

UI/Design System Guidelines

A set of guidelines to follow when creating or updating elements in the Cloud Console UI.

This is a living document. Expect that as issues come up outside this document or as comments are added, the contents of this document will be edited - this is meant to be an ongoing resource.

Two ways to use this document:

1. New to the Pulumi Cloud? Give it a full read through to understand what our typical patterns are for common UX situations and UI elements
2. Making a UI change? Search for the section specifically relevant to your changes to make sure you're aligning with our norms.

Following these guidelines helps create consistency in our UI, which helps our users. Like anything, there are exceptions to every rule - if following these guidelines for a specific use case doesn't make sense, bring it up with the immediate team working in that area and come to a shared consensus and document the decision.

Buttons

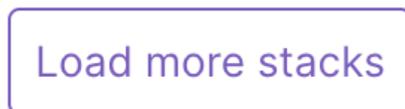
Variants

Use a button variant that matches your needs, as described by the below:

- **Primary:** Use only one Primary button on a single view. This should be the main action we expect or want to encourage users to take on this page.



- **Secondary:** An alternative to the primary action



- **Tertiary:** Additional actions users may need to take that are less common and thus need less visual emphasis

Cancel

Cancel

- **Destructive:** An action that would be very difficult or impossible for the user to come back to the original state from (often, deleting something).

Delete account kimberleyamackenzie

States

- **Disabled:** Provide clear indication of why the button is disabled via tooltip or description.

Delete organization

Unable to delete an organization with an active subscription. Visit your [Billing & usage page](#) to update your subscription.

Links

Variants

- **External link:** If a link is expected to open a new tab and take the user out of the Pulumi Cloud, use our `appExternalLink` directive, which will add a handy icon, the open in a new page behavior, as well as standard link styles for you. We should always let the user know visually that they will be taken somewhere new.

Additional examples can be found in the [documentation](#) 

- **Link styled as a button:** We use this sparingly, but there may be some instances where you have a series of buttons (dialog actions, for example)

Documentation 

Close

General

- Links should not include ending punctuation. For example in “We love Pulumi!”, a link would wrap “We love Pulumi” and “!” would be non-interactive.

Dialogs

Size

- Dialogs should adhere to our 3 standard widths, available as consts in the codebase:
Large: **872px**
Medium: **752px**
Small: **632px**
- The length of each dialog will vary based on contents but should be at least **400px** long.

Titles

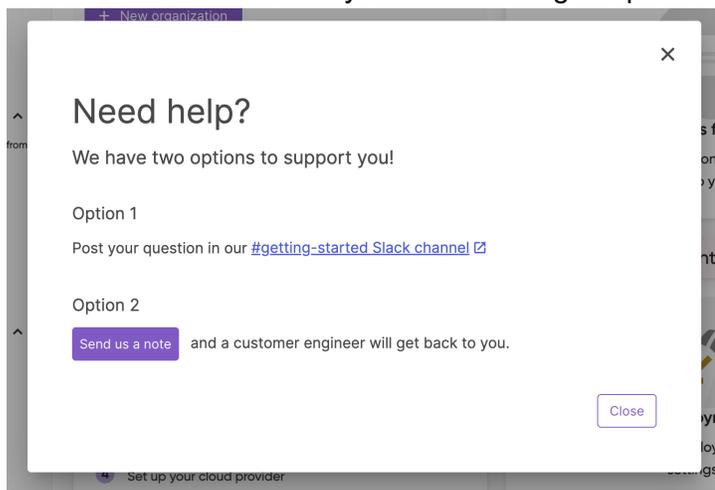
- Dialog titles should be left aligned. If there is an icon, the icon should appear to the left of the title. Titles should be in the Gilroy font family, font size 20, with a color of gray 900.

Choose your policy pack

Add a new policy pack to your organization using one of our policy pack templates.

Closing

- Every dialog should have two methods of closing: A tertiary “Cancel” button and an “X” icon in the upper right corner. The text of the button should always be “Cancel” to make it clear to the user that they are abandoning the process.



- Dialogs should have the behavior of closing on clicking outside the dialog (the gray-ed out background) enabled as well. Disabling this behavior is inaccessible.

Action buttons

- Buttons should be aligned on the **right** side of the dialog.
- The primary action button should always appear to the right of the tertiary “Cancel” button.

 Choose your policy pack

Add a new policy pack to your organization using one of our policy pack templates.

You can create an empty project and write your own policies, or use one of the pre-built compliance-ready policy templates to select from a subset of hundreds of existing policies for common compliance frameworks including PCI DSS, ISO27001, and CIS across a wide variety of cloud providers.

Policy pack

Choose a policy pack ▼

Cancel Next

Cards vs Expansion Panels

When to use a card

- “Card styling” in our UI refers to using a border around a section of content. When the console was first designed, we relied on this heavily. In the following years, we have started to move away from this styling. Current guidance is: if a page already uses card styling, and you are adding a new section of content, continue to use card styling. However, if you are creating a new page or working on a page without card styling, do not add new usages. Over time, we are working to move off card styling completely.

About Policy

Policy empowers you to set guardrails to enforce compliance for resources so developers within an organization can provision their own infrastructure while sticking to best practices and security compliance.

Organization admins can create **Policy Groups** that associate stacks with **Policy Packs**. Policy Packs are run on each stack update, with any violation blocking that update from proceeding.

[Policy Configuration Guide](#)

[CrossGuard Guide](#)

Policy Packs

Policy Packs must be published to your organization by an administrator. For more information on authoring Policy Packs, visit the [pulumi-policy repo](#)

+ New policy pack

Name	Latest Version
aws-iso27001-compliance-ready-policies-typescript	0.0.1
aws-typescript	0.0.1
continuous-policy	0.0.3
pulumi-internal-policies	0.0.6
pulumi-vanta-policies	0.0.1

Page sections

- This is the preferred alternative to card styling.

Billing & usage

Subscription

Status	Active
Edition	Business Critical Edition
Duration	Annual
Free Pulumi Credits	868m of 0
Extra Pulumi Credits Cost	\$0
Included Pulumi Deploy Minutes Used	47313 of 0
Extra Deploy Minutes Cost	\$0.01
Next payment	June 16, 2043 \$0.00
Support	Gold (24 x 7 portal + email + Slack support included), contact support

Change editions

Usage history

Resource Count

TIME RANGE

Last month

Pulumi credit consumption over time.

Date and Time formatting

Timezone

- All times should be presented in the user's local timezone
 - There may be very limited exceptions to this, like the Schedules API which uses cron syntax and is based on UTC. If you believe a certain area of the UI should also be an exception, work with Product and UX to determine the use case.

Date formatting

- When presenting an absolute date (versus a relative time, eg "3 days ago"), the date should be formatted as "Day of week, Month, Day, Year". Example: "Tuesday, June 6th, 2024"

Tooltips

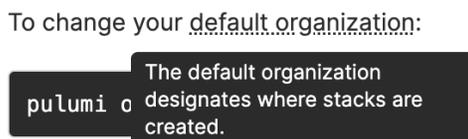
General

- Don't duplicate information already on the screen via tooltip

Definitions of specific terms

- When needing to include long descriptions of what particular things are. Opt for a tooltip, and use a dotted line under the text in which hovering on will display the tooltip. The text with the dotted underline should be the same color as surrounding text.

To change your default organization:



Alerts

- Use a tooltip in combination with an info icon as a trigger when adding extra information that gives context.

Resources:  57,587  

Resources:  57,587  

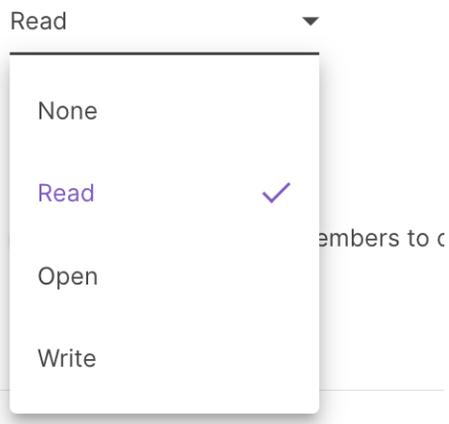
Only 10,000 results are visible here. Please use the API to see more than 10,000 results.

Capitalization and casing

General

- Use sentence case unless using a proper feature name. Examples: “Account settings” (non-feature) or “Organization admins can create Policy Groups that associate stacks with Policy Packs” (where “Policy Groups” and “Policy Packs” are both features).
- Permissions items in lists (see below example) should also be in sentence case, not all caps. We previously styled some of these items in all caps, and we have moved away from that.

Environment permissions



Empty states

General

- All lists and tables should have an empty state that includes either an action button to populate an item, or a clear directive about how to populate the list.



Try creating a new project using Pulumi AI

Create a template from natural language prompts and Deploy with Pulumi in seconds.

Create AI Template



No results found

Suggestions:

- Check for typos and spelling errors
- Try more general search terms
- Try different search terms



No activity yet!

Try creating a project. We'll keep a log of your latest activity here.

+ Create Project

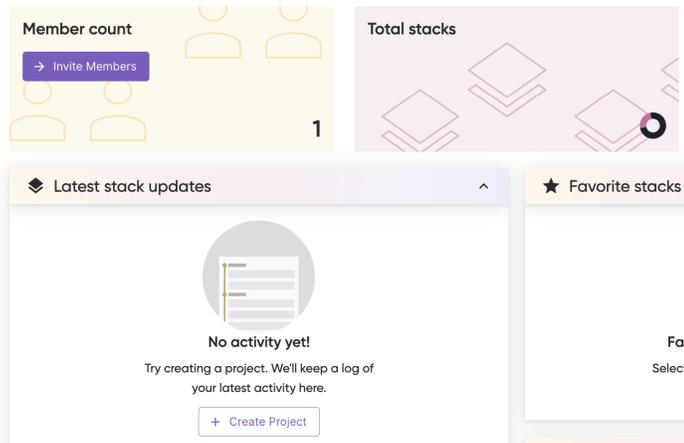
Loading states

General

- We should always visually indicate when something on the screen is loading. If the user can access some information on the screen in a useful way while other information is loading, you should use a partial loading state. If the page is essentially unhelpful or unusable without the loading data, you should use the full screen loading state.

Partial loading

- Use this state when the page still has helpful information available while it waits for other data to load.



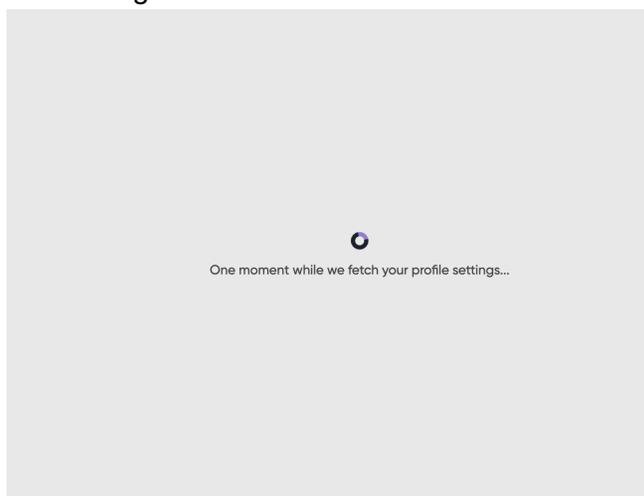
Loading after button interaction

- If a partial load state needs to occur after a button selection, you should show a loading state on that button.



Full screen loading

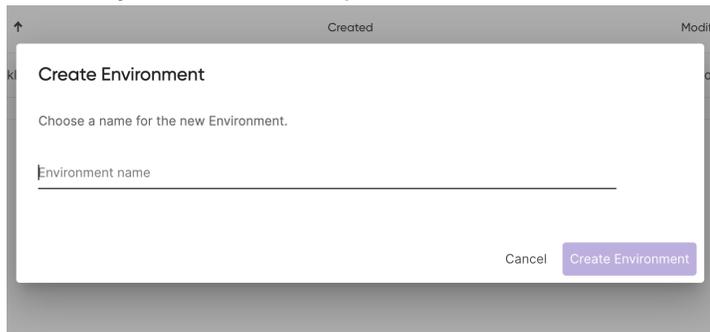
- Use this when the page is either non-interactive or lacking essential information without the loading data.



Actions

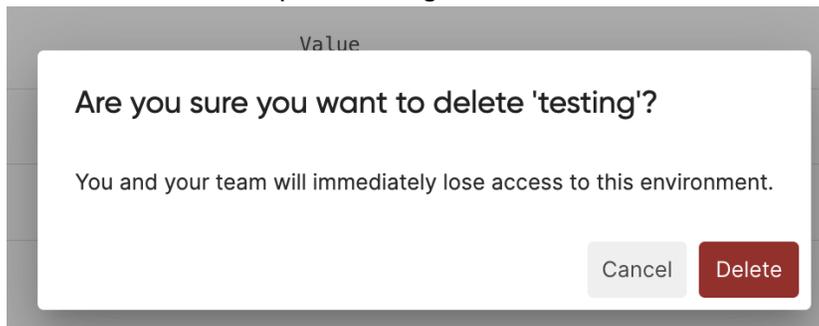
Create

- All “creates” should open a dialog to complete the action (i.e - “Create webwork”, “Create team”... etc.) There may be exceptions to this rule at the discretion of the UX and Product teams, based on the specific scenarios - some may be too complex for a dialog (creating a webhook is a current example of this). Consult with the UX and Product teams if you think an exception should be made.



Delete

- All “deletes” should open a dialog to confirm this destructive action



Typography

Page headings

- Use font family Gilroy, size 24px, weight 500, and color Gray 900.

Create a new project

Choose your source

Choose your cloud

All templates

Choose a cloud

Search templates



No results found

Suggestions:

OR



Try creating

Create a template

Section headings

- Use font family Gilroy, size 20px, weight 500, and color Gray 900.

Defaults

- Default body or paragraph text should be font family Inter, weight 400, size 14px and line height 21px.

Alignment

- By default, we left align most text unless a design specifically calls for center alignment.

Line height

- Line height for text should be the automatic line height given for that particular size unless the text is a paragraph. In the case of a paragraph, the line height should be 1.5 times the size of the font. For example, if the font size is 16pt, the line height within a paragraph should be 24pt.

Code

- Use our “code” component whenever adding code into the UI. This component adds backgrounds for contrast, uses our defined code fonts, supports a copy button, and supports both single lines and blocks..

```
brew install pulumi/tap/pulumi
```

```
# Pulumi.yaml  
echo "name: great-project" > Pulumi.yaml  
echo "runtime: nodejs" >> Pulumi.yaml  
  
# Pulumi.greatStack.yaml  
pulumi stack select greatStack  
pulumi config refresh
```

- If the code is meant to be copied (such as a command for use in the CLI), use the copy command button component.

```
brew install pulumi/tap/pulumi 
```

On screen text

General

- Keep on page descriptions short and focused on only the information necessary for users to trust the system and be successful.
- Favor “and” over short-hand like “+” or “&” unless it is part of a feature name.

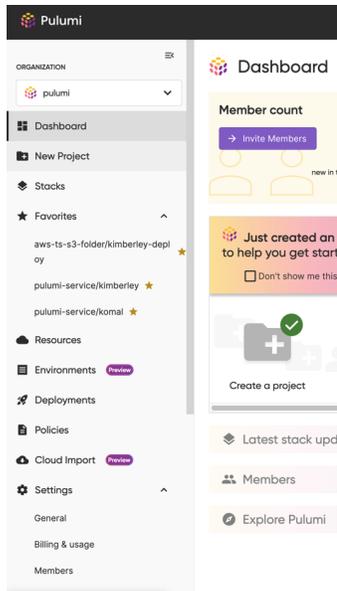
Navigation

General

- Note that all levels of console navigation may appear different to different users: we hide or show navigation items based on current organization subscription and user role (“admin” versus “member” versus “billing manager”).

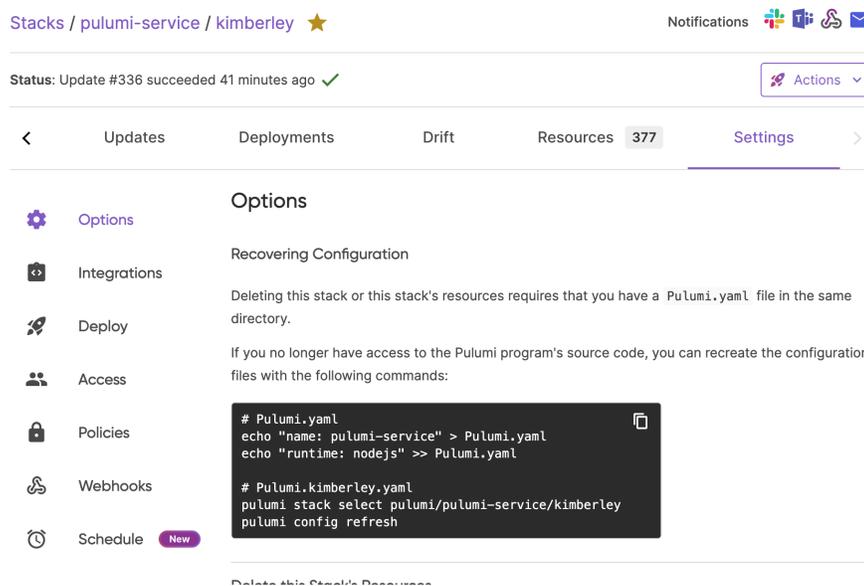
Main console navigation

- This is our primary app-wide navigation, and appears on the left side of the screen. It appears on nearly all Console pages, and can be in a collapsed or expanded format. Items here should be top-level features we expect users will want to navigate to quickly and often.



Vertical sub-navigation

- Used on pages where there is already a horizontal sub-navigation. Consult with the UX and Product teams if you think there is a case to add a new vertical sub-nav.



Horizontal sub-navigation (tabs)

- Used on pages that need further navigation than what is available via the primary Console navigation.

Status: Update #336 succeeded 40 minutes ago ✓

Actions ▾

< Overview README Updates Deployments Drift Re >

 **Update #336 succeeded** in 41 minutes, 39 seconds +1 -22 ~30 341
kimberley-pulumi-corp updated an hour ago ✓

Cleanup a lot of config already defaulted by config.ts (#20140) <> 5005f1c [Details](#)
🔗 master
🔗 #20140
🗑️

Breadcrumb navigation

- Used on pages to navigate between a list of items (eg. Environments) to single individual instances from that list (eg. My Environment).
- If there are multiple instances at the same level, there should be a caret that triggers a dropdown menu to navigate directly to those other instances. If there is only a single option (My Environment is the only environment I have defined), there is no menu functionality.
- The breadcrumbs visible that are links that will navigate to that page.
- Breadcrumb is not visible on the base list page. That page should use standard header styling.

Environments

Create Environment

Name ↑

Environments / vfsavfvs ▾

Editor

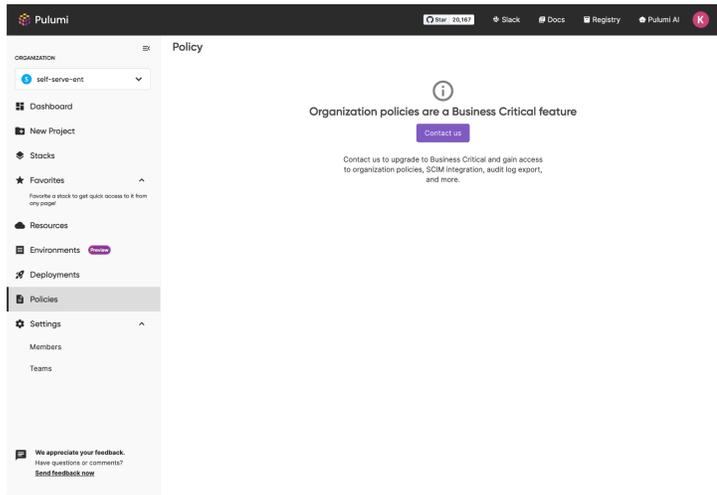
ddffds

Environments / vfsavfvs ▾

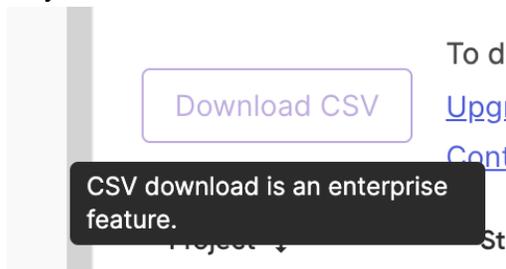
Feature highlighting and subscription gating

For features only available to certain subscription levels after trial period

- Some subscription-gated features are hidden from view if a customer does not have access (no links, page 404s). Others are available for navigation (may have a link in the console nav, may be accessible via direct navigation) in order to support an upsell. You should work with the engineering and product teams during development to determine the desired state for a specific feature.
- If a feature is accessible via navigation, and the feature is a whole page or a widget within a page, you should use the “Upgrade to unlock” component. That component will manage the text and actions relevant to the minimum product level.



- If the subscription-gated feature is an addition on a larger, fully enabled, page, you should disable the relevant actions with explanatory text about what is inaccessible and why.



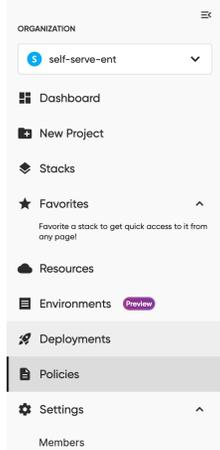
For features only available to certain subscription levels during trial period

- Use our “trial badge” component for features that users will lose access to after a trial if they don’t subscribe at a certain level because they are subscription gated.

Policy Packs Enterprise + Business Critical

For features in a preview state (Alpha, Beta, etc)

- Use a badge in the main left navigation item where applicable



For new features

- Use a badge in the main left navigation item where applicable, or on the primary navigation link if that is located within another page.

Environments / cxzcxz



To view the resolved value of this environment

Accessibility

General guidelines

- We should be striving for all our features to be accessible to all users. Accessibility on the web covers a wide range - when in doubt, a quick web search “is X accessible” can usually point you in the right direction.
- W3 is a great general resource as well as in-detail answers to specific elements: <https://www.w3.org/WAI/fundamentals/accessibility-intro/>
- [Our goal should be to achieve WCAG 2.2 conformance of AA as this is required for enterprise applications.](#)
 - [A \(Single A\) is viewed as the minimum level of requirement which all websites, apps, and electronic content such as documents should adhere to.](#)
 - [AA \(Double A\) is viewed as the acceptable level of accessibility for many online services, which should work with most assistive technology which is now widely](#)

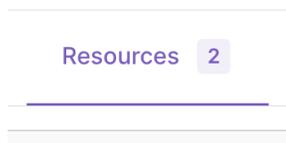
[available on both desktop and mobile devices, or which can be purchased as a third-party installation.](#)

- [AAA \(Triple A\) compliance is viewed as the gold standard level of accessibility, which provides everything for a complete accessible offering, including all the bells and whistles which make the difference between a very good experience and an excellent one. \(AAA is unobtainium - accessibility advocates essentially need to endorse the Pulumi service to meet AAA as it's as much a technical achievement as it is a marketing one\).](#)

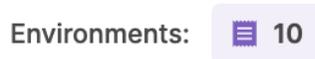
Showing item counts

General guidelines

- If you're showing item counts use our standard styles for the count. Counts are useful if a user would need to scroll, navigate to a new page, or expand in order to see a total number of items.



- For counts of items in a table, use the label and icon presentation.



Tables

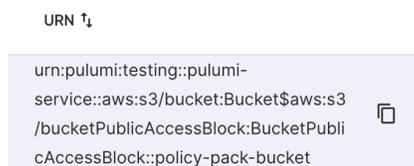
Actions column

- If each row in a table has actions that can be performed on it ("Remove member", "View details", etc), there should be an "Actions" column. This column should have a 3-dot menu that when expanded shows each option available to be performed. If a specific row does not have any actions available, the menu should be visible but disabled with explanatory text.



Copying values

- If we anticipate users frequently copying a specific value in a table, a copy button should be added directly next to the values being copied.



Sorting

- Any columns that the table can be sorted by should have sorting icons in their column header. These icons should indicate a current ascending sort, a current descending sort, or no sort on that column.

Stack ↑↓

Stack ↑

Stack ↓

Items that can be navigated to

- If an item in a table can be directly navigated to via select, the trigger should be styled as a link and should be an a tag - this ensures users can use standard keyboard navigation.

[uploads-bucket-acl](#)

Sorting and ordering

Default ordering of lists

- By default we sort all lists alphabetically, case-insensitive. For example, items “A, X, Y, a, b, c” would be sorted as “**A, a, b, c, X, Y.**” Depending on the context of the items you are sorting, you may also choose to have the default sort be time-based (eg, “last updated”).

Sorting on non-table lists

- In general, we do not support descending alphabetical sorting on non-table lists (lists of environments or stacks for example).
- Depending on what sorting we can support, we typically prompt for sorts of non-table lists via a select element on the page.

Stacks Project  Repository name 

Success, Error, Warning, and Information states

General

- We have three types of alerts: Global, Inline, and Floating (i.e., Toasts). All three types have four different categories: Information, Success, Warning, and Error. See correlating color info below. Each category has its own specialized Material icon representing info, success, warning, and error, which are also used within the rest of the Cloud. These icons, as well as a button component inside each alert, can be included or left out, depending on the desired message by UX/Product. Bold headings and/or text links can also be included. Buttons match the main color of each alert.

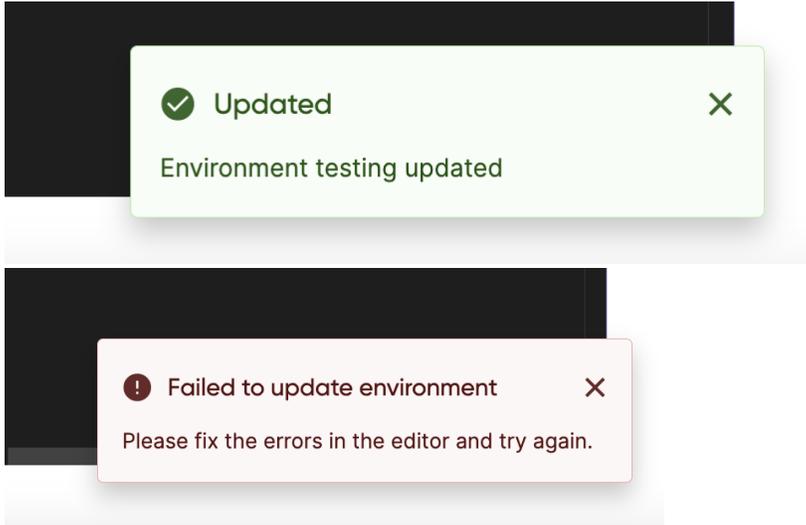
Colors

- You should use our defined palettes for each of these states, instead of a unique color:
 - **Error:** Red 700, #9e2626
 - **Success:** Green 600: #437E37
 - **Warning:** Orange 600: #e0771c
 - **Information:** Blue 600: #4d5bd9

Floating alerts (AKA toast notifications)

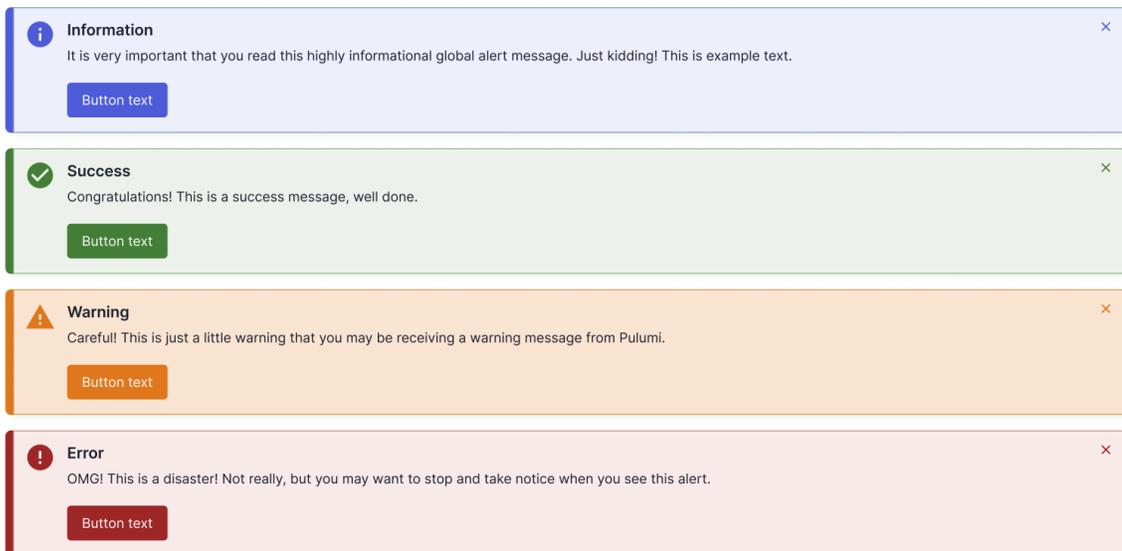
- On success or failure of an action, we should render a toast to inform the user of the status of their action. The exception to this is if the action takes the user to the newly

created item (example: creating a new environment navigates to that new environment, so the success is communicated without the toast).



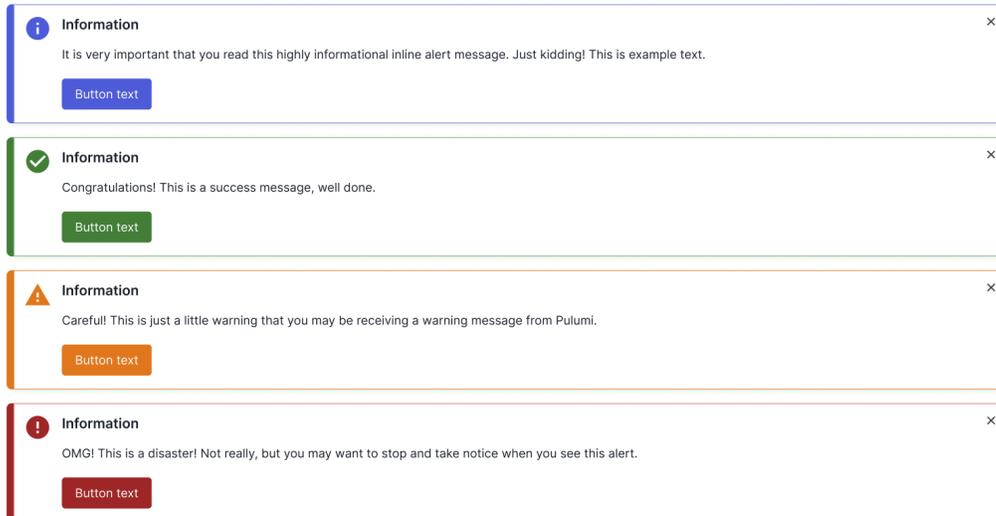
Global alerts

- Global alerts are used for messages that apply to the Cloud as a whole and appear at the top of a page. See other usage guidelines above in the General section.



Inline alerts

- Inline alerts can apply to a section of the site or appear in a dialog. See other usage guidelines above in the General section.



Avatars

General

- Users and organizations should be represented visually by avatars. These avatars draw from either a custom image uploaded by users, or in lieu of that, default to a random color paired with a letter representing the organization.
- If the avatar is used on its own, without accompanying text with the name,, hovering should show the organization or user name.



Update #1 succeeded in 0 seconds

kimberley-pulumi-corp updated a year ago ✓



[arun-nov623](#)



[arun-pulumi-corp-test](#)



[arun-noteam](#)

Colors

General

- We have a standard palette that includes our brand colors, as well as spectrums of colors for use in our Console. All colors should be taken from this palette. Any use of a color other than these should be discussed and defined with the UX and Product teams, and codified into a clear variable that defines its usages.
- You can find our palettes in Storybook as well as code.
- Note that the Marketing team brand colors are similar but slightly different than our Console colors. They are not to be used interchangeably.