of software products including operating systems, access methods, commercial off-the-shelf (COTS) packages, internally developed packages and utilities.
- **Database**: Changes to databases or files such as additions, reorganizations and major maintenance.
- **Application**: Application changes being promoted to production and the integration of new application systems and the removal of obsolete elements.
• **Moves, adds, changes and deletes**: Changes to system configuration.
• **Schedule Changes**: Requests for creation, deletion, or revision to job schedules, back-up schedules or other regularly scheduled jobs managed by the company's IT organization.

**Out of scope**

Many activities are performed on a routine basis as part of the operations and administration of your Splunk environment that are outside the scope and governance of the CM process. It's a best practice to identify and document these activities as "out of scope" for CM to maintain operational efficiency. Examples include the following:

- Contingency or disaster recovery
- Changes to non-production elements or resources such as changes in Dev or Test environments
- Changes under the governance of large-scale projects or initiatives have a separate CM process
- Changes made within the daily administrative process.

Examples of daily administrative tasks are:

- Password changes or resets
- User adds or deletes
- Adding, deleting, or revising security groups
- Rebooting machines when there is no change to the configuration of the system
- File permission changes

Consider defining a catch-all change pathway or a standard operating procedure where submissions for change are channeled when the requester is unsure if the change is in or out of scope.

**Change management process**

CM is a sequence of process steps with associated activities that one follows to govern a change to drive and ensure an intended outcome. The primary goal of the change management process is to ensure standardized methods, processes, and procedures are used to facilitate efficient and prompt handling of all changes. The CM process is grouped into two distinct phases, the approval phase and the execution phase. The approval phase consists of the process steps that support the decision-making and authorization activities for the change. The execution phase addresses the process steps that are related to the
actual work needed to build and deploy the change.

As you go through the exercise to define the process steps and associated activities to include in your change management process, we suggest you include every process step and the associated activities for the given process step that you may use. Use the change pathways to select which process steps and which activities to implement for a given change scenario. Think of the CM process as the checklist for defining each change pathway.

**Approval phase**

The approval phase has steps and activities to support the decision-making and authorization activities for the change. The activities in the table below are examples to reference and use to build and evolve to fit the needs of the organization.

<table>
<thead>
<tr>
<th>Steps</th>
<th>Activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identify change</td>
<td>Identify the reason for a change such as correction, new, or modification in requirements</td>
</tr>
</tbody>
</table>
| Qualify change   | • Assess scope of change such as new or add, change or replace, retire or remove  
|                  | • Identify change impact such as audience, use-case, platform, cost, and so on  
|                  | • Determine next process step                                               |
| Change request   | Documents and describes the change based on predefined criteria              |
| Triage change    | • Validate the completeness of the request information, return to the submitter if it's not sufficient.  
|                  | • Determine appropriate resources required to analyze the request           
|                  | • Perform quick assessment of the potential impacts to the overall Splunk Platform/Service |
| Analyze request  | • For analysis assign the change request to an authorized member             
|                  | • Deferred change re-enter this analysis step                               
|                  | • At this stage, rejected change request should stopped                     |
### Steps Activities

| Request approval | • Identify change risk and complexity level before approval  
|                  | • Identify the impact level of the change before approval  
|                  | • Review impact of change request to authorized person for approval  
|                  | • At this stage, rejected change request should stopped  
| Request implementation | • Initiate execution phase  
|                      | • Monitor progress of change request  
| Request close-out | • Update project procedure and management plans  
|                    | • Communicate change  
|                     | • Record the completion of change request  
|                     | • Close change request  

**Execution phase**

The execution phase has steps and activities that are relate to the actual work needed to build and deploy the change. The activities in the table below are examples to reference and use to build and evolve to fit the needs of the organization.

<table>
<thead>
<tr>
<th>Steps</th>
<th>Activities</th>
</tr>
</thead>
</table>
| Planning | • Identify the necessary skillsets and resources for the change request  
|         | • Identify and estimate the effort required to complete the change request  
|         | • Identify schedule and resources available to complete the change request  
|         | • Identify and estimate the cost of the change  
| Analysis | Gather, validate, and document both business requirements and technical requirements  
| Design | Document specifications, features and operations to satisfy the functional requirements of the change  
<p>| Implement |</p>
<table>
<thead>
<tr>
<th>Steps</th>
<th>Activities</th>
</tr>
</thead>
</table>
|       | • Change build-out  
|       | • Update SOP |
| Test  | • Unit testing  
|       | • Integration testing  
|       | • System testing  
|       | • User-acceptance testing |
| Deploy| • Complete promotion process from dev/test to production environment.  
|       | • Distribute SOP and other impacted documentation updates to production areas for general consumption |
| Close-out| • Monitor change to validate change success  
|         | • Communicate closure to appropriate parties  
|         | • Complete and update all documentation for historical preservation  
|         | • Backup all software components involved in change for historical preservation |

**Change management RACI matrix**

The RACI matrix defines the participation requirements for various roles to complete tasks or deliverables in a CM pathway. RACI is an acronym derived from the four key responsibilities most typically used: Responsible, Accountable, Consulted, and Informed.

- **Responsible** - Those who do the work to complete the task. There is at least one role with a participation type of responsible, although others can be delegated to assist in the work required.
- **Accountable** - The one ultimately answerable for the outcome. The role who ensures the prerequisites of the task are met and who delegates the work to those responsible. In other words, an accountable must sign off to approve the work that responsible provides. There must be only one accountable role specified for each task or deliverable.
- **Consulted** - Those whose opinions are sought. Typically, this applies to subject matter experts and with whom there is two-way communication.
- **Informed** - Those who are kept up to date. People in this role are informed of the latest progress. Often only on completion of the task or deliverable; and with whom there is just one-way communication.
There is a distinction between a role and individually identified people. A role is a descriptor of an associated set of tasks that can be performed by many people and one person can perform many roles. For example, an organization may have ten people who can perform the role of system administrator and one person who can perform the role of system administrator may also be able to perform the role of developer. It's a best practice to assign only one participation type for each task or deliverable per role in the CM pathway. Where more than one participation type is shown, generally implies that participation is fully resolved, which can impede the value of this technique in clarifying the participation of each role on each task. See Role-based data management best practices for a Splunk deployment.

**Change management pathways**

A set of defined change pathways is the outcome of the CM framework and is used to facilitate the CM process. Since not all changes are alike, having multiple defined change pathways provides the flexibility needed to best accommodate changes. The change pathway identifies the steps and associated activities from the CM process and the who-does-what-when from the CM RACI to properly govern a change request. For example, changing a source type definition can have a significant impact on the ingest of data and can impact many business processes and need analysis and approval from multiple parties. To contrast, a change to a dashboard panel may not require the same level of analysis nor approval. Change pathways provide the flexibility to address various scenarios of change, improve overall efficiencies, and avoid the one-size-fits-all approval pitfall. It's a best practice to define several CM pathways to accommodate change categories.

It's important to keep the quantity of change pathways to manageable number. It's better to start with a smaller set and expand as you learn more about how your organization addresses change. It's a best practice to define default pathways as catch-alls to use for multiple scenarios.

Use the following items to define your change pathways:

- Change guidance examples to define a set of change categories to develop your change pathways.
- CM process to identify the process steps to enact against each change grouping
- CM process to select the appropriate activities within each process step to complete for each change category
- CM RACI to define the roles and select the responsibilities for the people participants for each change category
• Use the identify deployment requirements for example Dev-Test-Prod Promotion, Segregation of Duties, and so on

**Identify deployment requirements**

Organizations do have special situations, or pathway variation considerations, that they need to address from time to time. In keeping with a flexible framework, it's a best practice to define a "variations" option for the effected pathway as opposed to creating specific pathways for each variation. This allows more flexibility over time as new conditions or constraints are identified. In environments that support a formal development life cycle, it is typical to see both promotion process (manual or automated) and requirements for segregation of duties (coders, testers, promoters are all different people). In these cases, you must consider deployment priorities and constraints to implement a change. Deployment priority levels may be similar to the following examples:

- **Emergency**: Significant risk if not implemented immediately, such as applying a security patch
- **High**: Implement soon to prevent significant negative impact to ability to conduct business
- **Routine**: Implement to gain benefit from the changed service
- **Low**: Not pressing, but implement to provide a positive outcome

There may be other events or circumstance that are known until they appear. You can use these variant options to create one-off or edge case events.

**Change guidance examples**

Download the change management process template and use it as a tool to plan and scope the full impact of changes to define your change management process for Splunk changes. This helps gain insight about how a change impacts people, services offerings, and businesses processes in the organization. Think about the reason for the change to identify the actions to complete in Splunk to support the change and consider the following examples to fully understand the impact and create a plan for the change:

- **Downstream impacts**: Potential outcomes to understand and evaluate
- **Change consideration**: Lists optional items such as governance documents, training, or testing to support the change
- **Risk score**: Use a number from 1 to 5, that corresponds to the level of difficulty, complexity, or significance of the change
- **Scenario category**: Type of change
• **Scope of impact**: Lists of rating of High, Medium, or Low to estimate the impact to the service, platform, audience, business process, and change difficulty
  
  - **Service**: Domain such as your IT Ops, SOC, NOC
  - **Platform**: Technology used in a domain such as Splunk
  - **Audience**: Impact to the community such as executives, external customer, or partner
  - **Business process**: Scope of a specific business process such as point of sales, security vulnerability, and so on
  - **Change difficulty**: Complexity, level of effort, and skill set

---

**Communication best practices for a Splunk deployment**

A communication plan outlines events and activities your user community should know about, and how you communicate those events. You can use the worksheet here to define what to communicate to your Splunk community and how.

**Audience**

- Executive sponsor
- Program manager
- Project manager
- User community

For more about these roles, see [Roles best practices](#).

**Guidelines for setting up a communication plan**

A communication plan is an essential component of a comprehensive Splunk program management plan. For more information about program management, see [Program best practices overview](#).

**Communication options**

Here are some ways you can communicate updates to your Splunk community:

- Splunk banner message
- Email
- Phone call
• SMS alert
• Dashboard panel
• Ticketing system
• Portal (wiki)
• Newsletter

**Communication plan template**

This list is an example of what kind of activities warrant communication to your Splunk community. Use these as a guideline, add and customize according to your own Splunk workflow.

<table>
<thead>
<tr>
<th>Event</th>
<th>Communication method</th>
<th>Communication audience</th>
<th>Communication frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>New use case</td>
<td>See Communication options above</td>
<td>N/A</td>
<td>After implementation</td>
</tr>
<tr>
<td>Scheduled maintenance</td>
<td>Email</td>
<td>All user communities</td>
<td>1 week prior to maintenance</td>
</tr>
<tr>
<td>In-progress maintenance</td>
<td>Email</td>
<td>All user communities</td>
<td>When maintenance begins</td>
</tr>
<tr>
<td>Completed maintenance</td>
<td>Email</td>
<td>All user communities</td>
<td>Upon completion</td>
</tr>
<tr>
<td>Outage/ongoing issue</td>
<td>Email</td>
<td>All user communities</td>
<td>Within 1 hour of event awareness</td>
</tr>
<tr>
<td>Periodic update</td>
<td>See Communication options below</td>
<td>N/A</td>
<td>Quarterly</td>
</tr>
<tr>
<td>Workshop</td>
<td>Meeting invites</td>
<td>All user communities</td>
<td>1 month prior to the workshop</td>
</tr>
<tr>
<td>User/team-created request</td>
<td>Email</td>
<td>Requester</td>
<td>After implementation</td>
</tr>
<tr>
<td>User/team-modified request</td>
<td>Email</td>
<td>Requester</td>
<td>1 week prior to starting the user/team modification</td>
</tr>
<tr>
<td>User/team-removed request</td>
<td>Email</td>
<td>Requester</td>
<td>1 week prior to removing the</td>
</tr>
</tbody>
</table>
### Event Communication

<table>
<thead>
<tr>
<th>Event</th>
<th>Communication method</th>
<th>Communication audience</th>
<th>Communication frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use case in progress</td>
<td>Email</td>
<td>Requester</td>
<td>When implementation begins</td>
</tr>
<tr>
<td>Use case implemented</td>
<td>Email</td>
<td>Requester</td>
<td>Upon completion</td>
</tr>
<tr>
<td>Use case removed</td>
<td>Email</td>
<td>Requester</td>
<td>1 week prior to removing the use case</td>
</tr>
</tbody>
</table>

**Communication plan next steps**

Post your communication plan for your community to see. Use your community portal, team wiki, or preferred platform.

### Community portal best practices for a Splunk deployment

A Splunk community portal is a great way to keep your community engaged and informed. A community portal serves several purposes:

- Exchange ideas and offer support
- Identify common goals and interests
- Build a thriving and growing Splunk community
- Provide space for open collaboration

**Audience**

- Knowledge manager
- Program manager
- Project manager
- User community

For more about these roles, see Roles best practices.
Guidelines for establishing a community portal

A community portal is a shared resource where users can get an overview of activities the Splunk community is engaged in, and resources they need to be successful with Splunk. There is no prescribed layout for a community portal. Design the community portal in the way that works best for your users on a platform they use regularly. Your community portal can feature Splunk resources, updates, announcements, key processes, user questions and answers, and contact information for submitting requests or feedback.

Start by identifying the needs of your users, then design and create a draft of your portal:

Identify the needs of your users

Who are my stakeholders?
Identify the Splunk user roles in your environment. Refer to the Staffing best practices and Roles best practices.

What kind of content do they need to see?
Consider what they need to know. For example, an executive sponsor and program manager may want to see a program status, and an expert user or knowledge manager may want to see discussions about how to model data from certain data sources.

What collaboration medium or application is best?
Platforms used by thriving Splunk communities include Jive Interactive Intranet, Microsoft Sharepoint, and Confluence. Some organizations start out with a central file repository before choosing a platform.

Splunk resources section

Your community portal should include a Splunk resources section with user roles and relevant content for each user type. The example below shows a matrix of relevant content sorted by user role.
Recommended actions

1. Once your portal is designed, try it out on some users to make sure your portal is meeting their needs.
2. Assign a team to add fresh content and monitor any questions posted to the forum.
3. Create a maintenance schedule for your community portal.
4. For inspiration, see the Splunk Community.

Back up and restore best practices for a Splunk deployment

A best practice for any software implementation is establishing regular backups. Identify backup and restore points, and make regular backups of your Splunk configuration files to ensure system continuity in case of a failure, outage, or mistake.

Audience

- Engineer
- Architect

For more about these roles, see Roles best practices.

Guidelines for establishing a Splunk backup policy

The $SPLUNK_HOME/etc/ directory and its subdirectories contain all the settings for
your Splunk installation and all apps, including saved searches, user accounts, tags, custom source type names, and other configuration information. Follow these guidelines to establish a Splunk backup policy.

Back up to a separate location
   Backing up to a separate location is a best practice for resiliency. Backing up to a different disk or mount than Splunk is installed on can reduce single points of failure.

Back up single points of failure
   Back up any single points of failure, such as a single indexer, single instance deployment, and single search head, and utility tier resources, such as the deployment server, deployer, master node, license server.

Back up at least one search head cluster (SHC) member periodically
   As a best practice, periodically back up the SHC state to ensure you can restore knowledge objects in their current state in case of a catastrophic failure. For details about what to back up on the SHC and how, see Back up and restore search head cluster settings in the *Splunk Enterprise Distributed Search* manual.

Implement some form of version control
   Having some way of preserving state between versions is a best practice so you can roll back in case of an error.
   **Standard**: Scripted input that kicks off a specific diagnostic (or just an $SPLUNK_HOME/etc/ directory) and cleans old copies to prevent filling up the file system. See Generate a diagnostic file in the *Splunk Troubleshooting* manual.
   **Intermediate**: Scripted input that checks into a source control system, such as Git.
   **Advanced**: A custom solution using source control that provides managed restoration.

**More resources**

- For your Unix environment, see the app Stateful Snapshot for Splunk on Splunkbase.
- For more about configuration files, see Backup configuration information in the *Splunk Enterprise Admin Manual*. 
Data onboarding best practices for a Splunk deployment

There are already many resources about how ingest data into Splunk. The guidelines in this article complement existing documentation by mapping the data onboarding process into five phases. These guidelines help you streamline data requests, define the use case, validate data, and properly communicate the availability of new data.

Audience

- Developer
- Knowledge manager
- Program manager
- Project manager

For more about these roles, see Roles best practices.

Guidelines for establishing a data onboarding workflow

The recommended data onboarding workflow consists of five steps: request data, define the use case, implement, validate, and communicate.

During the process, document your approach so your community is well informed. This can deflect many questions, establish user expectations, make your users more aware of their responsibilities, and teach your users how they can make effective contributions to the data onboarding process.

Step One: Request data

The data onboarding workflow begins with a request to add data. You can keep it as simple as an email, or you can use a format, such as the User and Data Request System app on Splunkbase (see the User and Data Request System app on Splunkbase), or you can even leverage an enterprise change control system.

Simplify your data requests
- Capture only the essentials. Avoid using Splunk-specific terms, such as index name, field extractions, and so on.

Ask for specific known information
Ask for concrete details, such as data source host names and IP addresses, path, location, and access information; retention requirements (how long they need to keep the data); and a brief description of what the data represents. This will help you prioritize the request and define source types.

Estimate data volume
The requester may know their estimated data volume, but it may be more efficient for you to review the source location and do the math yourself. A near-maximum of the data volume (such as 95th percentile) works well with the Splunk licensing model. Do not take the average or median data volume, since the actual data volume will exceed that threshold 50% of the time. For help estimating data volume, download the Data Volume Calculator from Splunkbase.

**Step Two: Define the data**

Hold a data definition meeting to clarify details of the request. Be thorough during this stage to reduce the chance of miscommunication or misunderstanding and to help the implementation phase go more smoothly.

Get a sample of the data prior to the data definition meeting
Review the sample data up front to verify the following:

- Splunk's ability to access the data
- Permissions to the data within Splunk
- Forwarders (if needed)
- Dependency on a modular input (if needed)
- Any data retention and storage considerations

Verify the requester's commitment
If the requester is enthused and prioritizes this meeting with you, you’ll know this request is truly important to them and the fruit of your labors will be enjoyed. If the requester does not make this meeting a priority, it may indicate they are not as invested in this use case as they could be.

Define the use case with the requester
Validate Splunk-relevant details about the information, such as event breaks, timestamps, and other critical source-type elements. Discussing the use case with the requester enables you to uncover searches or dashboards that will be immediately useful to them. The scope should be to assist the requester with their initial search and dashboard setup to get them going, not a commitment to own their use case.
Empower the requester to own the use case
   Make sure the requester has completed the appropriate education path to enable them to own their use case. The requester should be responsible to own further search-time activities. For more information about how to establish education paths, see User and Team Lifecycle.

**Step Three: Implement the use case**

Once the data is defined, proceed with technical implementation.

Build out search and reporting artifacts
   Use the information gathered in the define data step. Focus on value-add elements that only you can uniquely provide, such as tags, reports, saved searches, dashboards, forms, field extractions, and any other elements you have uncovered or nice-to-haves submitted by the requester.

Ask for clarifications as needed
   Ask the requester if you need more information about the data, details or objectives of the use case during implementation.

**Step Four: Validate**

After developing the use case artifacts, validate that they achieve the expected results.

Run through the use case in your lab
   Run the artifacts you created through testing in your own lab using sample data relevant to the use case.

Invite the requester to validate the use case
   Have the requester review the results you generated from your tests to make sure the use case meets the requester’s expectations. Make any adjustments needed.

**Step Five: Communicate**

This phase ensures that each diverse data point added to an analytic (or KPI) directly contributes to business value.

Send an announcement about the availability of the new data
   Communicate with the wider user community that the use case is available. This enables other users to consider how these data points might help them.
Help the community understand current and potential use case(s) for the data. In your announcement, suggest some creative applications of the data. Provide use case information that will help the community understand how this data can support stronger, data-driven decisions.

Include details in the announcement
Include details in your announcement, such as how to access the data (index, sourcetype, tag name), what the data represents (use information from the data request and/or data definition meeting, what knowledge objects exist for it already (fields, dashboards, saved searches, and so on).

More resources

- To learn about getting data into Splunk Enterprise, see What data can I index? in the Splunk Enterprise Getting Data In manual.
- For an introduction to data onboarding best practices, see the .conf2017 session Data Onboarding: Where do I begin?

User enablement best practices for a Splunk deployment

User enablement is about motivating your users to learn and grow. When you provide an environment of incentive-driven access, you encourage users to explore and implement best practices, which adds value to the whole user community.

Audience

- Search expert
- Program manager
- Project manager
- User community

For more about these roles, see Roles best practices.

Guidelines for enabling users

To maximize user enablement, focus on these main principles:

- Incentive-driven user access
• User experience
• User roles and capabilities

Don't be a "data butler"
Users often want to skip the required education. If you are providing users with access and the information they need, this leaves them with little incentive to expand their own knowledge. Make sure users are motivated to learn best practices. This means limiting their access until they've completed certification and education.

Encourage customer-facing groups to explore value-add activities
Empowering users become capable of manipulating the data themselves leads to richer, dynamic insights that enables users to answer their own questions and make data-driven decisions.

Require an education path prior to granting access
Users can own and drive their own basic searching when they have established education paths and certification requirements. For recommended education requirements, see Roles best practices.

Grant capabilities to advanced users only
Grant capabilities only to users who qualify with your certification or education requirements. This means limited or no access until users have completed certification. Limiting access ensures that users are empowered to learn and implement their own best practices.

Guidelines for managing user experience

Give each team their own app
Create an app for each team and set this as the default in the navigation. Use the app as the team's dedicated workspace.

Create a welcome page for each team
Set up a welcome page for each team to improve the user onboarding experience and facilitate easy access to the resources they need. To create welcome pages, download the Welcome Page Creator from Splunkbase. For more information, see Welcome page best practices.

Hide all other apps
Remove read permissions for apps the user does not need, and to apps the user is not certified to use. As a general best practice, ensure that users are not distracted by other items deployed to the Splunk environment.
**Guidelines for managing user roles and capabilities**

Split roles and capabilities
Create roles based on data access and roles based on capabilities. This enables you to customize user access many ways without needing to create new roles. For more information about separating roles and access to capabilities, see Role-based data management best practices.

Limit permissions
Consider limiting permissions for features such as acceleration, scheduled searches, and real-time searches. If necessary, use search limits. Limiting permissions will optimize your search capacity. When granting capabilities, consider whether the feature you are granting access to could impact Splunk performance in a recurring way, such as scheduled searches, report acceleration, or searches that exceed the Splunk timeout limit.

**Lab environment best practices for a Splunk deployment**

An essential enterprise software best practice is to set up a lab environment that simulates your production environment so you can test product features.

A lab is different from setting up an individual sandbox, which is intended to be a lightweight, high-volatility environment for individual development and experimentation. A lab is similar to your production environment, and should be stable and persistent, so you can effectively test and develop product features before implementing them in your production environment.

**Audience**

- Architect
- Developer
- Engineer
- Search expert
- Knowledge manager

For more about these roles, see Roles best practices.
Guidelines for establishing a lab

A lab environment should mirror your production topology to the extent possible, in functionality, if not in scale. You can host your lab a number of ways:

On-prem lab
You can host your lab on premises using dedicated hardware. This hardware should be managed and maintained under the same policy as your production environment to ensure stability, security, and up-time.

Cloud or VM lab
You can use a public cloud (such as Amazon Web Services?AWS) or one that is private to your company. Consider any concerns your organization may have about exfiltration, or unauthorized transfer of personal or company data to a cloud-based resource.
You can also set up a virtual machine on your local instance. Oracle VM VirtualBox is a free and open-source option. Be aware that installing a virtual machine can be involved, especially if you want to replicate a distributed environment.

Access control
Access control in your lab environment should mirror your production environment. See About configuring role-based user access in Securing Splunk Enterprise for details. See Role-based data management best practices for tips on how to fine tune your user access model.

Getting data into your lab
A lab is where you test product features and custom solutions on production-simulated data before pushing them to production, so ideally, you want the data in your lab to mirror production as closely as possible.

When you first set up your lab, you can start with sample data from Splunk. For sample data, see Upload the tutorial data in the Splunk Search Tutorial. You can also use the _internal index, which has logs about your own instance. Once you have your lab set up, however, use one of the other methods listed below so you can test your product features on data that closer simulates reality.

Export data
If you don't want to connect directly to a production indexer, you can use the export feature in Splunk to get a sample of raw events. Export data as raw events so you can experiment with defining the source type. You can
also use a semi-structured format, such as JSON or XML. For information about exporting data, see Export data using Splunk Web in the Splunk Search Manual.

When you export data from the production environment, be aware of the amount of data you are requesting. If the data set is too large, Splunk could become unresponsive while transmitting the export.

Production data
You can point to existing indexers, including those in your production environment. It is possible to have a many-to-many relationship between search heads and indexers (or index clusters). For information about connecting to production non-clustered indexers, see Add search peers to the search head in the Distributed Search Manual. For information about connecting to production indexer clusters, see Enable the search head in Managing Indexers and Clusters of Indexers. Before proceeding with this option, consider the impact this will have on the environment.

If you are using your lab for an indexer-heavy workload, connecting to production indexers may not be the best solution.

License management
Configure a license master and set up a license slave for your lab. This will ensure that your lab is stable and persistent, and has all the functionality available to it that is available in production. See Configure a license slave in the Splunk Enterprise Admin Manual.

Logging best practices for a Splunk deployment
Splunk does not need or require a logging standard. Splunk identifies an event using a few default fields from the incoming event’s raw data, then identifies and correlates common elements with other events on the fly at search time. That means there is no fixed schema, which makes searching with Splunk fast, easy, and flexible.

However, you can optimize how data is formed at the source so that Splunk can parse event fields easier, faster, and more accurately when the events do arrive.

The article Logging best practices in the Splunk developer forum provides specific guidelines for how to make the most of your log files. This article
provides additional guidelines and considerations.

**Audience**

- Architect
- Developer
- Search expert
- Knowledge manager

For more about these roles, see [Roles best practices](#).

**Guidelines for logging best practices**

"If you can read it, you can Splunk it." - the Splunk Best Practices Team

If the meaning is not codified in the log events, then it needs to be added, either when the log is created (optimize log files at the source), or on the fly (using fields in Splunk).

Semantic logging is writing event logs explicitly for gathering analytics that will be consumed and processed by software. Logs are generally written by developers to help them debug, or to form an audit trail, so they are often cryptic or lack the detail needed for data analysis.

Here are some guidelines for optimizing logs at the source.

**Only optimize event logs if it is practical**

Optimizing your logs at the source is not necessary or required, but it can streamline your Splunk experience. Optimizing event logs makes the most sense for systems in active use whose source code is easily accessible. It may not be practical to try rewriting the logs for a legacy application whose source code is no longer available. A better approach in that case is to use Splunk knowledge objects to add meaning to existing log information.

**Capture data from a variety of sources**

Think of a business scenario you might want to analyze. Think about what elements you would want to visualize, and consider what data might be needed to help answer basic questions about that scenario. For example:

- Graph transaction volume by hour, by day, by month
- How long are transactions taking during different times of the day and different days of the week?
- Are transactions taking longer than they did last month?
- What volume of transactions come from which geographical regions?
- How many transactions are failing? Graph these failures over time.
- Which specific transactions are failing?

To begin answering these questions, consider all the systems involved in a business transaction workflow. The more varied your sources are, the better your correlation will be. For example:

- Application logs
- Database logs
- Network logs
- Configuration files
- Cron jobs and other scheduled tasks
- Performance data (CPU, disk, memory, and so on)

_Treat your data source as part of your development software stack_

Work with your development team to establish detailed, organized, human-readable logs.

- Encourage development teams to create tags and notations in logs for easier identification
- Include creating custom reports, dashboards, and alerts in each application backlog
- Build analytics to support all code as part of its delivery criteria before releasing it

_Practice good log file management_

Log locally to files to create a persistent record. This avoids any data gap when there is a Splunk restart.

_Naming convention best practices for a Splunk deployment_

A naming convention is a system of using standard terms to classify categories of data so you can organize data in a way that makes sense to you.
Lexicographical order is alpha-numeric name sorting. Splunk uses lexicographical order to determine priority when processing knowledge object configurations. Many users new to Splunk are unaware of this, and incorrectly attribute faulty lexicographical ordering to buggy behavior. Once you understand how Splunk lexicographical order works, you can use it to your advantage to enhance performance.

When you design a naming convention for your apps and indexes, consider lexicographical order to optimize search efficiency and accuracy.

**Audience**

- Architect
- Developer
- Search expert
- Knowledge manager

For more about these roles, see [Roles best practices](#).

**How lexicographical order works**

The Splunk configuration (.conf) files define the logic that controls what Splunk does and in what order. Splunk determines configuration priorities based on factors such as the current user and current app (scope) and alpha-numeric name sorting (lexicographical naming). This enables you to blend configurations from different files of the same configuration type, tune your data's source type, and increase the performance of indexing and searching.

Lexicographical order is as easy as "1, 10, 3" and "A, B, c". Splunk prioritizes numbers first, then capital letters, then lowercase letters:
Guidelines for leveraging lexicographical order in naming conventions

Keep these guidelines in mind when applying lexicographical order to a naming convention for your knowledge objects.

Before working with configuration files, see these topics in the Admin Manual to get familiar with the nuances of configuration file structure, configuration file directories, configuration file precedence, and when to restart Splunk Enterprise after a configuration file change for a direct .conf edit to apply.

To help keep it all straight, Splunk provides btool, a command-line utility, to troubleshoot issues with .conf file interactions and precedence. See Use btool to troubleshoot configurations in the Troubleshooting Manual.

Apply lexicographical order to internal app naming

One of the best ways to use lexicographical order is in your internal app's naming conventions. For example, a naming convention such as <companyname>_ <purpose>_< (app|add-on)> enables you to easily differentiate your apps and configs from those downloaded from Splunkbase.

Lexicographical order effects the order in which apps are loaded, so you can control priority order with your naming convention. For example, if you want one group of objects to be evaluated before another, make sure the naming convention you apply sorts first according to the lexicographical hierarchy shown above.

Set up a global app to establish default behavior

You can use lexicographical order to set up a global app that will load before other apps. This enables you to set a general default behavior. The global app can contain basic settings. Apps that load after the global app can override (or fine tune) these settings as necessary.

For example, say you work for a company called 'Acme' and you created a global app named Acme_zglobal_ta. In that app, you have included global configurations, such as non-default port assignments and disabling the Splunk web server. On your search heads, you have deployed Acme_searchhead_ta, which enables the web server.

Acme_zglobal_ta loads first, which establishes that no Splunk instances will run the web server. Acme_searchhead_ta loads second and overrides that default
behavior just for your search heads. So the web server will run only on your search heads.

**Use unique names**

You might run into issues if multiple knowledge objects of the same type share the same name, even if they belong to different apps. Objects get applied in reverse lexicographical order of the app directories. To understand more about how this works, see Give knowledge objects of the same type unique names in the *Knowledge Manager Manual*.

To avoid these issues, give each knowledge object a unique name.

**Keep naming conventions for knowledge objects simple**

Keep it simple if you apply a naming convention to knowledge objects. Use a system others can easily understand and apply. A complex syntax is harder for other users to understand, and could discourage them from using it.

**Naming conventions for indexes**

Another best practice is to apply a naming convention to event indexes you create to manage user access, varying retention policies, or repositories for specific types of data or data from specific sources. A good naming convention contains descriptors to help identify the index, categories to classify the index, and a summary. This enables you to use wildcards in configuration files, such as `authorization.conf`, to manage access. Create a naming convention that is simple so it's more likely to stay accurate over time. For example,

```plaintext
<companyname>_<purpose>_<sensitivity>_<summary>.
```

*<companyname>*

Include your company name to help differentiate indexes you create from indexes provided by Splunk or other solutions. For example, `acme`.

*<purpose>*

The purpose descriptor indicates what the index was created to do. It should be generic enough to apply to any data source whose data is routed to that index. For example, `web`, or `firewall`. Keep names generic. Specific values, like product names, team names, organizations, people, and so on, are subject to change. For example, use `firewall` instead of a specific brand of firewall, such as `pan`.

*<sensitivity>*
Sensitivity is a category that indicates the degree to which data in that index should be protected or restricted to certain user groups, for example, topsecret, secret, and confidential, pci, or prod and nonprod. This enables you to leverage wildcards when assigning permissions to certain indexes to users in certain roles.

<summary>
If you have created a summary index to skim a large data set for a smaller subset of data to search, you can create an identifier for it in your naming scheme. See Use summary indexing for increased reporting efficiency in the Knowledge Manager Manual for more about summary indexing.

Using the example above, you might create the following naming scheme for indexes that serve different types of data:

- **acme_web_pci**: an index for the Acme web server for information that is subject to PCI regulations
- **acme_firewall_prod**: an index for the Acme firewalls in production
- **acme_av_nonprod**: an index for the non-production anti-virus server at Acme
- **acme_all_prod_summary**: a summary index of correlations produced from all of Acme's production indexes

The next example uses wildcards in the authorization.conf file to demonstrate how you can use your naming convention to manage access to specific indexes for specific user roles using the following options in the srchIndexesAllowed attribute, where * is the name of any index that matches the naming criteria.

- **Acme_***: For user roles that are allowed to access all data in all Acme indexes
- **Acme_*_nonprod**: For user roles that should never have access to production data
- **Acme_*_pci**: For user roles permitted to view PCI data

**Naming conventions for source types**

Source types do well by following the naming conventions outlined in Source types for add-ons in the Splunk Add-ons Manual.
Next steps

Try out the examples above using configurations and apps in your sandbox. Make up some scenarios of your own. Use `btool` with the `--debug` flag to explore how they be loaded.

Newsletter best practices for a Splunk deployment

A Splunk newsletter is an effective way to reach a broader audience with news and highlights about activities in your Splunk environment. A newsletter offers several benefits:

- Inspires new ideas from your Splunk community
- Demonstrates the value of Splunk to management and other stakeholders
- Compliments a Splunk Community Portal to help keep your organization informed and provide outreach (for more information, see Splunk Community Portal)
- Records your Splunk accomplishments and activities

Audience

- Program manager
- Project manager

For more about these roles, see Roles best practices.

Guidelines for implementing a Splunk newsletter

Here are some guidelines for putting together a successful Splunk newsletter.

Newsletter audience

Think about ways to inform and engage your entire user base. The content should be relevant to all Splunk roles. The newsletter should encourage everyone from basic users to management to participate in the Splunk community.

Frequency

Think about how often you would like to send out the Splunk newsletter, for example, once a month. If your community is particularly active, you can release
more often.

**Suggested content**

- Splunk use case highlights?insights gained using Splunk
- Calendar of events, such as workshops, trainings, or maintenance updates
- Announcements and important links
- Platform and user stats that indicate system health or other key metrics related to your Splunk goals
- Splunk tips and tricks

**Management and production**

- Embed statistics from Splunk reports and dashboards
- Cover events you want to bring to users’ attention
- Budget time in your project schedule to draft content for the newsletter in advance of the planned publication date

**Formatting and layout**

There is no prescribed layout for a Splunk newsletter. Create a design that works best for your organization. Here are some newsletter layout tips:

- Include a banner that identifies it as your Splunk newsletter with the following elements:
  - Time span of your newsletter (for example, "September 2018" for a monthly format, or "September 21, 2018" for a weekly format published on Fridays)
  - Key contacts for your Splunk team
- Identify the most important information you want your users to see, and make sure it's visible without having to scroll down
- Make sure that any action required of your users is clearly highlighted at the top
- Include images
  - Images, such as visualizations from Splunk, are worth 1,000 words
  - Pictures of people engage everyone in the content and the team's success
Quarterly business review best practices for a Splunk deployment

A Quarterly Business Review (QBR) is a quarterly status meeting for the stakeholders in your Splunk implementation. The purpose of a QBR is to provide updates on the overall plan, goals, and roadmap for your Splunk implementation. QBRs should include an executive dashboard and summaries of key activities and metrics. QBRs and executive dashboards provide your Executive Sponsor an opportunity to regularly engage with Splunk business goals and priorities.

Audience

- Architect
- Executive sponsor
- Search expert
- Knowledge manager
- Program manager
- Project manager
- User community

For more about these roles, see Roles best practices.

Plan your QBR

Planning a successful QBR requires some networking and outreach. Hosts should include the Splunk owner, architect, and program manager.

Encourage executive involvement
QBRs are most effective when your Executive Sponsor is involved as a key point stakeholder. This ensures visibility into the initiatives, highlights, and challenges associated with your Splunk implementation.

Bring in a diverse audience from different teams
Encourage leaders from all the different teams using Splunk to participate. Bringing in different teams ensures cross-functional review and collaboration, and provides access to diverse data sources for your Splunk correlation.

Involve your Splunk operations team
Consider including stakeholders from your computing and storage infrastructure team. This can drive their engagement in the Splunk infrastructure, help them understand the business criticality of Splunk
systems, and foster partnership for upcoming maintenance and scaling plans.

Design a QBR

A common format for a QBR is a presentation using a tool like Microsoft PowerPoint or Google Slides. A QBR presentation is a good opportunity to showcase the talents of your Splunk team, and to encourage the input and participation of key members. A QBR should contain the following information:

**Business objectives**

- Splunk project status
- Use case initiatives
- Cross-functional Splunk-related activities
- Key Performance Indicators (KPIs)
- Deployment overview

**License usage**

- Platform performance
- Capacity planning
- Service Level Objectives (SLOs)

**User enablement updates**

- Number of Splunk users
- Educational program status

**Value realization updates**

- Return on Investment (ROI)
- Accomplishments

**Use case showcase**

- Key success stories
- Live demos

**Risk assessment**

- Challenges and roadblocks
- Areas of concern
Your QBR should focus on quarterly and annual goals, and avoid tactical details, such as specific platform configurations, issue root cause analysis, or how to build a dashboard. Save implementation details for a Splunk team meeting.

Create an executive dashboard

A Splunk executive dashboard is useful for showing essential Splunk utilization metrics and system health statistics. For example:

For an example of how one company uses Splunk to populate business-critical dashboards for executive overviews and daily operations, see Keeping Your Medical Center CIO Engaged, a presentation given by a Splunk customer at Splunk .conf2017.

Responsibility assignment best practices for a Splunk deployment

A responsibility assignment matrix is a tool used in the field of project management to define roles. Understanding the tasks and activities associated with each role sets the expectations for the roles you assign in your staffing model. RACI is a popular model that defines:

- **Responsible** - Those who do the work to complete the tasks
- **Accountable** - Those who are ultimately answerable for the outcome
- **Consulted** - Those whose opinions are sought
- **Informed** - Those who need to be kept up to date

When filling out a RACI matrix for your Splunk community, consider the needs and priorities of all your stakeholders across the organization in addition to your
Splunk admins and users.

Audience

- Executive sponsor
- Program manager

For more about these roles, see Roles best practices.

Guidelines for RACI definitions

<table>
<thead>
<tr>
<th>Letter</th>
<th>RACI definition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>Responsible</td>
<td>Owner</td>
</tr>
<tr>
<td>A</td>
<td>Accountable</td>
<td>Sign-off</td>
</tr>
<tr>
<td>C</td>
<td>Consulted</td>
<td>Provide input</td>
</tr>
<tr>
<td>I</td>
<td>Informed</td>
<td>Notify</td>
</tr>
</tbody>
</table>

RACI template

Role names correspond with the roles outlined in Roles best practices?. For the most up to date list of activities and practices, refer to the coordinating About the Splunk Success Framework functional areas.

<table>
<thead>
<tr>
<th>Activity by functional area</th>
<th>Architect</th>
<th>Developer</th>
<th>Engineer</th>
<th>Executive sponsor</th>
<th>Search expert</th>
<th>Knowledge manager</th>
<th>Program manager</th>
<th>Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLATFORM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Admin education</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td></td>
<td>R</td>
<td>A</td>
<td>I</td>
</tr>
<tr>
<td>Change control processes, maintenance, and upgrades</td>
<td>C</td>
<td></td>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>Architecture</td>
<td>A</td>
<td>I</td>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td>I</td>
<td>I</td>
</tr>
<tr>
<td>Continuity Planning</td>
<td>A</td>
<td></td>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td>C</td>
<td>I</td>
</tr>
<tr>
<td>Support and incident management</td>
<td>C</td>
<td></td>
<td>R</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>A</td>
<td>I</td>
</tr>
<tr>
<td>Activity by functional area</td>
<td>Architect</td>
<td>Developer</td>
<td>Engineer</td>
<td>Executive sponsor</td>
<td>Search expert</td>
<td>Knowledge manager</td>
<td>Program manager</td>
<td>I mpor</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-----------</td>
<td>-----------</td>
<td>----------</td>
<td>-------------------</td>
<td>---------------</td>
<td>-------------------</td>
<td>----------------</td>
<td>--------</td>
</tr>
<tr>
<td>Capacity Management</td>
<td>C</td>
<td>R</td>
<td>C</td>
<td>C</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PROGRAM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Business Alignment</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td></td>
<td></td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Management of the buildout cycle</td>
<td>C</td>
<td>I</td>
<td>R</td>
<td>I</td>
<td></td>
<td></td>
<td>C</td>
<td>A</td>
</tr>
<tr>
<td>Governance</td>
<td>A</td>
<td>I</td>
<td>R</td>
<td></td>
<td>I</td>
<td>C</td>
<td>I</td>
<td>I</td>
</tr>
<tr>
<td>Value realization</td>
<td>C</td>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data request</td>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R</td>
<td>I</td>
<td></td>
</tr>
<tr>
<td>Data definition</td>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R</td>
<td>I</td>
<td></td>
</tr>
<tr>
<td>Implementation</td>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>Validation</td>
<td>R</td>
<td></td>
<td></td>
<td>R</td>
<td>I</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Communication</td>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>Use case retirement</td>
<td>C</td>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>USERS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Access request</td>
<td>I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>Role definition and capabilities</td>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>C</td>
<td>A</td>
</tr>
<tr>
<td>User access and user experience</td>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>User validation</td>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>I</td>
<td></td>
</tr>
<tr>
<td>User retirement</td>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>User education</td>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A</td>
<td></td>
</tr>
</tbody>
</table>
More resources

• For a staffing model template, see Staffing best practices.

Roles best practices for a Splunk deployment

A Splunk implementation team is made up of roles that demonstrate different strengths and skills with Splunk software and within your general business. These roles reflect the business skills needed to fulfill the associated duties, and do not necessarily map directly to the Splunk platform default user roles. One person on your Splunk team can fulfill more than one role. Roles and responsibilities are a good way to manage an incentive-based access model to encourage your user community to build and grow their Splunk software skills.

Audience

• Architect
• Developer
• Engineer
• Executive sponsor
• Knowledge manager
• Program manager
• Project manager
• User community

Splunk roles and responsibilities

The following table describes the common roles in a Splunk implementation, their general focus, and the recommended minimum level of Splunk education required for that role. A Splunk implementation team member can have a higher level of Splunk software certification than is required for that role.

<table>
<thead>
<tr>
<th>Splunk role and responsibilities</th>
<th>Required skills</th>
<th>Recommended education requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARCHITECT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Designs and optimizes Splunk platform architecture for</td>
<td>• Experience with interconnected, heterogeneous</td>
<td>• Splunk Certified Architect</td>
</tr>
<tr>
<td>Splunk role and responsibilities</td>
<td>Required skills</td>
<td>Recommended education requirements</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>----------------</td>
<td>------------------------------------</td>
</tr>
</tbody>
</table>
| large-scale and distributed deployments  
- Establishes best practices and development standards, and ensures that the team adopts them  
- Maintains a close partnership with Splunk on feature requests, upgrade planning, and product roadmap alignment | systems  
- Strong understanding of industry standards and technologies | |
| DEVELOPER                        |                |                                    |
| • Develops and customizes Splunk apps and dashboards  
• Implements integration with external systems  
• Builds advanced visualizations | • Basic web design  
• Scripting (such as Python or other) | • Splunk Certified Developer |
| ENGINEER                         |                |                                    |
| • Implements and maintains Splunk platform infrastructure and configuration  
• Undertakes day-to-day operational and user support  
• Executes new projects as well as data and user onboarding  
• Staffs help desk for Splunk platform system-related assistance | • WIN or *nix systems administration  
• Networking background  
• Familiarity with common infrastructure technologies | • Splunk Certified Admin |
<table>
<thead>
<tr>
<th>Splunk role and responsibilities</th>
<th>Required skills</th>
<th>Recommended education requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXECUTIVE SPONSOR</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| • Endorses and provides resources for the Splunk software investment | • Strong business acumen  
• Management experience  
• Considered an ‘influencer’ at the organization  
• Actively engaged in promoting Splunk software as a solution, service, or strategy | • Splunk Certified User |
| SEARCH EXPERT                    |                |                                     |
| • Customizes queries  
• Promotes advanced searching, forensics, analytics  
• Effects creative solutions to complex problems  
• Staffs help desk for search-related assistance | • Splunk Search Processing Language (SPL)  
• Splunk solution expert knowledge | • Splunk Certified Power User |
| KNOWLEDGE MANAGER               |                |                                     |
| • Manages data onboarding and defining configurations  
• Performs data interpretation, classification, and enrichment  
• Builds data models | • Experience with basic Splunk Search Processing Language (SPL)  
• Strong understanding of Splunk platform configuration, web | • Splunk Certified Power User |
<table>
<thead>
<tr>
<th>Splunk role and responsibilities</th>
<th>Required skills</th>
<th>Recommended education requirements</th>
</tr>
</thead>
</table>
| • Manages knowledge objects (fields, extractions, tags, event types, lookups, workflow actions, aliases, macros, and so on)  
• Configures summary-based reports and data model acceleration | UI and Common Information Model  
• Basic understanding of regular expressions | |
| **PROGRAM MANAGER** | | |
| • Manages the Splunk implementation team  
• Ensures Splunk implementation meets business requirements  
• Provides oversight on projects and cross-departmental initiatives  
• Spearheads communication to stakeholders  
• Facilitates maturity of the Splunk implementation team and user base through education programs | • Strong business acumen  
• Management experience  
• Some project management  
• There should be only one program manager | • Splunk Certified User |
| **PROJECT MANAGER** | | |
| • Scopes project requirements  
• Manages project timelines  
• Communicates progress and risks to stakeholders | • Project management expertise  
• Excellent communication skills | • Splunk Certified User |
### Splunk role and responsibilities

<table>
<thead>
<tr>
<th>Required skills</th>
<th>Recommended education requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bridges interactions between stakeholders and Splunk implementation team</td>
<td>Detailed knowledge of organizational and business process</td>
</tr>
<tr>
<td>Chief cat herder</td>
<td></td>
</tr>
</tbody>
</table>

### USER COMMUNITY

- Appreciates the value returned from Splunk analytics and reports
- Consumes reports, dashboards, alerts, and other use case-related dashboards
- May also include users who are Splunk Certified Power Users
- Varying levels of technical competencies and experience with SPL
- Can consume dashboards and alerts or write searches if inclined
- Experience with web browsers
- Splunk Certified User

### More resources

- For information on certifications and to access courses, see the Splunk Education homepage.
- For a staffing model template, see Staffing best practices.
- For the responsibility assignment matrix, see Responsibility best practices.

### Role-based data management best practices for a Splunk deployment

The Splunk roles feature enables you to define permissions and capabilities for a collection of users, for example, setting search limitations, providing access to product features, data, and knowledge objects, and setting the default app users land in when they log into Splunk. When data becomes more diverse and the number of users increases, it can be more complex to manage roles. This topic explores how you can manage roles, access to data, and access to product features in a more modular and scalable way.
Role separation is a process that segregates a user's access to data from their access to Splunk capabilities. Data governance is a process that controls access to certain data and capabilities for certain roles. Introducing a naming convention and separating roles and features can enable you to implement a highly flexible and scalable role-based access control (RBAC) solution.

This article does not discuss role inheritance. That is deliberate so you can clearly see direct roles and access permissions without having to consider layers of permissions inherited from other roles.

**Audience**

- Architect
- Engineer
- Search expert
- Knowledge manager

For more about these roles, see [Roles best practices](#).

**Guidelines for implementing role control and data management**

To make role control more granular, you can organize user access requirements into functional categories, such as:

- Data access
- Search restrictions
- Product feature capabilities
- Knowledge object permissions
- Default app

For example, a role classified as data access would define access to data, but not search restrictions, product feature capabilities, or knowledge object permissions. Here’s more about each of these five categories to better understand why such segregation can be useful.

**Data access (data)**

The `srchIndexesAllowed` parameter of `authorize.conf` is where you can specify which indexes a user is allowed to search. See Add and edit roles with `authorize.conf` in the *Securing Splunk Enterprise* for more information.

Rather than defining a role for data access per index, you can create collections based on commonality, like function. For example, `data_os`
where srchIndexesAllowed = windows, os, linux.

An index naming convention enables you to use wildcards, which becomes a boon to your solutions scalability. For details about how to establish a good naming convention, see In depth: Naming conventions.

Search restrictions (search)
Search restrictions enable you to develop a more flexible maturity model for managing user access. You can provide new users a role like search_new that restricts access to certain search functions (attributes of authorize.conf with the string arch) so that new users can’t make lasting mistakes on the platform as they learn.

For more information about authorization for product capabilities, search capabilities, and data, see About defining roles with capabilities in Securing Splunk Enterprise.

Product feature capabilities (feature)
A best practice with product feature capabilities is to limit access to features that can impact platform performance, such as scheduled searches and report accelerations, to those who have the technical training and architectural endorsement to use them. You can restrict access to these types of features for new users to limit risk to the system while they learn.

For more information about authorization for product capabilities, search capabilities, and data, see About defining roles with capabilities in Securing Splunk Enterprise.

Knowledge object permissions (ko)
As teams collaborate and create insights into their data, they will want to share knowledge objects with other users. A best practice to improve search performance is to establish more granular control over which groups can globally read knowledge objects. For example, if a security team uses many security-related tags and fields for their data models, they can limit the search parsing impact of sharing content with the web team by removing read access to the knowledge objects the web team doesn’t need.

The definition of the app and global knowledge object permissions is within the default.meta and local.meta from the app where the knowledge object was defined. For more about Defaultmeta.conf, see Defaultmeta.conf in the Splunk Enterprise Admin Manual.
For information on knowledge object permissions, see Manage knowledge object permissions in the Knowledge Manager Manual.

Default app (workspace)
When you create an app as a workspace, users enjoy a more comfortable and straightforward experience when they log into Splunk and land directly in an app tailored to their team or use case. They avoid having to hunt for the right place to start, and they can easily find the knowledge objects they need to do their jobs. See Workspace best practices.

To make an app a default workspace landing page, add the app name to the default_namespace attribute in user-prefs.conf.

For information on configuring a default app, see Configure Splunk Web to open directly to an app in the Splunk Enterprise Admin Manual.

Guidelines for using naming conventions to make access more modular

While Splunk's role inheritance helps manage the sprawl of roles, larger environments can struggle to keep connections and lines of access straight if there are layers of inheritance to sort. To address this, you can create roles that manage access to a particular capability without affecting other capabilities. To simplify this modular approach, you can use a naming convention for the roles you create like this: <capability>_<descriptor>.

Here are how the different <capability> of a role can be translated for this naming convention:

- data for data access
- search for search constraints
- feature for product feature capabilities
- ko for knowledge object permissions
- workspace for default app

A <descriptor> describes the purpose of that role. Putting it together with the <capability> you have examples like:

- data_operatingsystems for a role that defines operating system-related indexes allowed for searching
- workspace Middleware for a role that defines the default app the middleware team will see when they log in
- search_new for a role for new Splunk users that restricts access to limit system impact
• `feature_architecture` for a role that unlocks distributed deployment capabilities
• `ko_marketing` for a role that is used with Splunk permissions to provision read/write access to knowledge objects

**Provisioning a group’s role**

Once you establish modular groups like the example above, you can assign any given team within Splunk roles that are an aggregate of these different modular groups. Groups can be constructed from at least one selection of data, search, feature, ko, and workspace components.

Use this approach only if your Splunk implementation has scaled to the point that such differentiation makes sense. If your Splunk implementation is fairly simple, your needs may be addressed sufficiently with the traditional approach to role management.

**Promoting users**

Over time, business changes may require that a given user’s role evolve. This model supports such changes because users can just be made members of new groups to give them the new access they need.

For example, a user could be a member of the `data_operatingsystems` group, but now also needs access to web data. So you can just add them to the `data_web` group. In another example, a user is being promoted to perform more advanced search operations. So while the user is a member of `search_user`, you would also add the user to `search_alerting`, which provides the user the search capabilities to create and schedule searches that have alert functionality.

In both these scenarios, the user’s capabilities were changed without without needing to change the underlying Splunk configuration of these roles. That means you can implement this business need without a change approval or platform restart.

**Manage access to data and capabilities using Active Directory groups**

Membership to each of the groups may be implemented through third party solutions and connected to Splunk with SAML, SSO, and/or LDAP. You can map those user group definitions directly to Splunk roles.
Here is an example of some of the roles available within a hypothetical customer’s Active Directory platform. In this situation, users are made members of at least one of each type of group.

<table>
<thead>
<tr>
<th>Role Name</th>
<th>Group Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>capability_admin</td>
<td>Security Group - Global</td>
</tr>
<tr>
<td>capability_batch</td>
<td>Security Group - Global</td>
</tr>
<tr>
<td>capability_developer</td>
<td>Security Group - Global</td>
</tr>
<tr>
<td>capability_priveleged</td>
<td>Security Group - Global</td>
</tr>
<tr>
<td>capability_user</td>
<td>Security Group - Global</td>
</tr>
<tr>
<td>data_all</td>
<td>Security Group - Global</td>
</tr>
<tr>
<td>data_customerinfo</td>
<td>Security Group - Global</td>
</tr>
<tr>
<td>data_dreamhost</td>
<td>Security Group - Global</td>
</tr>
<tr>
<td>data_internet</td>
<td>Security Group - Global</td>
</tr>
<tr>
<td>data_maple</td>
<td>Security Group - Global</td>
</tr>
<tr>
<td>data_operatingsystems</td>
<td>Security Group - Global</td>
</tr>
<tr>
<td>ldap</td>
<td>Security Group - Global</td>
</tr>
<tr>
<td>workspace_admin</td>
<td>Security Group - Global</td>
</tr>
<tr>
<td>workspace_default</td>
<td>Security Group - Global</td>
</tr>
<tr>
<td>workspace_developer</td>
<td>Security Group - Global</td>
</tr>
<tr>
<td>workspace_power</td>
<td>Security Group - Global</td>
</tr>
<tr>
<td>workspace_user</td>
<td>Security Group - Global</td>
</tr>
</tbody>
</table>

**Before working in conf files**

The configuration file `authorize.conf` contains about 100 parameters you can use to fine tune permissions broadly, such as by role `[role_<roleName>]`, or narrowly, such as by list settings `[capability::list_settings]`. Splunk Web exposes some, but not all of these settings.

This article refers to making edits to `authorize.conf`. Before working with configuration files, see these topics in the Admin Manual to get familiar with the nuances of configuration file structure, configuration file directories, configuration file precedence, and when to restart Splunk Enterprise after a configuration file change for a direct `.conf` edit to apply. To help keep it all straight, Splunk provides `btool`, a command-line utility, to troubleshoot issues with `.conf` file interactions and precedence.

**Sandbox best practices for a Splunk deployment**

A best practice for establishing a stable and reliable production Splunk environment is to set up a workflow that includes individual sandboxes for development and innovation, a Splunk Lab environment for testing, and a safe push to production once things are ready.

A sandbox is a stand-alone Splunk Enterprise instance used by one person as a safe place to innovate and develop new ideas. Sandboxing is the process of establishing and using a sandbox. Everyone on your Splunk team should have their own sandbox so they feel safe to take risks and learn. With your own
sandbox, you won't be afraid to start over if you need to.

Encouraging a healthy sandbox culture for your Splunk team promotes learning and growth, and ensures that your innovators have the latitude to try new things without disrupting what already works, or each other.

**Audience**

- Architect
- Developer
- Engineer
- Search expert
- Knowledge manager
- User community

For more about these roles, see [Roles best practices](#).

**Guidelines for setting up a sandbox**

A sandbox should be easy to set up and easy to tear down so you feel free to innovate. Here are some guidelines and tips.

**Options for deploying a sandbox**

You have several options for setting up a Splunk sandbox. Whatever option you choose, use a basic Splunk setup, and keep customizations to a minimum. The more effort you put into it, the greater the fear of breaking it, and the less comfortable you will be taking risks and trying new things.

**Splunk Docker sandbox (recommended)**

Splunk has done all the thinking for you and published a Docker image so you can set up a Splunk sandbox in minutes. This is the quickest and easiest way to sandbox. See the Splunk blog Hands on Lab: Sandboxing with Splunk (with Docker) for complete instructions. To keep it simple, set up your Splunk Docker sandbox on your localhost.

**Splunk Search Tutorial sandbox**

Setting up a Splunk sandbox using the Splunk Enterprise Search Tutorial is a great way to get familiar with installing and launching Splunk. The instructions also lead you through the basics of getting data in, searching, and creating reports and dashboards. See About the Search Tutorial in the [Splunk Enterprise Search Tutorial](#).
Splunk sandbox on Cloud or VM
You can use a public cloud (such as Amazon Web Services?AWS) or one that is private to your company. Consider any concerns your organization may have about exfiltration, or unauthorized transfer of personal or company data to a cloud-based resource.

You can also set up a virtual machine on your local instance. Oracle VM VirtualBox is a free and open-source option. Be aware that installing a virtual machine can be involved, especially if you want to replicate a distributed environment.

**Sandbox licensing**

The Splunk Enterprise trial license will revert to a free license 60 days after download, which limits its capabilities. To learn more, see Splunk Platform Comparison on the Splunk web site. For more information about Splunk licenses, see Types of Splunk software licenses in the *Splunk Enterprise Admin Manual*.

Splunk notifies you when your 60 day period is about to expire. Think of this notification as a reminder that your sandbox is meant for experimentation, and you should not be afraid to break it. You can also switch to a free license from within Splunk: go to *Settings > Licensing* and click *Change license group*.

For a more formal development setup, you can also get a developer license for your sandbox. See Splunk Developer License Signup on dev.splunk.com.

**Getting data into your sandbox**

You have several options for getting data into your sandbox.

Sample data
The most basic way to get data into your sandbox is to download sample data from Splunk. For sample data, see Upload the tutorial data in the Splunk Search Tutorial?.

_internal index
You can also use the _internal index. It has logs about your own instance, which are often perfect for sandboxing.

Export data
If your organization's policies allow you to use production data in your sandbox, but you don't want to connect directly to a production indexer, you can use the export feature in Splunk to get a sample of raw events.
Export data as raw events so you can experiment with defining the source type. You can also use a semi-structured format, such as JSON or XML. For information about exporting data, see Export data using Splunk Web in the Splunk Search Manual.

When you export data from the production environment, be aware of the amount of data you are requesting. If the data set is too large, Splunk could become unresponsive while transmitting the export.

Production data
Wherever your sandbox is hosted, you can point to existing indexers, including those in your production environment. You can have a many-to-many relationship between search heads and indexers (or index clusters). Before proceeding with this option, think about the purpose of your sandbox and consider the impact this will have on the environment. If you are using your sandbox for an indexer-heavy workload, connecting to production indexers may not be the best solution.

- For information about connecting to production non-clustered indexers, see Add search peers to the search head in the Distributed Search Manual.
- For information about connecting to production indexer clusters, see Enable the search head in Managing Indexers and Clusters of Indexers.

Tips for advanced sandboxing

- The `| makeresults` command enables you to experiment with commands by fabricating results.
- The `|extract reload=true` command enables you to apply props or transforms without having to restart Splunk.
- The `/debug/refresh` endpoint or the `/en-US/_bump` endpoint command enables you to get some config or static content reloaded.
- Development mode enables you to add parameters to web.conf to prevent caching. For development mode settings, see Build a custom visualization > Getting Started in Developing Views and Apps for Splunk Web. Development mode settings are especially useful when developing a new app, as it will enable you to get static content reloaded without having to `_bump` every time.

When to avoid using a sandbox

Do not use a sandbox for transaction-based work. A sandbox is not intended for transaction-based work, such as defining a new search or field extraction.
are some other methods for experimenting in the production environment without incurring risk:

- Run your experiments on a reduced time range. This enables you to run highly complex searches without having a heavy impact on search heads, indexers, and other people using the environment. See Select time ranges to apply to your search in the Splunk Search Manual.

- Use the `| head` command to validate a search with 10 events by default. See Head in the Search Reference manual.

- Use the `| makeresults` command to generate data without worrying about the data set. See Makeresults in the Search Reference manual.

Do not use a sandbox for editing knowledge objects because of how long it takes to bring the data in and perform field extractions. Instead of using a sandbox, clone the knowledge object within the production environment. This enables you to edit the knowledge object without changing macros, saved searches, dashboards, or other settings.

- When cloning a knowledge object, set the permissions to private so only you can view it.
- Once you are done experimenting with the cloned knowledge object, copy the new configuration into the production knowledge object.
- When the updated knowledge object is transferred into production, delete the clone.

### Service-level best practices for a Splunk deployment

Service-level definitions are a contract between a service provider and the organization it serves that defines particular aspects of the service, such as quality, availability, and responsibilities. Service-level definitions consist of service-level objectives (SLOs), service-level agreements (SLAs), case priority levels, and incident response times. When Splunk is operated as a service offering, service-level definitions provide all teams and organizations assurance that Splunk operations and response models meet their needs without impacting other areas.

**Audience**
• Developer
• Engineer
• Search expert
• Program manager
• Project manager
• User community

For more about these roles, see Roles best practices.

Key terms

Mission critical
An outage impacting revenue, ability to hit agreed SLA/OLA, or a noted mission-critical data source or app.

Core service
Indexing, Searching, or Alerting.

Routine change
A low-impact, low-risk change not requiring a change review.

Emergency change
A change required to resolve a P1/P2 condition.

Service request
A request to add new capacity of lower complexity than a project, for example, new inputs or new add-on installs.

Guidelines for implementing service level definitions

There are many factors to consider when making a service-level commitment. The table below lists some guidelines to follow.

| Don't over-commit | • Consider what might be too fast for delivery.  
|                   | • Don't make a commitment that’s unreasonable or unfair to your team members. |
| Don't under-commit| • The requester should never feel that the response time is unreasonably long. |
Be as accommodating as possible when setting goals for turnaround

Consider the time it takes to gather the necessary information

- Many types of requests require follow-up with the requester. Recognize that there may be a waiting period for this additional information.
- Have reasonable expectations when requesting additional information. Make sure to communicate your expectations to the requester.
- The requester should be aware that a slow response time will have an impact on expected turnaround time.

Think about the process for incoming requests

- Think about your engagement model for incoming requests. Optimize the request process so teams can work together effectively.
- Create a process that is straightforward and effective.

SLO templates

SLOs provide expectations for maintenance planning, release planning, and communication with business partners. You can divide SLOs into administrative tasks (day-to-day activities) and implementation tasks.

Use the example provided below or make any updates as necessary.

<table>
<thead>
<tr>
<th>Administrative SLOs (day-to-day activities)</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete new user</td>
<td>5 business days</td>
</tr>
<tr>
<td>Add new user</td>
<td>5 business days</td>
</tr>
<tr>
<td>Uplifting user</td>
<td>25 business days</td>
</tr>
<tr>
<td>Dashboard creation support</td>
<td>10 business days</td>
</tr>
<tr>
<td>Report generation support</td>
<td>5 business days</td>
</tr>
<tr>
<td>Alert creation and changes support</td>
<td>5 business days</td>
</tr>
<tr>
<td>Create a new Active Directory group for access</td>
<td>25 business days</td>
</tr>
</tbody>
</table>
### Administrative SLOs (day-to-day activities) | Target
--- | ---
(External dependencies) |  
Create new role | 15 business days

### Implementation SLOs | Target
--- | ---
First response to new support request | 1 day
Data ingest (standard add-on) | 5 days
Data ingest (custom add-on) | 10 days
New app install | 1 day
Universal forwarder deployment (Does not include change control SLO) | 10 business days
Data source monitor (http, WMI, TCP/UDP) | 2 days
Implement new global knowledge object | 1 day
Upload data into Splunk (for example, static log, file, CSV) | 5 business days

### SLA templates

SLAs are key service definitions for platform availability and incident response.

Use the example provided below or make any updates as necessary.

| SLAs | Target |
--- | --- |
Platform availability | 99.9% uptime for all core services (< 8.76 hours unplanned downtime per year) |
Incident first response | Based on priority |
Incident status update | Based on priority |
Restore loss of data feed (ingestion) | 1 business day |
Restore universal forwarder not reporting (standard) | 5 business days |
Restore universal forwarder not reporting (mission critical applications) | 1 business day |
Case priorities

Case priorities are assigned based on the technical importance of the problem. The following case priorities are intended only as examples.

Use the examples provided below, or make any updates as necessary.

**Case priority levels**

Case priorities may vary by service or source. The following are general guidelines.

<table>
<thead>
<tr>
<th>Case priority level</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>A mission critical outage for which there is no workaround. This may be a complete service outage of a core service.</td>
</tr>
<tr>
<td>P2</td>
<td>A mission critical outage for which a less than ideal workaround exists. This may be a partial service outage of a core service.</td>
</tr>
<tr>
<td>P3</td>
<td>An outage or issue impacting a single user.</td>
</tr>
<tr>
<td>P4</td>
<td>Standard service requests or routine changes. For example, access requests, data onboarding, app installation, etc.</td>
</tr>
</tbody>
</table>

**Incident response times**

<table>
<thead>
<tr>
<th>Incident response</th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
</tr>
</thead>
<tbody>
<tr>
<td>First response</td>
<td>1 hour</td>
<td>2 hours</td>
<td>4 hours</td>
<td>1 business day</td>
</tr>
<tr>
<td>Communicated</td>
<td>Every 2 hours</td>
<td>Every 4 hours</td>
<td>Every business day</td>
<td>Every 5 days</td>
</tr>
<tr>
<td>updates</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resolution time</td>
<td>Within 4 hours</td>
<td>Within 2 business days</td>
<td>Within 3 business days</td>
<td>Agreement with the customer</td>
</tr>
<tr>
<td>Business hours</td>
<td>24 hours / 7 days per week</td>
<td>8:00am to 5:00pm / 5 days per week excluding holidays</td>
<td>8:00am to 5:00pm / 5 days per week excluding holidays</td>
<td>8:00am to 5:00pm / 5 days per week excluding holidays</td>
</tr>
</tbody>
</table>
Showback best practices for a Splunk deployment

Showback is a method of tracking Splunk utilization by an organization's business units or end users. Unlike chargeback, showback keeps track of utilization without actually cross-charging those costs back to the departments that use them.

A showback plan is a way to demonstrate that you are getting your money's worth from your Splunk investment. Metrics and cost tracking are a way to measure that value and allocate it to the users and beneficiaries of that value.

A showback practice is most effective for implementations with more than one constituency that are scoped as Splunk as a Service or Splunk as a Strategy. A showback plan may be more formal than is needed for Splunk-as-a-Solution deployments that serve only one team, group, or purpose. See Define a charter for your Splunk implementation for an explanation of charter levels.

Having a data-driven showback plan can be an effective way to foster positive competition among teams to make their Splunk usage more effective by challenging them to unlock more value per unit of cost. A showback plan should drive adoption and improve efficiency, not just track cost centers.

Audience

- Executive sponsor
- Program manager

For more about these roles, see Roles best practices.

Guidelines for implementing a showback plan

Here are some guidelines for implementing a successful showback plan.

Measure utilization on a scale

Start by establishing a benchmark for your Splunk usage. Your initial benchmark will be a guess, but as your user community spends time with Splunk, these estimates will become more accurate and detailed.

Next, design a measurement plan that matures over time. Start out with what you can easily measure, and add in things that are harder to measure over time when you have usage patterns on which to base your estimates. Communicate this
plan to your user community. For example:

- Years 1 and 2: measure license usage only
- Year 3: measure license and hardware costs
- Year 4: measure license, hardware, and support costs

This gives everyone in your organization time to establish their Splunk usage and value realization while getting licenses properly funded. Adjust the timing as needed.

**Establish baseline agreements**

Establish baseline data usage agreements
- Include storage retention solutions and staff overhead.

Base license usage requirements on near-maximum usage
- Using near-maximum measurements rather than averages will ensure that the baseline you establish has enough headroom for meeting daily volume obligations without exceeding license boundaries.

Provide a fixed chargeback cost and account for overages
- This provides an incentive for users to be good citizens.

Avoid providing discounts for under-usage
- This can drive less adoption.

**Include a variety of factors in usage measurements**

Keep measurements general
- Include general usage measurements, such as license and overall memory usage. Avoid measuring minute details, such as the search runtime or memory usage of individual use cases.

Account for indirect usage
- Factor in outside resources, such as storage retention solutions, hardware needs, and support usage.

Include predictions
- Predictions help stakeholders see their growth over time. Predictions help set expectations, and keep planning and communications proactive.
**Measure support time and grade on a scale**

When measuring support time, design a plan that encourages adoption. If you measure support time at a flat rate, it can drive less adoption (for example, users may think if they use Splunk less, they will need less support and get charged less). You can grade groups based on percentage of certified users, search maturity, add-on support, and support time. This can build an incentive for users to increase their Splunk adoption. For example:

- Group A wrote a power shell to pull exactly what they needed and maintained the app themselves so the support group didn't have to. Support time was trivial.
- Group B spent less time learning and relied on the support group to build tooling for them. Support and training time was greater.

**Next steps for implementing a showback plan**

The Chargeback Analysis for Splunk app from Splunkbase is a great resource to start implementing a showback plan.

**Utility tier disaster recovery best practices for a Splunk deployment**

Splunk provides product features to increase availability and recovery options for the search tier (search head clustering) and the indexing tier (indexer clusters and index replication). Administrative functions, such as the deployment server, deployer, and licensing server, rely on best practices to provide their resiliency. This article refers to those functions as the utility tier, and outlines the best practices to ensure their recoverability.

**Audience**

- Architect
  - Engineer

For more about these roles, see [Roles best practices](#).

**Impact of failures on the utility tier**

The components of the Splunk utility tier are used for Splunk administration. If any of these components are unavailable or destroyed, the respective functions
and resources become unavailable. Note this does not include search heads, indexers, nor forwarders.

<table>
<thead>
<tr>
<th>Component</th>
<th>Impact if offline</th>
<th>Impact if destroyed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deployment server</td>
<td>No impact to search and indexing functions</td>
<td>Source of truth of environment's configuration destroyed</td>
</tr>
<tr>
<td>Deployer</td>
<td>No impact to search and indexing functions</td>
<td>Default configuration for search head cluster is lost, but can be mostly rebuilt from a SHC member</td>
</tr>
<tr>
<td>Master node</td>
<td>No data redundancy requirements</td>
<td>Default configuration for indexer cluster member is lost, but can be mostly rebuilt from a member.</td>
</tr>
<tr>
<td>License server</td>
<td>No impact to indexing functions</td>
<td>72 concurrent hours shuts down search functions See About the connection between the license master and license slaves in the &quot;Splunk Enterprise Admin Manual&quot; for more information. System would need to be rebuilt. No impact to end users if the rebuild happens within 72 hours.</td>
</tr>
<tr>
<td>Monitoring console</td>
<td>No impact to search and indexing functions. Lost health and performance visibility and monitoring for search and indexing functions. System would need to be rebuilt. Risk to operations if health and performance visibility and monitoring for search and indexing functions is offline for a long period. Built-in summary data showing insights and long term patterns would be lost. No lasting impact to end users or the overall Splunk platform.</td>
<td></td>
</tr>
</tbody>
</table>

If any of these components are destroyed, it requires effort to rebuild a new instance and update references to the new host's information throughout the
environment. You can mitigate these laborious and error-prone efforts by applying best practices.

**Preserve component’s state**

Many customers use virtual machines instead of bare-metal hardware for utility-tier components because virtual machines provide two features that are valuable for utility-tier components:

- **Dynamic resource sizing**
  VMs change the hardware specifications of the host as load increases.

- **State preservation and transition**
  VMs provide host snapshots that preserve an image of the instance. Some VMs, such as VMotion from VMWare, enable you to instantiate the host image on a new virtual machine.

If you are unable to leverage these benefits from virtual machines, consider putting a configuration backup plan in place. For more information about configuration backups, see Back up and restore best practices.

**Preserve networking using DNS entries**

When a utility instance fails or is destroyed, the administrative task to update networking details to all clients, such as host name and IP, can be impractical in large and distributed data center environments. You may be able to avoid that labor by rebuilding a utility component with the same networking details the previous one used, but this is usually not possible. A best practice is to use DNS CName (canonical name) records as a translation service.

When you establish DNS CNames for your utility instances, you can direct all clients to those DNS entries, and thus never need to rely on the true host and IP of the host hardware. If you have to replace the host hardware, you do not have to try to reuse the same hostname and IP. This also enables you to build new utility instances in parallel to the old with a simple DNS toggle as a cutover.

*Applications for load balancing*

You can use a similar practice for load balancing on the data collection tier or search tier. In such scenarios, a DNS A record distributes traffic to multiple hosts, which provides you an easy way to scale. Even if you have a single instance acting as your search head or data collection tier, you can use this kind of networking for scalability and easy management.
For load balancing the indexing tier, however, Splunk's native load balancing feature is the best practice for forwarding data to indexers. For more information, see Set up load balancing in the Splunk Enterprise Forwarding Data Manual.

Next steps

Partner with someone who oversees networking at your organization and make sure they understand the goal and the technical details. Draft the disaster recovery plan and verify it with a non-impacting/non-production environment before implementing it in production.

Staffing best practices for a Splunk deployment

A role is not a person or job description, a role is a collection of responsibilities delegated to existing staff member(s). Think about roles and responsibilities before you think about the individual members on your team. When you approach staffing by role rather than by individual, you can better estimate your staffing needs.

The size of the staff you need to operate your Splunk implementation depends on how you set up your Splunk business model and the needs of your organization, not on how much data you ingest.

Follow these guidelines for setting up a staffing model and planning your staff needs.

Audience

- Executive sponsor
- Program manager
- Project manager

For more about these roles, see Roles best practices.

Guidelines for creating a staffing model

Create a resource that identifies who is on your Splunk team and the roles they fulfill. Post it in a place where stakeholders and team members can access it easily. Include contact information, such as an email link and a picture, so people know who to look for and where they are located. Consider this as information you share on your communication portal. For more information about a
Staffing model template

<table>
<thead>
<tr>
<th>Team member</th>
<th>Program manager</th>
<th>Developer</th>
<th>Engineer</th>
<th>Executive sponsor</th>
<th>Search expert</th>
<th>Knowledge manager</th>
<th>Architect</th>
<th>Project manager</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joe Smith</td>
<td></td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sally Brown</td>
<td></td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>John Doe</td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Recommendations for staff sizing

One person can usually manage a single instance of Splunk, a deployment server, and several forwarders. However, as your Splunk implementation grows and you add some advanced features to meet your data analysis needs, you may need more staff. The larger, more distributed, and service-oriented your implementation, the more people you will need to keep it running smoothly.

Technical drivers that influence staffing decisions

Increased complexity and mitigating risk are the two main drivers for increasing Splunk staff. Here is a closer look at some situations that can increase the demands on your team. For each of these advanced features, consider adding at least half a person’s time. The skills needed to address that demand fall within the roles of architect and engineer. For details about these roles and skills needed, see Roles best practices.

Distributed deployment

If an implementation shifts to a more distributed deployment model that separates indexers from the search head, you may want to add an
architect or engineer to help manage the expanded deployment. Another team member can provide peer review and help optimize and maintain a distributed deployment.

Indexer clustering
If you implement indexer clustering, your staff should have the necessary data management skills to maintain data fidelity between data sources and the indexer cluster nodes. You should also have sufficient staff to ensure timely response in case a problem arises--if you have high availability requirements for data or search, you also need high availability for people.

Search head clustering
If you implement search head clustering, your staff should have the necessary capacity tuning and optimization skills to maintain and optimize search head performance.

Data collection tier
If you establish a data collection tier, modular inputs and third-party data forwarding can add administrative complexity. Your staff should have the expertise with the systems your Splunk deployment integrates with.

Complex utility tier
Utility Splunk instances, such as Splunk deployers, masters, and deployment servers, are usually managed easily within normal operations. However, if you deploy complex redundancies, such as a pool of deployment servers, this can increase the team workload.

**Operational drivers that influence staffing decisions**

How you set up your operational model can also influence your staffing needs. Here are some considerations for how you set up Splunk in your organization, and how that can influence staffing decisions. The skills required for staff that interact with your customers and their use cases are filled by the developer, search expert, and user community roles.

Closed platform approach
In a closed platform setup, Splunk staff are responsible for managing and creating all knowledge objects. This model is more resource intensive. In an environment where Splunk is a service or a strategy, the Splunk staff are rarely the subject matter experts for a given use case's technical domain. That means your staff will need to spend more consultation time to understand what aspects of the data are important and worth getting
insights into. If this is your model, make sure you have enough staff to devote the necessary time to to explore the data and be inspired with other questions.

Open platform approach
In an open platform setup, end users are empowered and entitled to implement their own use cases. This enables them to provide their own subject matter expertise, and is generally less resource intensive. The focus of your team shifts from consultation to education, empowerment and community management.

Whether you adopt a closed or open platform approach, Splunk usage often grows virally as the user community begins to use it, experiments using their own SPL, and eventually becomes proficient enough to create their own knowledge objects. As one person learns, they share their knowledge with another. As this happens, you should have enough staff to support the more consultative skills of the closed platform approach, and the education, empowerment, and community management skills of the open platform approach.

**Staffing recommendations**

This topic has moved to Staffing best practices. Please update your bookmarks.

**Stakeholder best practices for a Splunk deployment**

A stakeholder register is a record of anyone in your organization who has an investment in Splunk, from the managers who approved the purchase to the user community who uses Splunk every day.

Creating a stakeholder register is important because it helps you keep track of your Splunk constituents and understand their needs. This can be an important benchmark so you know what to do for which stakeholders, and how to measure their success with Splunk, and ultimately, measure how Splunk helps the overall success of your organization.

**Audience**

- Executive sponsor
- Program manager
- Project manager
For more about these roles, see Roles best practices.

**Identify stakeholders**

To get started, make a comprehensive list of anyone who may have a stake in Splunk, either directly or indirectly, for example:

- Managers
- Executives
- IT and infrastructure team
- Security operations
- Your Splunk team
- Other departments
- Customer-facing organizations
- Legal

Anyone with an interest in the success of your company or organization is a potential stakeholder, even investors, future employees, and users of your products and services. So although you may not have to manage regular communication with all these stakeholders, you may have occasion to spotlight insights gained through your Splunk use cases to these outside audiences. Accounting for them on your stakeholder register can also remind you of the wide-reaching effects of your work with Splunk.

A stakeholder register should be a living document that you update regularly as people and priorities shift. The register includes sensitive details, such as names, contact information, and management strategies, so keep your register in a secure location with restricted access, and exercise caution when sharing it.

Program managers are responsible for tracking stakeholder goals and making sure they are met. The program manager should create the stakeholder register during customer onboarding and update it regularly.

**Guidelines for creating a stakeholder register**

At a minimum, the stakeholder register should contain three types of information for each stakeholder:

- General information - name and contact information
- Assessment - needs, expectations, and dependencies on Splunk
- Classification - attributes used to identify types of stakeholders

These are described in more detail in the following sections.
**Stakeholder general information**

For each stakeholder, capture the following information:

- Name
- Title
- Contact information
- How they prefer to communicate (for example, email, phone, instant messaging)
- General role and responsibilities in their organization
- Whether they are internal or external to the company
- What version of Splunk they’re using (if applicable)

**Stakeholder assessment**

Include a high level evaluation of each stakeholder, for example:

- What they need to know about Splunk
- How often they need communication
- What their expectations of Splunk are
- What influence they have on Splunk projects and activities
- What interest they have in Splunk and what authority they have to influence decisions

This can be in a narrative format, or organized tabularly depending on what tool you use for your stakeholder register.

**Stakeholder classification**

You can also manage groups of stakeholders by classifying them using various criteria. This resource is flexible; you can use classifications related to Splunk, or initiatives outside of Splunk. Here are some examples:

- Classify stakeholders based on their authority, responsibility, or interest in Splunk, for example, high, medium or low.
- Assign other attributes to stakeholders, such as internal, external, positive, supporter, detractor, neutral, champion, executive sponsor, contract signer, and so on.

**Stakeholder management strategy**

As an option, you can include stakeholder management strategy information in your stakeholder registry. Management strategy information includes ideas such
as how to get users more involved, increase their training knowledge, garner support for expanding Splunk operations and infrastructure, and so on.

**Stakeholder register template**

Below is a template you can use. The classification segment shown here is just an example. Use whatever classification criteria apply to the stakeholders in your Splunk community.

<table>
<thead>
<tr>
<th>General</th>
<th>Assessment</th>
<th>Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Title</td>
<td>Role</td>
</tr>
</tbody>
</table>

**Unix profile best practices for a Splunk deployment**

As a Splunk engineer, you may work with various Unix-based operating systems and correspondingly different standards and practices. In these environments, the Splunk installation may not be in the same location, which can make it difficult to use the Splunk command-line features needed for your job function. Since you cannot ask all operating systems to conform to one set of standards, a non-intrusive alternative is to use environment variables to mask such differences, which creates a consistent and effective environment to work in.

**Audience**

- Architect
- Engineer

For more about these roles, see [Roles best practices](#).

**Guidelines for implementing Splunk Unix profiles**

The code examples here are unsupported and can vary depending on the Unix shell you use. Refer to the documentation for your Unix shell to ensure these guidelines are valid for your environment.

Use the following variables and commands to provide a consistent shell experience regardless of differences in the underlying operating system.
Set SPLUNK_HOME

Splunk can be installed in many locations. To normalize the variation of installation directories across instances, you can use a script to define and map them to SPLUNK_HOME.

The first line in the following example is pre-populated with some common installation locations. Modify this to match your needs.

```bash
for SPLUNK_HOME in "/Applications/Splunk" 
  "Applications/SplunkForwarder" 
  "/opt/splunk" 
  "Applications/SplunkBeta" 
  ":${HOME}/splunkforwarder" ; do
  if [ -d ${SPLUNK_HOME} ]; then
    break
  fi
done
if [ "${SPLUNK_HOME}" == "" ]; then
  echo "WARNING: SPLUNK_HOME env variable undefined"
fi
export SPLUNK_HOME
```

Set history control

Unix keeps command line history. This can be troublesome from a security standpoint when Splunk commands require that you enter a username and password at the command line. The Unix variable HISTCONTROL controls how the shell terminal stores command history. Set the HISTCONTROL variable in your environment and use the flag ignorespace to ignore commands that have a leading space. Execute the Splunk commands that use a username and password with a leading space so they won't persist to the history.

```
HISTCONTROL=ignorespace
```

Add Splunk to PATH

You do not have to navigate to ${SPLUNK_HOME}/bin every time to run ./splunk! You can add items to your PATH to reference wherever your command line navigation takes you.

```
export PATH=$PATH:/usr/bin:$SPLUNK_HOME/bin
```

Add btool to PATH

The troubleshooting utility btool is powerful, but you might avoid using it because you have to remember how to navigate to it. By adding btool to your PATH, you can call upon it wherever and whenever you need it. For more about btool and how to use it, see Use btool to troubleshoot configurations in the Splunk

89
Enterprise Troubleshooting manual.

KERNEL=`uname -s`
case "x$KERNEL" in
  "xLinux")
    if [ -f /etc/lsb-release -o -d /etc/lsb-release.d ];
    then 
      export DISTRO=$(lsb_release -i | cut -d: -f2 |
        sed s/'^	'//)
    else 
      export DISTRO=$(ls -d 
        /etc/[A-Za-z]*[\-_][rv]e[lr]* | grep -v "lsb" | cut -d'/' -f3 | cut 
        -d'-' -f1 | cut -d'_' -f1)
    fi
    if [ "x$DISTRO" != "xUbuntu" ];then 
      export LD_LIBRARY_PATH=$SPLUNK_HOME/lib 
    fi 
    ;;
  "xDarwin")
    export DYLD_LIBRARY_PATH=$SPLUNK_HOME/lib
    ;;
  *)
    echo "ERROR: Unable to set LIBRARY_PATH"
    exit 1
    ;;
esac

Prompt string 1 (PS1)

PS1 is the primary prompt that is displayed before each command. If you have Splunk instances distributed on systems with OS variations, having consistent PS1 prompt information from system to system can save time and reduce mistakes. Here are some suggestions about what variables to set and why so you have the same prompt information for each OS variation you work with.

PS1="[\`date\`] [\u@\h \w]\n> "

- Use brackets and special characters as needed to format the prompt string, for example, '['], '@', and '>'
- Use > as the prompt character, since it resembles the Splunk logo
- Use [\`date\`] to automatically execute a timestamp on every command prompt so you can have an approximate timeframe of the commands you issue on that system
- \u includes the username of the current user
- \h includes the host name the user is working from up to to the first period '
- \w includes the fully qualified path of the current working directory
- \n establishes a new line to make the results easy to read
#History Control
HISTCONTROL=ignore_space

#Set SPLUNK_HOME
for SPLUNK_HOME in "/Applications/Splunk" "/Applications/SplunkForwarder" "/opt/splunk" "/opt/splunkforwarder" "/Applications/SplunkBeta" "/Applications/SplunkForwarderBeta" "/${HOME}/splunkforwarder"; do
  if [ -d ${SPLUNK_HOME} ]; then
    break
  fi
done
if [ "${SPLUNK_HOME}" == "" ];then
  echo "WARNING: SPLUNK_HOME env variable undefined"
fi
export SPLUNK_HOME

#Add splunk to PATH
export PATH=$PATH:/usr/bin:$SPLUNK_HOME/bin

#Add btool to PATH
KERNEL=`uname -s`
case "x$KERNEL" in
  "xLinux")
    if [ -f /etc/lsb-release-o -d /etc/lsb-release.d ];
    then
      export DISTRO=$(lsb_release -i | cut -d: -f2 | sed s/'^	'//)
    else
      export DISTRO=$(ls -d /etc/[A-Za-z]*_[rv][elr]* | grep -v "lsb" | cut -d'/' -f3 | cut -d'-' -f1 | cut -d'_' -f1)
    fi
    if [ "x$DISTRO" != "xUbuntu" ];then
      export LD_LIBRARY_PATH=$SPLUNK_HOME/lib
    fi
  ;;
  "xDarwin")
    export DYLD_LIBRARY_PATH=$SPLUNK_HOME/lib
  ;;
  *)
    echo "ERROR: Unable to set LIBRARY_PATH"
    exit 1
  esac
Simplify distribution of the shell profile using the deployment server

You can leverage these shell commands on all your deployment clients using the deployment server. You can push out the shell profile with an app that contains the shell script in a \texttt{bin} directory. To leverage the shell profile on each deployment client, add a line to the Unix account's existing shell profile to "source" (or load) your creation from its deployment target.

Append the following to the unix user's default profile:

\begin{verbatim}
#Load Splunk-specific profile
./path/to/splunk/etc/apps/my_unix_profile/bin/splunk_profile
\end{verbatim}

Welcome page best practices for a Splunk deployment

A welcome page is a default dashboard within a default app tailored to a specific role or user group that presents users with all the resources they need to get Splunking. A welcome page is a good way to focus users' attention on workflows that are relevant to them. You can create your own welcome pages from scratch, or use the Welcome Page Creator app on Splunkbase to help you quickly implement welcome pages that accelerate your users' time to effectiveness.

\textbf{Audience}

- Program manager
- Engineer

For more about these roles, see \texttt{Roles best practices}.
Guidelines for creating welcome pages

Use the Welcome Page Creator app from Splunkbase
- The Welcome Page Creator provides example dashboards, template workspaces, and more than 20 prebuilt panels to get your welcome page started. To download the Welcome Page Creator app, go to Welcome Page Creator for Splunk on Splunkbase.

Keep it simple
- Keep content minimal and focused on relevant tasks and information. Less content makes it easier for users to focus on what is important so they can use the platform more effectively.

Consider the audience
- The focus and communication style of your developers is dramatically different than that of your business users. Keep this in mind when selecting what panels to include on a welcome page. Use materials and language that is effective for the reader. In fact, that is why some of the panels in the Welcome Page Creator have similar content, but presented differently.

Tailor welcome pages to different roles
- Assign a unique role to each work group so you can segregate their workspaces and provide the most effective welcome page for each team. You can also change the permissions on other team’s apps so users only see the workspace (app) for them ? not apps that belong to other groups. For more about assigning access to apps for specific roles, see Restricting access to specific apps in the Securing Splunk Enterprise manual.

Listen to questions from your users
- After implementing welcome pages, users will still have questions. Put answers to repeated questions in a new panel on your welcome page so you can get back to the fun stuff. When necessary, edit any shared prebuilt panel or convert any panel to an inline panel so you can further customize it for that audience. For more about working with dashboard panels, see Add panels to dashboards in the Splunk Enterprise Dashboards and Visualizations manual.

Not a welcome app, but a welcome page within the team workspace app
- The Welcome Page Creator app is deliberately focused on creating welcome pages, and is not intended as a place to start a Splunk experience. The Welcome Page Creator’s panels are shared globally so your end users can continue to work within their workspace (app).
more about setting up an app as a team workspace, see Workspace best practices.

Reuse content with prebuilt panels and cloning knowledge objects
Prebuilt panels enable you to share content across dashboards. The Welcome Page Creator comes with many prebuilt panels. You can make your own prebuilt panels to create welcome pages for different teams. For more about using prebuilt panels, see Add prebuilt panels to a dashboard in the *Splunk Enterprise Dashboards and Visualizations* manual.

Cloning knowledge objects is another way to share content with different teams and audiences. For more about cloning knowledge objects, see Moving or cloning a knowledge object in the *Splunk Enterprise Knowledge Manager Manual*.

**Guidelines for creating a welcome page from scratch**

Here are the high-level steps to create a welcome page from scratch. As an alternative, you can use the instructions on the Welcome Page Creator’s Splunkbase page to accelerate this process.

1. Identify a role or group that is the target audience for the welcome page
   Identify the needs and objectives of that role or group, and design content that will meet those needs and answer their questions.

2. Create a welcome page
   Populate the dashboard with content that answers those common questions from the users. For more about creating a dashboard, see Create dashboards and panels in the *Splunk Enterprise Search Tutorial*.

3. Set the role or group’s default app
   Ideally, this would be an app tailored for that role or team. For more about default apps and how to provision them for certain user roles, see Configure Splunk Web to open directly to an app in the *Splunk Enterprise Admin Manual*. For more about apps as team workspaces, see Workspace best practices.

4. Set the default app’s default dashboard to your welcome page
   For more about default dashboards, see Add navigation to a Splunk app in *Splunk Dev*.

5. Validate your welcome page
Ask a member of the target team or role to log in and discover features using your welcome dashboard.

**Workspace best practices for a Splunk deployment**

A workspace is a Splunk app tailored to a specific role or user group that enables users to search, explore, and create without distractions from other teams and users.

In many Splunk deployments, all users work in the default Search and Reporting app. This can cause some users and teams to struggle as they sort through dashboards, saved searches and knowledge objects from other teams to figure out what to use. It can also stifle adoption and discovery, as users may feel intimidated about participating or concerned they may break knowledge objects created by other teams. The volume and diversity of knowledge objects can also have a negative effect on overall search performance.

Creating an app as a workspace for each role or user group helps teams focus on what is important to them without getting distracted by content from other teams. Workspaces also facilitate co-working, and faster learning and adoption. When teams have their own workspaces, users can feel more confident, discover more, and collaborate more.

Creating workspace apps can also help improve search performance. You can set permissions to limit searches to the knowledge objects of their workspace, and not others.

**Audience**

- Architect
- Engineer
- Program manager

For more about these roles, see Roles best practices.

**Guidelines for implementing a team workspace app**

Create a new app

Create a new app using the barebones template or the workspace template provided by the Welcome Page Creator on Splunkbase. For more information about creating an app, see Create a new app in Splunk.
Dev. To download the Welcome Page Creator app, go to the Welcome Page Creator for Splunk on Splunkbase.

Make the app the default app
Set the new app as the default app for a given role. See Add or edit a role and About configuring role-based user access in the Splunk Enterprise Securing Splunk Enterprise Manual.

Reuse knowledge objects
Clone or move knowledge objects to share content with different teams and audiences. For more about cloning knowledge objects, see Moving or cloning a knowledge object in the Splunk Enterprise Knowledge Manager Manual.

Assign permissions and restrict access
Assign permissions for users to access their team workspace, and restrict access to unrelated users and groups by removing their read permissions. For more information about managing permissions to knowledge objects, see Manage knowledge object permissions in the Splunk Enterprise Knowledge Manager Manual.

Add a welcome page to your team workspace
You can add a welcome page to your team workspace app to provide welcoming direction to a user to help them get started and answer common questions. For more about adding a welcome page, see Welcome page best practices.