* blocked — can't proceed with solving the issue right now due to external factors
* needs review — most of the work and testing is done, the issue is in the "review" column in the project
* needs testing — most of the work is done, but needs testing. also in the "testing" column in the project
* needs work — needs some improvements (usually after testing and/or review). basically the issue is still in progress, but this is to show that it has completed at least one test-review cycle
* on hold — currently not actively worked on, mostly due to other, more critical issues taking priority
* question — meant for discussion threads
* migration-blocking — specifies PRs that need some manual work before deployment
* issuelist — actual work is done with smaller issues, but overall progress is marked in this generic issuelist using markdown checkboxes.
* wontfix — issues currently not viable to solve or out of scope of the project
* We also use issues for non-development tasks and discussion threads, so it makes sense to have the following columns in our Project: TODO (development), TODO (non-development), In progress, Needs testing, Needs review, Done. Issues move from the TODO columns to the right.
* We also use the Github issue references and links pretty extensively: All issues mention other related issues in their description (See #123 etc.). Pull requests always mention the issue they close (Closes #123) and the branches also have the issue number as the prefix (123\_new\_api\_feature). This way it's always very easy to find all related issues, discussion threads, pull requests and branches.

***Accepted***

Issues with this label are issues that the core team has accepted on to the roadmap. The core team focuses on accepted bugs, features, and improvements that are on our immediate roadmap and will give priority to these issues. Pull requests could be delayed or closed if the pull request doesn’t align with our current roadmap. An issue or pull request that has not been accepted should either eventually move to an accepted state, or should be closed. As an issue is accepted, we will find room for it on our roadmap, either on an upcoming release (point release milestones), or on a future milestone project (named milestones).

***Bug***

An issue describing unexpected or malicious behaviour of the readthedocs.org software. A Bug issue differs from an Improvement issue in that Bug issues are given priority on our roadmap. On release, these issues generally only warrant incrementing the patch level version.

***Design***

Issues related to the UI of the readthedocs.org website.

***Feature***

Issues that describe new features. Issues that do not describe new features, such as code cleanup or fixes that are not related to a bug, should probably be given the Improvement label instead. On release, issues with the Feature label warrant at least a minor version increase.

***Good First Issue***

This label marks issues that are easy to get started with. The issue should be ideal for beginners to dive into the code base.

***Priority: high***

Issues with this label should be resolved as quickly as possible.

***Priority: low***

Issues with this label won’t have the immediate focus of the core team.

***Improvement***

An issue with this label is not a Bug nor a Feature. Code cleanup or small changes to existing features would likely have this label. The distinction for this label is that these issues have a lower priority on our roadmap compared to issues labeled Bug, and aren’t implementing new features, such as a Feature issue might.

***Needed: design decision***

Issues that need a design decision are blocked for development until a project leader clarifies the way in which the issue should be approached.

***Needed: documentation***

If an issue involves creating or refining documentation, this label will be assigned.

***Needed: more information***

This label indicates that a reply with more information is required from the bug reporter. If no response is given by the reporter, the issue is considered invalid after 2 weeks and will be closed. See the documentation about our [triage process](https://docs.readthedocs.io/en/stable/contribute.html#triage-not-enough-information) for more information.

***Needed: patch***

This label indicates that a patch is required in order to resolve the issue. A fix should be proposed via a pull request on GitHub.

***Needed: tests***

This label indicates that a better test coverage is required to resolve the issue. New tests should be proposed via a pull request on GitHub.

***Needed: replication***

This label indicates that a bug has been reported, but has not been successfully replicated by another user or contributor yet.

***Operations***

Issues that require changes in the server infrastructure.

***PR: work in progress***

Pull Requests that are not complete yet. A final review is not possible yet, but every Pull Request is open for discussion.

***PR: hotfix***

Pull request was applied directly to production after a release. These pull requests still need review to be merged into the next release.

***Sprintable***

Sprintable are all issues that have the right amount of scope to be handled during a sprint. They are very focused and encapsulated.

***Status: blocked***

The issue cannot be resolved until some other issue has been closed. See the issue’s log for which issue is blocking this issue.

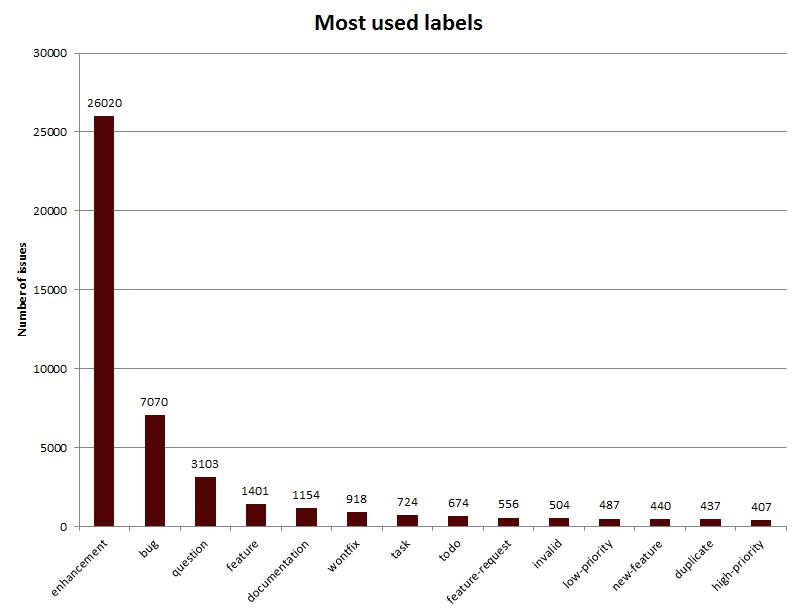
***Status: stale***

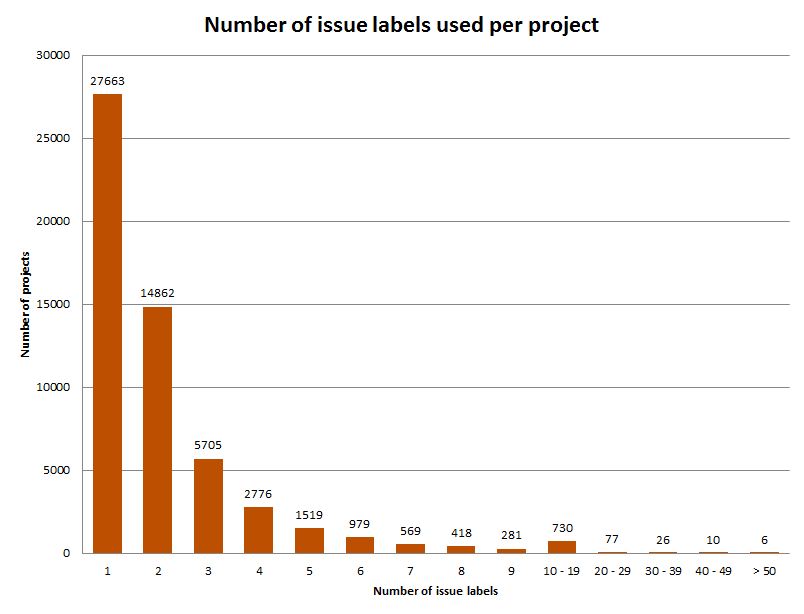
A issue is stale if it there has been no activity on it for 90 days. Once a issue is determined to be stale, it will be closed after 2 weeks unless there is activity on the issue.

***Support***

Questions that needs answering but do not require code changes or issues that only require a one time action on the server will have this label. See the documentation about our [triage process](https://docs.readthedocs.io/en/stable/contribute.html#triage-support-tickets) for more information.









Most issues will have labels for at least one of the following:

* Type: ~feature, ~bug, ~tooling, ~documentation, etc.
* Stage: ~"devops::plan", ~"devops::create", etc.
* Group: ~"group::source code", ~"group::knowledge", ~"group::editor", etc.
* Category: ~"Category:Code Analytics", ~"Category:DevOps Reports", ~"Category:Templates", etc.
* Feature: ~wiki, ~ldap, ~api, ~issues, ~"merge requests", etc.
* Department: ~UX, ~Quality
* Team: ~"Technical Writing", ~Delivery
* Specialization: ~frontend, ~backend, ~documentation
* Release Scoping: ~Deliverable, ~Stretch, ~"Next Patch Release"
* Priority: ~"priority::1", ~"priority::2", ~"priority::3", ~"priority::4"
* Severity: ~"severity::1", ~"severity::2", ~"severity::3", ~"severity::4"

1. LABELS

Labels are used to sort and describe issues and pull requests. Some labels are usually reserved for one or the other, though most labels may be applied to both.

New issues will receive at least one label and a milestone, and new pull requests will receive at least one label. Except for the [functional area](https://docs.saltproject.io/en/latest/topics/development/labels.html#functional-area-labels) and [functional group](https://docs.saltproject.io/en/latest/topics/development/labels.html#functional-group-labels) label categories, issues will generally receive only up to one label per category.

1. TYPE

Issues are categorized into one of several types. Type labels are almost never used for pull requests. GitHub treats pull requests like issues in many ways, so a pull request could be considered an issue with an implicit Pull Request type label applied.

* Feature - The issue is a request for new functionality including changes, enhancements, refactors, etc.
* Bug - The issue documents broken, incorrect, or confusing behavior. This label is always accompanied by a [severity label](https://docs.saltproject.io/en/latest/topics/development/labels.html#bug-severity-labels).
* Duplicate - The issue is a duplicate of another feature request or bug report.
* Upstream Bug - The issue is a result of an upstream issue.
* Question - The issue is more of a question than a request for new features or a report of broken features, but can sometimes lead to further discussion or changes of confusing or incongruous behavior or documentation.
* Expected Behavior - The issue is a bug report of intended functionality.

1. PRIORITY

An issue's priority is relative to its [functional area](https://docs.saltproject.io/en/latest/topics/development/labels.html#functional-area-labels). If a bug report, for example, about gitfs indicates that all users of gitfs will encounter this bug, then a P1 label will be applied, even though users who are not using gitfs will not encounter the bug. If a feature is requested by many users, it may be given a high priority.

* P1 - The issue will be seen by all users.
* P2 - The issue will be seen by most users.
* P3 - The issue will be seen by about half of users.
* P4 - The issue will not be seen by most users. Usually the issue is a very specific use case or corner case.

1. SEVERITY

Severity labels are almost always only applied to issues labeled Bug.

* Blocker - The issue is blocking an impending release.
* Critical - The issue causes data loss, crashes or hangs salt processes, makes the system unresponsive, etc.
* High Severity - The issue reports incorrect functionality, bad functionality, a confusing user experience, etc.
* Medium Severity - The issue reports cosmetic items, formatting, spelling, colors, etc.

1. FUNCTIONAL AREA

Many major components of Salt have corresponding GitHub labels. These labels are applied to all issues and pull requests as is reasonably appropriate. They are useful in organizing issues and pull requests according to the source code relevant to issues or the source code changed by pull requests.

* Execution Module
* File Servers
* Grains
* Multi-Master
* Packaging Related to packaging of Salt, not Salt's support for package management.
* Pillar
* RAET
* Returners
* Runners
* SPM
* Salt-API
* Salt-Cloud
* Salt-SSH
* Salt-Syndic
* State Module
* Tests
* Transport
* Windows
* ZMQ

1. FUNCTIONAL GROUP

These labels sort issues and pull requests according to the internal SaltStack engineering teams.

* Core - The issue or pull request relates to code that is central or existential to Salt itself.
* Platform - The issue or pull request relates to support and integration with various platforms like traditional operating systems as well as containers, platform-based utilities like filesystems, command schedulers, etc., and system-based applications like webservers, databases, etc.
* RIoT - The issue or pull request relates to support and integration with various abstract systems like cloud providers, hypervisors, API-based services, etc.
* Console - The issue or pull request relates to the SaltStack enterprise console.
* Documentation - The issue or pull request relates to documentation.

1. STATUS

Status labels are used to define and track the state of issues and pull requests. Not all potential statuses correspond to a label, but some statuses are common enough that labels have been created for them. If an issue has not been moved beyond the Blocked milestone, it is very likely that it will only have a status label.

* Cannot Reproduce - The issue is a bug and has been reviewed by a SaltStack engineer, but it cannot be replicated with the provided information and context. Those involved with the bug will need to work through additional ideas until the bug can be isolated and verified.
* Confirmed - The issue is a bug and has been confirmed by a SaltStack engineer, who often documents a minimal working example that reproduces the bug.
* Fixed Pending Verification - The issue is a bug and has been fixed by one or more pull requests, which should link to the issue. Closure of the issue is contingent upon confirmation of resolution from the submitter. If the submitter reports a negative confirmation, this label is removed. If no response is given after a few weeks, then the issue will be assumed fixed and closed.
* Info Needed - The issue needs more information before it can be verified and resolved. For a feature request this may include a description of the use cases. Almost all bug reports need to include at least the versions of salt and its dependencies, the system type and version, commands used, debug logs, error messages, and relevant configs.
* Pending Changes - The pull request needs additional changes before it can be merged.
* Pending Discussion - The issue or pull request needs more discussion before it can be closed or merged. The status of the issue or pull request is not clear or apparent enough for definite action to be taken, or additional input from SaltStack, the submitter, or another party has been requested.

If the issue is not a pull request, once the discussion has arrived at a cogent conclusion, this label will be removed and the issue will be accepted. If it is a pull request, the results of the discussion may require additional changes and thus, a Pending Changes label.

* won't-fix - The issue is legitimate, but it is not something the Salt core team is currently able or willing to fix or implement. Issues having this label may be revisited in the future, or solved by a Salt community member.

1. TEST STATUS

These labels relate to the status of the automated tests that run on pull requests.

* has-failing-test - The PR currently has one or more failing tests that prevent the PR from being merged.
* Needs Testcase - The PR has code changes, but lack any automated tests. These PRs need automated tests written before they may be merged.

1. OTHER

These labels indicate miscellaneous issue types or statuses that are common or important enough to be tracked and sorted with labels.

* Awesome - The pull request implements an especially well crafted solution, or a very difficult but necessary change.
* Help Wanted - The issue appears to have a simple solution. Issues having this label should be a good starting place for new contributors to Salt.
* Regression - The issue is a bug that breaks functionality known to work in previous releases.
* Story - The issue is used by a SaltStack engineer to track progress on multiple related issues in a single place.
* Stretch - The issue is an optional goal for the current sprint but may not be delivered.
* ZD - The issue is related to a Zendesk customer support ticket.
* <Release> - The issue is scheduled to be implemented by <Release>. See [here](https://docs.saltproject.io/en/latest/topics/releases/version_numbers.html#version-numbers) for a discussion of Salt's release codenames.

## Label Groups

We group labels by color, according to broad themes. Labels are consistent across repositories, except for a few language specific topics. This makes switching between projects easy, since you don't need domain expertise in order to write an issue. **New team members can learn the system once, and use it everywhere.**

### **Platform**

If the repository covers multiple parts, this is how we designate where the issue lives. (i.e. iOS and Android for cross-platform tablet app).

### **Problems**

Issues that make the product feel broken. High priority, especially if its present in production.

### **Mindless**

Converting measurements, reorganizing folder structure, and other necessary (but less impactful) tasks.

### **Experience**

Affect user’s comprehension, or overall enjoyment of the product. These can be both opportunities and “UX bugs”.

### **Environment**

Server environment. With good QA, you’ll identify issues on test and staging deployments.

### **Feedback**

Requires further conversation to figure out the action steps. Most feature ideas start here.

### **Improvements**

Iterations on existing features or infrastructure. Generally these update speed, or improve the quality of results. Adding a new “Owner” field to an existing “Calendar” model in the API, for example.

### **Additions**

Brand new functionality. New pages, workflows, endpoints, etc.

### **Pending**

Taking action, but need a few things to happen first. A feature that needs dependencies merged, or a bug that needs further data.

### **Inactive**

No action needed or possible. The issue is either fixed, addressed better by other issues, or just out of product scope.See a [version of this in action](https://github.com/robinpowered/swolebot/labels) on one of our public repos. Want to see what we're building? Check out our [product features](https://robinpowered.com/features).

<https://gitlab.com/gitlab-org/gitlab/-/labels>

<https://docs.gitlab.com/ee/development/contributing/issue_workflow.html>

<https://medium.com/@dave_lunny/sane-github-labels-c5d2e6004b63>

<https://docs.saltproject.io/en/latest/topics/development/labels.html>

<https://robinpowered.com/blog/best-practice-system-for-organizing-and-tagging-github-issues>

[2 open issues and pull requests](https://github.com/robinpowered/swolebot/issues?q=label%3A%22bug%22+is%3Aopen)

[chore](https://github.com/robinpowered/swolebot/labels/chore)

[copy](https://github.com/robinpowered/swolebot/labels/copy)

[2 open issues and pull requests](https://github.com/robinpowered/swolebot/issues?q=label%3A%22copy%22+is%3Aopen)

[discussion](https://github.com/robinpowered/swolebot/labels/discussion)

[4 open issues and pull requests](https://github.com/robinpowered/swolebot/issues?q=label%3A%22discussion%22+is%3Aopen)

[duplicate](https://github.com/robinpowered/swolebot/labels/duplicate)

[enhancement](https://github.com/robinpowered/swolebot/labels/enhancement)

[4 open issues and pull requests](https://github.com/robinpowered/swolebot/issues?q=label%3A%22enhancement%22+is%3Aopen)

[feature](https://github.com/robinpowered/swolebot/labels/feature)

[1 open issue or pull request](https://github.com/robinpowered/swolebot/issues?q=label%3A%22feature%22+is%3Aopen)

[in progress](https://github.com/robinpowered/swolebot/labels/in%20progress)

[invalid](https://github.com/robinpowered/swolebot/labels/invalid)

[optimization](https://github.com/robinpowered/swolebot/labels/optimization)

[question](https://github.com/robinpowered/swolebot/labels/question)

[rfc](https://github.com/robinpowered/swolebot/labels/rfc)

[security](https://github.com/robinpowered/swolebot/labels/security)

[ux](https://github.com/robinpowered/swolebot/labels/ux)

[watchlist](https://github.com/robinpowered/swolebot/labels/watchlist)

[wontfix](https://github.com/robinpowered/swolebot/labels/wontfix)