Human Interface Guidelines

Table of Contents

OVERVIEW

What is a HIG?

The Human Interface Guidelines document was created for OpenStack designers in order to direct the creation of new OpenStack experiences and assist the evaluation of design blueprints.

> The goal for the OpenStack project is to create a single, unified product that is easy to use yet robust enough to support immense amounts of data and complexity.

Because the OpenStack project is open source, one of the highest priorities for the OpenStack design community is to maintain consistency across both the visual and interaction design of the OpenStack experience. With hundreds of contributors to the project, the possibility for fragmentation is great, but by using the HIG to guide the design process this fragmentation can be minimized.

The Principles section defines the high level priorities to be considered when designing for the OpenStack dashboard.

The Core Architecture section describes what types of screens are currently used and how these screens relate to one another.

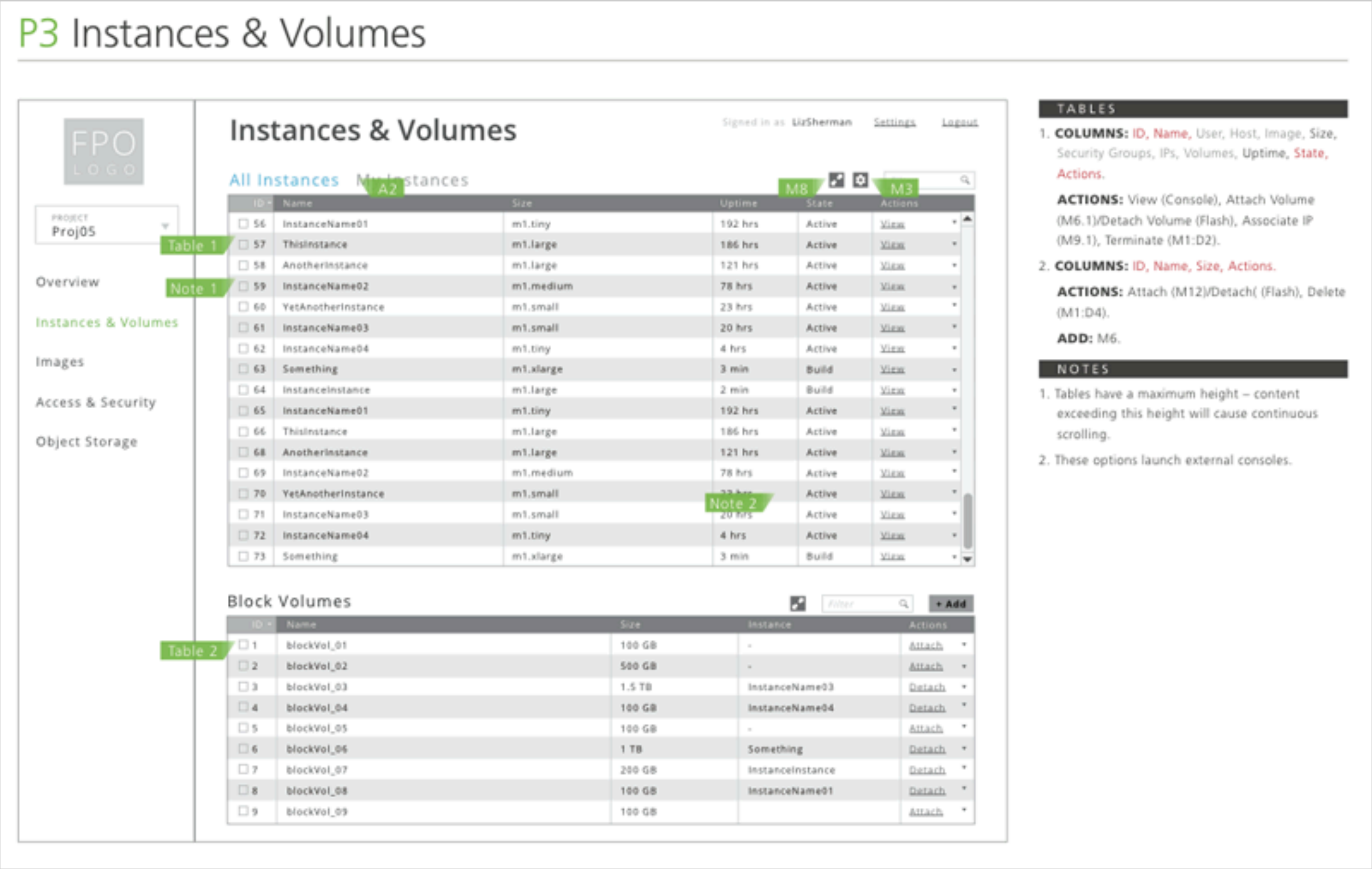
The Screen Examples section provides several wireframes to model some of the implementations of screens described in the Core Architecture section.

The Core Elements section describes the types of controls and content used in the OpenStack dashboard and how each should be implemented. This document focuses on the design of the OpenStack experience and will not discuss how to code for the OpenStack project.

What is a design blueprint?

Similar to the engineering blueprints used by the OpenStack community to document proposed code additions and modifications, design blueprints document proposed design additions and modifications. These documents will usually be submitted as wireframes: schematic diagrams representing the way in which proposed screens will work. Visual designs may also be used when new visual elements must be created to support the wireframes. These blueprints are submitted to the OpenStack design community for feedback and sign off before they are implemented in the OpenStack dashboard.

Further details on how to create wireframes may be found in the Getting Started section.



P3 Instances & Volumes

Instances & Volumes

ID	Name	Size	Uptime	State	Actions
□ 56	InstanceName01	m1.tiny	192 hrs	Active	View
□ 57	ThisInstance	m1.large	186 hrs	Active	View
□ 58	AnotherInstance	m1.large	121 hrs	Active	View
□ 59	InstanceName02	m1.medium	78 hrs	Active	View
□ 60	YetAnotherInstance	m1.small	23 hrs	Active	View
□ 61	InstanceName03	m1.small	20 hrs	Active	View
□ 62	InstanceName04	m1.tiny	4 hrs	Active	View
□ 63	Something	m1.xlarge	3 min	Build	View
□ 64	InstanceInstance	m1.large	2 min	Build	View
□ 65	InstanceName01	m1.tiny	192 hrs	Active	View
□ 66	ThisInstance	m1.large	186 hrs	Active	View
□ 68	AnotherInstance	m1.large	121 hrs	Active	View
□ 69	InstanceName02	m1.medium	78 hrs	Active	View
□ 70	YetAnotherInstance	m1.small	23 hrs	Active	View
□ 71	InstanceName03	m1.small	20 hrs	Active	View
□ 72	InstanceName04	m1.tiny	4 hrs	Active	View
□ 73	Something	m1.xlarge	3 min	Build	View

Block Volumes

ID	Name	Size	Instance	Actions
□ 1	blockVol_01	100 GB	-	Attach
□ 2	blockVol_02	500 GB	-	Attach
□ 3	blockVol_03	1.5 TB	InstanceName03	Detach
□ 4	blockVol_04	100 GB	InstanceName04	Detach
□ 5	blockVol_05	100 GB	-	Attach
□ 6	blockVol_06	1 TB	Something	Detach
□ 7	blockVol_07	200 GB	InstanceInstance	Detach
□ 8	blockVol_08	100 GB	InstanceName01	Detach
□ 9	blockVol_09	100 GB		Attach

TABLES

1. **COLUMNS:** ID, Name, User, Host, Image, Size, Security Groups, IPs, Volumes, Uptime, State, Actions.

   ACTIONS: View (Console), Attach Volume (M6.1)/Detach Volume (Flash), Associate IP (M9.1), Terminate (M1:D2).

2. **COLUMNS:** ID, Name, Size, Actions.

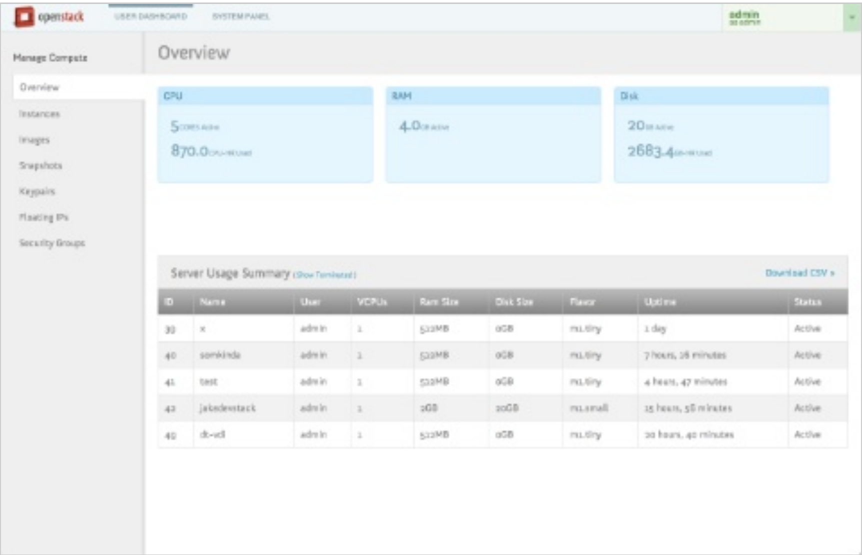
   ACTIONS: Attach (M12)/Detach((Flash), Delete (M1:D4).

   ADD: M6.

NOTES

1. Tables have a maximum height – content exceeding this height will cause continuous scrolling.
2. These options launch external consoles.

Figure 1.1 An example of a wireframe



Overview

CPU
5 Cores Active
870.0 CPU-HR Used

RAM
4.0 GB Active

Disk
20 GB Active
2683.4 GB-HR Used

Server Usage Summary (Show Terminated) Download CSV »

ID	Name	User	VCPUs	Ram Size	Disk Size	Flavor	Uptime	Status
39	x	admin	1	512MB	0GB	m1.tiny	1 day	Active
40	somkinda	admin	1	512MB	0GB	m1.tiny	7 hours, 18 minutes	Active
41	test	admin	1	512MB	0GB	m1.tiny	4 hours, 47 minutes	Active
42	jakedevstack	admin	1	2GB	20GB	m1.small	15 hours, 58 minutes	Active
49	dk-vdl	admin	1	512MB	0GB	m1.tiny	20 hours, 40 minutes	Active

Figure 1.2 An example of a visual design

When does a design blueprint need to be created?

A design blueprint is required for any engineering blueprint in which new functionality must be represented in the OpenStack dashboard. Once a design blueprint has been created and the design itself has been documented, the OpenStack community will evaluate the design blueprint against criteria defined in the HIG. When the design blueprint has been cleared, the OpenStack Model document will be updated to include the new design.

How does the OpenStack design process work?

The two core documents for the OpenStack design process are the Human Interface Guidelines document (this document) and the OpenStack Model document.

As described previously, the HIG is used to guide and evaluate design proposals.

Once a design has been approved, it is added to the OpenStack Model document which serves as the OpenStack community's definition for how the current OpenStack dashboard experience functions from a design perspective. Think of the Model as the equivalent of the Master codebase on the engineering side of OpenStack.

> Both the HIG and Model are living documents. As new blueprints propose better design, both documents may be changed to support a stronger user experience.

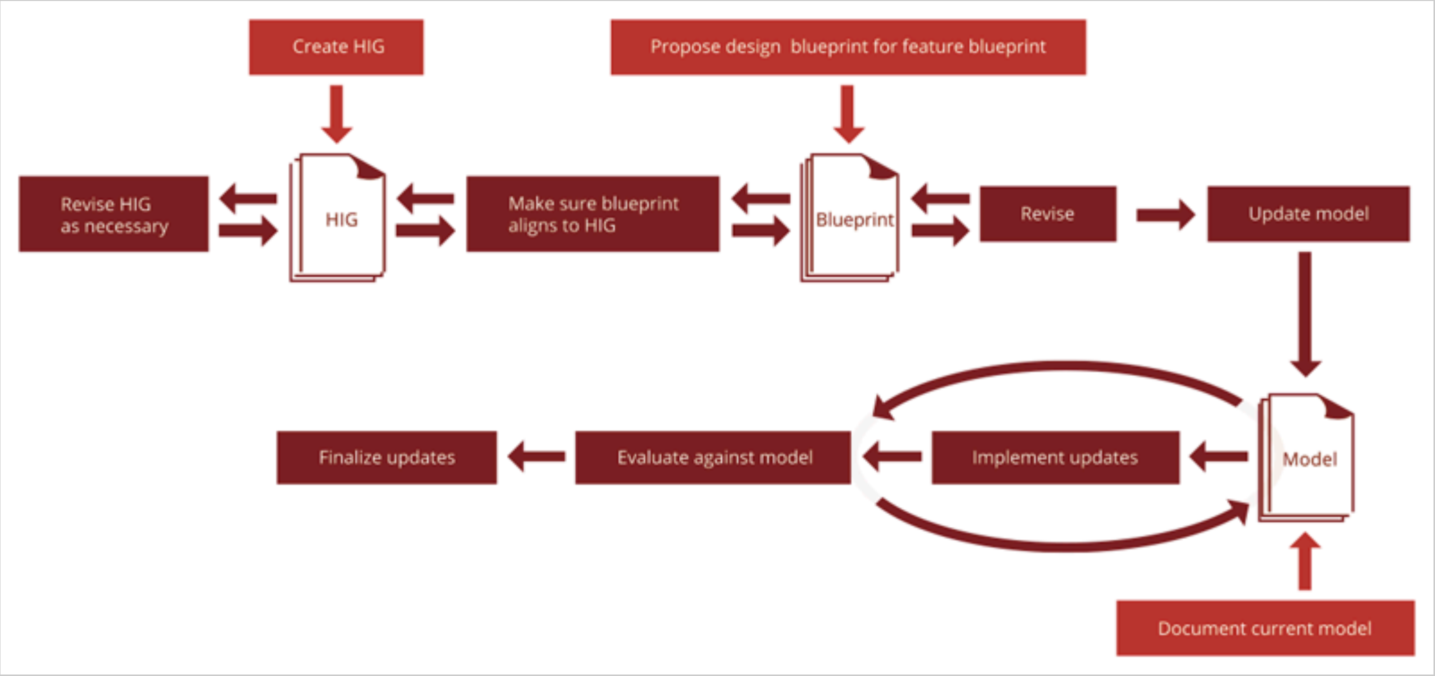


Figure 1.3 An overview of the OpenStack design process, including the use of the HIG and Model documents

> NOTE: The wireframes and visual designs presented in this document are for example purposes only and do not represent the current design of the OpenStack dashboard.

PRINCIPLES

The principles established in this section define the high-level priorities to be used when designing and evaluating interactions for the OpenStack dashboard. Principles are broad in scope and can be considered the philosophical foundation for the OpenStack experience; while they may not describe the tactical implementation of design, they should be used when deciding between multiple courses of design.

A significant theme for designing for the OpenStack experience concerns focusing on common uses of the system rather than adding complexity to support functionality that is rarely used.

In general, one should design for the 90% use case and support the other 10%.

Each of the following principles defines the intent of each concept and provides a few example questions that may be used to help validate design thinking for that concept.

Consistency

Consistency between OpenStack experiences will ensure that the dashboard feels like a single experience instead of a jumble of disparate products. Fractured experiences only serve to undermine user expectations about how they should interact with the system, creating an unreliable user experience. To avoid this, each interaction and visual representation within the system must be used uniformly and predictably. The architecture and elements detailed in this document will provide a strong foundation for establishing a consistent experience.

Example Review Criteria

- Does the design adhere to the structural model of the core experience? (See Core Architecture.)
- Has a new type of screen been introduced?
- Does the design use interactive elements as defined? (See Core Elements.)
- Are any data objects displayed or manipulated in a way contradictory to how they are handled elsewhere in the core experience?
- Does the design use the OpenStack dashboard visual design language? (See Core Design Language.)
- Can any newly proposed interactions be accomplished with existing interactions?

Simplicity

To best support new users and create straight forward interactions, designs should be as simple as possible. When crafting new interactions, designs should reduce the amount of noise present on a screen: large amounts of nonessential data, overuse of highlight colors, overabundance of possible actions, etc. Designs should focus on the intent of the screen, revealing only the necessary components and either removing superfluous elements or making them accessible through a secondary action. An example of this principle occurs in OpenStack’s use of tables: only the most often used columns are shown by default. Further data may be accessed through the Table View control, allowing users to specify the types of data that they find useful in their day-to-day work.

Example Review Criteria

- Are any screens visually overwhelming?
- How many of the displayed actions/data are relevant to the majority of users?
- If multiple actions are required for the user to complete a task, is each step required or can the process be more efficient?

User-Centered Design

Interactions should be design based on how a user will interact with the system and not how the system’s backend is organized. While database structures and APIs may define what is possible, they often do not define good user experience; consider user goals and the way in which users will want to interact with their data, then design for these work flows and mold the interface to the user, not the user to the interface.

Example Review Criteria

- How quickly can a user figure out how to accomplish a given task?
- Has content been grouped and ordered according to usage relationships?
- Do work flows support user goals or add complexity?

Transparency

Make sure users understand the current state of their infrastructure and interactions. For example, users should be able to access information about the state of each machine/virtual machine easily, without having to actively seek out this information. Whenever the user initiates an action, make sure a confirmation is displayed to show that an input has been received. Upon completion of a process, make sure the user is informed. Ensure that the user never questions the state of their environment.

Example Review Criteria

- Does the user receive feedback when initiating a process?
- When a process is completed?
- Does the user have quick access to the state of their infrastructure?

CORE ARCHITECTURE

This section documents the types of pages used in the OpenStack dashboard and the way in which they relate to one another.

Basic Structure

There are three types of screens used in the OpenStack dashboard: the Sign In Screen, Content Screens, and Modals. After a user has signed in, the majority of the content is displayed as Content Screens grouped within sections. Modals may occur on top of any screen in order to perform ad hoc tasks.

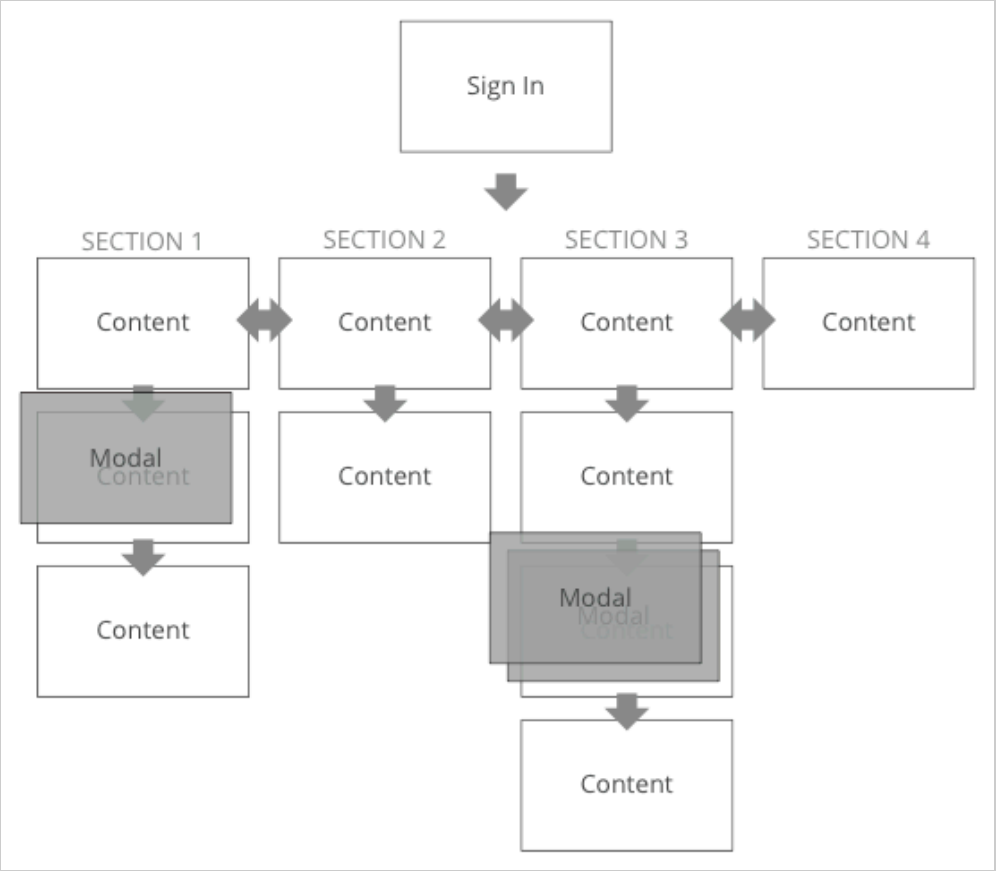
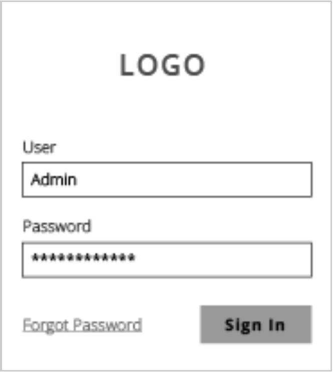


Figure 3.1 The Sign In screen provides access to Content Screens on which Modals may overlay

Sign In Screen

The Sign In Screen is a very straightforward screen: a simple form allowing the user to log into the OpenStack dashboard. This screen is not considered a Modal screen because it does not overlay another screen and several required Modal elements (such as close buttons) are not used in this screen.



Admin

****

Figure 3.2 An example of a Sign In Screen

Content Screens

Content Screens make up the bulk of the OpenStack dashboard. These screens are grouped by section under an overview screen. For example, upon navigating to the Instances & Volumes section the Instances & Volumes overview screen would be shown. This screen would list currently running instances and available/utilized volumes. From here, subsection screens may be accessed, in this case screens that further describe the Instances & Volumes of the system. Each Content Screen in the OpenStack dashboard consists of a Navigation Bar, Header, and Content Area:

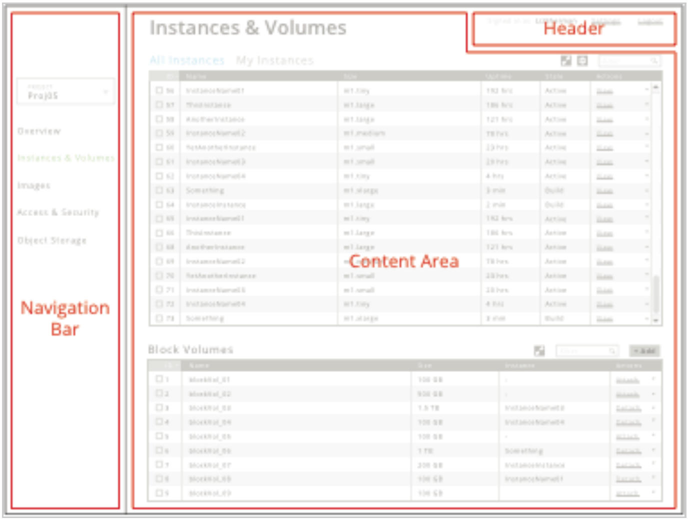


Figure 3.3 Each Content Screen contains a Navigation Bar, Header, and Content Area

Navigation Bar

The Navigation Bar contains the OpenStack logo, panel dropdown menu, role dropdown menu, project dropdown menu, and section titles (determined by the selected dropdown menu values). Plugin titles are also included in the Navigation Bar. To determine which sections of content should be surfaced to the current user, the Navigation Bar may contain several dropdown menus to help set the context for the tasks the user is attempting to accomplish. These dropdown menus cascade: by changing the value of a higher level dropdown, all lower level dropdowns may change available values accordingly. The order in which content is determined is according to the following hierarchy:

Panel > Role > Project

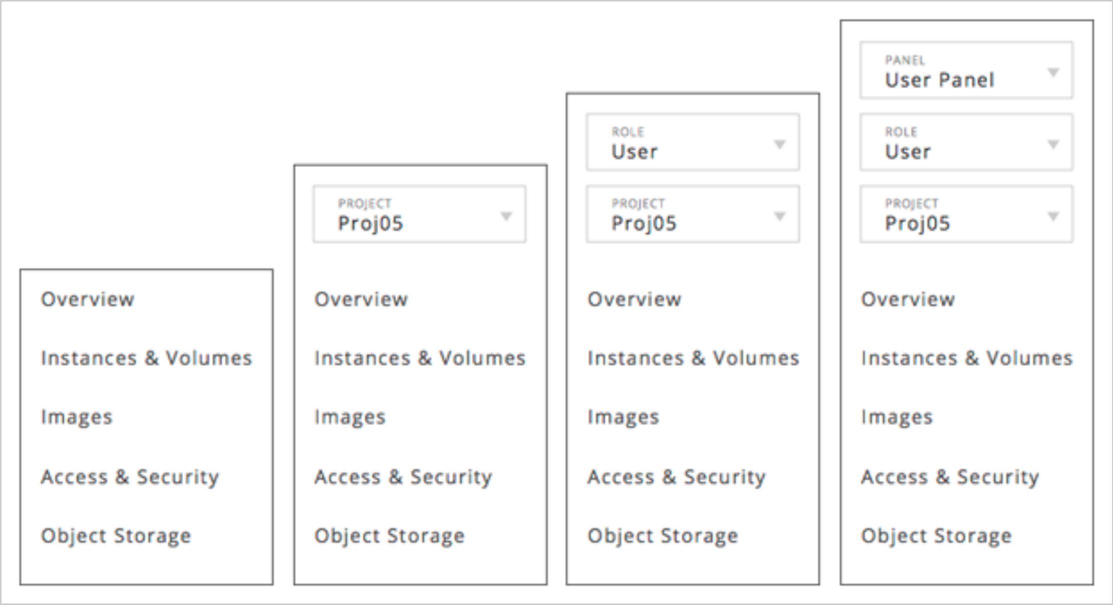


Figure 3.4 Various dropdown menus may or may not appear in the Navigation Bar depending on user privileges

The Panel Dropdown determines the superset of content the user has access to, such as the User Dashboard and the System Panel. The User Dashboard contains functionality related to setting up, running, and terminating virtual machines, while the System Panel contains admin functionality to manage users, project groups, and other high-level tasks. If a user only has access to one panel, then this dropdown does not appear for that user. The Role Dropdown determines next level of content groups and is affected by both the roles available to the user and the current panel selected in the Panel Dropdown. Examples of Role Dropdown values are project admin, system admin, user, and security admin. If only one role is available for the user for the given panel, then this dropdown does not appear for that user. The Project Dropdown specifies which project the user is currently working within. This dropdown may be omitted if a specific panel or role does not target a specific project, such as the System Admin role. If a user only has access to one project, then this dropdown does not appear for that user. The sections and plugins listed in the rest of the Navigation Bar are updated according to the selected values for these dropdowns. Sections and plugins that are not available for the selected panel, role, or project are omitted from the Navigation Bar. By selecting a section or plugin, the Header and Content Area are updated with associated content.

Header

The Header contains login information, navigation to the settings section and the ability to logout.

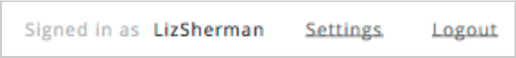


Figure 3.5 An example of the Header

Content Area

The Content Area, as the name implies, contains all the content for the current screen, including the current screens title and a navigational breadcrumb trail if required.

Modals

Modals take place on top of non-modal screens and act as temporary spaces in which users may perform actions and inspect objects. Once a modal is closed, the user is returned to the initial non-modal screen.

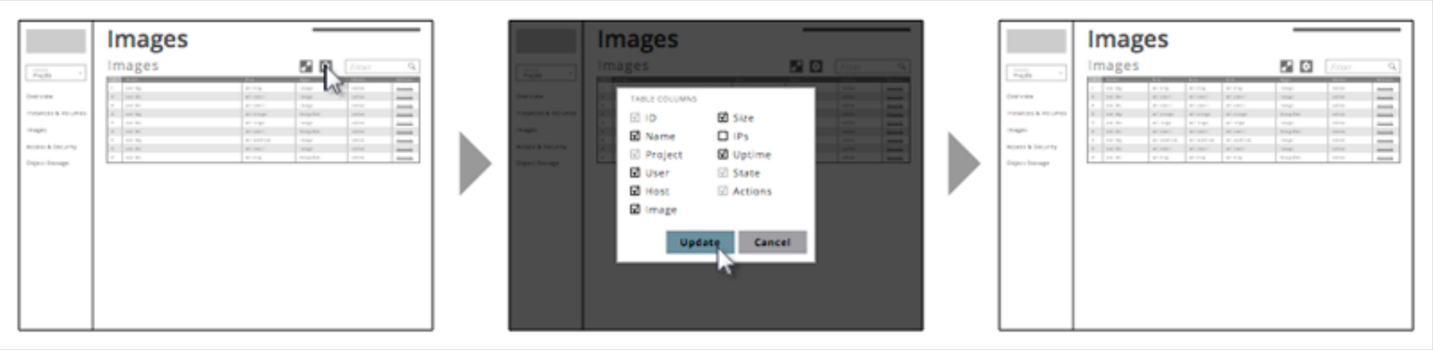


Figure 3.6 An example of a simple modal flow: the user clicks the Table Options button, adjusts settings in the resulting modal, and returns to an updated screen

Modals may also be linked together to create modal flows, such as a “Create New User” flow or “Edit Security Group” modal.

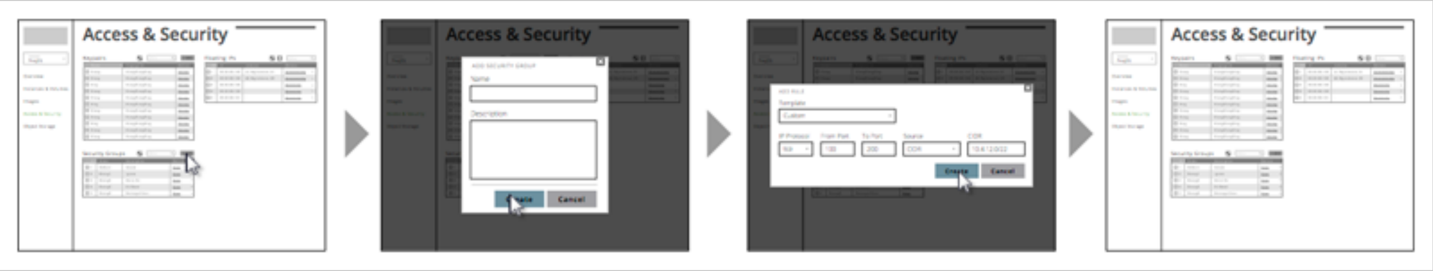
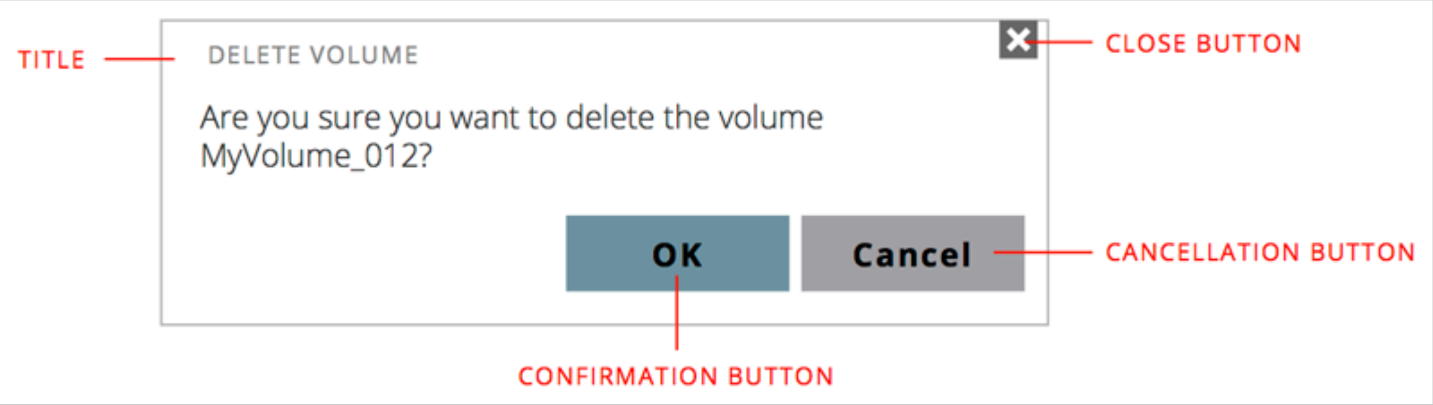


Figure 3.7 An example of a multiple-modal flow: the user clicks the Add Security Group button, creates a group, adds a rule, and returns to an updated screen

Each modal must have a title, confirmation/cancellation buttons, and a close button in the top-right corner of the modal window.



Are you sure you want to delete the volume MyVolume_012?

Figure 3.7 An example of a modal

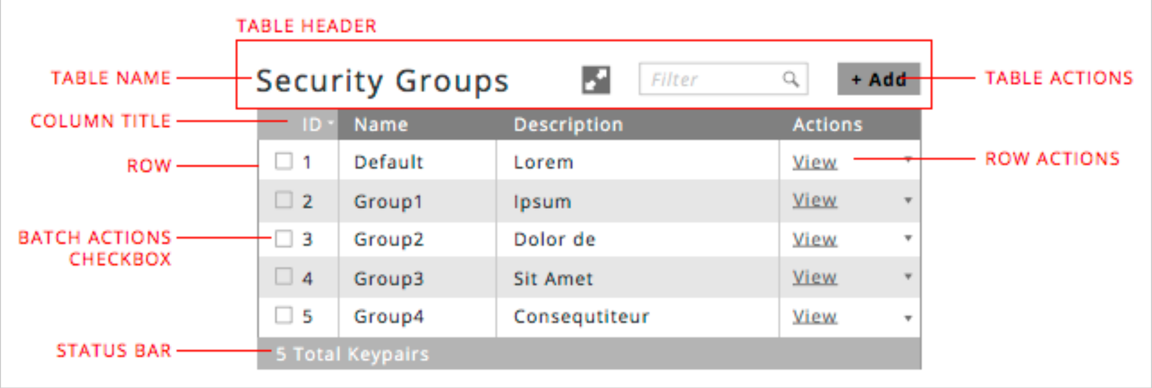
Modal screens should be used to perform actions that require further user input. For example, after clicking the “Add Security Group” button on the Security Group table, a modal should be used to allow the user to enter the values used to create the new security group.

CORE ELEMENTS

This section will detail the uses, requirements, and recommendations for common components.

Tables and Table Elements

Table elements are used to display a matrix of information, such as describing the ID, name, uptime, and state of running instances. All tables require a table name, a table header, and one or more rows.



Security Groups

ID	Name	Description	Actions
□ 1	Default	Lorem	View ▾
□ 2	Group1	Ipsum	View ▾
□ 3	Group2	Dolor de	View ▾
□ 4	Group3	Sit Amet	View ▾
□ 5	Group4	Consequtiteur	View ▾

5 Total Keypairs

Figure 4.1 An example of a table

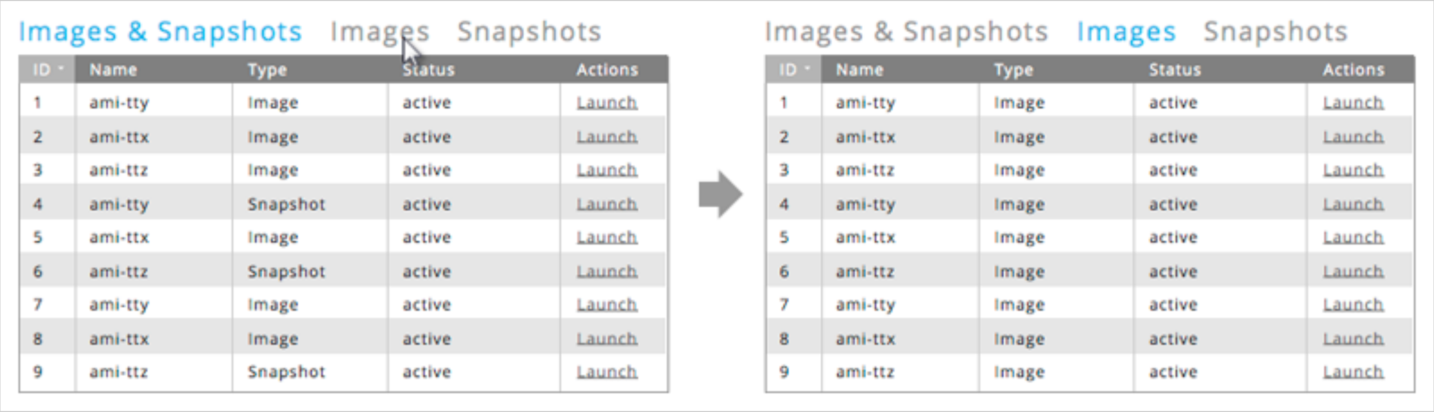
Table height may be fixed or variable depending on the context of the table. If no content exists below or beside a table, the length of the table may grow to accommodate the full data set. If multiple tables exist on a screen, allowing these tables to use variable height may result in situations in which some tables are not visible upon initial loading of the screen (meaning the user will not know that table exists) or navigating one table may result in further difficulty comparing and navigating the other content available on the screen. Because of this, tables may use fixed height to accommodate all tables on the screen. The full set of data for these tables may be viewed using either continuous scrolling (content loads as the user reaches the end of the currently visible content) or pagination (the user manually navigates through sets of data). In either case, content for these screens should be designed to allow the user to view the full range of content groups on the screen.

Table Name

The Table Name is displayed in the table header and succinctly describes the type of data displayed in the table.

Table Tabs and Table Dropdowns

In some cases it is useful to combine tables that display different sets of the same data, such as Images and Snapshots. To combine these tables, either Table Tabs or Table Dropdowns are used. In either case, the Table Name is replaced with Table Tabs or a Table Dropdown. If the number of sets is fixed and can fit in the table header, then Table Tabs are used to toggle between the various sets of data:



Images & Snapshots Images Snapshots

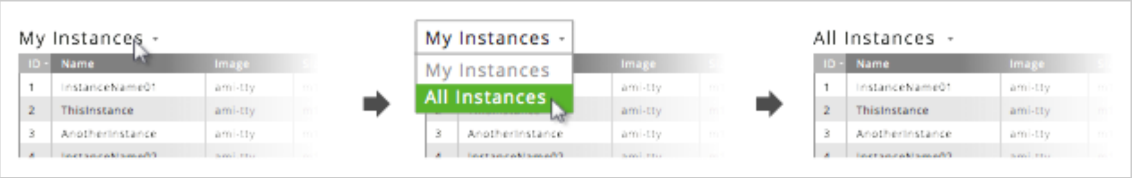
ID	Name	Type	Status	Actions
1	ami-tty	Image	active	Launch
2	ami-ttx	Image	active	Launch
3	ami-ttz	Image	active	Launch
4	ami-tty	Snapshot	active	Launch
5	ami-ttx	Image	active	Launch
6	ami-ttz	Snapshot	active	Launch
7	ami-tty	Image	active	Launch
8	ami-ttx	Image	active	Launch
9	ami-ttz	Snapshot	active	Launch

Images & Snapshots Images Snapshots

ID	Name	Type	Status	Actions
1	ami-tty	Image	active	Launch
2	ami-ttx	Image	active	Launch
3	ami-ttz	Image	active	Launch
4	ami-tty	Image	active	Launch
5	ami-ttx	Image	active	Launch
6	ami-ttz	Image	active	Launch
7	ami-tty	Image	active	Launch
8	ami-ttx	Image	active	Launch
9	ami-ttz	Image	active	Launch

Figure 4.2 Selecting a Table Tab refreshes the table with appropriate content

If the number of sets is not fixed or the sets do not fit within the Table Header then a Table Dropdown is used instead. This provides the user to select a data set from a list:



ID	Name	Image
1	InstanceName01	ami-tty
2	ThisInstance	ami-tty
3	AnotherInstance	ami-tty
4	InstanceName02	ami-tty

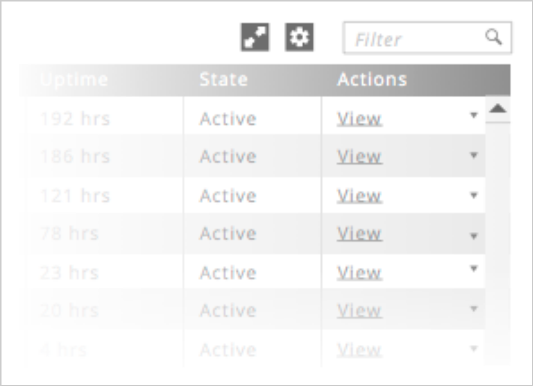
ID	Name	Image
		ami-tty
		ami-tty
3	AnotherInstance	ami-tty
4	InstanceName02	ami-tty

ID	Name	Image
1	InstanceName01	ami-tty
2	ThisInstance	ami-tty
3	AnotherInstance	ami-tty
4	InstanceName02	ami-tty

Figure 4.3 Some tables use Table Dropdowns to allow the user to determine the data to be displayed in the table

Table Actions

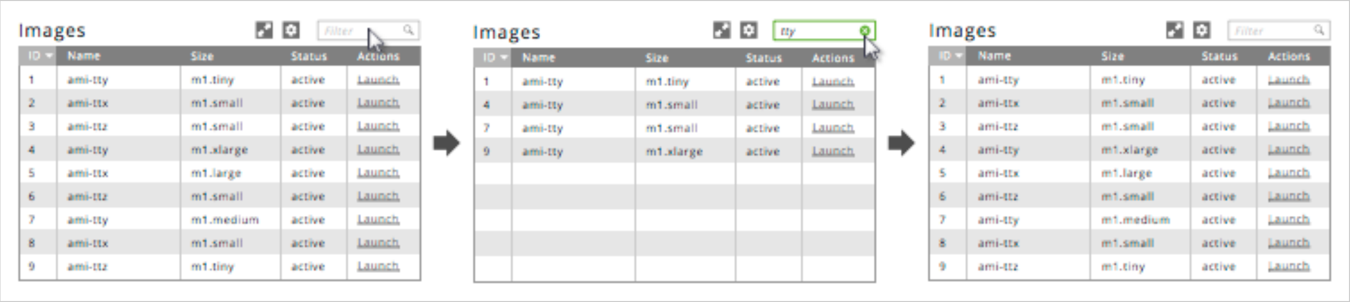
Table Actions are displayed above the Table Header and aligned to the Table Header's right edge. Actions placed here may be used to modify the table itself or the group of data contained within the table.



Uptime	State	Actions
192 hrs	Active	View
186 hrs	Active	View
121 hrs	Active	View
78 hrs	Active	View
23 hrs	Active	View
20 hrs	Active	View
4 hrs	Active	View

Figure 4.4 Table Actions are displayed above the table

The Search field acts as a filter mechanic: as the user types characters in this field, the table responds by showing only rows that match those characters. The user may delete these characters or use the Clear button at any time to refresh the table with the entire data set.



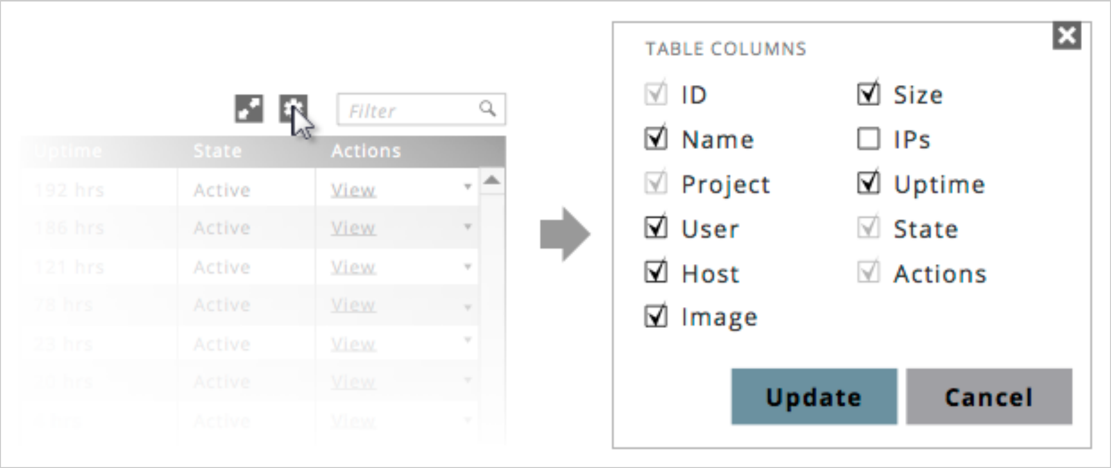
ID	Name	Size	Status	Actions
1	ami-tty	m1.tiny	active	Launch
2	ami-ttx	m1.small	active	Launch
3	ami-ttz	m1.small	active	Launch
4	ami-tty	m1.xlarge	active	Launch
5	ami-ttx	m1.large	active	Launch
6	ami-ttz	m1.small	active	Launch
7	ami-tty	m1.medium	active	Launch
8	ami-ttx	m1.small	active	Launch
9	ami-ttz	m1.tiny	active	Launch

ID	Name	Size	Status	Actions
1	ami-tty	m1.tiny	active	Launch
4	ami-tty	m1.small	active	Launch
7	ami-tty	m1.small	active	Launch
9	ami-tty	m1.xlarge	active	Launch

ID	Name	Size	Status	Actions
1	ami-tty	m1.tiny	active	Launch
2	ami-ttx	m1.small	active	Launch
3	ami-ttz	m1.small	active	Launch
4	ami-tty	m1.xlarge	active	Launch
5	ami-ttx	m1.large	active	Launch
6	ami-ttz	m1.small	active	Launch
7	ami-tty	m1.medium	active	Launch
8	ami-ttx	m1.small	active	Launch
9	ami-ttz	m1.tiny	active	Launch

Figure 4.5 Entering characters into the Search Field filters the tables contents; clearing the Search Field shows all content for that table

The View Options button opens a modal allowing the user to toggle table column visibility on and off. Some columns may not be hidden and others may be hidden by default.



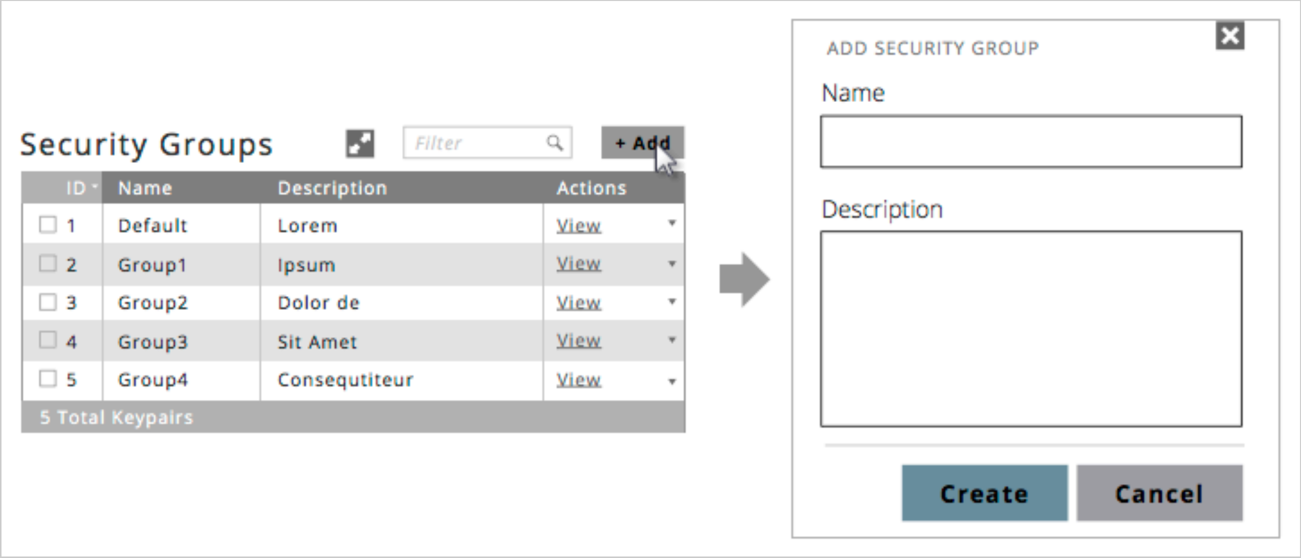
Uptime	State	Actions
192 hrs	Active	View ▾
186 hrs	Active	View ▾
121 hrs	Active	View ▾
78 hrs	Active	View ▾
23 hrs	Active	View ▾
20 hrs	Active	View ▾
6 hrs	Active	View ▾

TABLE COLUMNS

☑ ID ☑ Size
☑ Name □ IPs
☑ Project ☑ Uptime
☑ User ☑ State
☑ Host ☑ Actions
☑ Image

Figure 4.6 Clicking the view options button opens the view options modal

New objects may be added to the table through use of the Add button. The Add button launches a corresponding modal to specify the creation of a new object.



Security Groups

ID	Name	Description	Actions
□ 1	Default	Lorem	View ▾
□ 2	Group1	Ipsum	View ▾
□ 3	Group2	Dolor de	View ▾
□ 4	Group3	Sit Amet	View ▾
□ 5	Group4	Consequtiteur	View ▾
5 Total Keypairs			

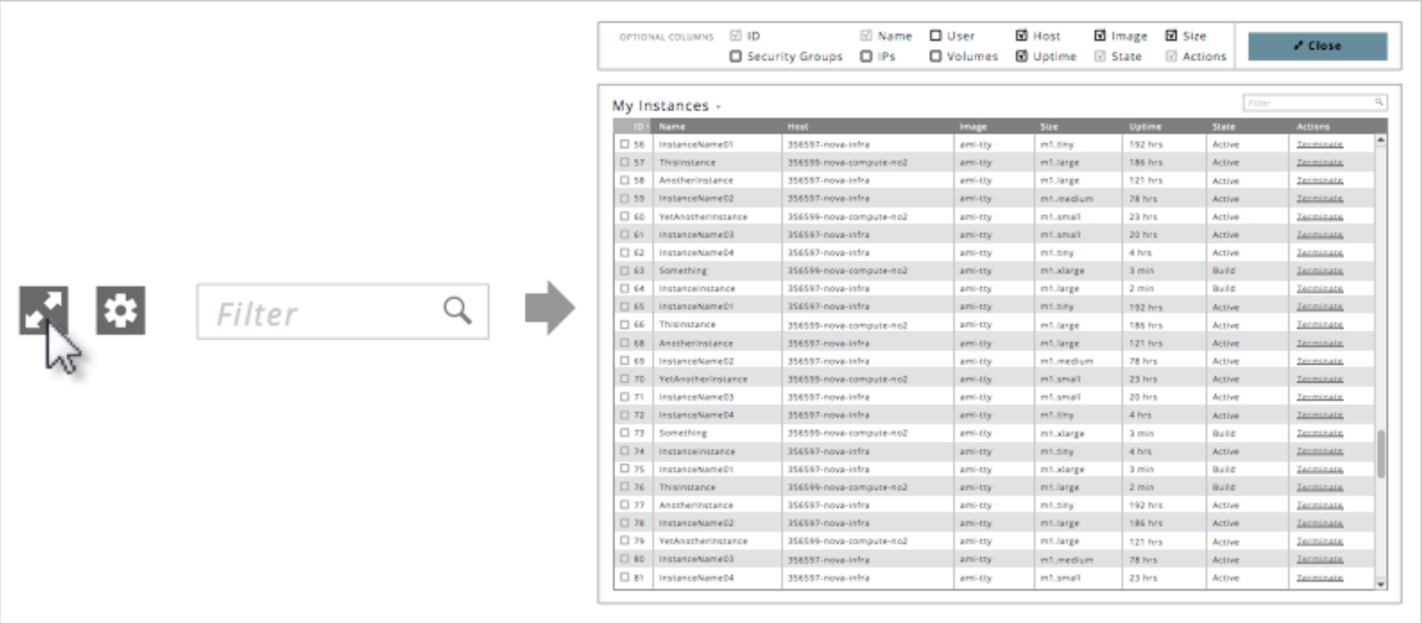
ADD SECURITY GROUP

Name

Description

Figure 4.7 Clicking the Add button for a table opens a modal to allow the user to create a new object

The Fullscreen Table button allows any table to be viewed as an immersive modal, allowing the user to display more data and interact with table data more fully.



OPTIONAL COLUMNS ☑ ID ☑ Name □ User ☒ Host ☒ Image ☒ Size
□ Security Groups □ IPs □ Volumes ☒ Uptime ☑ State ☑ Actions

Close

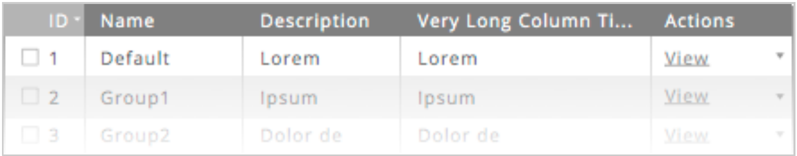
My Instances

ID	Name	Host	Image	Size	Uptime	State	Actions
□ 56	InstanceName01	356597-nova-infra	ami-tty	m1.tiny	192 hrs	Active	Terminate
□ 57	ThisInstance	356599-nova-compute-no2	ami-tty	m1.large	186 hrs	Active	Terminate
□ 58	AnotherInstance	356597-nova-infra	ami-tty	m1.large	121 hrs	Active	Terminate
□ 59	InstanceName02	356597-nova-infra	ami-tty	m1.medium	78 hrs	Active	Terminate
□ 60	YetAnotherInstance	356599-nova-compute-no2	ami-tty	m1.small	23 hrs	Active	Terminate
□ 61	InstanceName03	356597-nova-infra	ami-tty	m1.small	20 hrs	Active	Terminate
□ 62	InstanceName04	356597-nova-infra	ami-tty	m1.tiny	4 hrs	Active	Terminate
□ 63	Something	356599-nova-compute-no2	ami-tty	m1.xlarge	3 min	Build	Terminate
□ 64	InstanceInstance	356597-nova-infra	ami-tty	m1.large	2 min	Build	Terminate
□ 65	InstanceName01	356597-nova-infra	ami-tty	m1.tiny	192 hrs	Active	Terminate
□ 66	ThisInstance	356599-nova-compute-no2	ami-tty	m1.large	186 hrs	Active	Terminate
□ 68	AnotherInstance	356597-nova-infra	ami-tty	m1.large	121 hrs	Active	Terminate
□ 69	InstanceName02	356597-nova-infra	ami-tty	m1.medium	78 hrs	Active	Terminate
□ 70	YetAnotherInstance	356599-nova-compute-no2	ami-tty	m1.small	23 hrs	Active	Terminate
□ 71	InstanceName03	356597-nova-infra	ami-tty	m1.small	20 hrs	Active	Terminate
□ 72	InstanceName04	356597-nova-infra	ami-tty	m1.tiny	4 hrs	Active	Terminate
□ 73	Something	356599-nova-compute-no2	ami-tty	m1.xlarge	3 min	Build	Terminate
□ 74	InstanceInstance	356597-nova-infra	ami-tty	m1.tiny	4 hrs	Active	Terminate
□ 75	InstanceName01	356597-nova-infra	ami-tty	m1.xlarge	3 min	Build	Terminate
□ 76	ThisInstance	356599-nova-compute-no2	ami-tty	m1.large	2 min	Build	Terminate
□ 77	AnotherInstance	356597-nova-infra	ami-tty	m1.tiny	192 hrs	Active	Terminate
□ 78	InstanceName02	356597-nova-infra	ami-tty	m1.large	186 hrs	Active	Terminate
□ 79	YetAnotherInstance	356599-nova-compute-no2	ami-tty	m1.large	121 hrs	Active	Terminate
□ 80	InstanceName03	356597-nova-infra	ami-tty	m1.medium	78 hrs	Active	Terminate
□ 81	InstanceName04	356597-nova-infra	ami-tty	m1.small	23 hrs	Active	Terminate

Figure 4.8 Selecting the fullscreen table button opens that table as a fullscreen modal

Column Title

Column Titles are displayed in the Table Header and describe the type of data contained within that column. If a Column Title is longer than its column width allows, the Column Title is truncated with an ellipse at the end (e.g. “Total Capacity” would become “Total Cap...”).

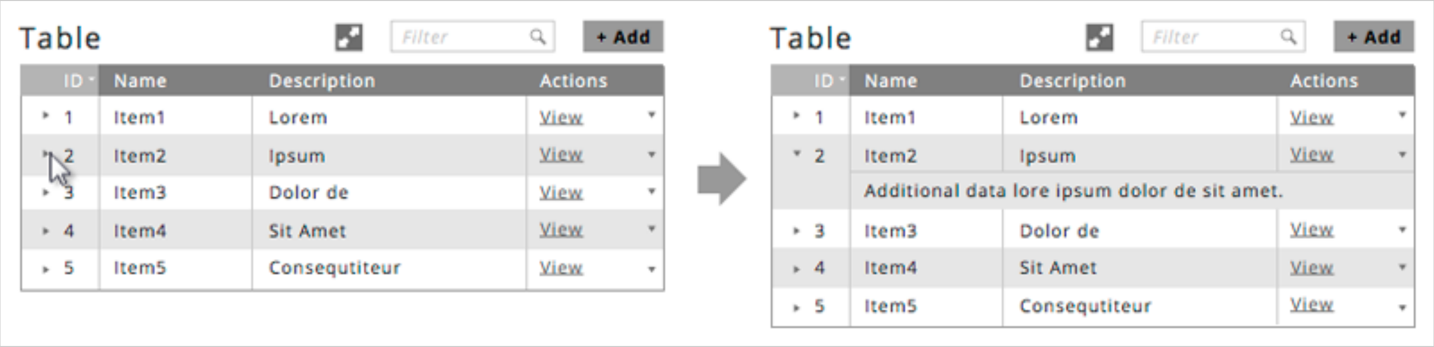


ID	Name	Description	Very Long Column Ti...	Actions
□ 1	Default	Lorem	Lorem	View
□ 2	Group1	Ipsum	Ipsum	View
□ 3	Group2	Dolor de	Dolor de	View

Figure 4.9 Examples of column titles

Row

Each row displays data for each column. Cells for which no data is available use single dashes (e.g. “-”) to indicate the absence of relevant data. Avoid displaying multi-line data in a cell. In some cases, a row may use a chevron to indicate that the row is expandable. By clicking on the chevron the row can be expanded and collapsed to reveal/hide additional information.



Table

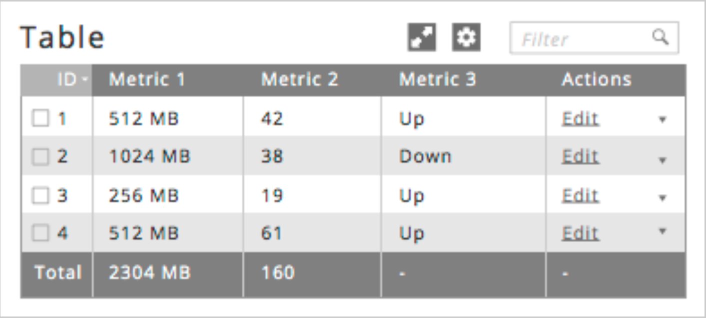
ID	Name	Description	Actions
▸ 1	Item1	Lorem	View
▸ 2	Item2	Ipsum	View
▸ 3	Item3	Dolor de	View
▸ 4	Item4	Sit Amet	View
▸ 5	Item5	Consequtiteur	View

Table

ID	Name	Description	Actions
▸ 1	Item1	Lorem	View
▾ 2	Item2	Ipsum	View
	Additional data lore ipsum dolor de sit amet.		
▸ 3	Item3	Dolor de	View
▸ 4	Item4	Sit Amet	View
▸ 5	Item5	Consequtiteur	View

Figure 4.10 Some rows may be expanded by clicking the chevron for that row

A row may use additional visual styling to differentiate itself. These rows are different from Status Bars in that the data they provide remains structured by the table’s columns.



Table

ID	Metric 1	Metric 2	Metric 3	Actions
□ 1	512 MB	42	Up	Edit ▾
□ 2	1024 MB	38	Down	Edit ▾
□ 3	256 MB	19	Up	Edit ▾
□ 4	512 MB	61	Up	Edit ▾
Total	2304 MB	160	-	-

Figure 4.11 An example of a row showing the total for a table

Row Actions

The actions for a row are displayed in the right-most column of the table. If only a single action is available, this action is listed by itself. If more than one action exists for a row then one action is promoted by being displayed in the Action column. The rest of the actions are shown in a More Actions menu accessed by clicking the More Actions chevron.

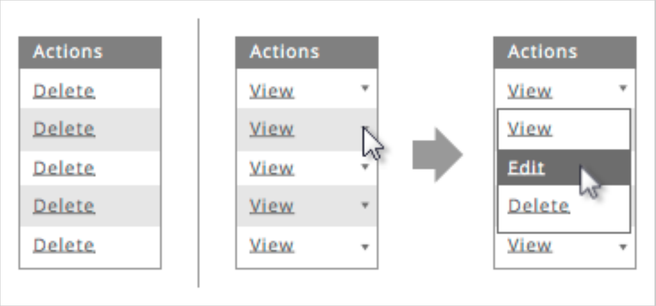
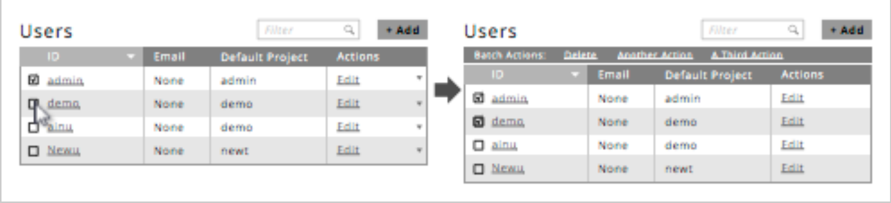


Figure 4.12 An example of an action row containing one action, an action row containing more than one action, and the action dropdown

Checkboxes and Batch Actions

Checkboxes are shown in the first column of a table for objects that may have batch actions taken upon them. Once one or more checkboxes have been selected, the Batch Actions Bar is shown. The Batch Actions Bar lists all applicable actions that may be performed on the selected item/s.



Users

ID	Email	Default Project	Actions
☑ admin	None	admin	Edit ▾
□ demo	None	demo	Edit ▾
□ ainu	None	demo	Edit ▾
□ Newu	None	newt	Edit ▾

Users

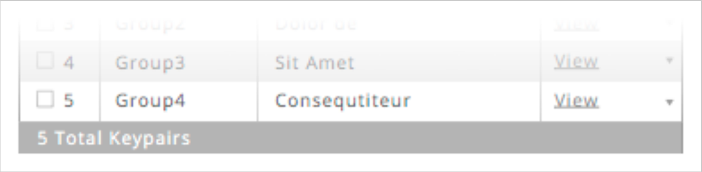
Batch Actions: Delete Another Action A Third Action

ID	Email	Default Project	Actions
☑ admin	None	admin	Edit
☑ demo	None	demo	Edit
□ ainu	None	demo	Edit
□ Newu	None	newt	Edit

Figure 4.13 After selecting more than one time, the Batch Actions menu is shown, allowing the user to perform a batch action on the selected items

Status Bar

The last row of a table may be used as a Status Bar to show aggregate data about the table, such as total number of items, total amount of storage capacity for the listed items, etc.

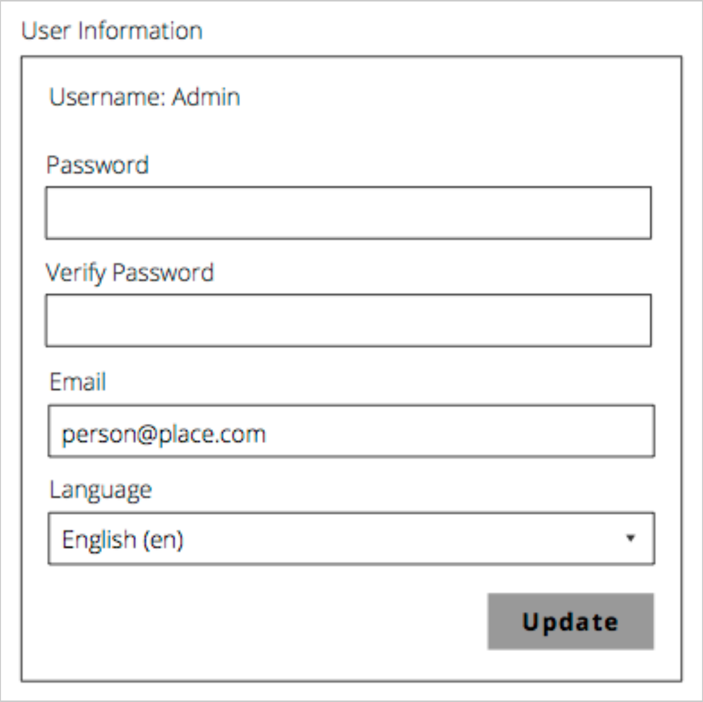


□ 4	Group3	Sit Amet	View ▾
□ 5	Group4	Consequtiteur	View ▾
5 Total Keypairs			

Figure 4.14 An example of a status bar depicting the total number of Keypairs listed in a table

Form and Form Elements

Forms are collections of fields, checkboxes, dropdowns, and other controls used to allow the user to specify one or more values and perform an action. All forms require a title, either as a Screen Title if the form is the only content for the screen or a Form Title if other content is displayed on the screen, and Confirmation/Cancellation buttons.



User Information

Username: Admin

Password

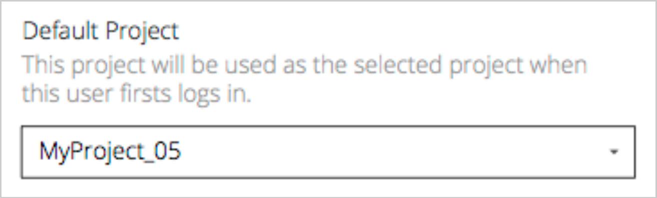
Verify Password

Email: person@place.com

Language: English (en)

Figure 4.15 An example of a form, including form title, text fields, a dropdown, and a confirmation button

Each control must include a Control Name. In some cases, a Control Description should be included to clarify the type of value to be entered.



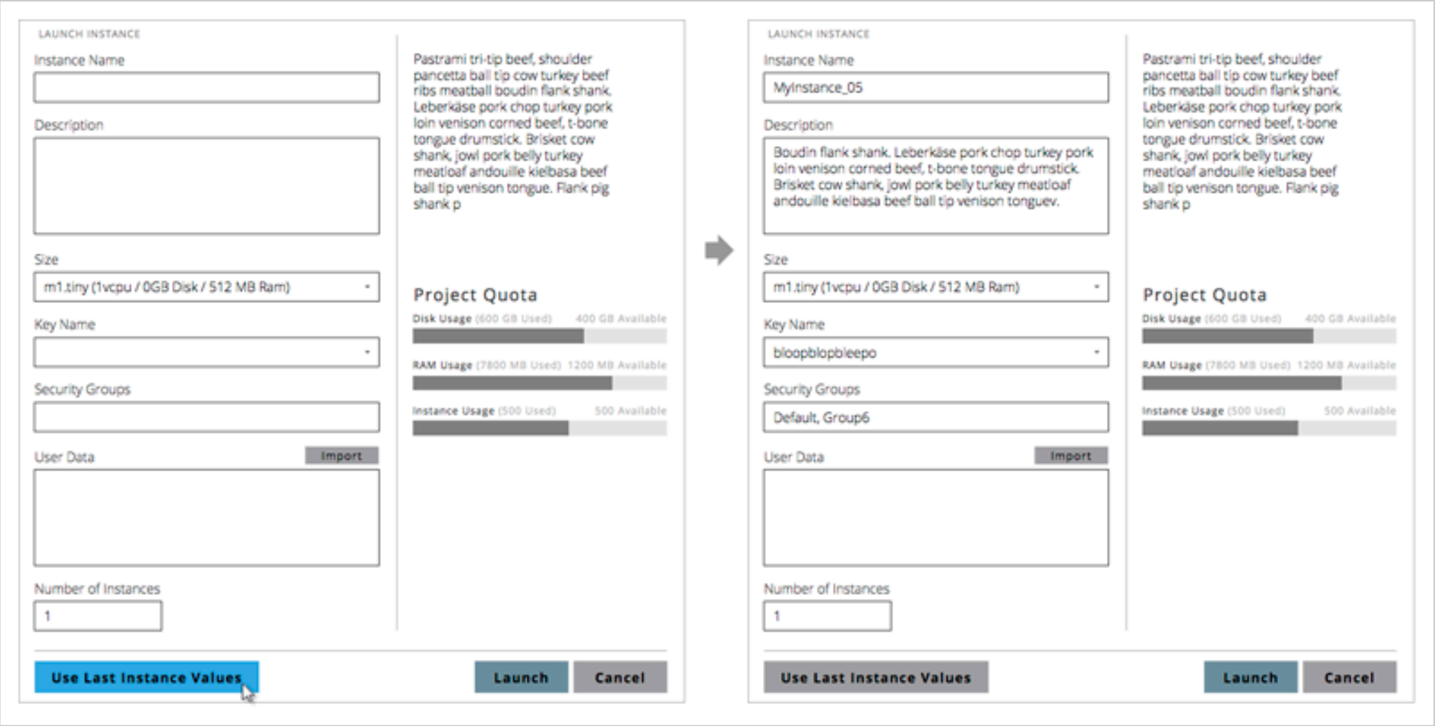
Default Project

This project will be used as the selected project when this user firsts logs in.

MyProject_05

Figure 4.16 The control name is a required element for a form control and the control description is useful when explaining complex values

If the form is extensive, consider including a “Populate with last values” button to allow the user to quickly fill the form with the values used the last time they submitted that form.


LAUNCH INSTANCE
Instance Name
Description
Size
m1.tiny (1vcpu / 0GB Disk / 512 MB Ram)
Key Name
Security Groups
User Data
Import
Number of Instances
1
Use Last Instance Values
Launch
Cancel
Project Quota
Disk Usage (600 GB Used) 400 GB Available
RAM Usage (7800 MB Used) 1200 MB Available
Instance Usage (500 Used) 500 Available

Pastrami tri-tip beef, shoulder pancetta ball tip cow turkey beef ribs meatball boudin flank shank. Leberkäse pork chop turkey pork loin venison corned beef, t-bone tongue drumstick. Brisket cow shank, jowl pork belly turkey meatloaf andouille kielbasa beef ball tip venison tongue. Flank pig shank p

MyInstance_05

Boudin flank shank. Leberkäse pork chop turkey pork loin venison corned beef, t-bone tongue drumstick. Brisket cow shank, jowl pork belly turkey meatloaf andouille kielbasa beef ball tip venison tonguev.

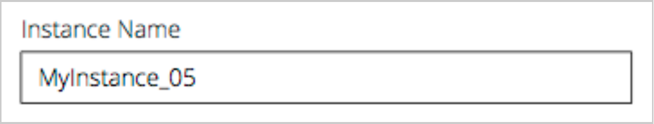
Size
m1.tiny (1vcpu / 0GB Disk / 512 MB Ram)
Key Name
bloopblopbleepo
Security Groups
Default, Group6
User Data
Import
Number of Instances
1
Use Last Instance Values
Launch
Cancel
Project Quota
Disk Usage (600 GB Used) 400 GB Available
RAM Usage (7800 MB Used) 1200 MB Available
Instance Usage (500 Used) 500 Available

Pastrami tri-tip beef, shoulder pancetta ball tip cow turkey beef ribs meatball boudin flank shank. Leberkäse pork chop turkey pork loin venison corned beef, t-bone tongue drumstick. Brisket cow shank, jowl pork belly turkey meatloaf andouille kielbasa beef ball tip venison tongue. Flank pig shank p

Figure 4.17 Selecting the Use Last Instance Values button populates the form with the previously submitted data

Text Fields

Text Fields are used to define single-line, relatively short strings. If the user may wish to enter extensive amounts of text, a Text Box should be used instead.



MyInstance_05

Figure 4.18 An example of a text field

Text Boxes

Text Boxes are used to capture extensive, multiple-line strings. In some cases, Text Boxes may be used in conjunction with Import Buttons to allow the user to import text files into the Text Box.

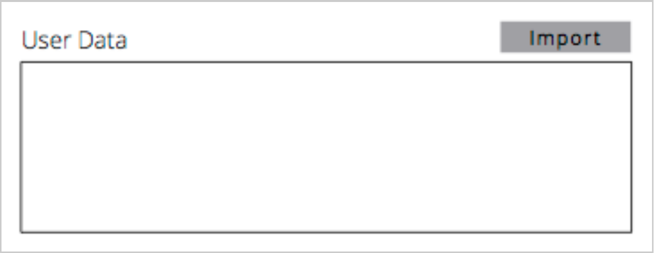
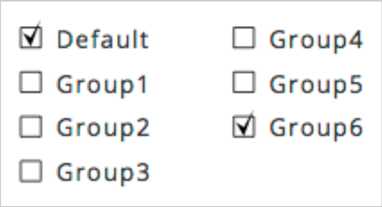


Figure 4.19 An example of a text box with Import button

Checkboxes

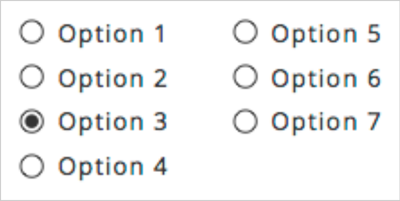
Checkboxes should be used to capture multiple-selection data in which one or more values may be selected.



☑ Default
□ Group1
□ Group2
□ Group3
□ Group4
□ Group5
☑ Group6

Figure 4.20 An example of checkboxes

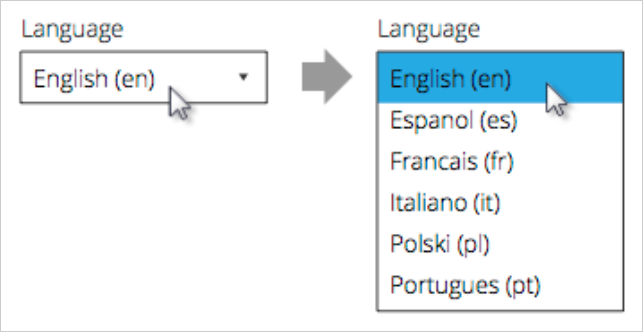
Radio Buttons
Radio Buttons should be used for single-selection data only.



○ Option 1
○ Option 2
● Option 3
○ Option 4
○ Option 5
○ Option 6
○ Option 7

Figure 4.21 An example of radio buttons

Dropdowns
Drop Downs should be used for single-selection data only.

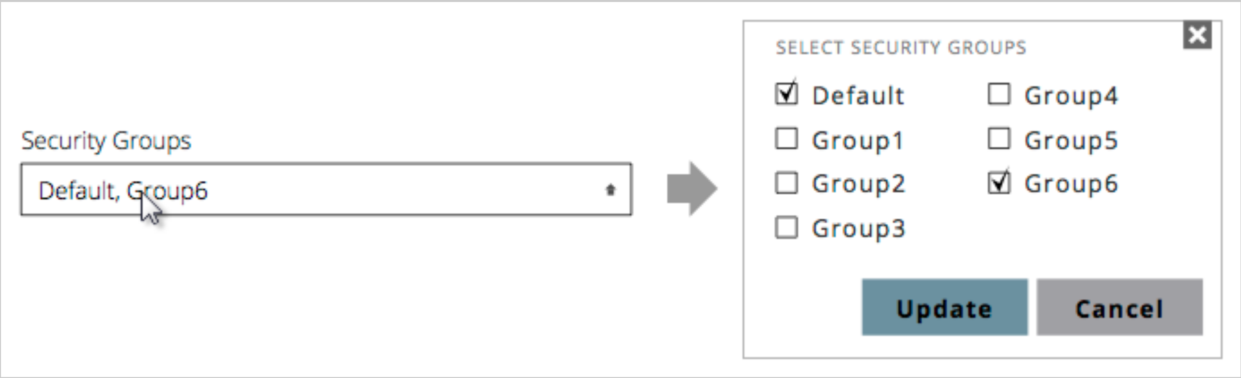


Language
English (en) ▾

Language
English (en)
Espanol (es)
Francais (fr)
Italiano (it)
Polski (pl)
Portugues (pt)

Figure 4.22 An example of a dropdown and its corresponding dropdown menu

Multiple Selection Fields
Multiple Selection Fields should be used when the user needs to select one or more values from and extensive or dynamic list. They are preferred over Check Boxes when the amount of content may dominate a form. Multiple Selection Fields display currently selected values in the field. Clicking the field launches a modal from which the user may view and select all possible values for that field.



Security Groups
Default, Group6

SELECT SECURITY GROUPS
☑ Default
□ Group1
□ Group2
□ Group3
□ Group4
□ Group5
☑ Group6

Figure 4.23 Selecting a multiple selection field opens a selection modal

Confirmation/Cancellation Buttons

Confirmation buttons are used to submit a form.

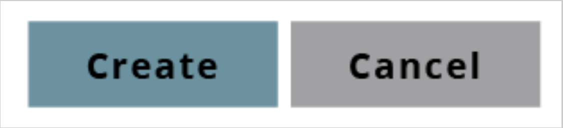


Figure 4.24 An example of confirmation and cancellation buttons

Breadcrumbs

Breadcrumbs are used above screen titles and in some Table Headers in order to provide hierarchical navigation within a section. When used above a screen title, a Breadcrumb shows the directory structure that led to the current screen.

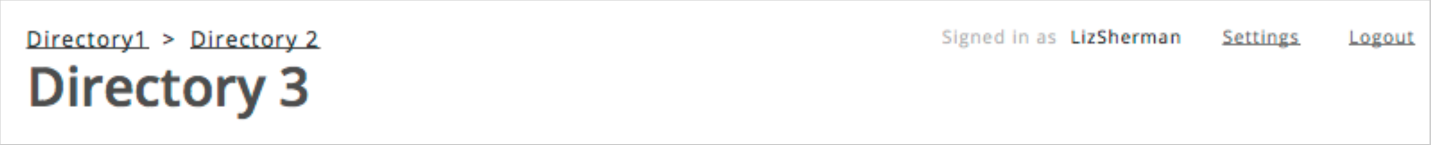


Figure 4.25 An example of a breadcrumb header

When used in a Table Header, Breadcrumbs describe the directory structure that led to the content currently displayed in the Table itself.

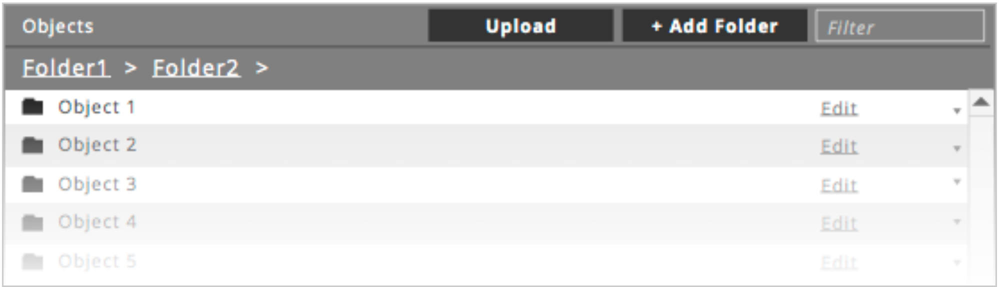


Figure 4.26 An example of a table breadcrumb

By selecting a directory in the breadcrumb, the user navigates to that directory. This is the preferred navigational structure for content trees requiring multiple levels of hierarchy.

Flash Notifications

Flash Notifications are temporary bars containing status updates and may contain buttons directing the user to a screen in which they can resolve an issue related to the notification. Flash Notifications should be displayed upon initiation and completion/failure of actions the user has initiated (such as terminating an instance) but may also be used to surface any critical message not directly related to the users current activity.

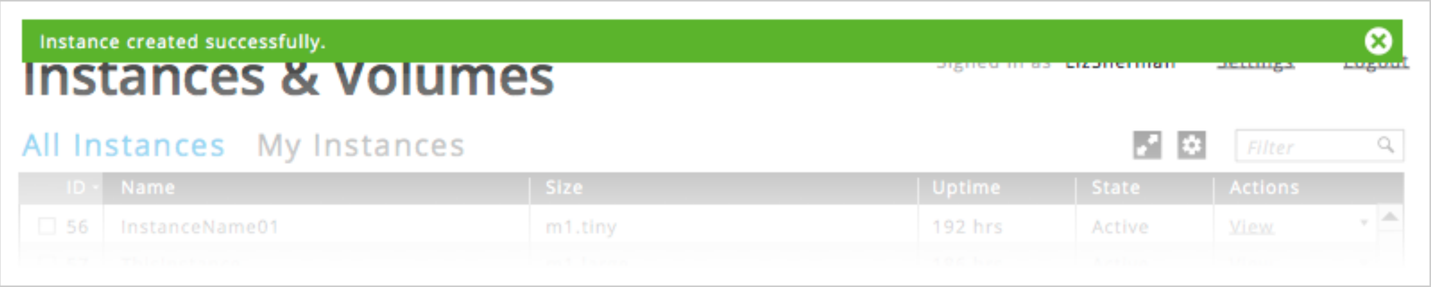
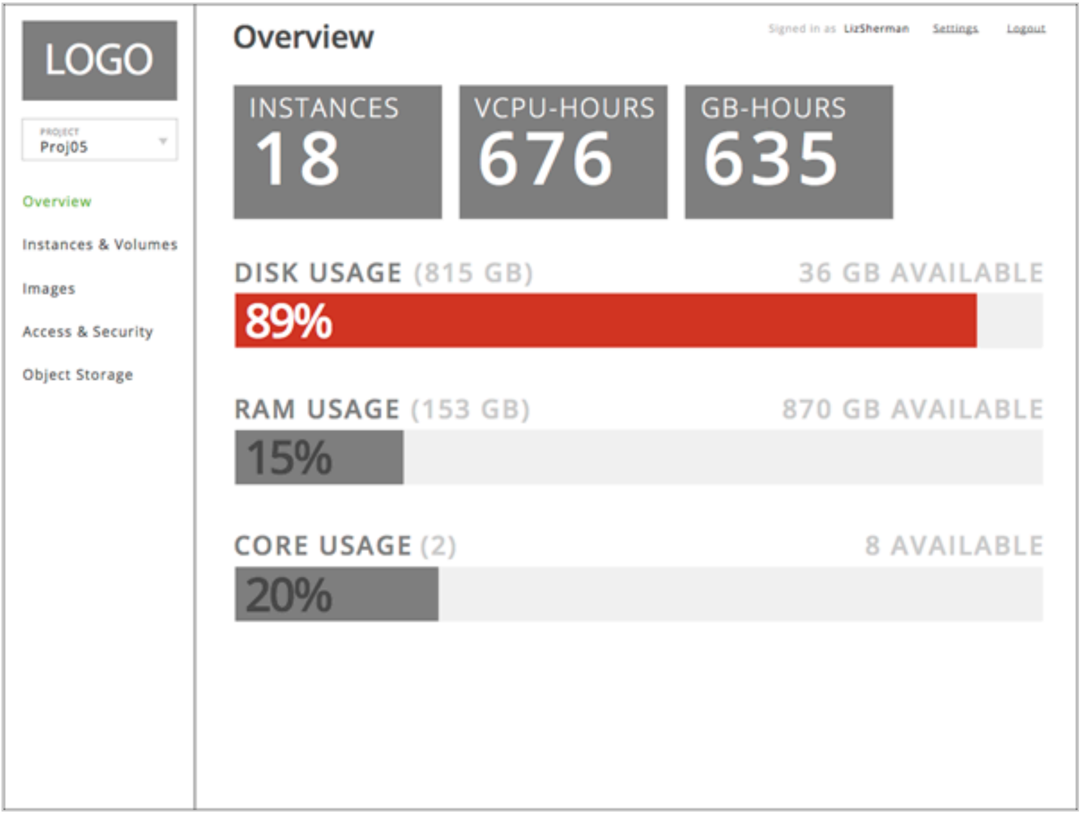


Figure 4.27 An example of a flash notification

SCREEN EXAMPLES

This section will showcase several screens in the OpenStack dashboard and the ways in which they adhere to the HIG. They may be used as the foundation for new designs.

Dashboard Overview Screen



Description
The Dashboard Overview Screen is an example of a relatively free-form Content Screen.

Design Reasoning
As an overview for the entire system, high-level statistics have been pulled out for visual emphasis. Numbers with no upper limits (Instances, VCPU-Hours, and GB-Hours) are shown as simple, labeled numbers while known percentages (Disk Usage, RAM Usage, and Core Usage) are displayed as large bar graphs visualizing the total amount of system usage. Color is used only to emphasize statistics displaying critical statuses.

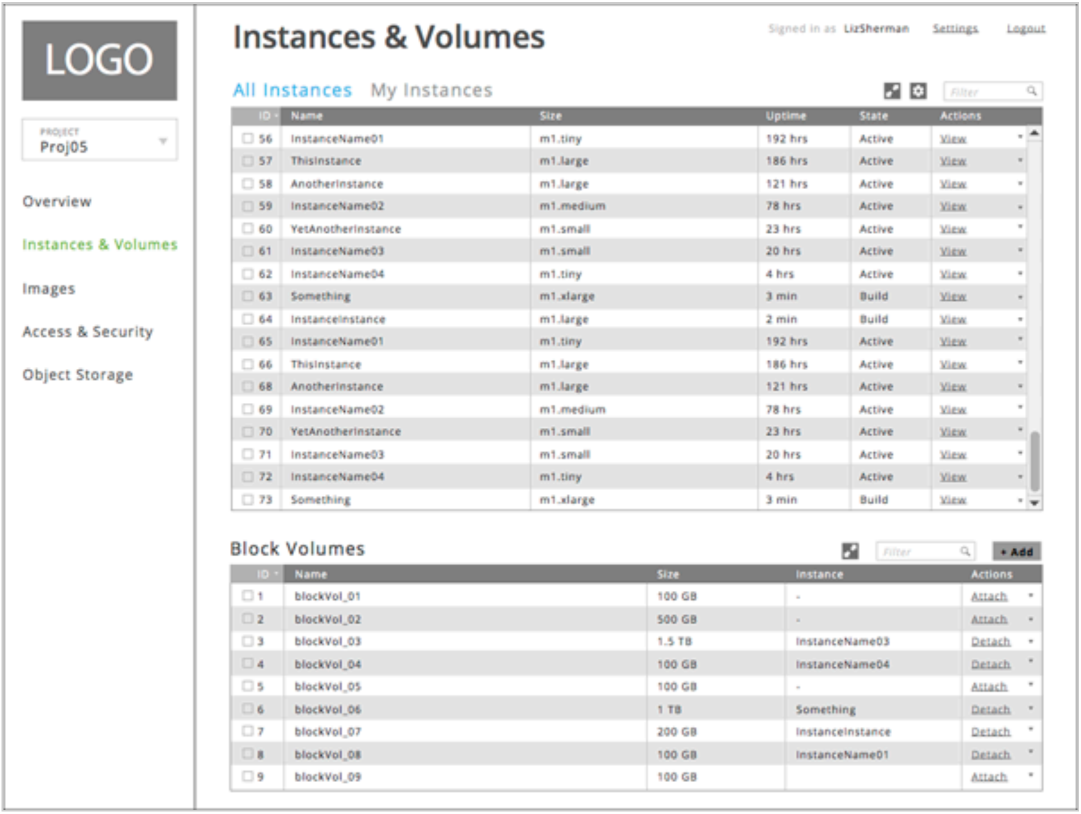
Evaluation
Consistency: This screen does not use any of the major components. Consistency, in this case, would be evaluated on the adherence to the visual design language (color and typography). Simplicity: Data is displayed in a straightforward, quickly accessible manner.

The amount of information displayed is not overwhelming.

User-Centered Design: The data that is shown represent the important, high-level statistics a user would wish to know.

Transparency: As the opening screen, an overview on system statistics provides a good level of transparency for the user.

Instances & Volumes Overview



Instances & Volumes

	ID	Name	Size	Uptime	State	Actions
□	56	InstanceName01	m1.tiny	192 hrs	Active	View
□	57	ThisInstance	m1.large	186 hrs	Active	View
□	58	AnotherInstance	m1.large	121 hrs	Active	View
□	59	InstanceName02	m1.medium	78 hrs	Active	View
□	60	YetAnotherInstance	m1.small	23 hrs	Active	View
□	61	InstanceName03	m1.small	20 hrs	Active	View
□	62	InstanceName04	m1.tiny	4 hrs	Active	View
□	63	Something	m1.xlarge	3 min	Build	View
□	64	InstanceInstance	m1.large	2 min	Build	View
□	65	InstanceName01	m1.tiny	192 hrs	Active	View
□	66	ThisInstance	m1.large	186 hrs	Active	View
□	68	AnotherInstance	m1.large	121 hrs	Active	View
□	69	InstanceName02	m1.medium	78 hrs	Active	View
□	70	YetAnotherInstance	m1.small	23 hrs	Active	View
□	71	InstanceName03	m1.small	20 hrs	Active	View
□	72	InstanceName04	m1.tiny	4 hrs	Active	View
□	73	Something	m1.xlarge	3 min	Build	View

Block Volumes

	ID	Name	Size	Instance	Actions
□	1	blockVol_01	100 GB	-	Attach
□	2	blockVol_02	500 GB	-	Attach
□	3	blockVol_03	1.5 TB	InstanceName03	Detach
□	4	blockVol_04	100 GB	InstanceName04	Detach
□	5	blockVol_05	100 GB	-	Attach
□	6	blockVol_06	1 TB	Something	Detach
□	7	blockVol_07	200 GB	InstanceInstance	Detach
□	8	blockVol_08	100 GB	InstanceName01	Detach
□	9	blockVol_09	100 GB		Attach

Description
The Instances & Volumes Overview screen exemplifies the typical Content Screen initially shown for a selected section.

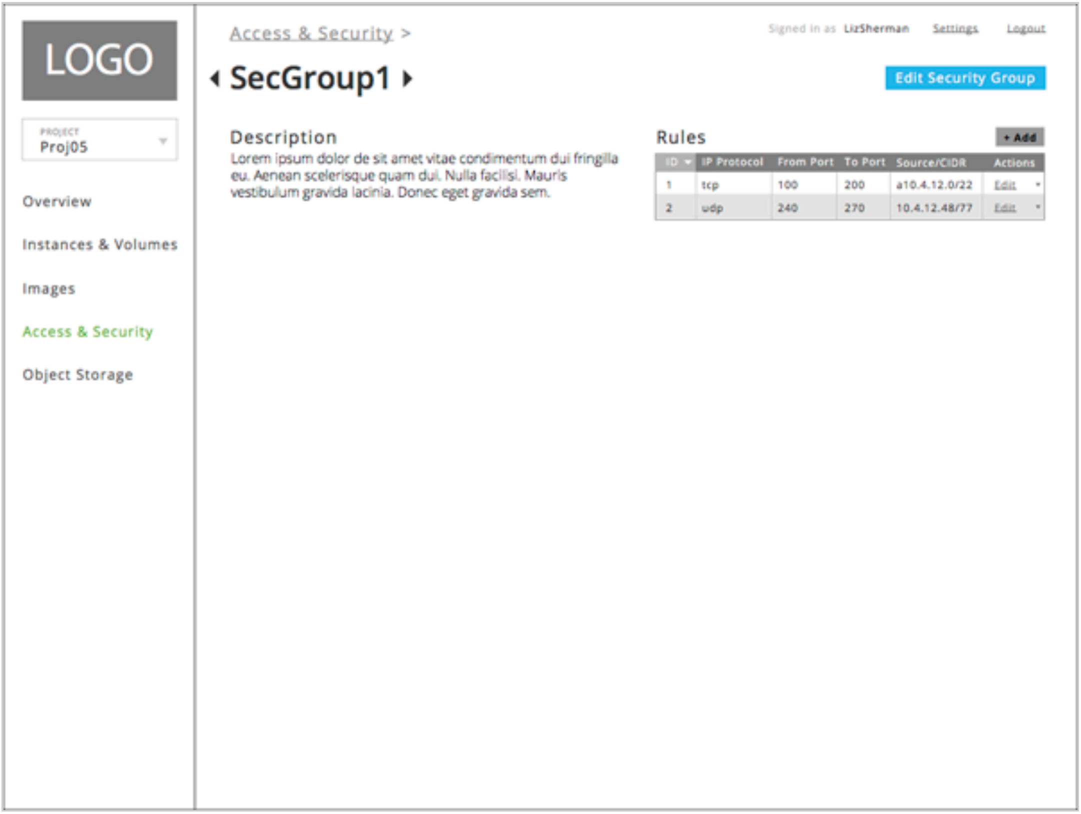
Design Reasoning
To describe specific details about the instance and volume objects, two tables have been used with fixed heights to ensure that both tables are always visible on this screen. The instances table uses Header Tabs to allow the user to either view all instances or only the instances that they have initiated.

Evaluation
Consistency: This type of screen, because it primarily uses an established component and does not introduce new user interactions, should be evaluated for consistency with documented table behavior/layout. Simplicity: Several available columns in the instances table have been omitted to simplify the amount of data shown and may be accessed through the View Option button.

User-Centered Design: From this screen, the user can take direct actions upon instances and volumes, such as terminating an instance or attaching a volume to an instance.

Transparency: The tables on this screen surface the status for all instances and volumes in the system.

Security Group Detail



Lorem ipsum dolor de sit amet vitae condimentum dui fringilla eu. Aenean scelerisque quam dui. Nulla facilisi. Mauris vestibulum gravida lacinia. Donec eget gravida sem.

ID	IP Protocol	From Port	To Port	Source/CIDR	Actions
1	tcp	100	200	a10.4.12.0/22	Edit
2	udp	240	270	10.4.12.48/77	Edit

Description

This screen exemplifies a Content Screen for a detailed view of an object, in this case a security group.

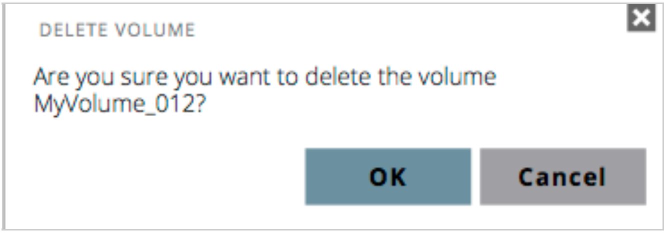
Design Reasoning

The Breadcrumb allows for navigation back to the Access & Security overview screen and the navigational arrows on either side of the SecGroup1 title allow the user to navigate between the next and previous peers to this object. This screen also uses description text and a table to describe the specifics of the security group.

Evaluation

Consistency: Because this screen acts as a breadcrumbed detail screen, it should be evaluated for consistency with the way other breadcrumbed detail screens have been designed as well as use of the table component. Simplicity: This screen displays a manageable level of content. User-Centered Design: To avoid navigating back and forth between the Access & Security Overview screen and the security group detail screens, the use of peer navigation arrows allows the user to easily browse their security groups. Transparency: N/A

Confirmation Modal



Are you sure you want to delete the volume MyVolume_012?

Description

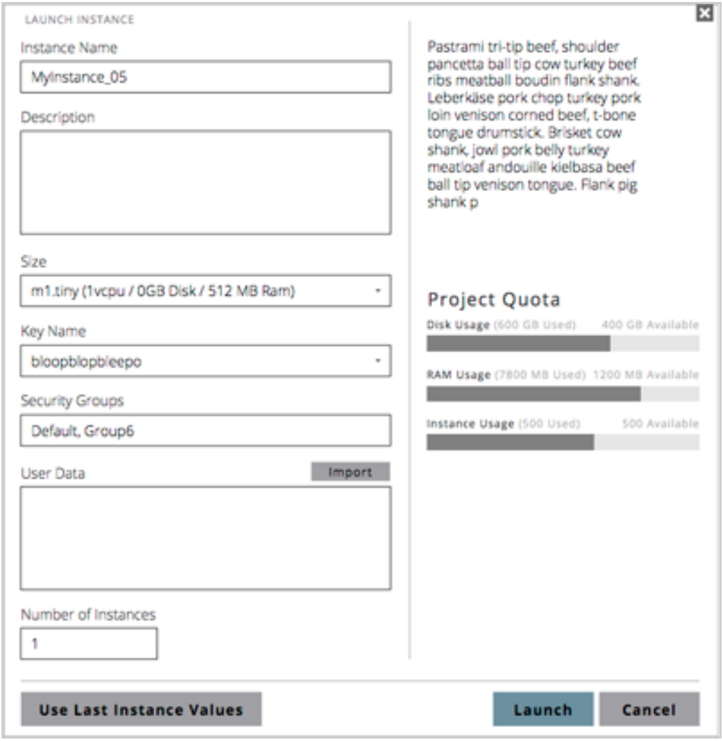
An example of a simple modal, this screen shows how a simple confirmation modal should work.

Design Reasoning

As with all modals, the presence of a close button and confirmation/cancellation buttons provides a redundant structure for exiting the modal.

Evaluation
Consistency: Evaluation of this screen would focus on making sure all required modal elements are present. Simplicity: N/A User-Centered Design: N/A Transparency: N/A

Launch Instance Modal



LAUNCH INSTANCE

Instance Name: MyInstance_05

Description:

Size: m1.tiny (1vcpu / 0GB Disk / 512 MB Ram)

Key Name: bloopblopbleepo

Security Groups: Default, Group6

User Data:

Number of Instances: 1

Pastrami tri-tip beef, shoulder pancetta ball tip cow turkey beef ribs meatball boudin flank shank. Leberkäse pork chop turkey pork loin venison corned beef, t-bone tongue drumstick. Brisket cow shank, jowl pork belly turkey meatloaf andouille kielbasa beef ball tip venison tongue. Flank pig shank p

Project Quota

Disk Usage (600 GB Used) 400 GB Available

RAM Usage (7800 MB Used) 1200 MB Available

Instance Usage (500 Used) 500 Available

Description
The Launch Instance Modal provides an example of a complex modal screen that includes more than a simple form. The form itself uses a variety of elements such as text fields, text boxes, and dropdown menus.

Design Reasoning
The User Data text box includes an Import button to allow the user to reuse already existing User Data rather than recreating this data each time an instance is launched. The Use Last Instance Values button also allows the user to quickly launch an instance identical or similar to their previously launched instance. The information on the right side of the modal provides guiding information about how to launch an instance and the current usage of the project quota.

Evaluation
Consistency: This modal would be evaluated based on its adherence to similar complex modals and the use of form components.

Simplicity: This modal asks for the minimum number of user inputs in order to launch an instance. Also, all interactive, required elements are displayed on the left side of the modal while supporting information is providing on the right, providing clear distinction between the two sections.

User-Centered Design: Use of the Import and Use Last Instance Values buttons prevents repetitious interactions with this often-used modal.

Transparency: Supplying information about the current quota usage ensures that the user allocates the appropriate amount of resources before attempting to launch the instance.

CORE VISUAL DESIGN LANGUAGE

This section defines the use of color, typography, and iconography to be used when designing new elements and screens. In general, components already existing in the OpenStack dashboard should be used as they are currently implemented.

Color Palette

The OpenStack palette is defined as follows. Additional grays may be used, but each design should minimize the number of colors it introduces.

OpenStack Red

🟥

#D03229 This color should be used sparingly for selected states and important elements. Red should also be used on buttons that denote destructive actions, such as terminating an instance or deleting a user.

OpenStack Teal

🟦

#6C92A0 This color should be used as a highlight color for elements requiring differentiation. Avoid using this color for large amounts of text. Teal should be used on buttons that denote the confirmation of a modal screen, such as "Update" or "Ok." If this confirmation is destructive in nature, such as "Delete," red should be used instead.

Dark Gray

⬛

#6E6E6E This general purpose gray can be used as the primary color for all elements.

Light Gray

⬜

#F3F8FA This general purpose light gray should be used mostly as a background color.

White

⬜

#FFFFFF White should be used mostly as a background color but may be used on top of a dark color for some content.

Typography

The OpenStack dashboard uses Anivers font family. Download Anivers for free: http://www.exljbris.com/anivers.html

Iconography

OpenStack iconography uses simple, two-dimensional, monochromatic images to represent various actions and statuses. Iconography within the OpenStack dashboard should not be reused outside of their original context. When developing new icons, align the style as close as possible to existing OpenStack iconography.

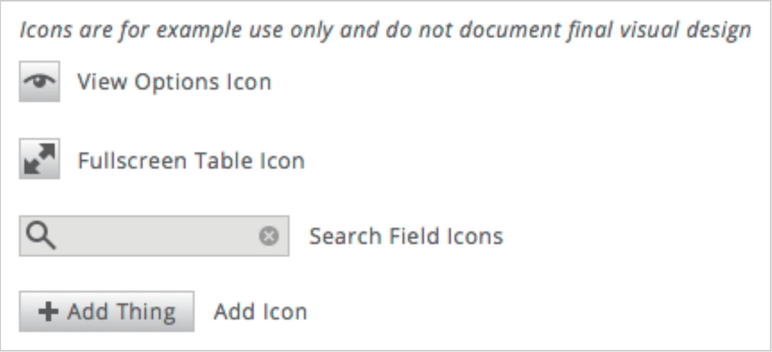

Icons are for example use only and do not document final visual design

Figure 6.1 An example of OpenStack iconography

GETTING STARTED

To create a design blueprint, download one of the design templates:

Illustrator: [link to illustrator file] Omnigraffle: [link to omnigraffle file]

These templates provide a structure to document the screens associated with your design and flows to help explain how new features work. When you've finished documenting your proposal, export a PDF and post it to the community. That's it! If you have any questions about the HIG or the OpenStack design process, feel free to contact [contact information].