
Docs Documentation

Release latest

Srikanth V

Aug 24, 2020

RLCatalyst Overview

1	About RLCatalyst	3
1.1	RLCatalyst DevOps Platform	4
2	Getting Started	5
2.1	Install RLCatalyst	5
2.2	Working with RL Catalyst	6
2.3	Basic Scenarios - Using RLCatalyst Instance	6
2.4	Advanced Features with Cloud Providers	7
2.5	Advanced Continuous Integration & Continuous Deployment [CI/CD] Features	7
3	Introduction - RLCatalyst	9
4	Settings	11
4.1	Organization Configuration	11
4.2	Config Management	17
4.3	Users Configuration	27
4.4	Provider Configuration	30
4.5	Gallery Setup	38
4.6	DevOps Roles	47
4.7	Track Setup	51
4.8	CMDB Configuration	52
4.9	BOTs	53
4.10	BOT Executor	53
5	Design	57
6	Workzone	67
6.1	Launch New Instances	67
6.2	Update Configuration	69
6.3	Connect to the Instance	71
6.4	Start Instance	72
6.5	Stop Instance	73
6.6	Import By IP	73
6.7	Cloud Formation Templates	75
6.8	Docker Blueprints	77
6.9	Control Panel	79
6.10	Orchestration	81

6.11	Application Deployment	84
6.12	App Blueprint	85
6.13	Launching Blueprint	86
6.14	Deploy New App	88
7	Track	93
8	Cloud	95
8.1	Cost	95
8.2	Capacity	99
8.3	Usage	103
9	BOTs	107
10	Installation	111
10.1	Install Using QuickInstaller	111
10.2	Install using Docker	112
10.3	Install using Vagrant	112
10.4	Install from AWS AMI	113
10.5	Install Directly From Source	113
10.6	Working With Chef	121
11	Code Contributions	127

The main documentation for the site is organized into a couple sections:

- *RLCatalyst Overview*
- *User Documentation*
- *Developer Documentation*

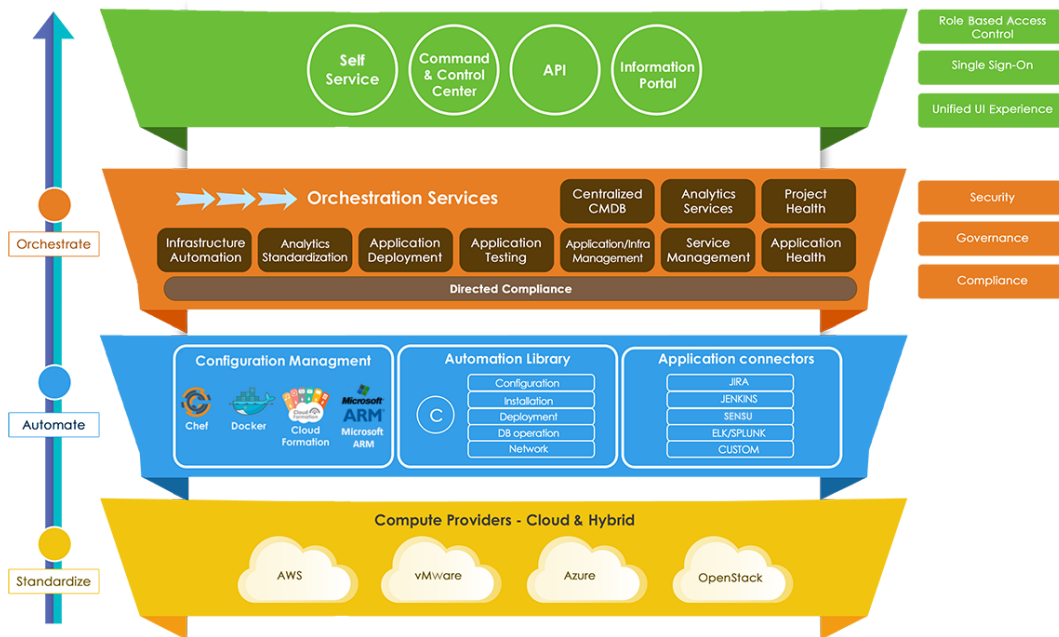
CHAPTER 1

About RLCatalyst

RLCatalyst is an end-to-end automation platform that helps enterprises adopt devops maturity and benefits. Catalyst is powered by Chef and integrated with all major cloud providers like AWS, Azure, Openstack, VMware etc. It provides seamless Infrastructure Automation across data centers, environments, applications with Configuration Management & Service Orchestration to help enterprises achieve end-to-end IT DevOps Service Automation and being prepared for Web-scale IT. Few of the highlights of Catalyst are:

- It helps in adopting intelligent devops – from Unmanaged->Managed->Self-Service->Self-Aware->Self Heal Infrastructure
- It does Infrastructure automation, provisioning, Orchestration and management
- It helps automating the entire ALM cycle from Continuous Integration->Testing->Continuous Deployment and works with all major CI/CD tools
- It gives realtime-dashboard and alerts-based monitoring and remediation of cost, usage , health and performance of all IT assets
- Powered by Chef and integrated with Docker

1.1 RLCatalyst DevOps Platform



How you can make use of Catalyst:

- **Infrastructure Provisioning and Management:** Do you need to provision infrastructure dynamically? Do you want to manage your heterogeneous environments? Do you want to control usage and cost? or Do you want to identify and retire your unused infrastructure? RLCatalyst has the solutions to all these challenges. It helps you towards a more efficient capacity planning and improved utilization
- **Application Deployment:** RLCatalyst provides you a seamless experience of managing your ALM lifecycle, with its one-click application deployment, on any of your cloud providers. You can reduce your deployment time from weeks to days to hours with better quality with a focus on performance and health of the application. It works with all latest CI/CD tools- Jenkins, JIRA, BitBucket, Github, SonarQube etc to name a few.
- **Monitoring and Tracking :** RLCatalyst provides you the near-real time information on the cost and usage of your infrastructure . This helps you to keep track of Cloud Capacity and to optimize your resources to ensure better utilization.

2.1 Install RLCatalyst

RLCatalyst can be installed either directly from source or using one of our quickinstallers

Quick Installer

RLCatalyst can be installed using one of our Quick Installers

Quick install using RLCatalyst Installer Script

RLCatalyst can be installed quickly using a chef-based installer. This installer will be available for installing on Centos and Ubuntu . The catalyst installer will install RLCatalyst and Open Source Chef. Basic seed data to start the application will also be taken care by the installer

Ubuntu 14.04 and Centos

You can install RLCatalyst using the installer script

```
wget https://raw.githubusercontent.com/ROpenCatalyst/installer/master/installer.sh
chmod +x installer.sh
./installer.sh
```

Install using Docker

Download the docker image from:

```
https://hub.docker.com/r/relevancelab/rlcatalyst/
```

and run these commands:

```
docker pull relevancelab/rlcatalyst:3.5.0
docker run -t -i -p 3001:3001 --name catalyst331 relevancelab/rlcatalyst:3.5.0 /bin/
↵bash
```

Quick install using Vagrant

Setup RLCatalyst on your dektop/laptop using vagrant. You need to have vagrant installed in your machine:

RL Catalyst Installation on Vagrant with chef:

```
git clone https://github.com/ROpenCatalyst/installer.git
cd vagrantwithchef
vagrant up
```

RL Catalyst Installation on Vagrant without chef:

```
git clone https://github.com/ROpenCatalyst/installer.git
cd vagrant
vagrant up
```

Quick install using RLCatalyst Public AWS Image

If you have an AWS account, you can bring up RLCatalyst using the public AMI available. The public image is currently available for US east(N.Virginia) region. This comes with some basic configurations required by RLCatalyst

1. From your EC2 dashboard, select N.California region . In the Images/AMI link, choose “Public Images” in the dropdown . Search for image with AMI ID `ami-f4270294` and AMI Name `RLCatalyst4.0.0-stage`
2. Select the image and hit Launch
3. On the “Instance Type” page ,choose the instance size as t2.medium or bigger . We recommend atleast 4 GB RAM
4. On the “Configure Instance Details” page, choose your preferred Network and Subnets. If you want to assign a public IP to RLCatalyst, then enable “Auto-assign Public IP”
5. On “Tag Instance” , name your instance
6. On “Security Group” , add rule to open ports 443, 8080,8081 and 3001.
7. Review and launch . Once the instance is in “Running” state , you can access RLCatalyst at <http://<ip>:3001> . Login using `superadmin/superadmin@123`
8. This image includes the devops role like jenkins at <http://<ip>:8080> (default credentials)
9. This image includes the devops role Nexus at <http://<ip>:8081> (default credentials)

Following video demonstrates how to Quick install using RLCatalyst public AWS Image:

Install from Source

Refer to *Install Directly From Source* to install RLCatalyst directly from source

2.2 Working with RL Catalyst

2.3 Basic Scenarios - Using RLCatalyst Instance

Once you install RLCatalyst , basic data to quick-start your experience will be available within the application - Organization setup, chef server , environments, templates, services etc . You can start exploring by importing RLCatalyst instance into Catalyst

Listed below are few features which RLCatalyst offers out of the box:

- Import Linux Node and Install latest version of-Tomcat

- Install Apache on imported node and use service to stop, start and restart-Apache
 - Deploy Pet-clinic application in the imported node
 - Update Petclinic application-version
 - View History of App deployments &-upgrades
 - Deploy a multi-tier application using docker-container
-

2.4 Advanced Features with Cloud Providers

Prerequisite : An AWS account should be available

RLCatalyst comes with the flexibility to create blueprints to automate dynamic provisioning on the cloud provider of your choice . Currently AWS, Azure, VMware and Openstack are supported. To start experiencing, add your provider account details in RLCatalyst

- Launch New Ubuntu Instance and Install-Jboss
 - Deploy Windows App on-IIS
 - Launch New ubuntu Instance, Install Tomcat, upgrade to-v8.0[attribute]
 - Provider Sync and-Import Instances
 - AWS Cost, Usage-Dashboards
 - Deploy Wordpress on multiple docker-container
 - Create CentOS instance and launch-Liferay
 - Create and Launch an AWS CFT Blueprint for 2-node Petclinic
 - Install LAMP Stack on a single-node
 - Allocate and Track AWS-Resources
-

2.5 Advanced Continuous Integration & Continuous Deployment [CI/CD] Features

Jenkins is CI/CD tool which can be used for build and deployment automation. It also allows you to continuously deliver your software by providing powerful ways to define your build pipelines and integrating with a large number of testing and deployment technologies.

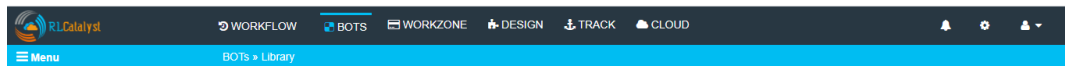
How to Configure, Create, Execute Jenkins Jobs and View History in RLCatalyst ?

- Configure and Execute a Jenkins-Jobs
 - Deploy and Promote-a Java Application
 - Deploy-a 3-tier application
 - Design and Launch-Application
 - Design and Launch 3-tier-Application
-

Introduction - RLCatalyst

RL Catalyst has four main menu options -

- *Settings* : Create the Organization structure and manage DevOps assets across data center and cloud
- *BOTs* : This Dashboard contains BOTs library and BOTs report and audit trail page.
- *Design* : Design re-usable templates and blueprints which are used to automate your various DevOps requirements
- *Workzone* : Manage and Control all your assets from Workzone. This is the execution zone in RLCatalyst
- *Track* : CI/CD metrics and Monitoring dashboards will be part of Enterprise and PAAS versions
- *Cloud* : This dashboard consists Infrastructure Cost and Usage



Add all your organization and asset details in Settings before using RLCatalyst. You can use the system only if you provide all the relevant options provided in the Settings.

Settings provides you eight different options:

4.1 Organization Configuration

To start using RL Catalyst, the first step is to setup an organization. While you setup an organization in RLCatalyst, you should also set up the Business Groups and Projects under the organization. An Organization can be an enterprise or a business that can have multiple sub groups and projects. An organization can have multiple Business Groups and each Business Group can have multiple Projects

4.1.1 Organization

An Organization can be an enterprise or a business that can have multiple sub groups and projects.

Follow the steps below to create an organization

Through Setup & Configuration Wizard:

- From the main menu click on Settings
- Click on **+New OrgSetup** in **Setup & Configuration Wizard**
- In Create Organization panel enter Name (Organization Name), Domain Name (Website Address) and Planned Cost.

RLCatalyst WORKZONE DESIGN TRACK ANALYTICS SUPERADMIN [Admin] SETTINGS ? Help

Menu Settings

Setup & Configuration Wizard Cancel Wizard

Start 1 2 3 4 5 6

Introduction Org Configuration Config Management User Configuration Provider Configuration Gallery Setup Devops Roles

1 Organization 2 Business group 3 Project

Create Organization

Name: * Domain Name:

Planned Cost: ?

Save Next

- Click on Save button
- Now Organization is created

Through Menu:

- From the main menu click on Settings
- Once you click on Settings, from the side menu click on Organization Configuration
- Click Organizations
- Click on New button provided
- In Create Organization page enter Name (Organization Name) and Domain Name (Website Address)

RLCatalyst WORKZONE DESIGN TRACK ANALYTICS SUPERADMIN [Admin] SETTINGS ? Help

Menu Settings

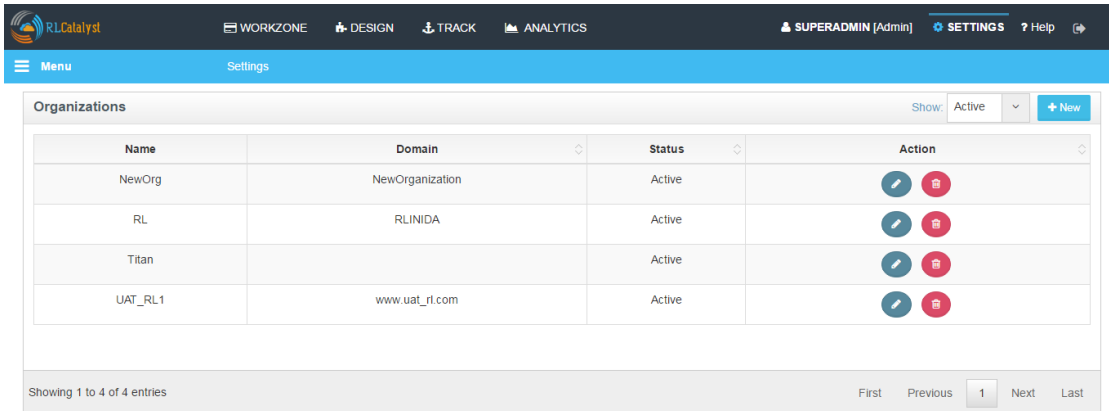
Create Organization

Name: *available Domain Name:









Planned Cost: ?

Cancel Save

- Click on Save button
- Now your Organization is setup and listed in the Organizations page



The screenshot shows the RLCatalyst interface with the 'Settings' tab selected. The 'Organizations' section displays a table with the following data:

Name	Domain	Status	Action
NewOrg	NewOrganization	Active	 
RL	RLINIDA	Active	 
Titan		Active	 
UAT_RL1	www.uat_rl.com	Active	 

Below the table, it says 'Showing 1 to 4 of 4 entries'. Navigation links for 'First', 'Previous', '1', 'Next', and 'Last' are visible.

Edit and Remove an Organization You can edit and remove an Organization.

To Edit the Organization

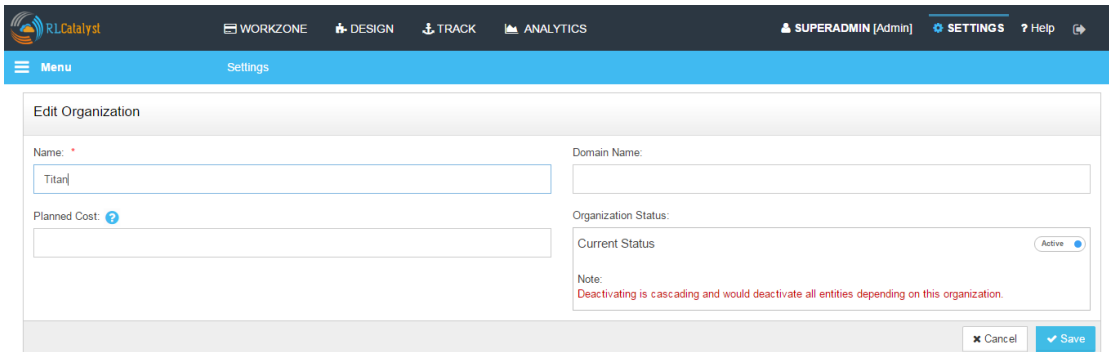
- Click on edit button to edit the organization details

To Remove the Organization

- Click on delete button to remove the organization from the list

Activate or Inactivate an Organization

You can activate or inactivate an Organization by using the cursor button provided. Remember, once you inactivate the organization all the entities relating to that organization will be disabled. You can enable the organization at any time, which will activate all the entities associated with that organization



The screenshot shows the 'Edit Organization' form in the RLCatalyst interface. The form contains the following fields and controls:

- Name:** A text input field containing 'Titan'.
- Domain Name:** An empty text input field.
- Planned Cost:** An empty text input field with a help icon.
- Organization Status:** A section containing a 'Current Status' label and a radio button labeled 'Active'.
- Note:** A red text note stating 'Deactivating is cascading and would deactivate all entities depending on this organization.'
- Buttons:** 'Cancel' and 'Save' buttons at the bottom right.

4.1.2 Business Groups

Setup Business Groups for an Organization

In an Organization you can create multiple Business Groups. A Business Groups can be a business division in your organization

Follow the steps below to create a new Business Group in an Organization

Through Setup & Configuration Wizard:

- From the main menu click on Settings
- After saving the organization click on Next button
- Click on **+Add New BG** button in **Create Business Group** panel
- Enter the Business Group name in the Business Group name field
- Click on Save button

The screenshot displays the RLCatalyst Setup & Configuration Wizard. The top navigation bar includes the RLCatalyst logo, menu items (WORKZONE, DESIGN, TRACK, ANALYTICS), user information (SUPERADMIN [Admin]), and settings/help links. The wizard progress bar shows six steps: Start, 1 Org Configuration, 2 Config Management, 3 User Configuration, 4 Provider Configuration, 5 Gallery Setup, and 6 Devops Roles. Below the progress bar, a green header indicates the current step is '1 Organization'. The main content area is titled 'Create Business Group' and features a text input field for 'Business Group Name' (containing 'RelBG') and a dropdown menu for 'Organization' (showing 'Rel'). A '+ Add New BG' button is located in the top right corner of the form. At the bottom right, there are 'Save' and 'Next' buttons.

Through Menu:

- From the main menu click on Settings
- Once you click on Settings, from the side menu click on Organization Configuration
- Click on Business Groups
- Click on New button provided
- Enter the Business Group name in the Business Group name field
- Select the Organization from the Organization drop down list
- Click on Save button

RLCatalyst WORKZONE DESIGN TRACK ANALYTICS SUPERADMIN [Admin] SETTINGS ? Help

Menu Settings

Create Business Group

Business Group Name *available

Organization *

Select an Organization

Cancel Save

- Now the Business Group is setup and listed in the Business Groups page

RLCatalyst WORKZONE DESIGN TRACK ANALYTICS SUPERADMIN [Admin] SETTINGS ? Help

Menu Settings

Business Groups + New

Name	Organization	Action
Demo	RL	
NewBG	NewOrg	
Product	RL	
RLBG	UAT_RL1	
titanbg	Titan	

Showing 1 to 5 of 5 entries First Previous 1 Next Last

Edit and Remove a Business Group You can edit and remove a Business Group.

To Edit the Business Group

- Click on edit button to edit Business Group details

To Remove the Business Group

- Click on delete button to remove Business Group from the list

4.1.3 Projects

Setup Projects for a Business Group

In a Business Group you can create multiple Projects. A Project in RLCatalyst can be a running project in your business group. Each project can run one or multiple applications

Follow the steps below to create a new Project associating with Business Group and Organization

Through Setup & Configuration Wizard:

- From the main menu click on Settings
- In Create Project panel enter Name of the project, Description and select Organization and Business Group from the drop down

The screenshot displays the RUCalyst Setup & Configuration Wizard. The top navigation bar includes 'WORKZONE', 'DESIGN', 'TRACK', 'ANALYTICS', and 'BOTS'. The user is logged in as 'SUPERADMIN [Admin]' and is in the 'SETTINGS' section. The wizard progress bar shows six steps: 1. Introduction, 2. Org Configuration, 3. Config Management, 4. User Configuration, 5. Provider Configuration, and 6. Gallery Setup. The 'Create Project' panel is active, showing a form with the following fields: 'Name' (text input), 'Description' (text input), 'Organization' (dropdown menu with 'Rel' selected), and 'Business Group' (dropdown menu with 'RelBG' selected). There are 'Save' and 'Next' buttons at the bottom right of the form.

- Click on Save button

Through Menu:

- From the main menu click on Settings
- Once you click on Settings, from the side menu click on Organization Configuration
- Click on Projects
- Click on New button provided
- Enter the Project name in the Name field
- Provide a brief description about the project in the Description field
- Select the Organization from the Organization drop down list
- Select the Business Group from the Business Group drop down list
- Environments is not mandatory while creating project and will be explained in later section

Create Project

Name: *available
RelPrj

Description:

Organization: *
Rel

Business Group: *
RelBG

Environments:

Cancel Save

- Click on Save button

Projects

Name	Description	Organization	Business Group	Environments	Action
AZURE	For Azure related demo setup	RL	Demo	View	
Catalyst	Catalyst project	RL	Product	View	
Catalyst	Catalyst_UAT_RL	UAT_RL1	RLBG	View	
DOT_NET	For .NET project	RL	Demo	View	
JAVA	JAVA project	RL	Demo	View	
NewProj	desc	NewOrg	NewBG	View	
RelPrj		Rel	RelBG	View	

Showing 1 to 7 of 7 entries

First Previous 1 Next Last

Edit and Remove a Project You can edit and remove a Project.

To Edit the Project

- Click on edit button to edit Project details

To Remove the Project

- Click on delete button to remove Project from the list

The following video will help you to setup an Organization, BusinessGroup, Project in RLCatalyst:

4.2 Config Management

In Config Setup we will cover Chef Server, Puppet Server and Environment:

The Chef server is the highly scalable foundation of the Chef automation platform. Create a flexible, dynamic infrastructure across multiple datacenters, public and private clouds, and in heterogeneous environments.

Puppet master is a Ruby application that compiles configurations for any number of Puppet agent nodes, using Puppet code and various other data sources. (For more info, see Overview of Puppet's Architecture.)

An Environment is a way to map an organization's real-life workflow to what can be configured and managed.

4.2.1 Chef Server

The Chef server acts as a hub for configuration data. The Chef server stores cookbooks, the policies that are applied to nodes, and metadata that describes each registered node that is being managed by the chef-client.

A node is any machine—physical, virtual, cloud, network device, etc.—that is under management by Chef.

A cookbook is the fundamental unit of configuration and policy distribution.

A policy file allows you to specify in a single document the cookbook revisions and recipes that should be applied by the chef-client.

RLCatalyst allows you to configure your own chef server. You can add either a hosted chef server or an on-premise installation of chefserver. If you don't have either of these, please create an account at <https://getchef.opscode.com/signup>. For more details on chef, please go to Chef Setup.

In RLCatalyst only one chef server can be configured for one organization. The same chef server cannot be associated to multiple organizations. Each chef server account will have a URL(Hosted/On-Premise), a User pem file, a Validator pem file and a knife file. You will get all these files when you create accounts in Chef.

Data/File needed for adding a Chef Server account in RLCatalyst

- Name : Alias or name of the chef server, to be identified in RLCatalyst
- User Name : User name of your chef server account
- URL : URL of the hosted or on-premise chef server
- User Pem File : User file to access your Chef Server account
- Knife.rb : Configuration file that you get while setting up the Chef server
- Validator Pem File : During the first chef-client run, this private key does not exist. Instead, the chef-client will attempt to use the private key assigned to the chef-validator, located in /etc/chef/validation.pem.
- Key File : It is used to encrypt the contents of the data bag item.
- Template File : The default bootstrap operation relies on an Internet connection to get the distribution to the target node. If a target node cannot access the Internet, then a custom template i.e., template file can be used to define a specific location for the distribution so that the target node may access it during the bootstrap operation.

To configure a new chef server follow the steps below:

- From the main menu click on Settings
- Once you click on Settings, from the side menu click on Config Management
- Click on Chef Server
- Click on New button provided
- Enter the configuration name in the Name field
- Enter the Chef user name in User Name field
- Provide or specify the Chef URL for the server to be configured
- Choose the organization from the Organization drop down list
- Upload the user PEM file provided by the Chef Server in the User PEM File field

- Upload the validator PEM file provided by the Chef Server in the Validator PEM File field
- Upload the Knife.rb file provided by the Chef Server in the Knife.rb File field
- Upload the key file which is used for Databag
- Upload the Template file which is used to Bootstrap node

Edit Configuration Management

Name:

User Name:

URL:

Organization:

User Pem File:

Knife.rb File:

Validator Pem File:

Key File:

Template File:

- Click on Save button
- Now your chef server is configured successfully and listed in the Chef server management page.

Chef Server Management

Chef Server Name	User Name	URL	Organization	Action
Cat1ChefServer	pcjoshi9	https://api.opscode.com/organizations/cat1	Phoenix	<input type="button" value="edit"/> <input type="button" value="add"/> <input type="button" value="share"/> <input type="button" value="folder"/> <input type="button" value="delete"/>
UAT_RL1	mycatqa	https://api.opscode.com/organizations/cattest	UAT_RL1	<input type="button" value="edit"/> <input type="button" value="add"/> <input type="button" value="share"/> <input type="button" value="folder"/> <input type="button" value="delete"/>

Showing 1 to 2 of 2 entries

First Previous 1 Next Last

Edit and Remove Chef Server configuration You can edit and remove a chef server configuration.

To Edit the Chefserver

- Click on edit button to edit chef server configuration details

To Remove the Chefserver

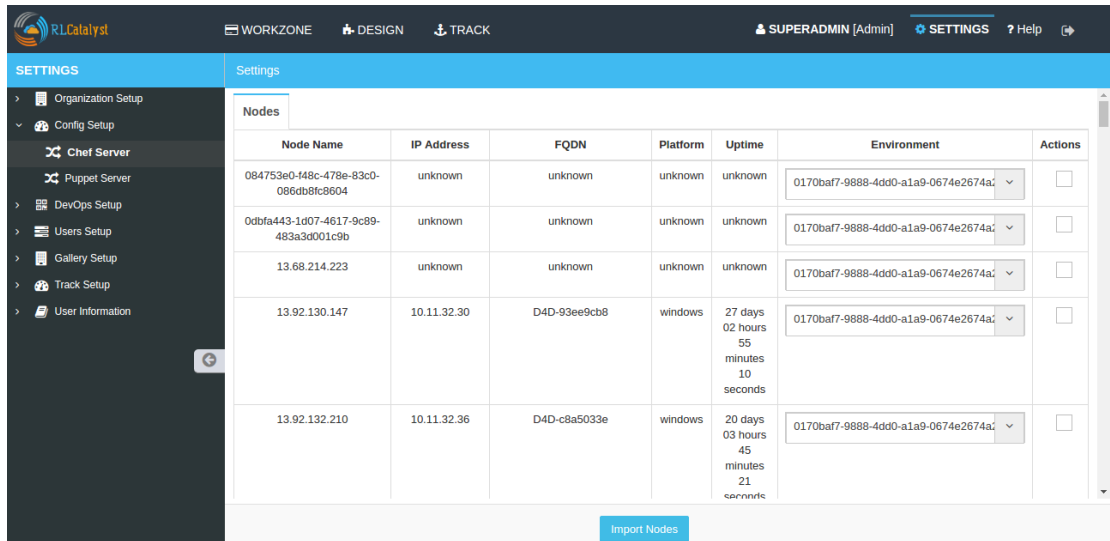
- Click on delete button to remove chef server configuration from the list

Following video demonstrates how to configure a chef server in RLCatalyst:

Import nodes from Chef Server

You can import existing nodes from the configured chef server into RLCatalyst by selecting the required Business Group and Project. These imported nodes can be operated from the Workzone.

- To import the existing nodes click on import button
- Select the environment from Environment drop down list for the node to be imported . These are the environments available in your Chef account
- Select the respective checkbox in the Action column



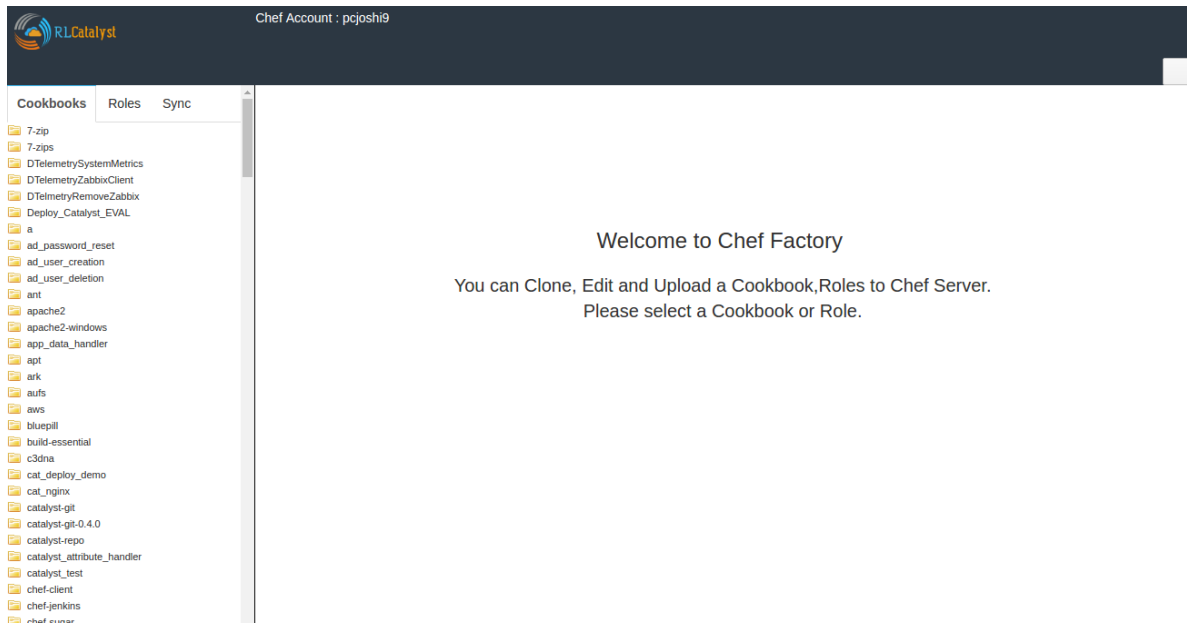
Node Name	IP Address	FQDN	Platform	Uptime	Environment	Actions
084753e0-448c-478e-83c0-086db8fc8604	unknown	unknown	unknown	unknown	0170ba7f-9888-4dd0-a1a9-0674e2674a	<input type="checkbox"/>
0dbfa443-1d07-4617-9c89-483a3d001c9b	unknown	unknown	unknown	unknown	0170ba7f-9888-4dd0-a1a9-0674e2674a	<input type="checkbox"/>
13.68.214.223	unknown	unknown	unknown	unknown	0170ba7f-9888-4dd0-a1a9-0674e2674a	<input type="checkbox"/>
13.92.130.147	10.11.32.30	D4D-93ee9cb8	windows	27 days 02 hours 55 minutes 10 seconds	0170ba7f-9888-4dd0-a1a9-0674e2674a	<input type="checkbox"/>
13.92.132.210	10.11.32.36	D4D-c8a5033e	windows	20 days 03 hours 45 minutes 21 seconds	0170ba7f-9888-4dd0-a1a9-0674e2674a	<input type="checkbox"/>

Import Nodes

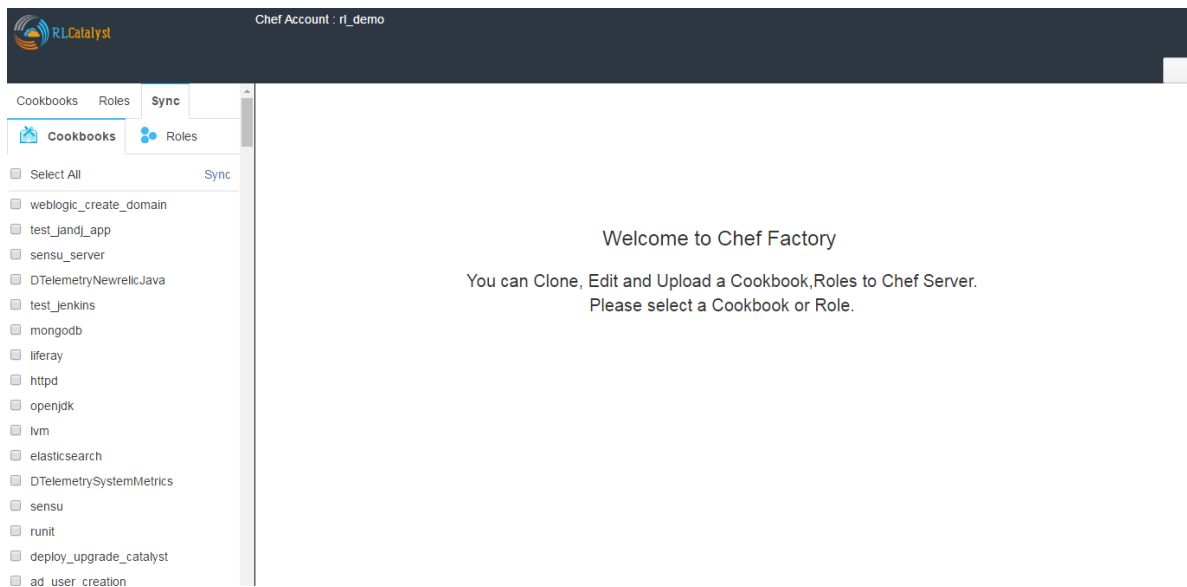
- Click on the Import Nodes button
- Select the business group from Business Group drop down list
- Select the project from the Project drop down list
- Enter the user name to access the chef server for import in the User Name field
- Choose authentication type from the Choose Authentication Type drop down list. RLCatalyst provides two types of authentication, you can either choose Password or by uploading the PEM file
- Type Password or upload PEM file
- Click on Import button
- Close the popup containing the success message 'Node imported'
- Click on Workzone
- The imported node will be available in the respective Environment of the Workzone

Chef Factory It consists of common and re-usable recipes and cookbooks.

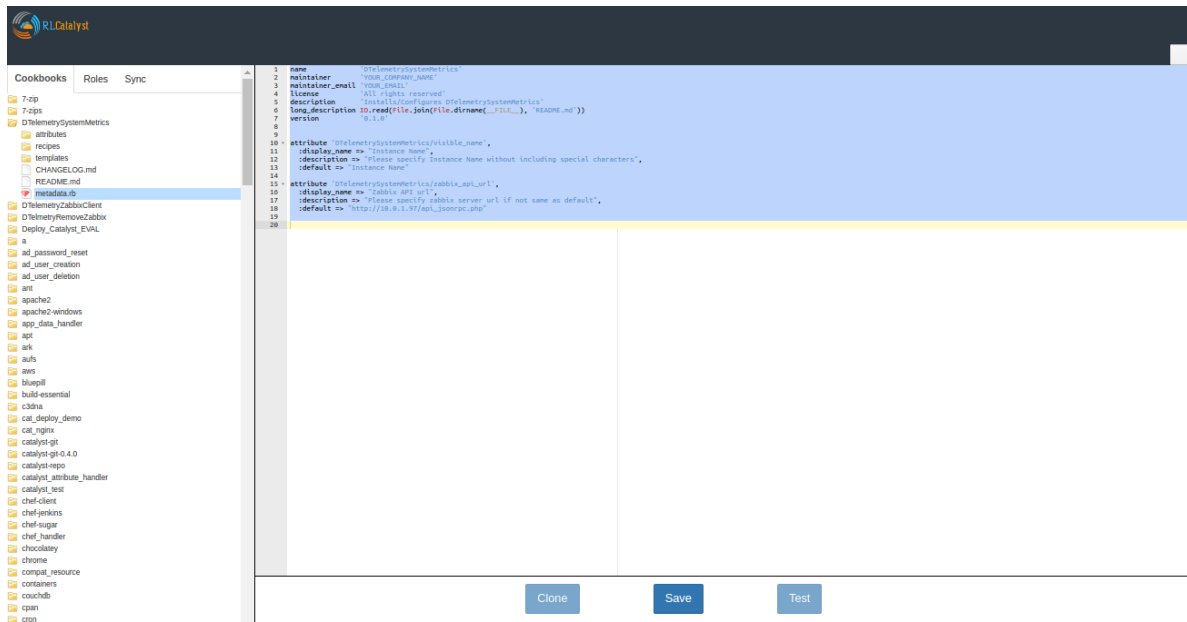
- Click on Chef factory icon present in the Action column , Chef factory page will open.



- Go to Sync tab, here all the cookbooks and roles which are present in the chef server will be listed.



- Select the Cookbook and click on Sync
- Close the popup window
- Go to Cookbooks tab, here you can find the downloaded (Synched) cookbook



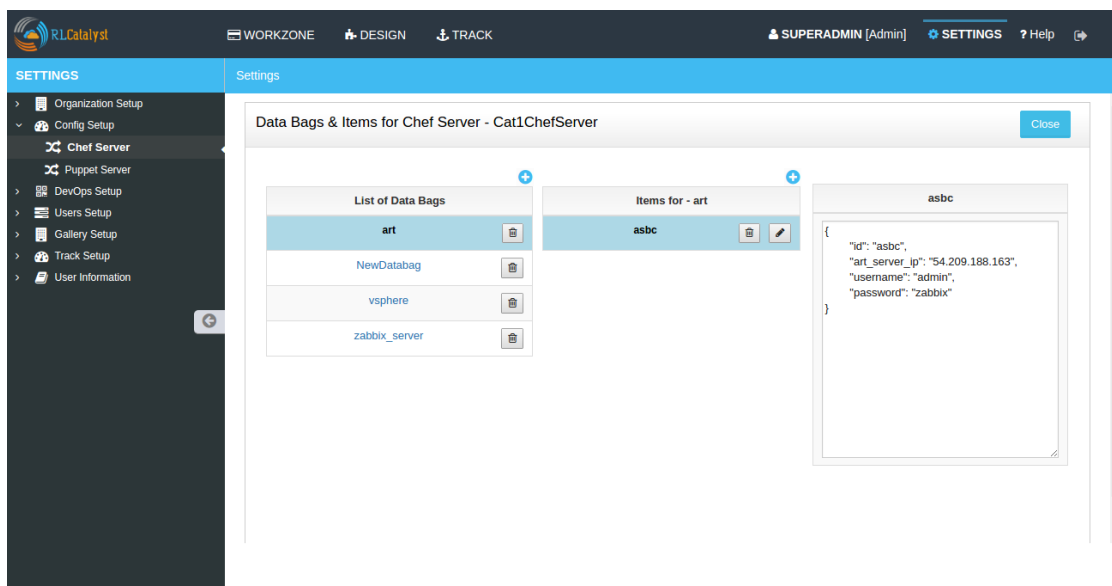
Databags and Items for Chef server

A data bag is a global variable that is stored as JSON data and is accessible from a Chef server. A data bag is indexed for searching and can be loaded by a recipe or accessed during a search.

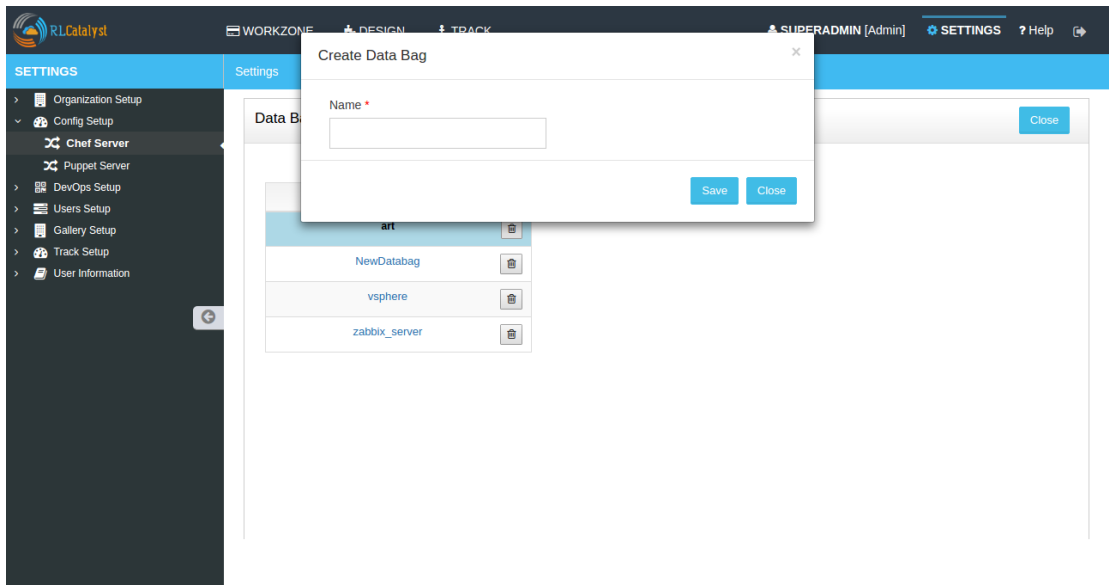
A data bag item may be encrypted using shared secret encryption. This allows each data bag item to store confidential information (such as a database password) or to be managed in a source control system (without plain-text data appearing in revision history). Each data bag item may be encrypted individually; if a data bag contains multiple encrypted data bag items, these data bag items are not required to share the same encryption keys.

How to create Databag and Items for Chef Server?

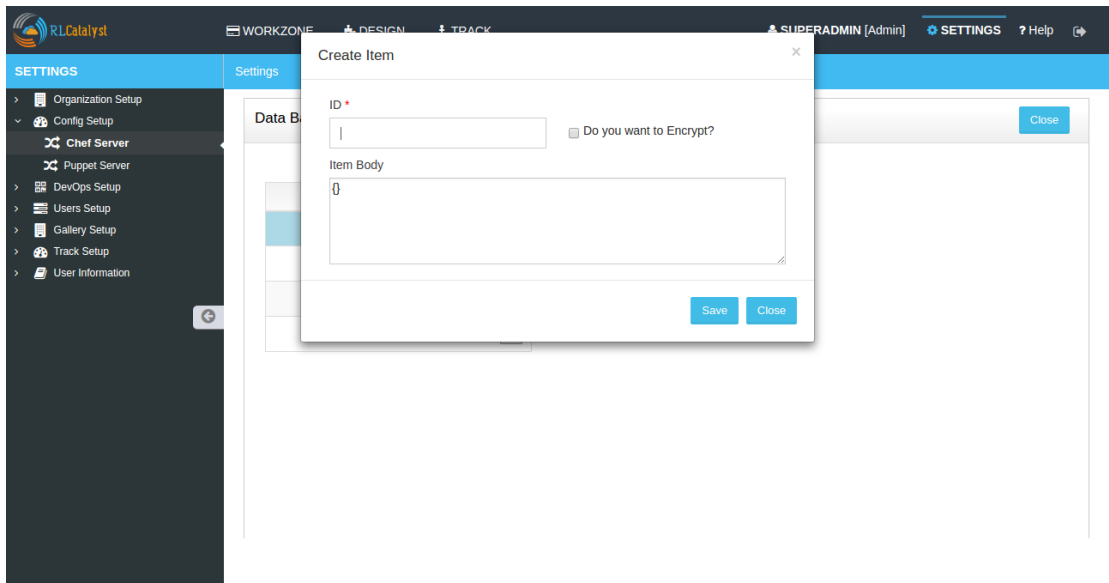
- In the Chef Server Page, Click on Databag icon in the Action column of your chef server
- Click on + icon above the List of Data Bags column header



- Enter the name of the Databag in the Name field



- Click on Save button
- Select the Created Databag and create an item by clicking + icon above the 'Items for -Databagname' column header
- Enter the ID and Item body



- Select the checkbox Do you want to Encrypt
- Click on Save button
- Now Databag and its item is created. Item body is shown in last column
- Click on Close button to navigate back to Chef server management page

4.2.2 Puppet Server

Puppet Server is an application that runs on the Java Virtual Machine (JVM) and provides the same services as the classic Puppet master application. It mostly does this by running the existing Puppet master code in several JRuby interpreters, but it replaces some parts of the classic application with new services written in Clojure.

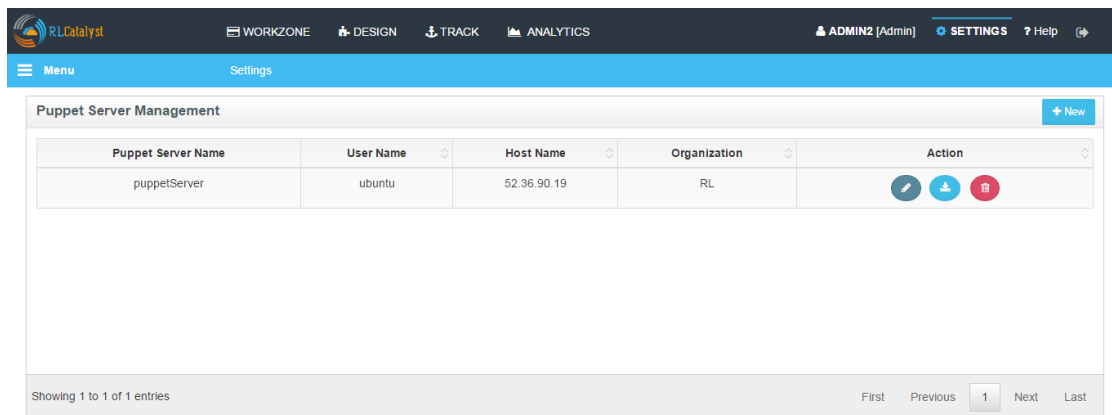
If you are using Puppet for your configuration management requirements, configure it in RLCatalyst. You can configure only one puppet server for one organization. Same puppet server cannot be associated with multiple organizations.

To configure a new puppet server follow the steps below:

- From the main menu click on Settings
- Once you click on Settings, from the side menu click on Config Management
- Click on Puppet Server
- Click on New button provided
- Enter the puppet server name in the Name box
- Enter the user name in User Name field
- Enter the Hostname in Host Name field
- Choose the organization from the Organization drop down list
- Choose the Authentication by selecting Password / Pem file
- Enter Password / Upload the pem file

The screenshot shows the 'New Configuration Management' form in the RLCatalyst application. The form is titled 'New Configuration Management' and is located under the 'Settings' menu. It contains several input fields and a dropdown menu. The 'Puppet Server Name' field is labeled 'Puppet Server Name: *available' and has the value 'puppetServer'. The 'User Name' field is labeled 'User Name: *' and has the value 'ubuntu'. The 'Host Name' field is labeled 'Host Name: *available' and has the value '52.36.90.19'. The 'Organization' dropdown menu is labeled 'Organization: *' and has the value 'RL'. The 'Choose Authentication Type: *' dropdown menu is labeled 'Choose Authentication Type: *' and has the value 'Pem File'. The 'User Pem File' field is labeled 'User Pem File: *' and has the value 'C:\fakepath\bootstrapncal.pem'. There is a 'Browse' button next to the 'User Pem File' field. At the bottom right of the form, there are 'Cancel' and 'Save' buttons.

- Click on Save button
- Now your Puppet server is configured and listed in the Puppet Server Management page



Import nodes from Puppet Server

You can import existing nodes from the configured puppet server into RLCatalyst by selecting the required Business Group and Project. These imported nodes can be operated from the Workzone.

- To import the existing nodes click on Import button
- Select the node and the respective Environment for the dropdown
- Select the environment from Environment drop down list for the node to be imported
- Select the respective checkbox in the Action column
- Click on the Import Nodes button
- Select the business group from Business Group drop down list
- Select the project from the Project drop down list
- Enter the user name to access server for import in the User Name box
- Choose authentication type from the Choose Authentication Type drop down list. RLCatalyst provide two types of authentication, you can either choose Password or by uploading the PEM file
- Type Password or upload PEM file
- Click on Import button
- Close the popup containing the success message 'Node imported'
- The imported node will be available in the respective Environment of the Workzone

Note: For the imported node using puppet server , Puppet client run icon will be shown.

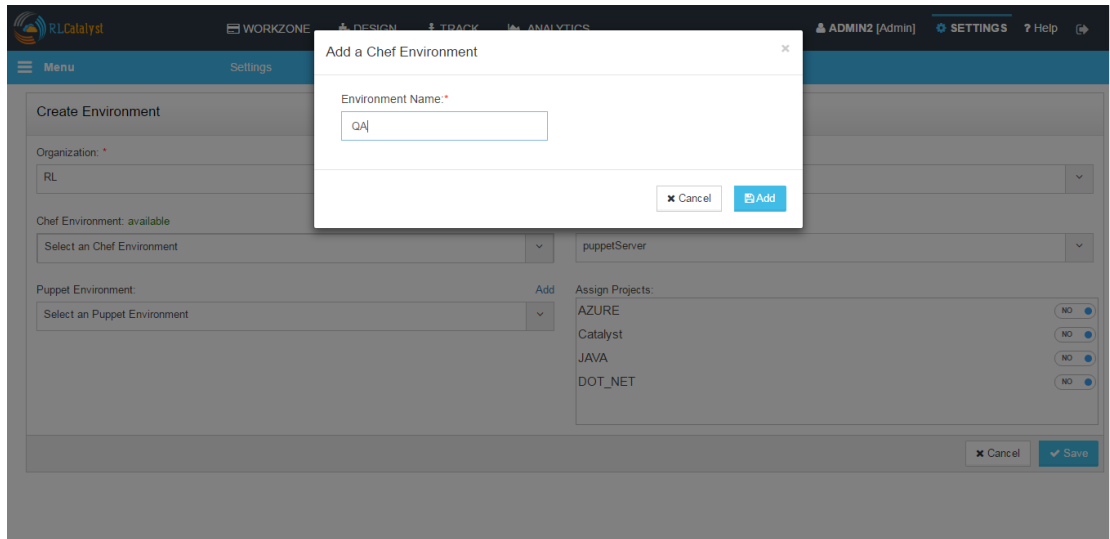
Hereby attaching a video which will demonstrate as in how to Create Puppet in RLCatalyst:

4.2.3 Environments

In an Organization you can create multiple Environments. These environments need to be linked to Projects back. For example: Production, Development, Testing and so on.

Follow the steps to setup a new Environment in an Organization:

- From the main menu click on Settings
- Once you click on Settings, from the side menu click on Config Management
- Click on Environments
- Click on New button provided
- Select the Organization from the Organization drop down list
- Select the server from the Chef Server drop down list
- You can see a list of environments in the drop down. These are the environments defined in your chef server account. You can select one from this drop down list **OR** you can Add new Environments to chef server by clicking on **Add** link provided right above the select an Chef Environment drop down
- Now Enter the Environment name to be created



- Click on Add button
- Now select the environment you added to the chef server from the Chef Environment drop down list

Create Environment

Organization:

Chef Server:

Chef Environment: available

Puppet Server:

Puppet Environment:

Assign Projects:

- AZURE ☐
- Catalyst ☐
- JAVA ☐
- DOT_NET ☐

- Assign the project by toggling to ‘Yes’
- Click on Save button.
- Now the environment is setup and listed in the Environments page

Environments

Name	Organization	Chef Server Name	Projects	Action
Azure	RL	RLChefServer	View	
QA	RL	RLChefServer	View	
RL_Azure_Dev	RL	RLChefServer	View	
RL_CatalystOS	RL	RLChefServer	View	
RL_Customer	RL	RLChefServer	View	
RL_Devops	RL	RLChefServer	View	
RL_EVL	RL	RLChefServer	View	
RL_Prod	RL	RLChefServer	View	

Hereby attaching a video which will demonstrate as in how to Create Environment in RLCatalyst:

4.3 Users Configuration

In Users Configuration we can create a user with particular role and assign teams.

4.3.1 Teams

As soon as an Organization is created 4 teams will get automatically created with organization name as Prefix. (Orgname_Admin, Orgname_DEV, Orgname_Devops and Orgname_QA).

You can add, update and remove Teams in RL Catalyst. Follow the steps below to Add, Update and Remove teams.

How to add a Team?

- From the main menu click on Settings
- Once you click on Settings, from the side menu click on Users Configuration
- Click on Teams
- Click on New button provided
- Enter the name for the team that you want to create in the Name field
- Provide a brief description of the team in the Description field
- Select the organization from the Organization drop down list
- Select the users that you want to assign to the Team
- Assign the projects to the team by using the button provided in the Assign Projects section
- New Team is created

The screenshot shows the 'Create Team' form in the RL Catalyst application. The form is titled 'Create Team' and has a light blue header. It contains the following fields and sections:

- Name:** A text input field with a green hint 'available' and the value 'teamRL'.
- Description:** A text input field.
- Organization:** A dropdown menu showing 'RL'.
- Users:** A list of users with checkboxes for assignment. The users listed are: Admin2, Eng1, Eng2, Tester1, Tester2, appuser1, pi, johnsmith, consumer, designer, superadmin, userc, userd, and testcons. Each user has a 'NO' button next to their name.
- Assign Projects:** A section with four projects: AZURE, Catalystst, JAVA, and DOT_NET. Each project has a 'NO' button next to it.

The form is part of a larger interface with a top navigation bar containing 'Menu' and 'Settings' tabs, and a bottom bar with 'Cancel' and 'Save' buttons.

How to Update or Remove a Team You can update or remove a Team in RL Catalyst

To Edit the Team

- Click on edit button to update Team details

To Remove the Team

- Click on delete button to remove Team from the list

Hereby attaching a video which describes how to Create Teams in RLCatalyst:

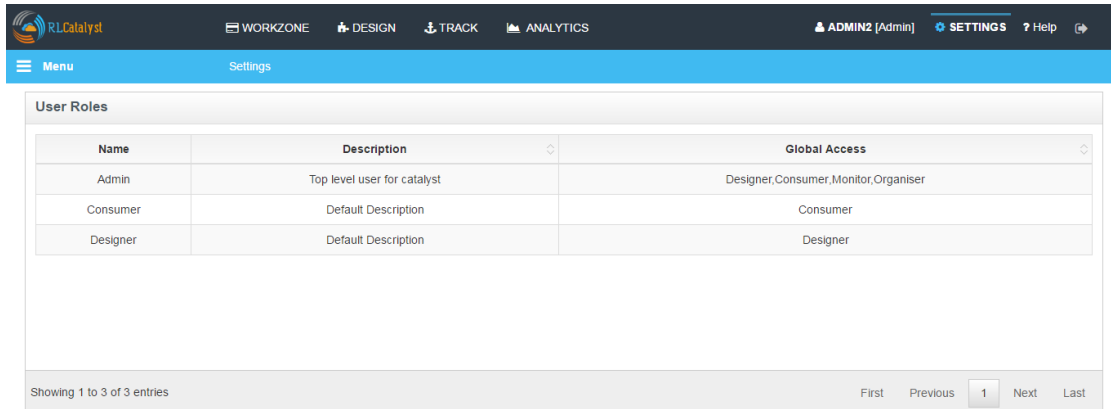
4.3.2 User Roles

Below are the permission sets currently available for the user roles in Catalyst.

RL Catalyst provides three types of user roles.

1. Admin : Top level user of RL Catalyst
2. Consumer : Access rights on Design and Workzone with all permissions
3. Designer : Access rights on Workzone with only Execute permission

These roles cannot be modified or deleted at any point of time.



The screenshot shows the RL Catalyst interface with the 'Settings' menu selected. The 'User Roles' section displays a table with three roles: Admin, Consumer, and Designer. Each role has a description and a list of global access permissions.

Name	Description	Global Access
Admin	Top level user for catalyst	Designer, Consumer, Monitor, Organiser
Consumer	Default Description	Consumer
Designer	Default Description	Designer

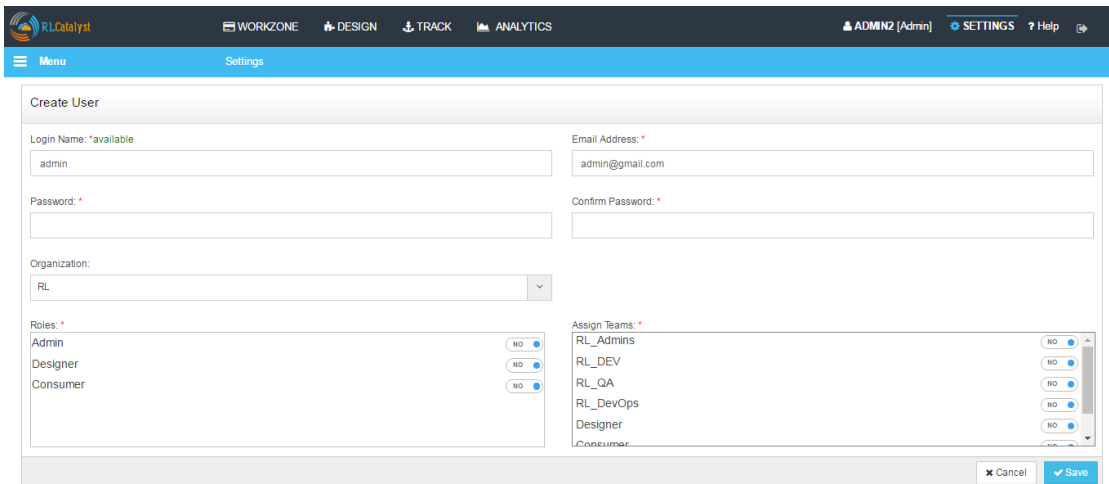
Showing 1 to 3 of 3 entries

4.3.3 Users

You can add, update and remove users in RL Catalyst. Follow the steps below to Add, Update and Remove users.

How to add a new user?

- From the main menu click on Settings
- Once you click on Settings, from the side menu click on Users Configuration
- Click on Users
- Click on New button provided
- Enter a login name in the Login Name field
- Enter the email address of the new user in the Email Address field
- Enter the password and confirm password fields.
- Choose the organization from the Organization drop down list
- Select the appropriate role (Admin / Designer / Consumer)
- Select the team.



- Click on Save button
- New user is created and is available in Users page

Hereby attaching a video which describes how to Create Users in RLCatalyst:

4.4 Provider Configuration

In Provider Configuration we can create a user with particular role and assign teams.

4.4.1 Providers

RLCatalyst supports infra automation across providers like AWS, Azure, VMware, Openstack. Each provider needs to be configured with the proper credentials and other required details before it can be used.

You can configure multiple cloud provider accounts of type **AWS**, **AZURE**, **OPEN STACK** and **VMWARE** within RLCatalyst.

To configure the Providers setup follow the steps below:

- From the main menu click on Settings
- Once you click on Settings, from the side menu click on Provider Configuration
- Click on Providers
- Click on New button provided
- Select the provider from the Provider Type drop down list (e.g. AWS)
- Enter the name of the provider in the name field
- To add **AWS** Provider Account

RLCatalyst supports 2 types of authentication into the AWS account.

IAM Role - RLCatalyst supports authentication using IAM roles, in cases when user doesn't want to enter secret keys and access keys. This requires the RLCatalyst instances to be on AWS and is launched with an IAM role. The credentials will be acquired by rlcatalyst from its instance metadata. Only such provider account can be added per

organization in the current version of RLCatalyst . All AWS API requests made from catalyst by the default provider will be signed with security credentials fetched from instance metadata.

Prerequisites - RLCatalyst should be deployed on an AWS instance. This AWS instance should be launched with an IAM role with permissions to launch EC2 instances and CFTs. Also, new instances from rlcatalyst can only be launched in the same AWS account in which catalyst is running.

Refer to <http://docs.aws.amazon.com/AWSSdkDocsJava/latest/DeveloperGuide/java-dg-roles.html> for more details

To choose this mode, unselect the checkbox **Requires Access Credentials**

Secret key and Access key - This method requires the access key and secret key to authenticate access to the AWS provider account.

To choose this mode select the checkbox **Requires Access Credentials**

- Provide the access key in the Access Key field
- Provide the secret key in the Secret Key field
- Provide the bucket name in the S3 Bucket Name field
- Select the organization from the Organization drop down list
- Select the region from the Region drop down list where your provider is located
- Select the key pair for the provider from the Key Pair drop down list
- Upload the .pem file for Provider

- Click on Save button
- Now Provider is successfully configured to RLCatalyst
- To add **Azure** Provider Account
 - Provide the Subscription ID
 - Provide the Client ID
 - Provide the Client Secret Key
 - Provide the Tenant ID
 - Upload the Pem file
 - Upload the private Key file
 - Select the organization from the Organization drop down list

- Click on Save button
 - Now Provider is successfully configured to RLCatalyst
- To add **OpenStack** Provider Account
 - Provide the Username
 - Provide the password
 - Provide the Host
 - Provide the Project name
 - Provide the Tenant ID
 - Provide the Tenant Name
 - Provide the Compute Service Endpoint
 - Provide the Identity Service Endpoint
 - Provide the Network Service Endpoint
 - Provide the Image Service Endpoint
 - Provide the Instance key Name
 - Upload the pem file for Instance
 - Select the organization from the Organization drop down list
 - Click on Save button
 - Now Provider is successfully configured to RLCatalyst
- To add **VMWare** Provider Account
 - Provide the Username
 - Provide the password
 - Provide the Host
 - Enter the DC
 - Select the organization from the Organization drop down list
 - Click on Save button
 - Now Provider is successfully configured to RLCatalyst

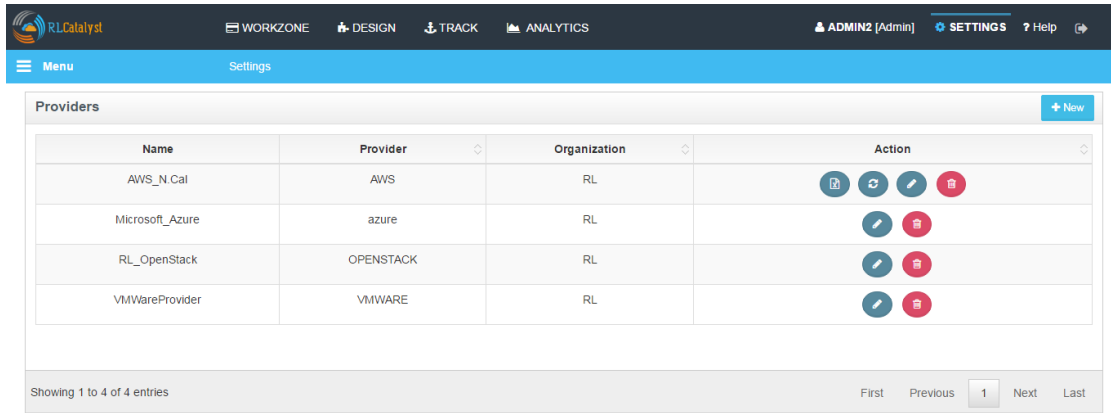
Hereby attaching a video which will demonstrate as in how to Create Providers in RLCatalyst:

Provider Sync for AWS

Provider sync allows you to import unmanaged instances to Catalyst.

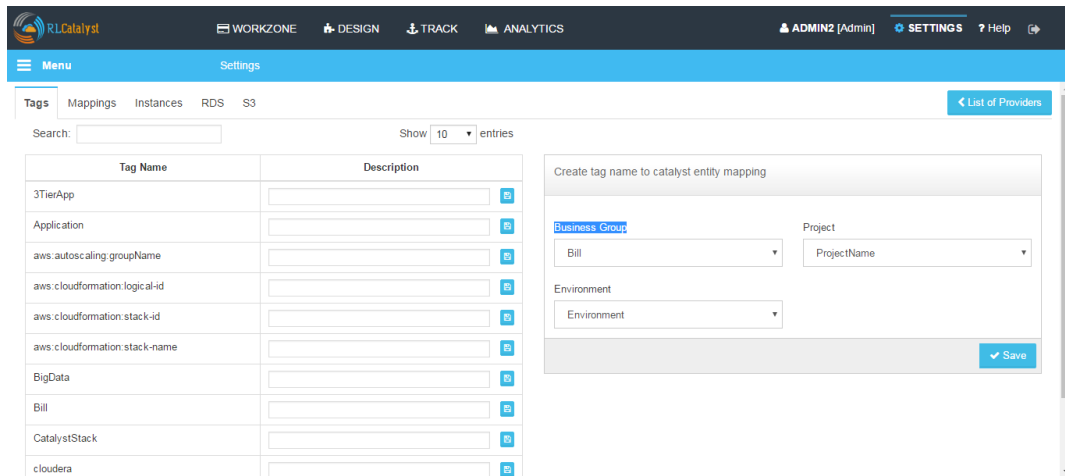
Follow the below steps for AWS provider sync:

- Create an **AWS** provider as shown above. The Created provider will be available in Providers page



- Click on the Sync Instance button of the provider.

1. Tags: you have two sections, left side you will get the tags which are present in ur AWS account will shown here and you can add description for your reference only. And right side you can map the tags with Business Group, Project and Environment, Specify which tags represent Business Group, project name and the environment name. Once you will save it, you can see the refelection in Mapping tab.



2. Mappings: In Mapping, all the mapped Tag Values would be visible with respect to Business Group, Projects and Environment. Select one tag name for Business Group, project from drop down as well as Environment tag name for Environment And save the changes. Now go to Instances tab.

The screenshot shows the 'Settings' page with the 'Mappings' tab selected. It contains three tables for mapping tags to values:

Catalyst Business Group	Business Group Tag Value
Demo	-- None --
Product	Catalyst

[Save](#)

Catalyst Project	Project Tag Value
AZURE	-- None --
Catalyst	Catalyst
DOT_NET	-- None --
JAVA	-- None --

[Save](#)

Catalyst Environment	Environment Tag Value
Azure	-- None --
QA	-- None --
RL_Azure_Dev	-- None --
RL_CatalystOS	-- None --
RL_Customer	-- None --
RL_Devoos	-- None --

3. Instances: You have 3 catalyst status:

- **Managed:** If catalyst status is 'Managed', you will get all "Bootstrapped successful Instances". You can delete the instances from here.
- **Assigned:** If you want to assign some unassigned Instance then you have to update tags corresponding to the mapping.
- **Unassigned:** Here you will get all other Instances available in your AWS account. Here you can update the tag value by selecting the node.

Unassigned Instances:

The screenshot shows the 'Instances' tab with the 'Catalyst Status' dropdown set to 'Unassigned'. The table displays the following instances:

InstanceId	OS	IP Address	Status	Bg Tag Value	Project Tag Value	Env. Tag Value	Actions
i-26c126fa	linux	35.160.23.177	stopped				
i-2711692	linux	10.1.2.165	stopped	J&J	J&J Team	Non-Production	
i-66f721d4	linux	54.153.44.89	stopped	J&J		Non-Production	
i-69ae48e8	linux	54.169.106.84	running				
i-7dcbd165	linux	172.31.16.16	stopped				
i-93fc2a21	linux	54.153.96.40	stopped	J&J		Non-Production	

[Update Tags](#)

Assigned Instances:

InstanceId	OS	IP Address	Region	Status	Org	Bg	Project	Environment	Actions
i-3133c385	linux	10.0.0.24	us-west-1	stopped	RL	-	Catalyst	-	
i-4ec938fa	linux	10.0.0.147	us-west-1	stopped	RL	-	Catalyst	-	
i-538face6	linux	10.0.0.76	us-west-1	stopped	RL	-	-	RL_EVL	
i-5c03de9f	linux	10.0.1.122	us-west-1	running	RL	-	-	RL_EVL	
i-69e979dd	windows	10.0.0.211	us-west-1	stopped	RL	Product	-	-	
i-7bc992b9	linux	54.67.35.103	us-west-1	running	RL	-	Catalyst	-	
i-7bd58ece	linux	52.8.66.52	us-west-1	running	RL	-	-	RL_EVL	
i-c2d18a00	linux	10.0.0.214	us-west-1	stopped	RL	-	Catalyst	-	
i-ca1f6d01	linux	10.0.1.94	us-west-1	running	RL	-	Catalyst	-	

Managed Instances:

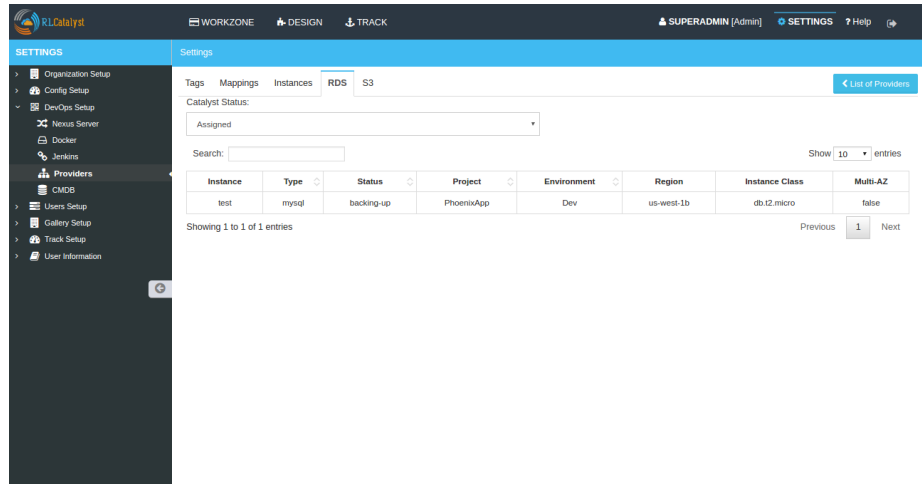
InstanceId	OS	IP Address	Region	Status	Org	Bg	Project	Environment	Actions
i-433be444	linux	10.0.1.121	us-west-1	running	RL	Product	Catalyst	RL_EVL	

4. RDS: Amazon Relational Database Service (Amazon RDS) is a web service that makes it easier to set up, operate, and scale a relational database in the cloud. It provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks.

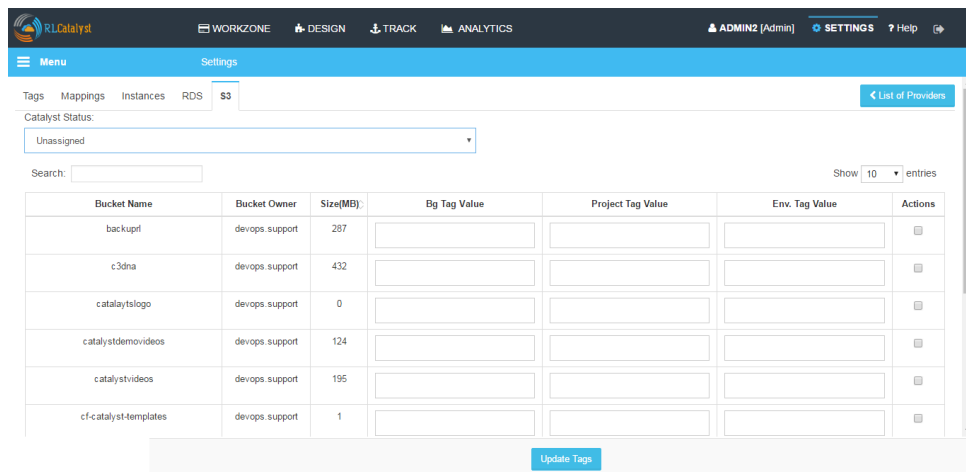
Unassigned RDS:

Instance	Type	Region	Project Tag Value	Env. Tag Value	Actions
test	mysql	us-west-1b			

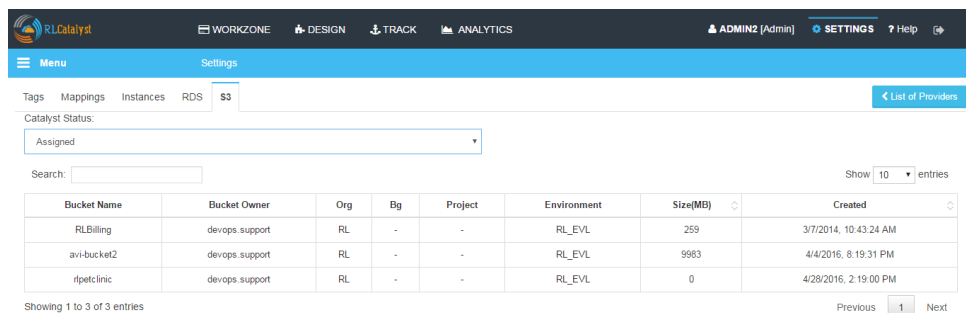
Assigned RDS:



5. S3: A bucket is a logical unit of storage in Amazon Web Services (AWS) object storage service, Simple Storage Solution S3. Buckets are used to store objects, which consist of data and metadata that describes the data. It is the account bucket name, where we store all the information regarding AWS Provider like instances cost and usage.



Assigned S3:



Following video demonstrates how to do provider sync in RLCatalyst:

4.4.2 VM Images

An image of a virtual machine is a copy of the VM, which may contain an OS, data files, data to be installed on multiple VMs and applications. It is usually tested for security, reliability and has the best tested configurations.

Adding a New VM Image

- From the main menu click on Settings
- Once you click on Settings, from the side menu click on Provider Configuration
- Click on VM Images
- Click on New button provided
- Enter the image name in the Name field
- Select the organization from the Organization drop down list
- Choose the provider from the Choose Provider drop down list
- Select the operating system type from the Operating System drop down list
- Provide the image identifier name in the Image ID field
- Provide the admin user name in the Admin User Name field
- Provide the admin password in the Admin Password field
- Click on Save button

- Now new VM Image is added and available in the VM Image list

Hereby attaching a video which will demonstrate as in how to Create VM Images in RLCatalyst:

How to add Windows VM Image? Before onboarding Windows nodes into RLCatalyst, we need to ensure that WinRM is configured on the windows guest node, the two ports 5985 & 5986 are opened for communication between RLCatalyst and node.

The settings below must be added to your base server image or passed in using some sort of user-data mechanism provided by your cloud provider.

Steps (To be performed from a windows host):

1. Use remote desktop to connect to the node (Start->Run->MSTC).
2. Provide the IP Address / Host name of the node along with the username and password.
3. Once connected to the node,
 - a. Open / Run powershell as an administrator
 - b. Execute the below commands (you could copy and paste all the commands together)

```
winrm quickconfig -q
winrm set winrm/config/winrs '@{MaxMemoryPerShellMB="300"}'
winrm set winrm/config '@{MaxTimeoutms="1800000"}'
winrm set winrm/config/service '@{AllowUnencrypted="true"}'
winrm set winrm/config/service/auth '@{Basic="true"}'
winrm set winrm/config/client/auth '@{Basic="true"}'
netsh advfirewall firewall add rule name="WinRM 5985" protocol=TCP dir=in
localport=5985 action=allow
netsh advfirewall firewall add rule name="WinRM 5986" protocol=TCP dir=in
localport=5986 action=allow
net stop winrm
Set-Service WinRm -StartupType Automatic
net start winrm
```

Note: Press enter to execute the last command, if you have copy - pasted the above commands.

4. To create an image from this node, follow the instructions given by the cloud service provider for image creation.
 - a. Remember to create a local admin user before generating an image, as image generation wipes out existing administrator account, which will be manageable only from the server's console and not remotely.
 - b. Install all necessary updates before creating the image.
 - c. Use the Windows sysprep utility to create the image.
 - d. Details about using the sysprep utility can be found here (<https://technet.microsoft.com/en-in/library/hh824938.aspx>)

4.5 Gallery Setup

The Gallery Setup option in RL Catalyst provides you four options:

- **Template Types:** RL Catalyst supports the following template types **SoftwareStack**, **OSImage**, **CloudFormation**, **Docker**, **ARMTemplate**. You can create various templates using these template types.
- **Templates:** You can create different templates in RL Catalyst for all supported Cookbooks. You can also add new templates for newly added Cookbooks. To know more on what is a Cookbook please click: <http://docs.chef.io/cookbooks.html>
- **Service Command:** Service command allows you to setup the service running options. You can choose between Chef Cookbooks based services or System service based commands.

- Script Gallery: In Script Gallery we can keep bash script.
-

4.5.1 Template Types

RL Catalyst supports the following template types **SoftwareStack**, **OSImage**, **CloudFormation**, **Docker**, **ARMTemplate**. You can create various templates using these template types.

Software Stack: Software Stack Template type allows you to create a blueprint with set of cookbooks that work sequentially to produce a result.

OSImage: OSImage template type allows you to create blueprint with the help of VMimages

CloudFormation: It simplifies provisioning and management on AWS. You can create templates which is in JSON format for the service or application architectures you want and have AWS CloudFormation use those templates for quick and reliable provisioning of the services or applications (called “stacks”).

Docker: Docker provides an integrated technology suite that enables development and IT operations teams to build, ship, and run distributed applications anywhere.

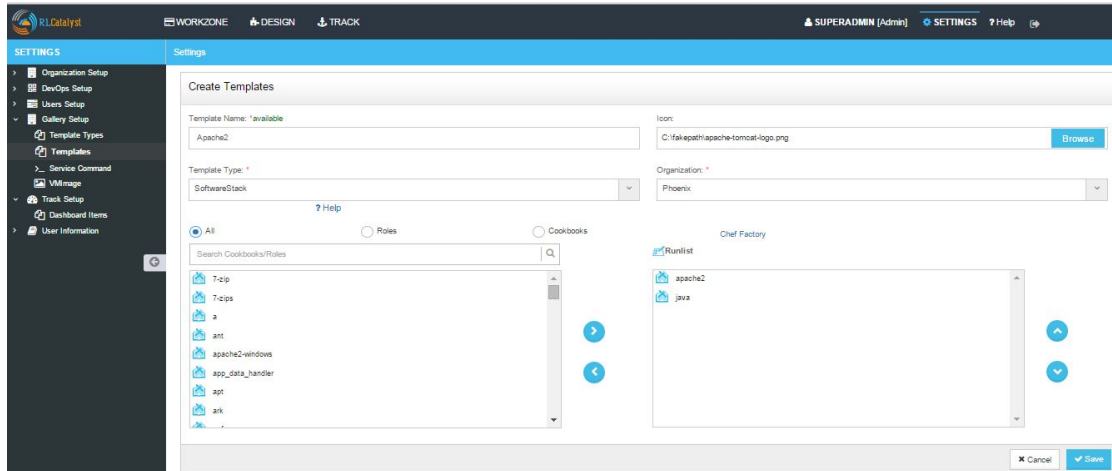
ARM Template: An Azure ARM Template is a JSON file that captures the infrastructure requirements of your application. An Azure Template defines a number of related resources that are required by your application. When an Azure Template is deployed on Azure, Azure Resource Manager ensures that all resources are brought to the specified states so that your application runs under the exact desired environments.

4.5.2 Templates

Templates will help to deploy applications on existing infrastructure without having need of launching new instances. Templates internally contain roles or cookbooks. With the help of templates also you can deploy different applications.

Adding a New Template For Software Stack Template Type

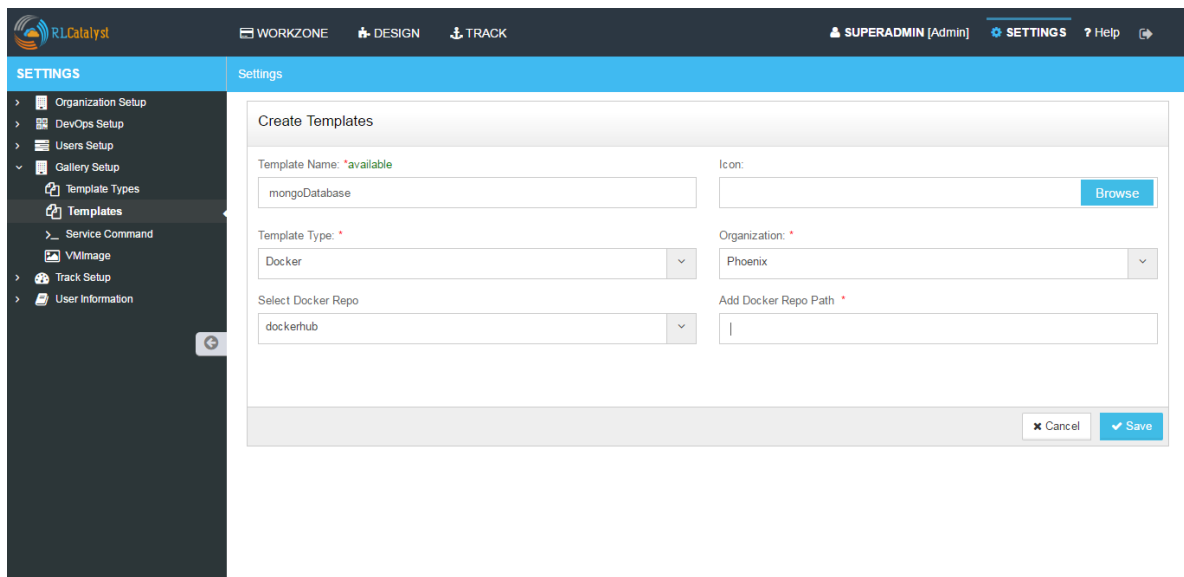
- From the main menu click on Settings
- Once you click on Settings, from the side menu click on Gallery Setup
- Click on Templates
- Click on New button provided
- Enter the template name in the Template Name field
- Upload an icon by clicking on the Browse button.
- Choose the template type ‘Software Stack’ from the Template Type drop down list
- Choose the Organization from the Organization drop down list
- Select the Cookbooks from left frame and Click the arrow button to add to the Order Runlist. You can deselect the selected Cookbooks from the Order Runlist by clicking the arrow button to the Select Runlist box again
- You can customize the order of Order Runlist. First select the item and then use the up and down arrows provided to change the order of the Order Runlist.



- Now new template is added and available in the Templates list

Adding a New Template For Docker Template Type

- From the main menu click on Settings
- Once you click on Settings, from the side menu click on Gallery Setup
- Click on Templates
- Click on New button provided
- Provide a template name in the Template Name field
- Upload an Icon by clicking on the Browse button.
- Choose the template type 'Docker' from the Template Type drop down list
- Choose the Organization from the Organization drop down list
- Choose the Docker repo
- Add Docker Repo Path



- Click on Save button
- Now new template is added and available in the Templates list

Adding a New Template For Cloud Formation Template Type

- From the main menu click on Settings
- Once you click on Settings, from the side menu click on Gallery Setup
- Click on Templates
- Click on New button provided
- Enter the template name in the Template Name field
- Upload an Icon by clicking on the Browse button
- Choose the template type 'CloudFormation' from the Template Type drop down list
- Choose the Organization from the Organization drop down list
- Browse the Template File
- Click on Save button

- Now new template is added and available in the Templates list

Adding a New Template For ARMTemplate Type

- From the main menu click on Settings
- Once you click on Settings, from the side menu click on Gallery Setup
- Click on Templates
- Click on New button provided
- Enter the template name in the Template Name field
- Upload an Icon by clicking on the Browse button
- Choose the template type 'ARMTemplate' from the Template Type drop down list
- Choose the Organization from the Organization drop down list
- Browse the Template File
- Click on Save button

Hereby attaching a video which will demonstrate as in how to Create Templates in RLCatalyst:






4.5.3 Service Command

Service Commands helps user to create a service associated with cookbooks which can run on the instance with the following actions **Start**, **Stop** and **Restart**.

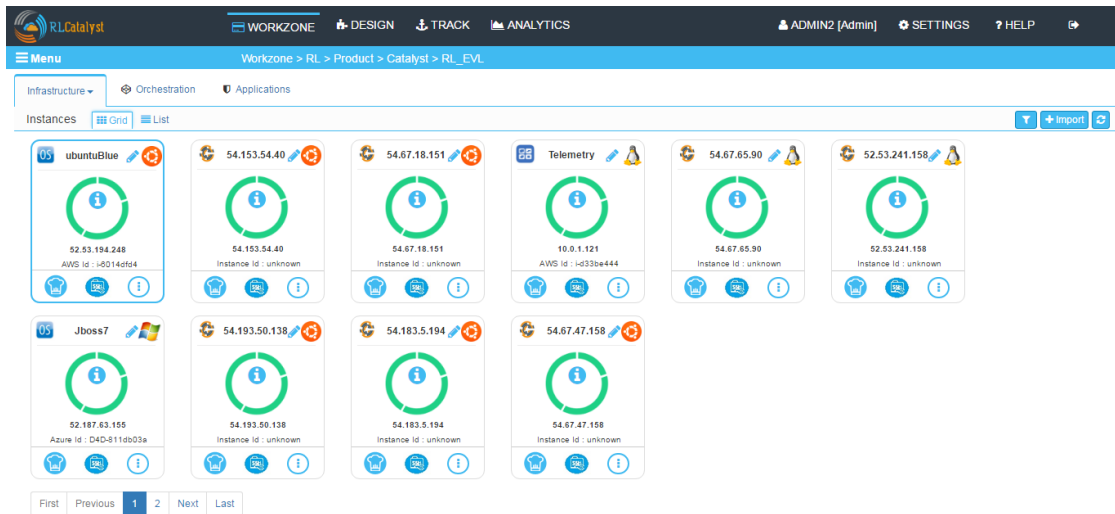
Adding a new Service Command

- From the main menu click on Settings
- Once click on Settings, from the side menu click on Gallery Setup
- Click on Service Command
- Click on New button provided
- On Create Services page Select Organization, Enter Name, Choose Service Command Type as Chef Cookbook/Recipe , Select Chef Server, Service cookbooks as 'service_apache'.
- Select the Actions.

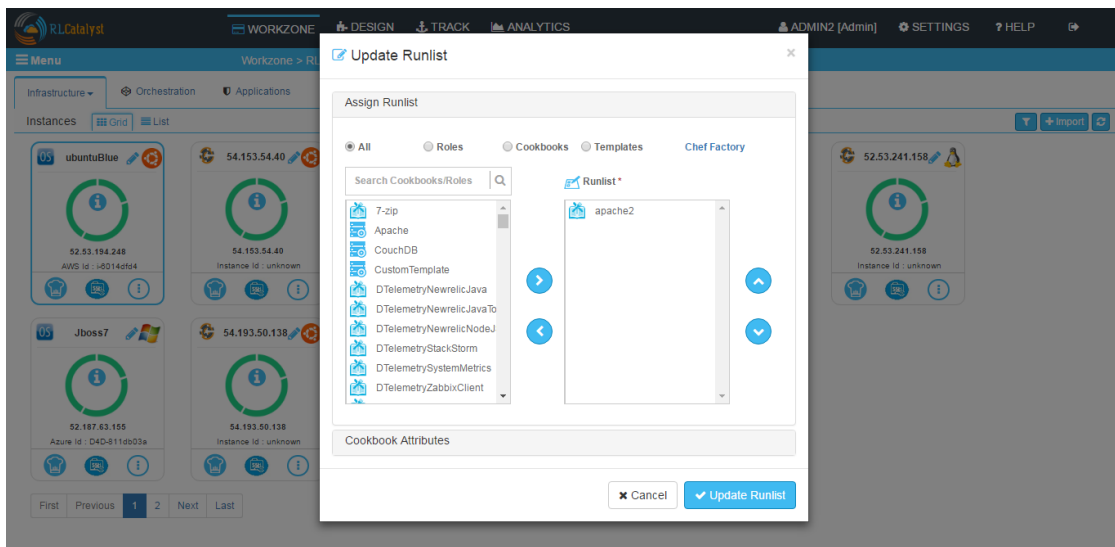
- Click on Save button
- Now your Service Command is setup and listed in the Services Page

Name	Action Type	Service Action	Organization	Action
serviceapache	Chef Cookbook/Recipe	  	Phoenix	 

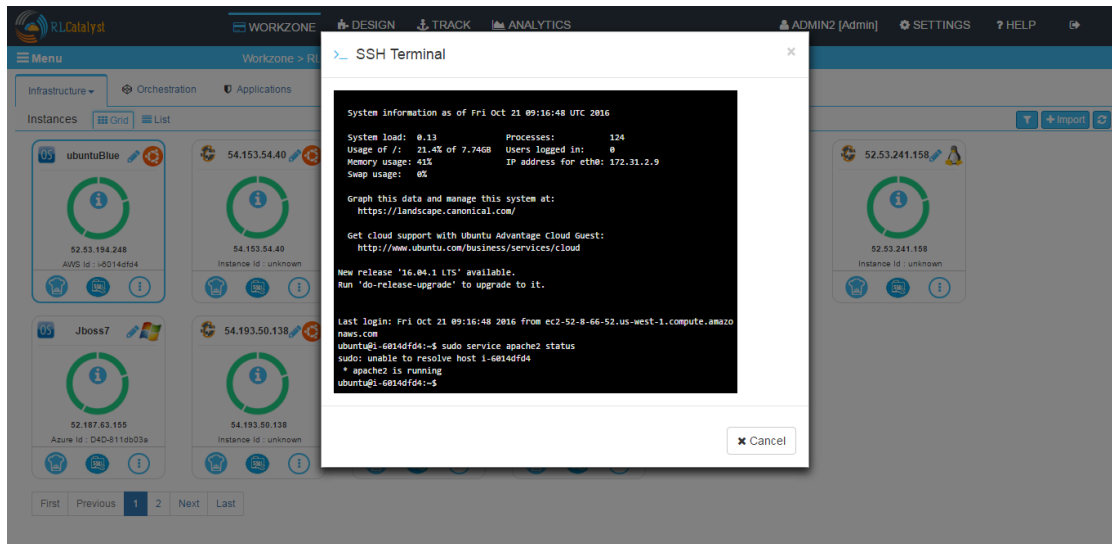
- Go to Workzone and Launch or Import a Node



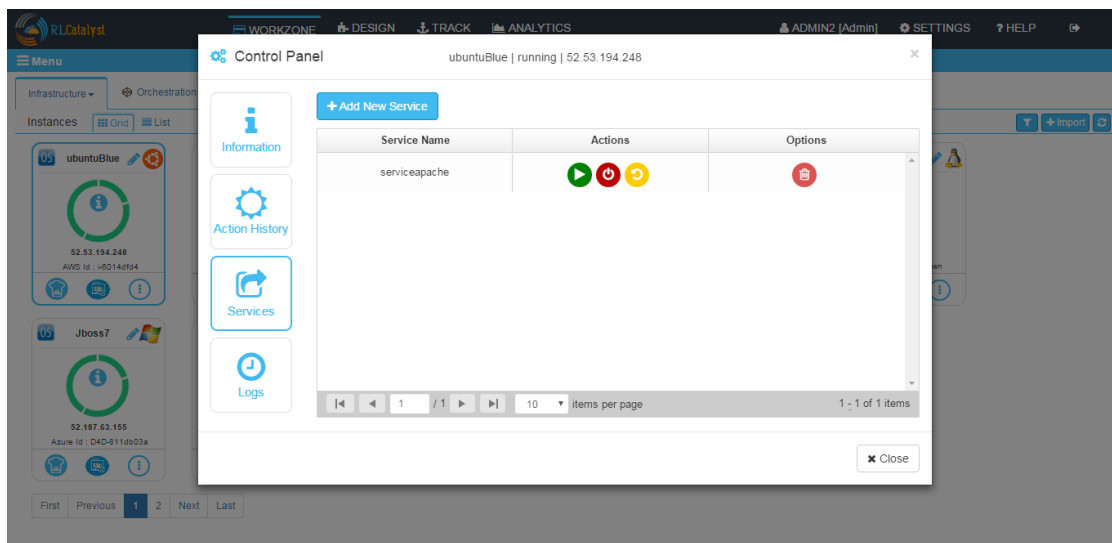
- Click on Chef Client run icon , add Apache2 cookbook to the runlist and click Update button. Wait until chefclient is success.



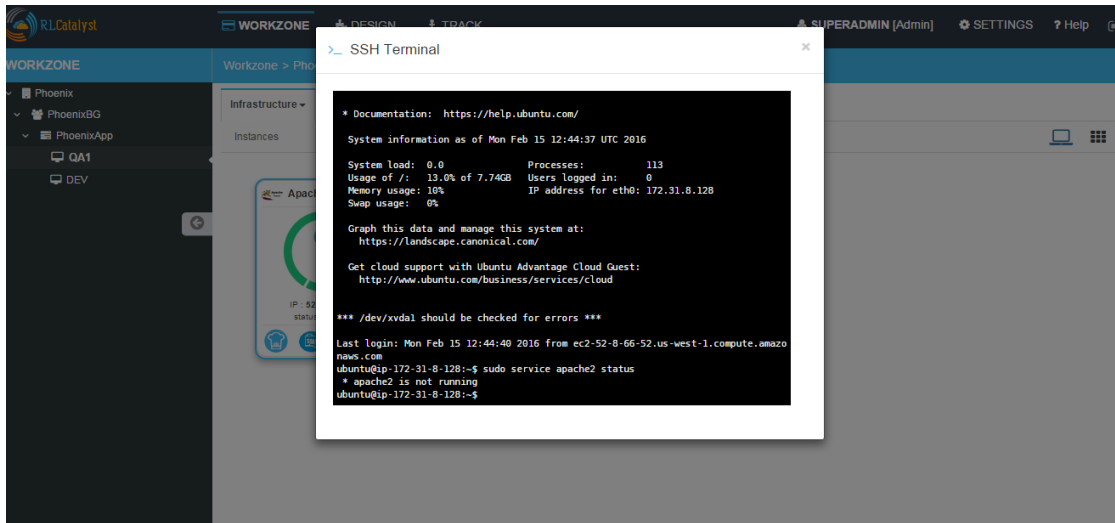
- When apache2 cookbook run successfully by default service will be running. Click on SSH icon and execute **sudo service apache2 status** command and verify apache2 is running.



- Close the SSH window
- Go to Instance control panel
- Go to Services tab and add the apache service and click on Save button



- Service is added to the Instance and Start, Stop and Restart buttons will be shown
- Click on Stop button (Red color) and wait until it succeeds
- Click on SSH icon
- Execute command **sudo service apache2 status** and verify apache2 is not running



4.5.4 Script Gallery

Script Gallery where we can add bash script file.

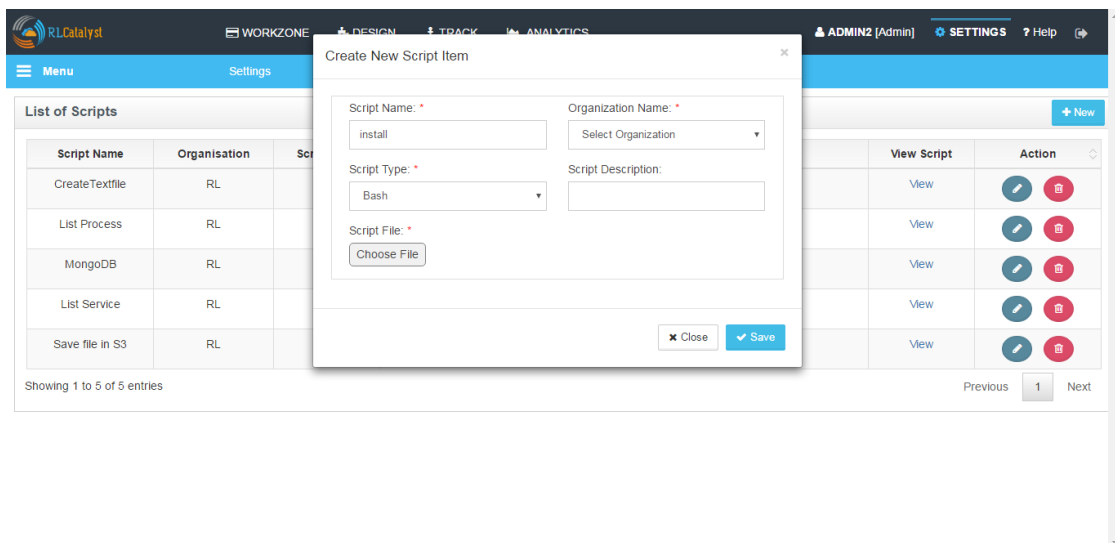
Currently we are supporting only bash scripts for installation.

To add your script:

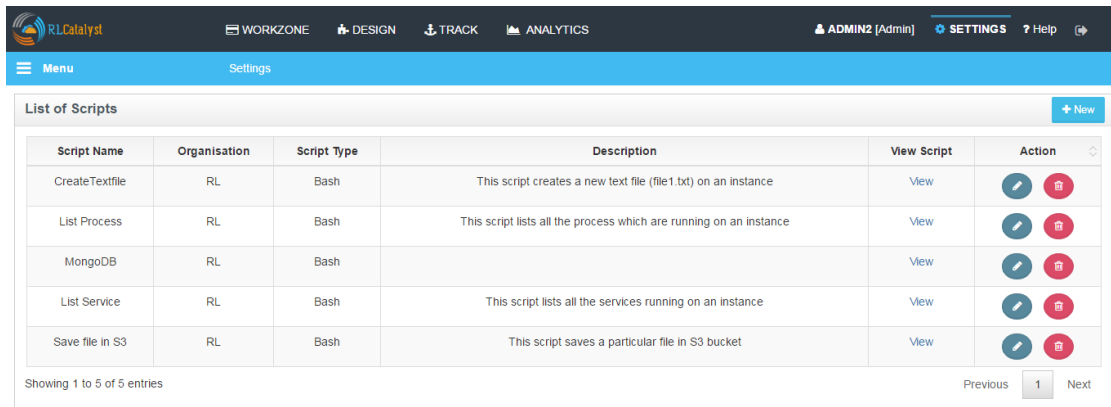
Navigate to Design -> Gallery Setup -> Script Gallery -> +New

Now fill all required details:

- Script Name could be any name
- Choose Organization Name from drop down
- choose Script Type to Bash
- enter some Script Description
- choose Script file



Once you will save it, it will be visible in the List of Scripts



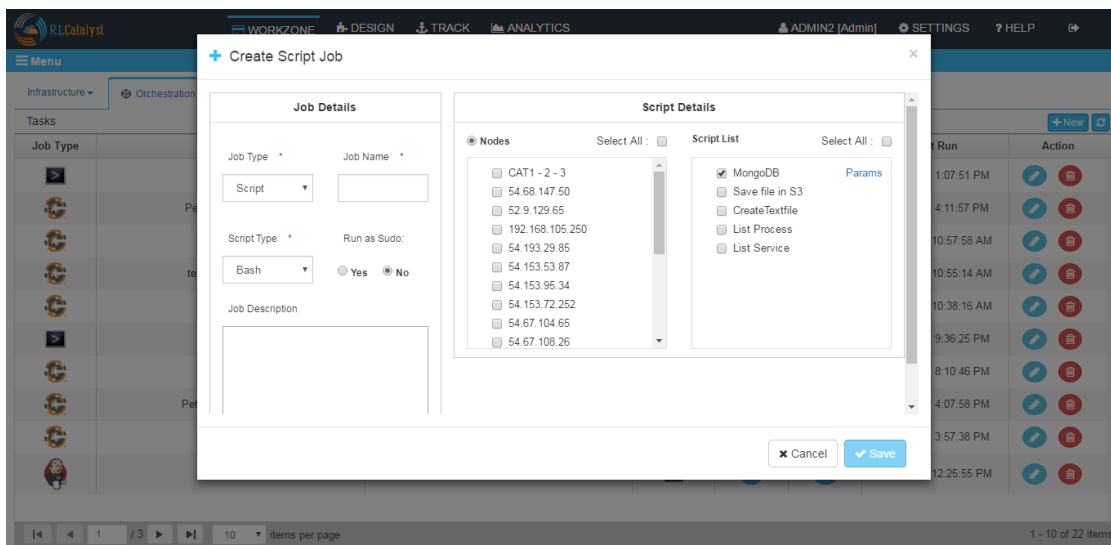
Script Name	Organisation	Script Type	Description	View Script	Action
CreateTextfile	RL	Bash	This script creates a new text file (file1.txt) on an instance	View	
List Process	RL	Bash	This script lists all the process which are running on an instance	View	
MongoDB	RL	Bash		View	
List Service	RL	Bash	This script lists all the services running on an instance	View	
Save file in S3	RL	Bash	This script saves a particular file in S3 bucket	View	

Showing 1 to 5 of 2 entries

Previous 1 Next

By clicking on edit button, you can see the script which you added and it would be Read Only.

Now, create Script Job by choosing Job type, Job Name, select nodes and select script which you want to add in your job



Create Script Job

Job Details

Job Type *

Script

Job Name *

Script Type *

Bash

Run as Sudo:

☐ Yes
 ☒ No

Job Description

Script Details

Nodes

☐ CAT1 - 2 - 3
☐ 54.68.147.50
☐ 52.9.129.65
☐ 192.168.105.250
☐ 54.193.29.85
☐ 54.153.53.87
☐ 54.153.95.34
☐ 54.153.72.252
☐ 54.67.104.65
☐ 54.67.108.26

Script List

☒ MongoDB
☐ Save file in S3
☐ CreateTextfile
☐ List Process
☐ List Service

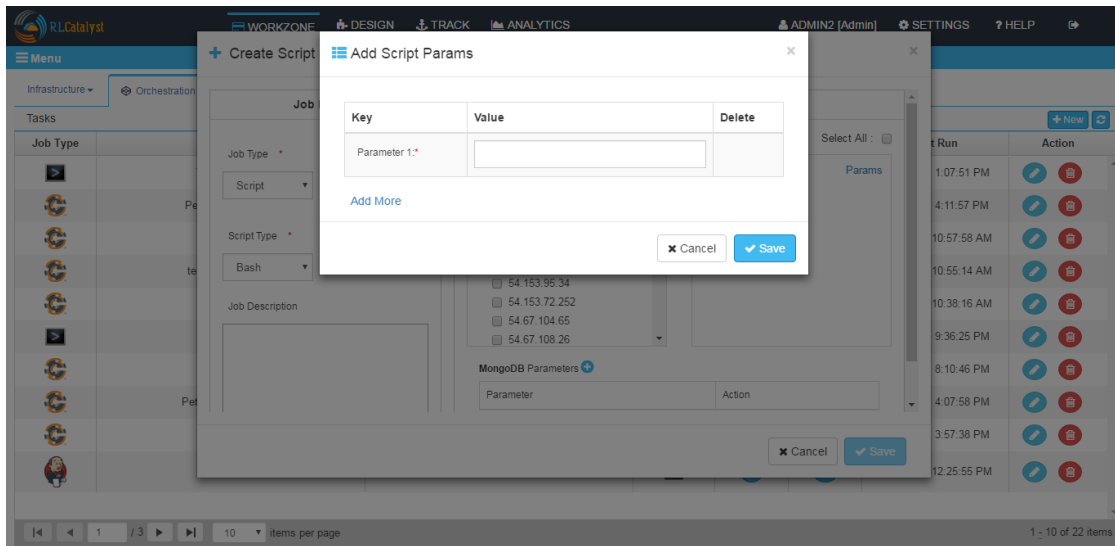
Cancel

Save

You can add parameters

46

Chapter 4. Settings



Click on Save button and you can see your Script Job in Tasks under Orchestration tab in WORKZONE

Job Type	Name	Job Description	Job Links	Execute	History	Last Run	Action
Script	TextFileCreate					9/21/2016, 1:07:51 PM	
Script	PetclinicAppDeploy					10/4/2016, 4:11:57 PM	
Script	testpet					9/21/2016, 10:57:58 AM	
Script	testcatalystdeploy					9/21/2016, 10:55:14 AM	
Script	deploycat					9/21/2016, 10:38:16 AM	
Script	MongoDB					8/24/2016, 9:36:25 PM	
Script	Appdeploy					8/24/2016, 8:10:46 PM	
Script	PetclinicAppUpgrade					8/18/2016, 4:07:58 PM	
Script	AppUpgrade					8/18/2016, 3:57:38 PM	
Script	t5askhist					8/16/2016, 12:25:55 PM	

Now from here you can execute your task. By clicking execute button script will be executed on the selected node.

4.6 DevOps Roles

Add all your DevOps accounts and tools data here. This includes your Nexus Server, Docker and Jenkins . All of these are optional in RLCatalyst

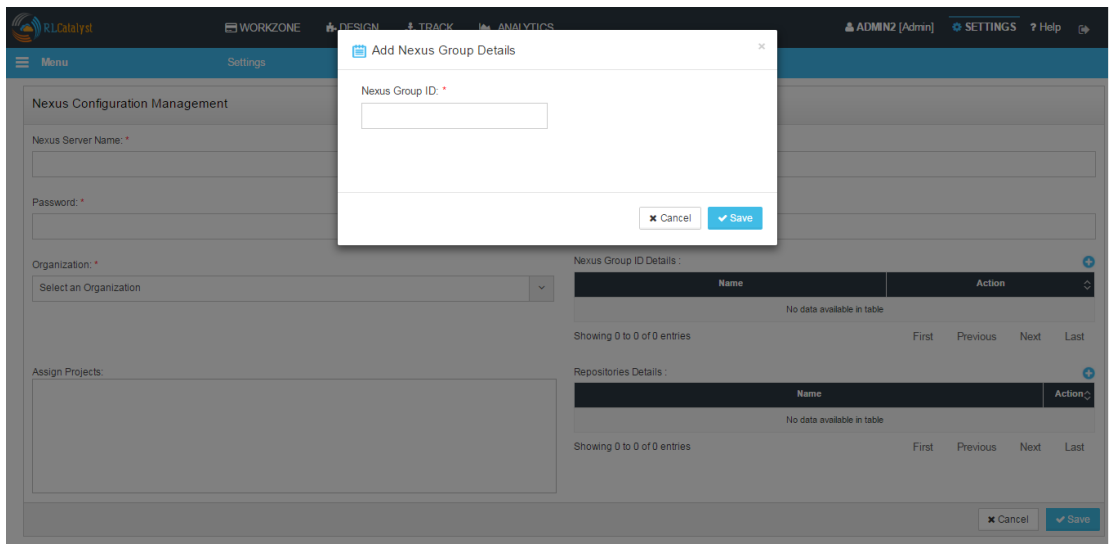
4.6.1 Nexus Server

Nexus is a repository manager. It allows you to proxy, collect, and manage your dependencies so that you are not constantly juggling a collection of JARs. It makes it easy to distribute your software. Internally, you configure your build to publish artifacts to Nexus and they then become available to other developers. You get the benefits of having your own 'central', and there is no easier way to collaborate.

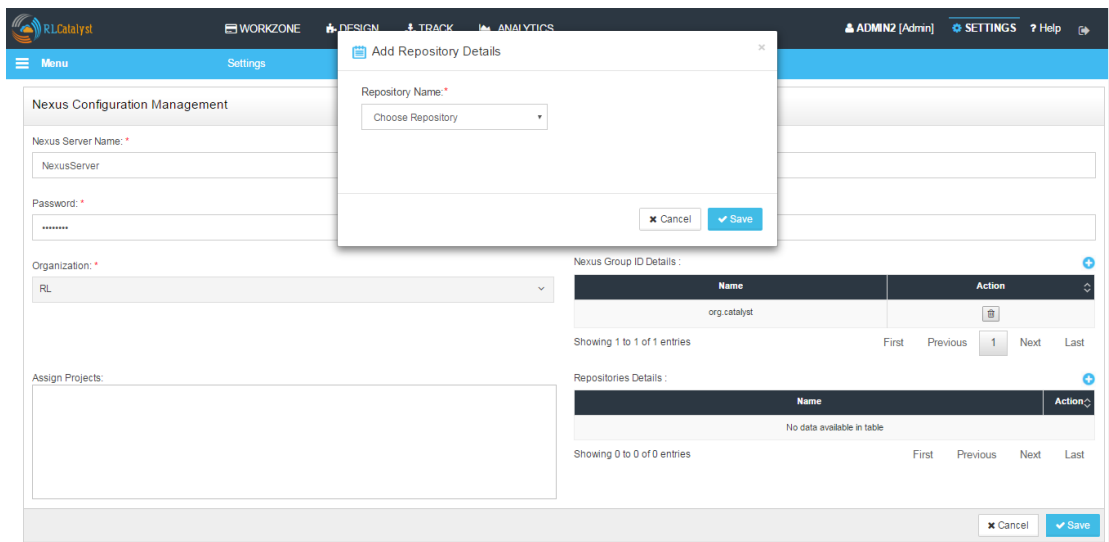
RLCatalyst works with different repositories where you have kept your artifacts . If you have a nexus repository, add it here

To configure a new Nexus server follow the steps below:

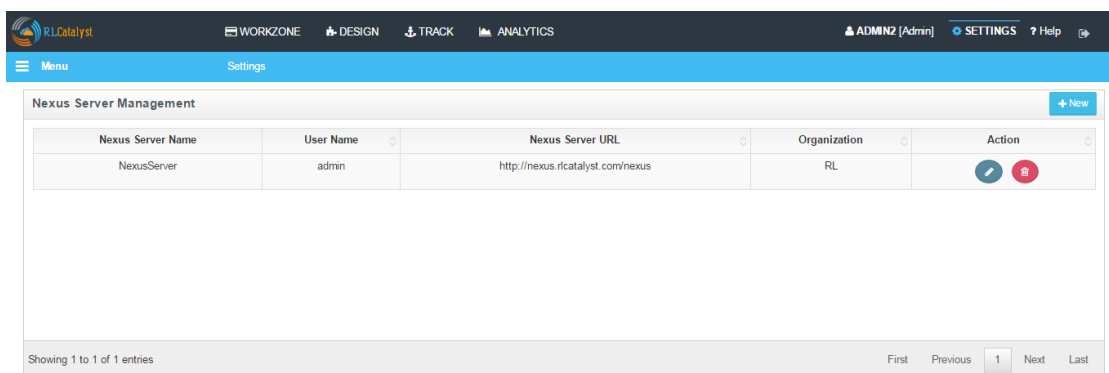
- From the main menu click on Settings
- Once you click on Settings, from the side menu click on Devops Roles
- Click on Nexus Server
- Click on New button provided
- Enter the Nexus server name in the Nexus Server Name field
- Enter nexus server username in the User Name field
- Enter nexus server password in the Password field
- Enter nexus server URL in the Nexus Server URL field
- Choose the organization from the Organization drop down list
- Click on + icon present next to Nexus Group ID Details and enter valid Nexus GroupID and save



- Assign Projects are optional
- Click on + icon present next to Repository Details and select valid Repository Name from the drop down and save



- Click on Save button
- Now the Nexus server setup is ready and listed in the Nexus Server page



Hereby attaching a video which will demonstrate as in how to Create Nexus in RLCatalyst:

4.6.2 Docker

Docker is all about making it easier to create, deploy, and run applications by using containers. Containers allow a developer to package up an application with all of the parts it needs, such as libraries and other dependencies, and ship it all out as one package.

You can configure the Docker setup with RLCatalyst. To configure the Docker setup follow the steps below:

- From the main menu click on Settings
- Once you click on Settings, from the side menu click on Devops Roles
- Click on Docker

- Click on New button provided
- Select the Organization from the Organization drop down list
- Provide a reference name in the Reference Name field
- Provide the registry in the Docker Hub Registry field provided
- Provide the Docker user ID in the User ID field
- Provide the email address to connect to the Docker in the Email Id field
- Enter the Docker password in the Password field
- All assigned projects will display in Assign Projects field
- Add Repositories Details

The screenshot shows the 'Edit Docker' configuration page in the RL Catalyst application. The page has a top navigation bar with 'WORKZONE', 'DESIGN', 'TRACK', and 'ANALYTICS' tabs. Below this is a 'Menu' bar with 'Settings' selected. The main content area is titled 'Edit Docker' and contains several form fields: 'Organization' (dropdown menu with 'RL' selected), 'Name' (text input with 'rldocker'), 'Docker Hub Registry Name' (text input with 'relevancelab'), 'User ID' (text input with 'relevancelab'), 'Email ID' (text input with 'puneeet@relevancelab.com'), and 'Password' (password input). There is also an 'Assign Projects' section with a list of projects (AZURE, Catalyst, JAVA, DOT_NET) and a 'Repositories Details' table. The table has a 'Name' column and an 'Action' column. The table is currently empty, showing 'No data available in table'. At the bottom, there are 'Cancel' and 'Save' buttons.

- Click Save button
- Now Docker is successfully configured to RL Catalyst

Hereby attaching a video which will demonstrate as in how to Create Docker in RLCatalyst:

4.6.3 Jenkins

Jenkins is CI/CD tool which can be used for build and deployment automation. You can configure the Jenkins setup with RLCatalyst. To configure the Jenkins setup follow the steps below:

- From the main menu click on Settings
- Once you click on Settings, from the side menu click on DevOps Roles
- Click on Jenkins
- Click on New button provided
- Select the Organization to which the Jenkins server will be attached to
- Enter the name in the Name field
- Enter the Jenkins Server URL

- Enter the user ID in the User ID field
- Enter the Jenkins password in the Password field

- Click Save button
- Now Jenkins is successfully configured to RLCatalyst

Hereby attaching a video which will demonstrate as in how to Create Jenkins in RLCatalyst:

4.7 Track Setup

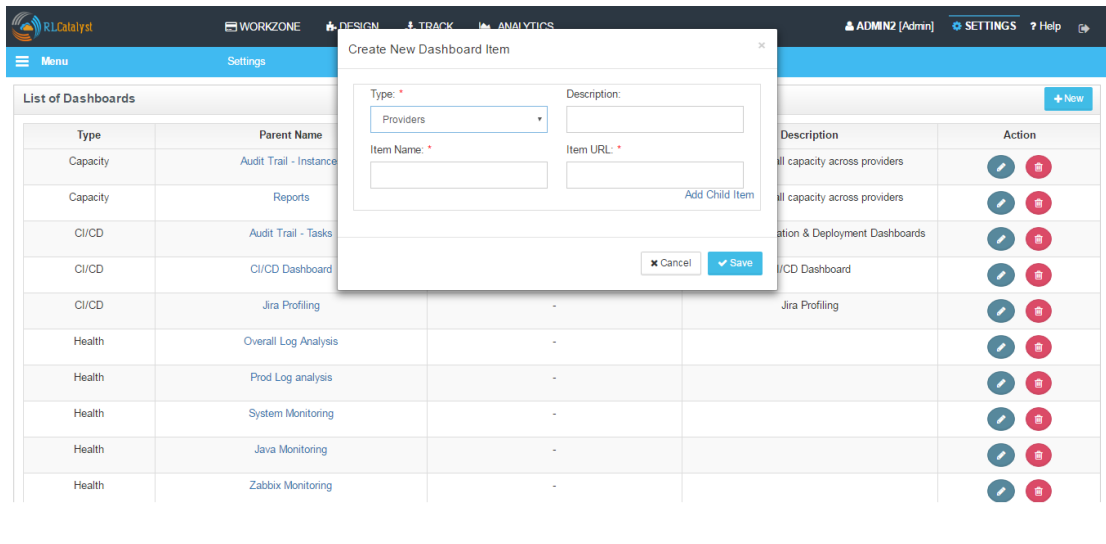
Configuring Track Setup

The Track Setup option in RL Catalyst provides you to add new dashboard items of type Provider, Monitoring, Logs and Notification options.

Adding New Provider Dashboard in Track setup

In Provider dashboard you will be able to see Total number of Instances , Total number of Managed Instances and Total number of Unmanaged Instances for each provider.

- From the main menu click on Settings
- Once you click on Settings, from the side menu click on Track Setup
- Click on Dashboards
- Click on New button provided
- Choose the Type as **Provider**
- Enter the Description and Item Name
- Enter the Item URL in this format (<http://ipaddress:port/public/Dashboard.html>)



- Click on Save button

Adding New AWS Summary Dashboard in Track setup

In Summary Dashboard you will be able to see the Total Cost, Today's Cost, Active Instances details

- From the main menu click on Settings
- Once you click on Settings, from the side menu click on Track Setup
- Click on Dashboards
- Click on New button provided
- Choose the Type as **Provider**
- Enter the Description and Item Name
- Enter the Item URL in this format (<http://ipaddress:port/public/dashing.html>)
- Click on Save button.

Once You save the **list of dashboards** will be displayed in table

4.8 CMDB Configuration

A configuration management database (CMDB) is a repository that acts as a data warehouse for information technology (IT) installations. It holds data relating to a collection of IT assets (commonly referred to as configuration items (CI)), as well as to descriptive relationships between such assets. To configure the CMDB setup follow the steps below:

- From the main menu click on Settings
- Once you click on Settings, from the side menu click on CMDB Configuration
- Click on CMDB
- Click on New button provided
- Enter Configuration Name
- Enter ServiceNow URL
- Enter user name in the UserName field

- Enter password in the Password field
- Select the Organization to which the Jenkins server will be attached to

- Click Save button
- Now ServiceNow CMDB is successfully configured to RLCatalyst

4.9 BOTs

BOTs are ready-made automated scripts.

If you have not configured your BOTs setup, ask your admin to set up BOTs.

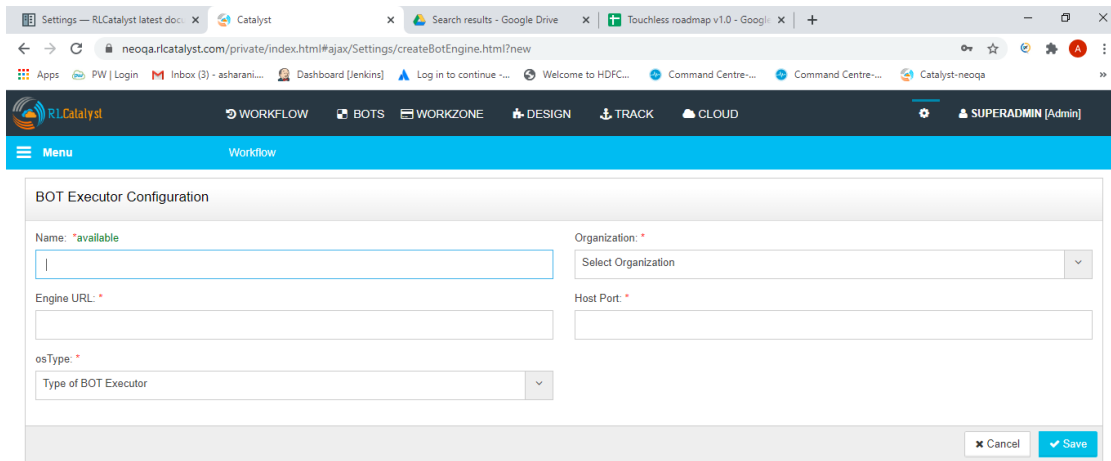
4.10 BOT Executor

BOT is nothing but a piece of an Automation. It would be in the form of scripts, cookbooks etc..It can start the work<<Execute the task<<stop.RLCatalyst BOTs Factory work as the job scheduler.

Connect RLCatalyst to BOT Executor

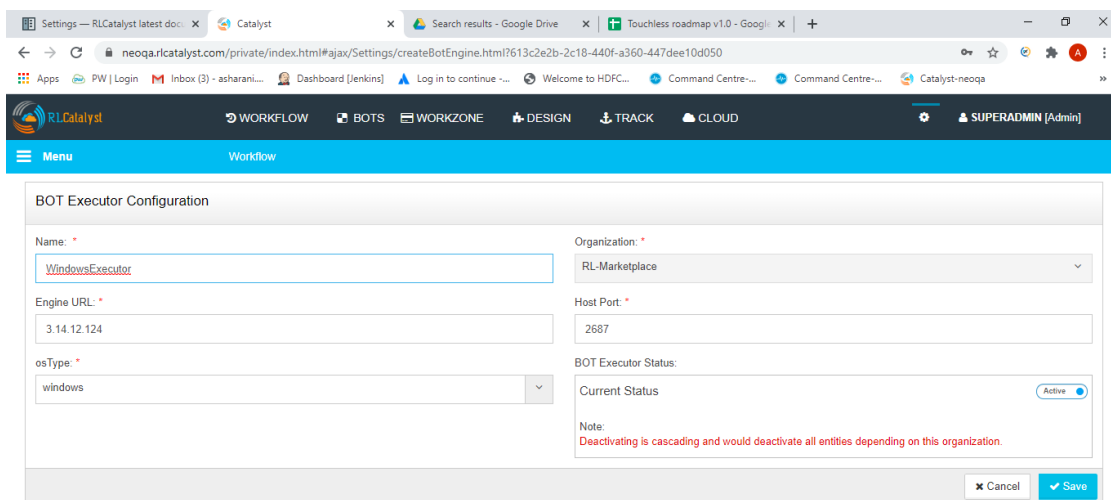
Please follow the below steps in order to configure the Executor from Catalyst UI

- Login to the Catalyst UI >> Go to the Settings >> Move your cursor on Menu (Top left side) >> Click on BOTs >> Click BOTs Executor
- Now Click on New (Top Right side) >> Give the Name of the Executor >> Select Organisation >> Put the BOT engine IP in BotEngine URL section >> Put 2687 in Host Port >> Select the same OS of the Botengine Machine. >> Click Save.
- When the BOTs Server executes a BOT, it chooses the appropriate executor based on the type of BOT. E.g. Linux BOT will run on Linux executor only. If an appropriate executor is not configured or available, the execution will fail.



Follow the steps to make your executor Active and Inactive

- Login to the Catalyst UI >> Go to the Settings >> Move your cursor on Menu (Top left side) >> Click on BOTs >> Click BOTs Executor >> Click on the edit button (Pencil)
- Now click active or inactive as per your requirement >> Click on the Save button



ServiceNow is a software platform which supports IT Service Management (ITSM). It helps you to automate IT Business Management. This is cloud-based platform. We are configuring ServiceNow with catalyst to get the BOT ticket details. BOTs Server can integrate with ITSM platform like ServiceNow to integrate with the ITSM process.

Please follow the below steps in order to configure the ServiceNow from Catalyst UI

- Login to the Catalyst UI >> Go to the Settings >> Move your cursor on Menu (Top left side) >> Click on BOTs >> Click ServiceNow server

- Now Click on New (Top Right side) >> Give the Name of configuration >> give URL of ServiceNow >> provide ServiceNow username and password details << select organization << click on save button

The screenshot shows a web browser window displaying the 'ServiceNow CMDB Setting' form in the RL Catalyst application. The browser's address bar shows the URL: `neoqa.rlcatalyst.com/private/index.html#ajax/Settings/ConfigCMDB.html`. The application's top navigation bar includes links for WORKFLOW, BOTS, WORKZONE, DESIGN, TRACK, and CLOUD, along with a user profile for SUPERADMIN [Admin]. The form itself has a light gray header and contains the following fields:

- Configuration Name:** A text input field with a red asterisk indicating it is required.
- ServiceNow URL:** A text input field with a red asterisk indicating it is required.
- UserName:** A text input field with a red asterisk indicating it is required.
- Password:** A text input field with a red asterisk indicating it is required.
- Organization:** A dropdown menu with the text 'Select an Organization' and a downward arrow.

At the bottom right of the form, there are two buttons: 'Cancel' and 'Save'.

The Windows taskbar at the bottom of the screen shows the search bar, taskbar icons for various applications, and the system clock indicating 10:36 on 13-07-2020.

User defined Blueprints can be created in the Design section associated with the respective **providers** and **Template Types**.

If you want to know about **Template types** please refer to [Gallery Setup](#) .

Blueprints are predefined templates which can be used by service consumers to Launch the instances. Blueprints are designed by Service Designers. Each blueprint stores the metadata of the instance, variables, actions and activity.

RL Catalyst Design options allows you to create Blueprints by using predefined templates.

Creating BluePrints for AWS provider

1. Software Stack

Provider: It offers some form of IT infrastructure that is commercially distributed and sourced across several subscribers - typically businesses. Cloud providers deliver cloud solutions through on-demand, pay-as-you-go systems as a service to customers and end users. Cloud provider customers access cloud resources through Internet and programmatic access and are only billed for resources and services used according to a subscribed billing method.

Images: An image of a virtual machine is a copy of the VM, which may contain an OS, data files and applications.

Region: An AWS account provides multiple regions so that you can launch Amazon EC2 instances in locations that meet your requirements.

VPC: Amazon Virtual Private Cloud, lets you provision a logically isolated section of the Amazon Web Services (AWS) Cloud where you can launch AWS resources in a virtual network that you define.

Subnet: A subnet is a range of IP addresses in your VPC. You can launch AWS resources into a subnet that you select.

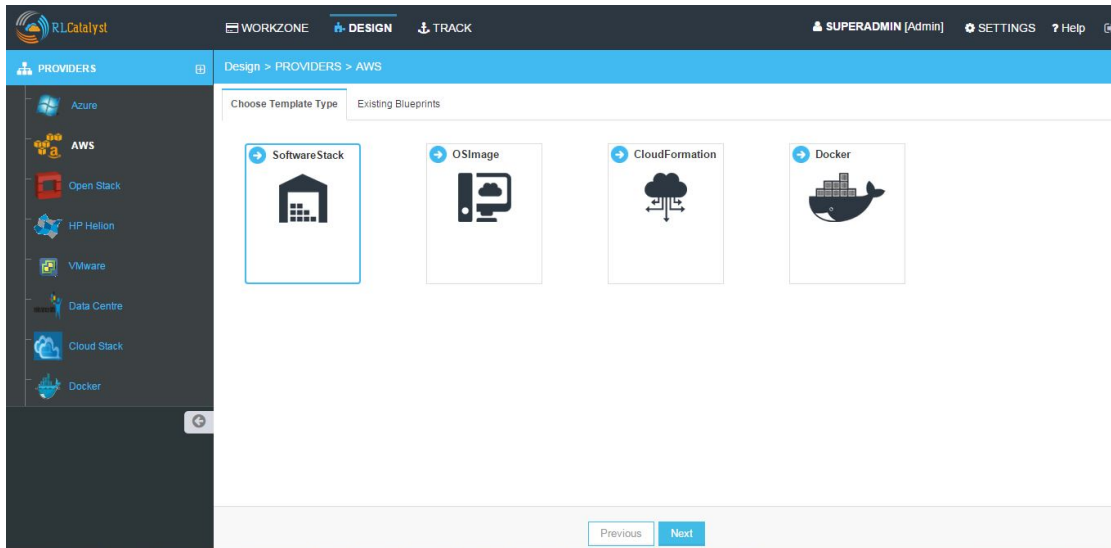
KeyPair: Amazon EC2 uses public-key cryptography to encrypt and decrypt login information. Public-key cryptography uses a public key to encrypt a piece of data, such as a password, then the recipient uses the private key to decrypt the data. The public and private keys are known as a key pair.

Instance Type: It specifies the hardware of the host computer used for your instance.

Security group: A security group acts as a virtual firewall that controls the traffic for one or more instances. When you launch an instance, you associate one or more security groups with the instance. You add rules to each security group that allow traffic to or from its associated instances.

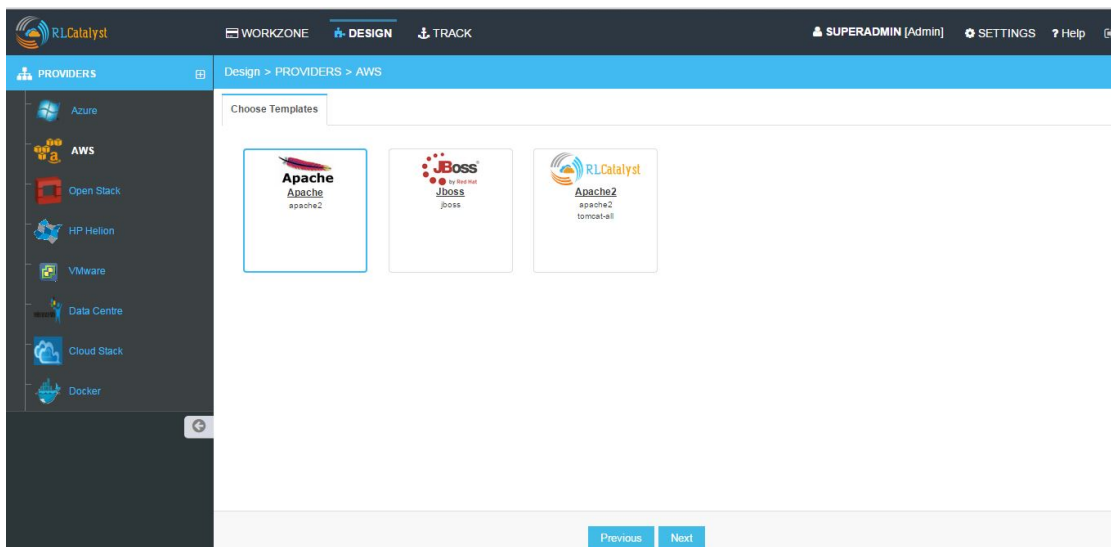
Instances to Launch: Indicates how many instances you want on launch of that particular blueprint.

- In the Design page select **AWS Provider** and choose **Software Stack Template Type** and click Next



- From the Template cards choose template and click Next

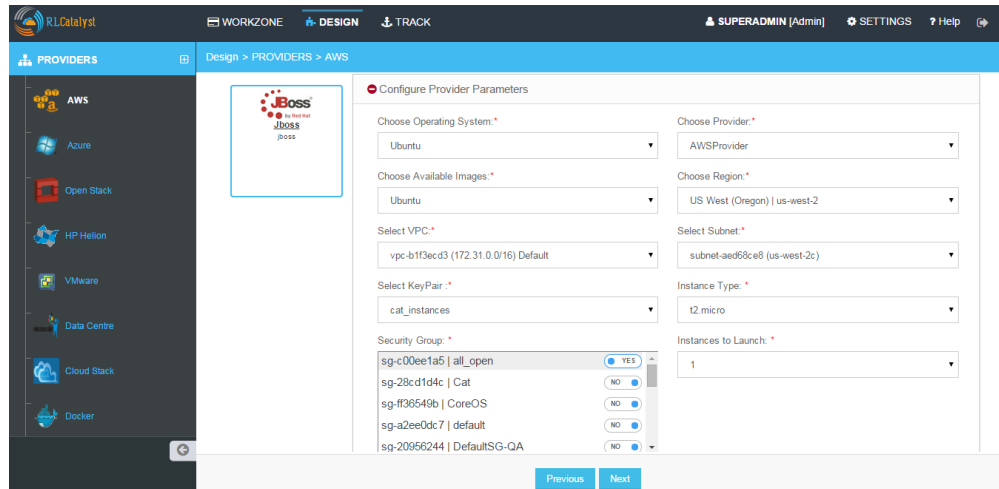
NOTE - If you have not created a template of Software Stack template type in Settings, follow the instructions at *Template Types*.



- Once you choose the Template, Enter the details for creating the BluePrint

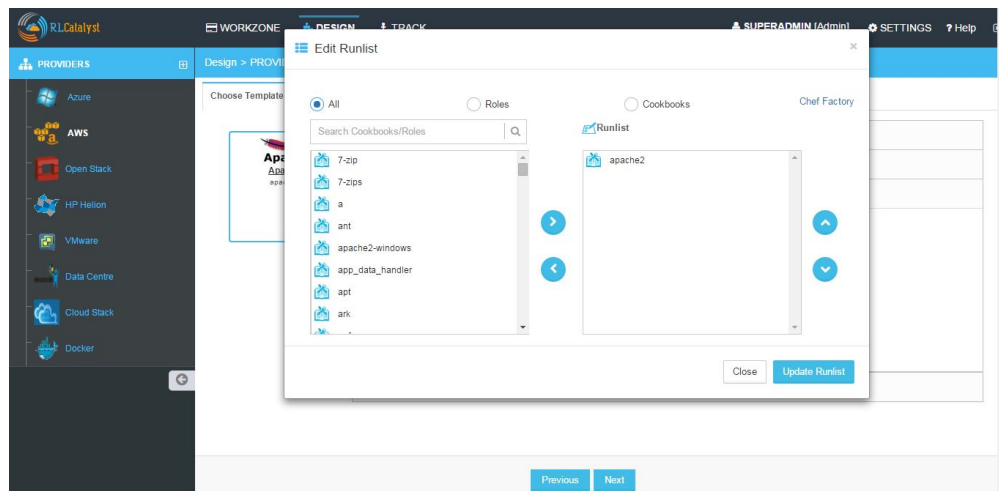
Configure Provider Parameters - Choose **OS, Provider, Image, Region, VPC, Subnet, Keypair, Instance Type, Security Group, No of Instances**.

Configure Organization Parameters - Choose Organization, Business Group, Project and Enter Blueprint name.



Configure Runlist Parameters - Click + icon (Edit Runlist), select Cookbooks / Roles and add to Runlist and update runlist.

NOTE - You can customize the order of Runlist after adding to runlist and then clicking on up and down arrows.



Configure Application - Application URL allows you to navigate to the application from the Instance Card.

Application URL - Add the Application Name and URL in below format `[http://protect\T1\textdollarhost:port/appname]` and save.

Following video demonstrates how to create Blueprint for AWS provider in RLCatalyst:

2. Docker

- In the Design page select **AWS** provider and choose **Docker Template Type** and click Next
- Choose the Docker Template and click Next

NOTE - If you have not created a template of Docker template type in Settings, follow the instructions at [Template Types](#).

- **Configure Organization Parameters** - Choose **Organization**, **Business Group**, **Project** and Enter **Blueprint name**

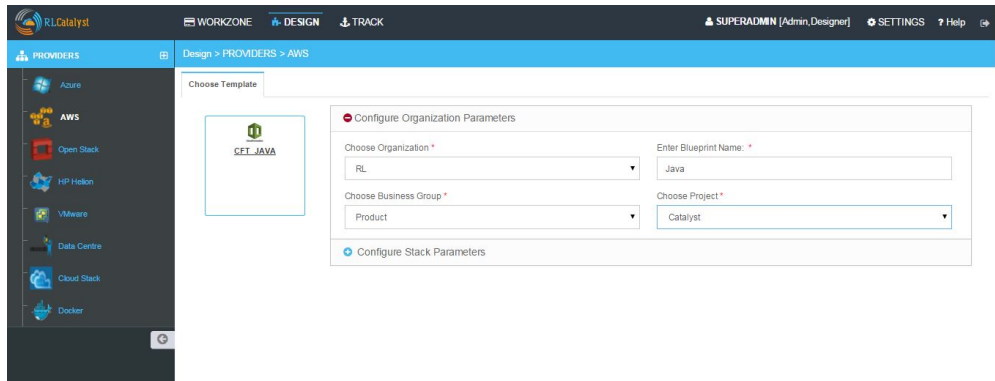
- Click **Launch Parameters** icon and Enter **Container Name**, **Port mappings**, **Volumes**, **Volumes-from**, **Linked Container name**, **Environment variables**, **Start up Command**, **Additional StartUp command** and **Add**.

3. Cloud Formation

- In the Design page select **AWS** provider and choose **Cloud Formation Template Type** and click Next
- Choose the Cloud Formation Template and click Next

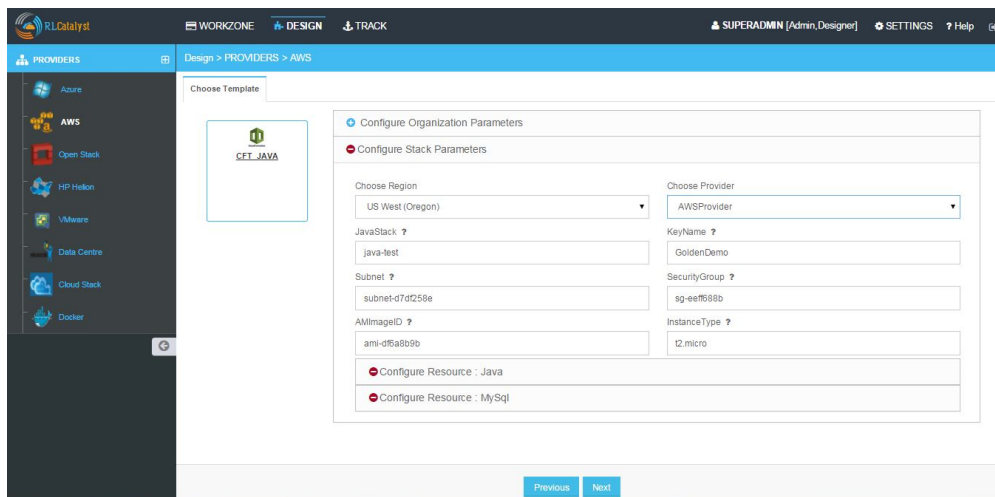
NOTE - If you have not created a template of Cloud formation template type in Settings, follow the instructions at [Template Types](#).

- **Configure Organization Parameters** - Choose **Organization**, **Business Group**, **Project** and Enter **Blueprint name**



- **Configure Stack Parameters - Choose Region, Provider**

NOTE - Remaining fields are the attributes present in the JSON file which you uploaded while creating Cloud Formation Template.



Following video demonstrates how to create and launch cloud formation template in RLCatalyst:

4. OSImage

- In the Design page select **AWS Provider** and choose **OSImage Template Type** and click Next
- From the Template cards choose OSImage Template and click Next

NOTE-

1. In OSImage template type the VMImages which you created in Settings -> *VM Images* page will be listed, based on the selection of the provider in the provider tree.
2. Suppose If you select ubuntu image, In Choose provider parameters section **Choose operating system, Choose Provider, choose available image** is selected by default.
3. Similarly when you select Centos or Windows image, In Choose provider parameters section corresponding operating system, provider and available image will be selected by default.

Configure Provider Parameters - Choose Region, VPC, Subnet, Keypair, Instance Type, Security Group, No of Instances.

Configure Organization Parameters - Choose Organization, Business Group, Project and Enter Blueprint name.

Configure Runlist Parameters - Click + icon (Edit Runlist), select **Cookbooks / Roles** and add to Runlist and update runlist.

Configure Application - Application URL allows you to navigate to the application from the Instance Card.

Application URL - Add the Application Name and URL in below format [<http://protect\T1\textdollarhost:port/appname>] and Save.

Creating BluePrints for AZURE provider

1. SoftwareStack

Security Group Ports: Security group ports contains a list of Access Control List rules that allow or deny network traffic to your VM instances in a Network.

Locations: It enables Azure customers to achieve higher performance and it supports their requirements and preferences regarding data location.

- In the Design page choose **AZURE** provider and **SoftwareStack Template Type** and click Next
- Choose the SoftwareStack Template and click Next

NOTE - If you have not created a template of SoftwareStack template type in Settings, follow the instructions at [Template Types](#).

Configure Provider Parameters - Choose **OS, Provider, Image, Security Group Ports, Location, VPC, Subnet, Keypair, Instance Size, No of Instances**.

Configure Organization Parameters - Choose **Organization, Business Group, Project** and Enter **Blueprint name**.

Configure Runlist Parameters - Click + icon (Edit Runlist), select **Cookbooks / Roles** and add to Runlist and update runlist.

Configure Application - Application URL allows you to navigate to the application from the Instance Card.

Application URL - Add the Application Name and URL in below format [<http://protect\T1\textdollarhost:port/appname>] and Save.

2. OSImage

- In the Design page select **AZURE** Provider and choose **OSImage Template Type** and click Next
- From the Template cards choose OSImage Template and click Next.

NOTE-

1. In OSImage template type the VMImages which you created in Settings -> [VM Images](#) page will be listed, based on the selection of the provider in the provider tree
2. Suppose If you select ubuntu image, In Choose provider parameters section **Choose operating system, Choose Provider, choose available image** is selected by default.
3. Similarly when you select Centos or Windows image, In Choose provider parameters section corresponding operating system, provider and available image will be selected by default.

Configure Provider Parameters - Choose **Security Group Ports, Location, VPC, Subnet, Keypair, Instance Size, No of Instances**.

Configure Organization Parameters - Choose **Organization, Business Group, Project** and Enter **Blueprint name**.

Configure Runlist Parameters - Click + icon (Edit Runlist), select **Cookbooks / Roles** and add to Runlist and update runlist.

Configure Application - **Application URL** allows you to navigate to the application from the Instance Card.

Application URL - Add the Application Name and URL in below format `[http://\protect\T1\textdollarhost:port/appname]` and Save.

3. ARMTemplate

Resource group: It is a container that holds related resources for an application. The resource group could include all of the resources for an application, or only those resources that are logically grouped together.

- In the Design page select **AZURE** Provider and choose **ARMTemplate Type** and click Next
- Choose the ARMTemplate and click Next

NOTE - If you have not created a template of ARM template type in Settings, follow the instructions at [Template Types](#).

NOTE - Make sure You have created Resource group in your Azure portal

- **Configure Organization Parameters** - Choose **Organization, Business Group, Project** and Enter **Blueprint name**
- **Configure Template Parameters** - Choose **Provider, Resource Group**

NOTE - Remaining fields are the attributes present in the JSON file which you uploaded while creating ARMTemplate.

Following video demonstrates how to create ARM Template, Blueprint, Launch, Update Configuration, Connect to the Instance in RLCatalyst:

Creating BluePrints for OpenStack provider

1. SoftwareStack

Flavours: Virtual hardware templates are called “flavors”. It defines sizes for RAM, disk, number of cores, and so on.

- In the Design page select **OpenStack** Provider and choose **SoftwareStack Template Type** and click Next
- Choose the SoftwareStack Template and click Next

NOTE - If you have not created a template of SoftwareStack template type in Settings, follow the instructions at [Template Types](#).

Configure Provider Parameters - Choose **OS, Provider, Image, Flavour, Network, Subnet, Security Group**.

Configure Organization Parameters - Choose **Organization, Business Group, Project** and Enter **Blueprint name**.

Configure Runlist Parameters - Click + icon (Edit Runlist), select **Cookbooks / Roles** and add to Runlist and update runlist.

Configure Application - **Application URL** allows you to navigate to the application from the Instance Card.

Application URL - Add the Application Name and URL in below format `[http://\protect\T1\textdollarhost:port/appname]` and Save.

2. OSImage

- In the Design page select **OpenStack** Provider and choose **OSImage Template Type** and click Next
- Choose the OSImage Template and click Next

NOTE-

1. In OSImage template type the VMImages which you created in Settings -> [VM Images](#) page will be listed, based on the selection of the provider in the provider tree.
2. Suppose If you select ubuntu image, In Choose provider parameters section **Choose operating system, Choose Provider, choose available image** is selected by default.
3. Similarly when you select Centos or Windows image, In Choose provider parameters section corresponding operating system, provider and available image will be selected by default.

Configure Provider Parameters - Choose **Flavour, Network, Subnet, Security Group**.

Configure Organization Parameters - Choose **Organization, Business Group, Project** and Enter **Blueprint name**.

Configure Runlist Parameters - Click + icon (Edit Runlist), select **Cookbooks / Roles** and add to Runlist and update runlist.

Configure Application - **Application URL** allows you to navigate to the application from the Instance Card.

Application URL - Add the Application Name and URL in below format [[http://protect\T1\textdollarhost:port/appname](#)] and Save.

Following video demonstrates how to create Openstack Blueprint, Launch, Update Configuration, Connect to Instance, Add Application URL in RLCatalyst:

Creating BluePrints for VMware provider

Datastore: A datastore is a manageable storage entity, usually used as a repository for virtual machine files including log files, scripts, configuration files, virtual disks, and so on.

1. SoftwareStack

- In the Design page select **VMware** Provider and choose **SoftwareStack Template Type** and click Next
- Choose the SoftwareStack Template and click Next

NOTE - If you have not created a template of SoftwareStack template type in Settings, follow the instructions at [Template Types](#).

Configure Provider Parameters - Choose **OS, Provider, Image, Data Store, No of Instances**.

Configure Organization Parameters - Choose **Organization, Business Group, Project** and Enter **Blueprint name**.

Configure Runlist Parameters - Click + icon (Edit Runlist), select **Cookbooks / Roles** and add to Runlist and update runlist.

Configure Application - **Application URL** allows you to navigate to the application from the Instance Card.

Application URL - Add the Application Name and URL in below format [[http://protect\T1\textdollarhost:port/appname](#)] and Save.

2. OSImage

- In the Design page select **VMware** Provider and choose **OSImage Template Type** and click Next
- Choose the OSImage Template and click Next

NOTE-

1. In OSImage template type the VMImages which you created in Settings -> *VM Images* page will be listed, based on the selection of the provider in the provider tree
2. Suppose If you select ubuntu image, In Choose provider parameters section **Choose operating system, Choose Provider, choose available image** is selected by default.
3. Similarly when you select Centos or Windows image, In Choose provider parameters section corresponding operating system, provider and available image will be selected by default.

Configure Provider Parameters - Choose **Data Store, No of Instances**.

Configure Organization Parameters - Choose **Organization, Business Group, Project** and Enter **Blueprint name**.

Configure Runlist Parameters - Click + icon (Edit Runlist), select **Cookbooks / Roles** and add to Runlist and update runlist.

Configure Application - **Application URL** allows you to navigate to the application from the Instance Card.

Application URL - Add the Application Name and URL in below format [[http://protect\T1\textdollarhost:port/appname](#)] and Save.

RL Catalyst Workzone is an option all the settings take action. Workzone has below options:

Infrastructure:

- a. Instances : It displays the numbers of instances that currently launched and running.
- b. Blueprints : This options allows you to launch instances.
- c. Cloud Formation : It displays the list of stacks launched from cloud formation blueprints.
- d. AzureARM: It displays the list of Deployments launched from Azure ARM Blueprints.
- e. Containers : Container details launched from docker blueprints.

Orchestration: This options allows you to execute one or more tasks/actions on multiple nodes.

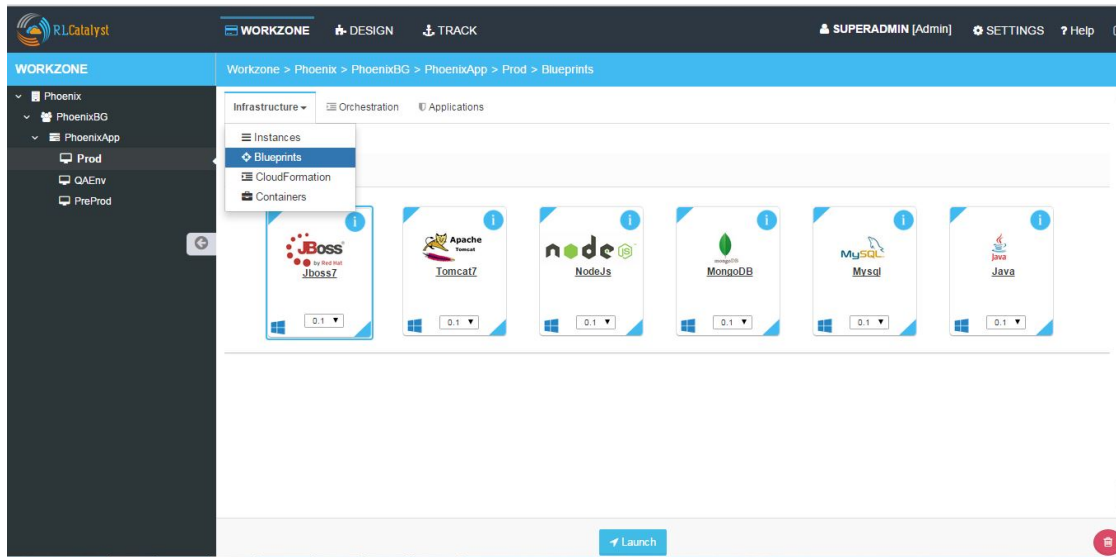
Applications: Here user can deploy new applications and deployed applications will be shown in card or table view will all details.

Here user can deploy new applications and deployed applications will be shown in pipeline view and table view with all details.

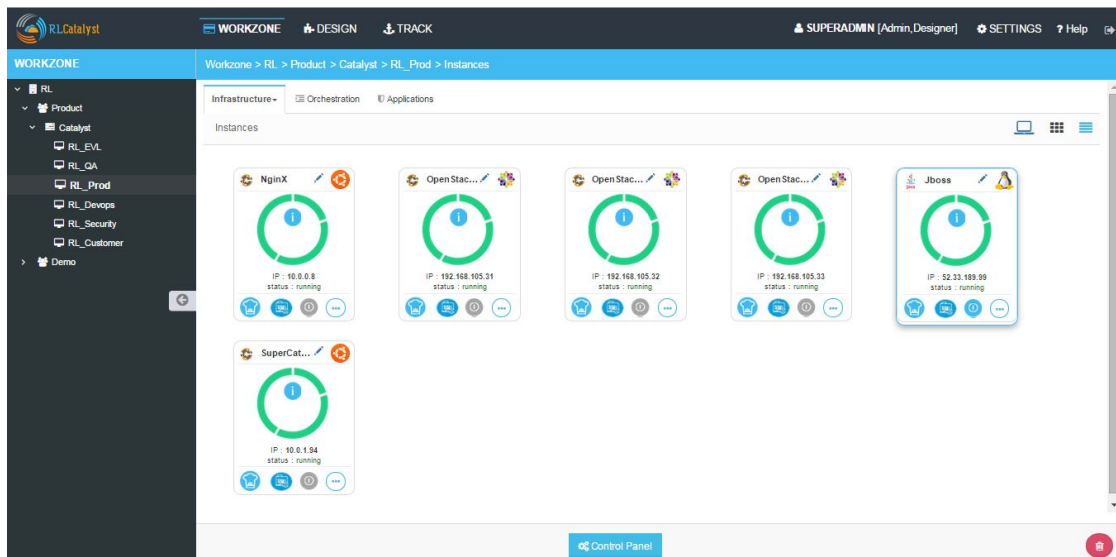
6.1 Launch New Instances

Instances are launched from Blueprints. Follow below steps to launch Instances from blueprints.

- Go to Workzone → Click on Infrastructure drop down → Choose Blueprints



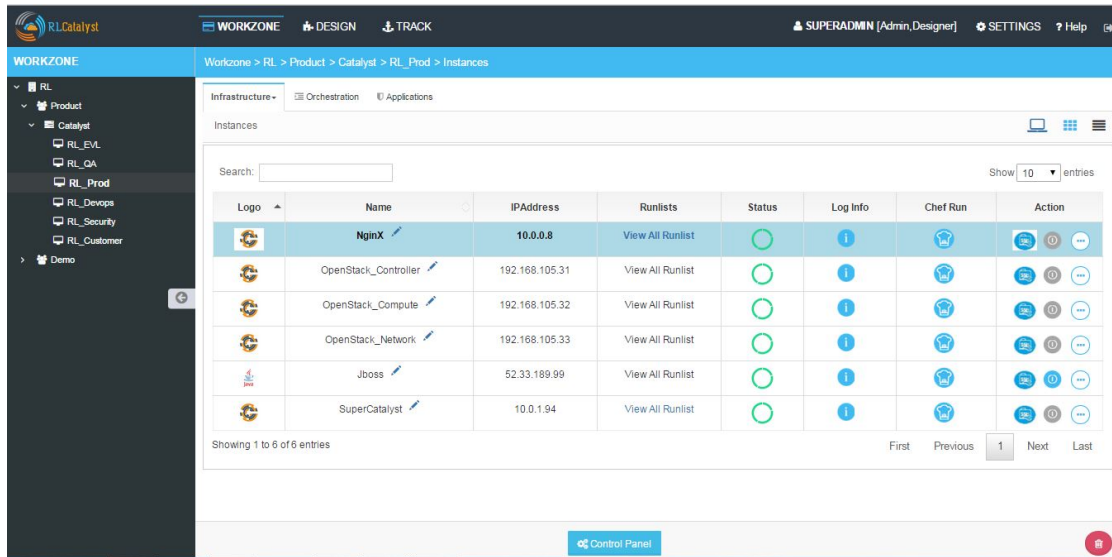
- Select the Blueprint and click on Launch button. Once the 'Launch' button is clicked, the instance will be launched in 5-8 minutes
- Go to Instances tab to view the launched instance



View Instances in Table View RL Catalyst on the Instances page provides you two kinds of views.

- Grid View[Above Image]
- Table View[Below Image]

You can click on the respective button to view the instances.



Instance Actions:

On the launched instance user can perform below actions.

- Chef Client Run
- SSH
- Stop / Start
- MoreInfo
- **Chef Client Run** option allows you to update the Chef Client Runlist. You can add new software to the Runlist and click on **Update**
- **SSH**, this option allows you to open the command prompt Terminal. You can execute your commands directly using this option
- **Stop**, This option allows you to start and stop an Instance
- **More Info**, This option displays the information about the Instance. You can view the logs of the Instance

6.2 Update Configuration

Any extra action on the running instance can be performed via the Chef client Run icon, which is displayed at the bottom left corner. Installing new software, upgrading a software etc can be performed using this icon.

A **Cookbook** is the fundamental unit of configuration and policy details that Chef uses to bring a node into a specific state. This just means that Chef uses cookbooks to perform work and make sure things are as they should be on the node.

A **Role** is a way to define certain patterns and processes that exist across nodes in an organization as belonging to a single job function. Each role consists of zero (or more) attributes and a run-list. Each node can have zero (or more) roles assigned to it.

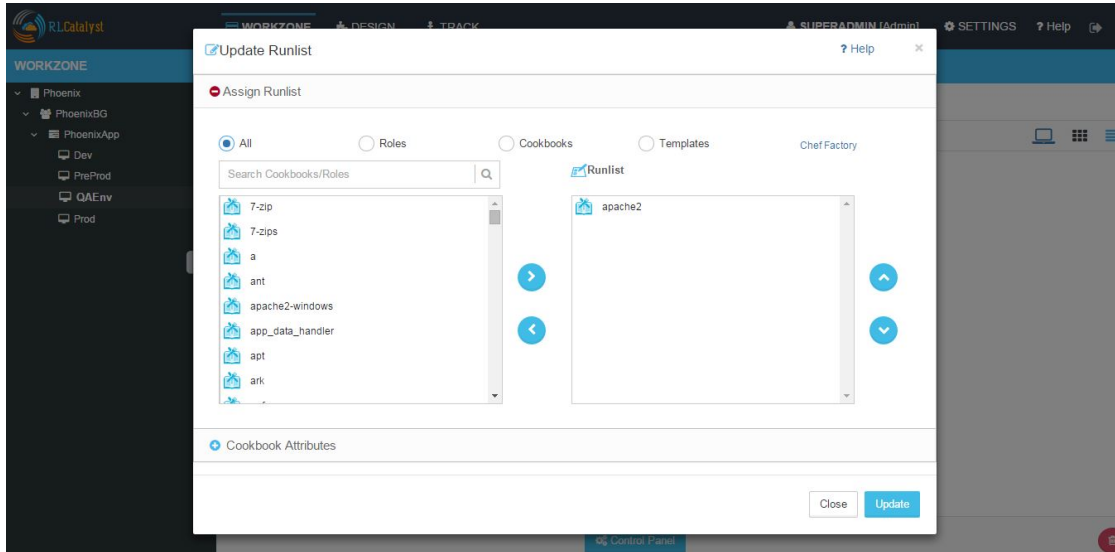
A **Run-list** defines all of the information necessary for Chef to configure a node into the desired state.

A **Template** is user defined skeleton structure which helps the user to add Cookbooks, Roles that can be run on the Instance, which you created in Settings.

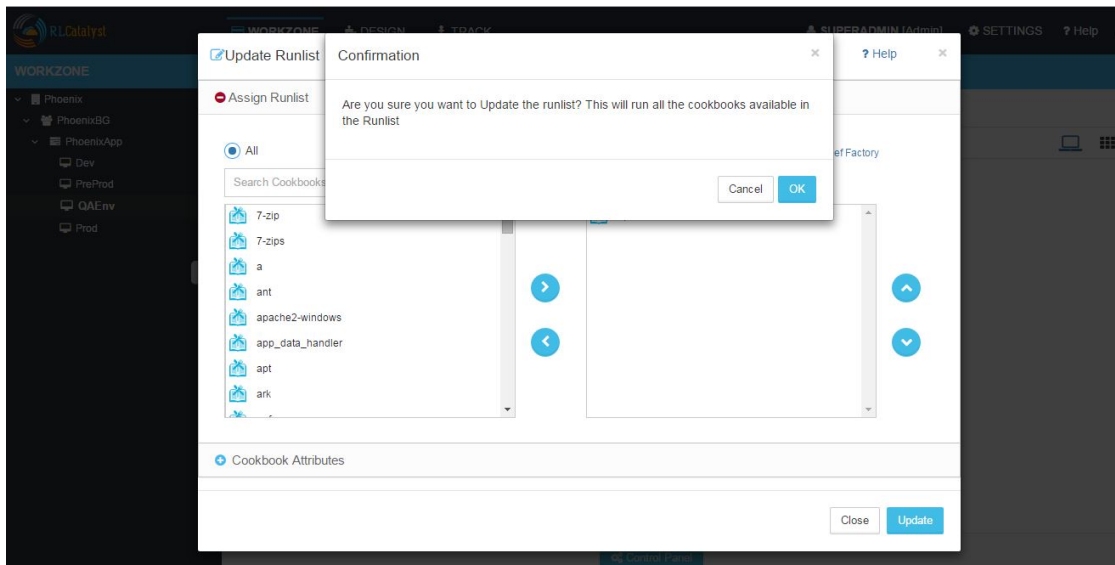
Follow below steps to run chef client run:

- Click on Chef client run icon

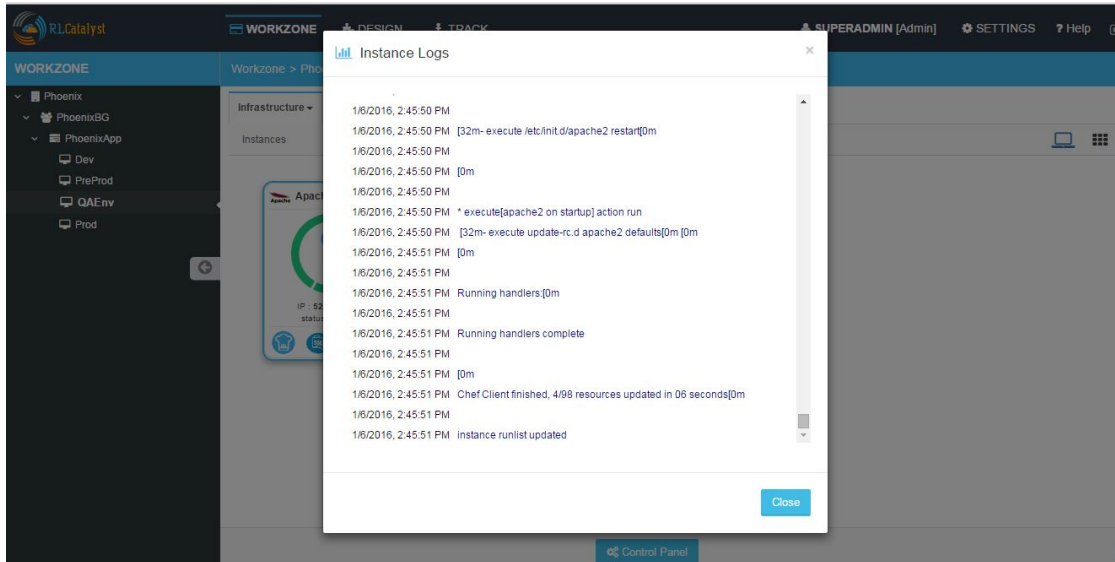
From here you can add Roles, Cookbooks [Present in your chef server] and Templates [Created in Settings] to the Runlist.



- Select any cookbook and move to Runlist by clicking > icon
- Click on Update button.
- Click OK on confirmation popup to update runlist



- Instance Logs window will be displayed to see the logs

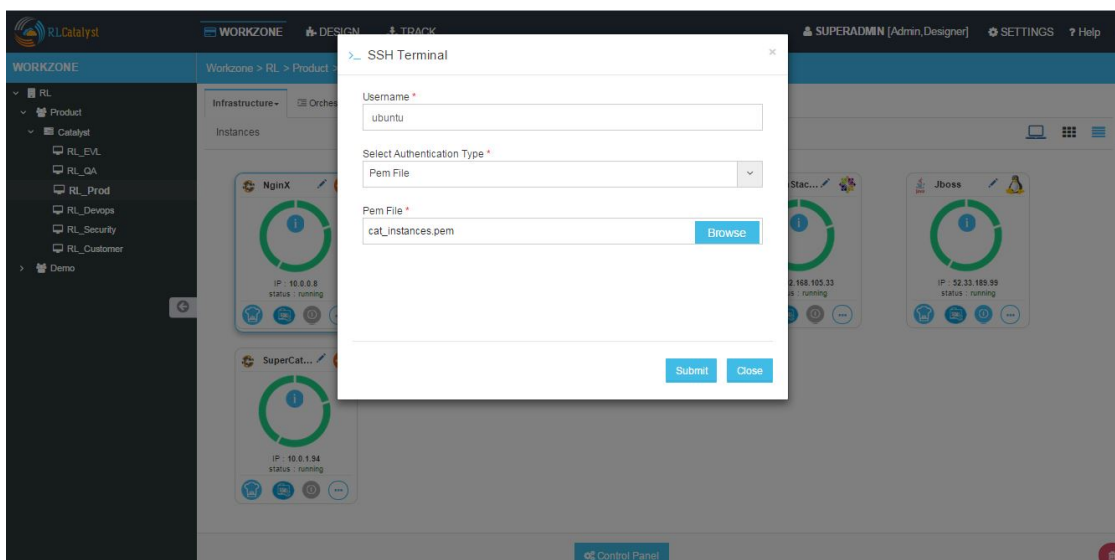


- Click on Close button to close the Instance logs window

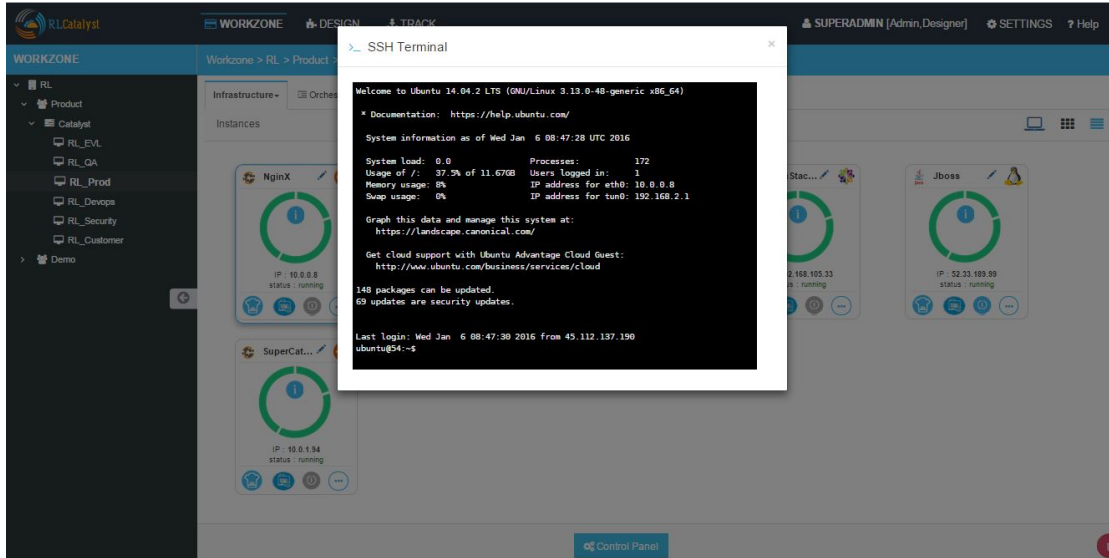
6.3 Connect to the Instance

Follow below steps to connect to Instance:

- Click on the SSH icon present on the instance
- Enter the Username of the Instance
- Select the authentication type by selecting password / pem file
- Enter the Password or Browse the Pem file



- Click on Submit button



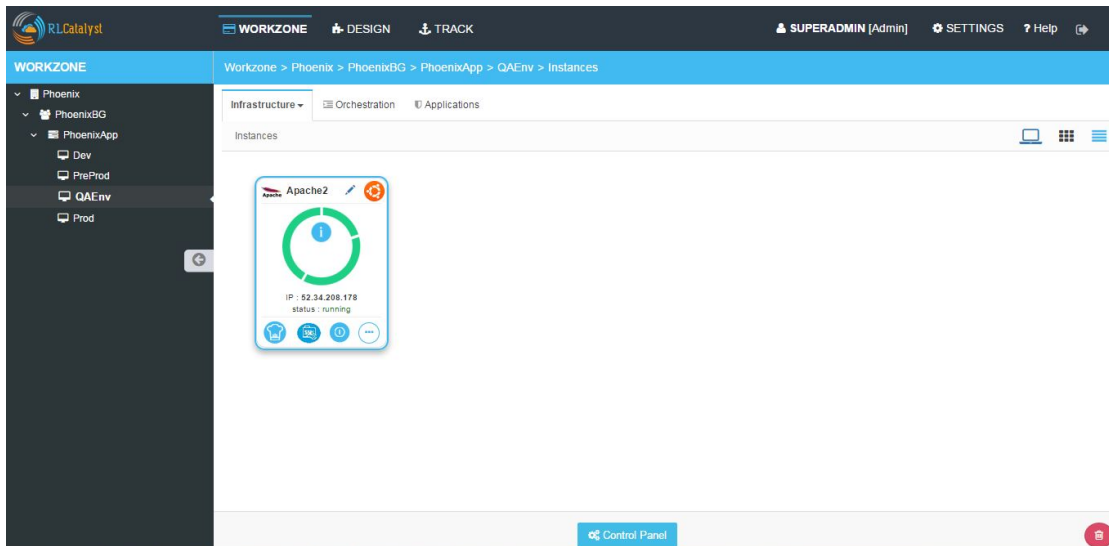
Following video demonstrates how to Launch Blueprint, Update Configuration and Connect to the Instance in RLCatalyst:

6.4 Start Instance

Catalyst allows you to start the instance which is already stopped.

Follow below steps to start the Instance:

- Click on the **Start** icon of the stopped instance
- Instance will be started and turn to Green color. Chef Client , SSH button will be enabled

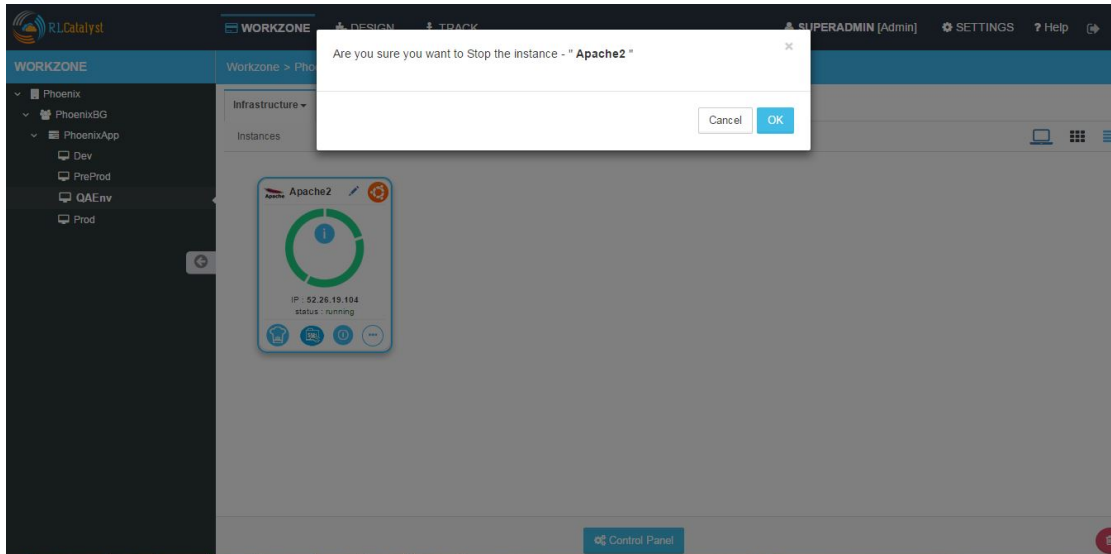


6.5 Stop Instance

Catalyst allows you to stop the instance which is already Running.

Follow below steps to stop the Instance:

- Click on the Stop icon present on the instance
- Click OK on confirmation popup

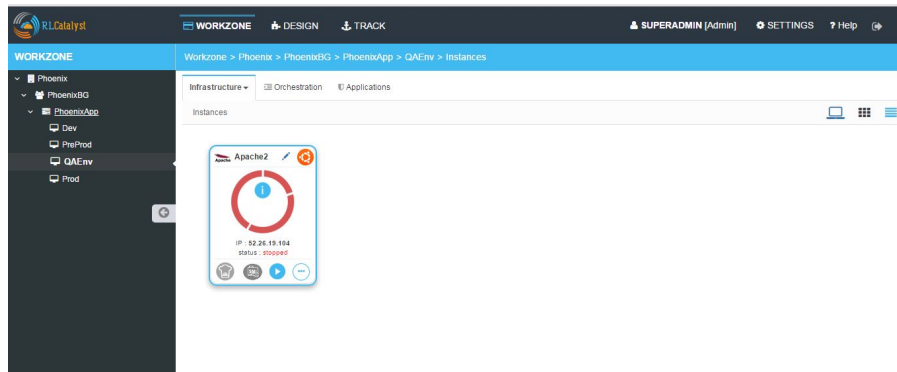


- Instance status is showing as stopped and red icon will be shown

Note: User can perform stop / start action only for the launched node from catalyst.

Note: For the imported node from IP address Stop button will be grayed out will be shown later.

Note: For the Stopped Instance, Chef client SSH buttons will be disabled.

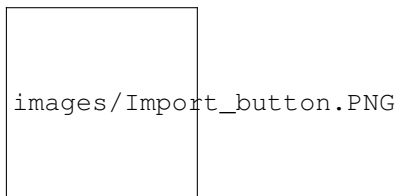




6.6 Import By IP

In the Instances page, you can import any running instances to the catalyst application using Import By IP option, follow the below steps to import:

- Click on Import button in righttop corner of the page.

- **Import Instance By IP** window opened.
- Provide the IP address which needs to be Imported
- Choose the operating system from **Choose Operating System** drop down list
- Provide the user name in the **Username** box
- Provide Configuration Management option like RL Chef Server.
- Choose authentication type from the **Choose Authentication Type** drop down list. RL Catalyst provide two types of authentication, you can choose Password or by uploading PEM file
- Type **Password or upload PEM file**
- Provide the application name in the **Name** box and the host **URL** in the URL box
- You can also Add application with name and URL details.
- Click **Import** to start importing the Instance

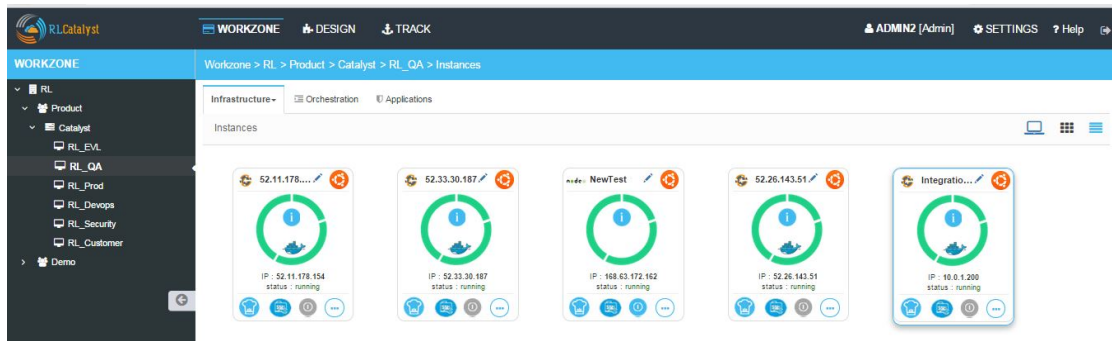


 **Import Instance By IP** 

Enter IP Address *	Choose Operating System *
<input type="text" value="56.35.36.14"/>	<input type="text" value="Ubuntu"/>
Username *	Config Management *
<input type="text" value="ubuntu"/>	<input type="text" value="ChefServer"/>
Choose Authentication Type *	Password *
<input type="text" value="Password"/>	<input type="text" value="....."/>
Application Name	Application URL
<input type="text" value="catalyst"/>	<input type="text" value="http://\$host:3001 "/>

[Add Application URL](#)

- Node will be imported and displayed in the instances tab. For the imported node Stop button will be disabled

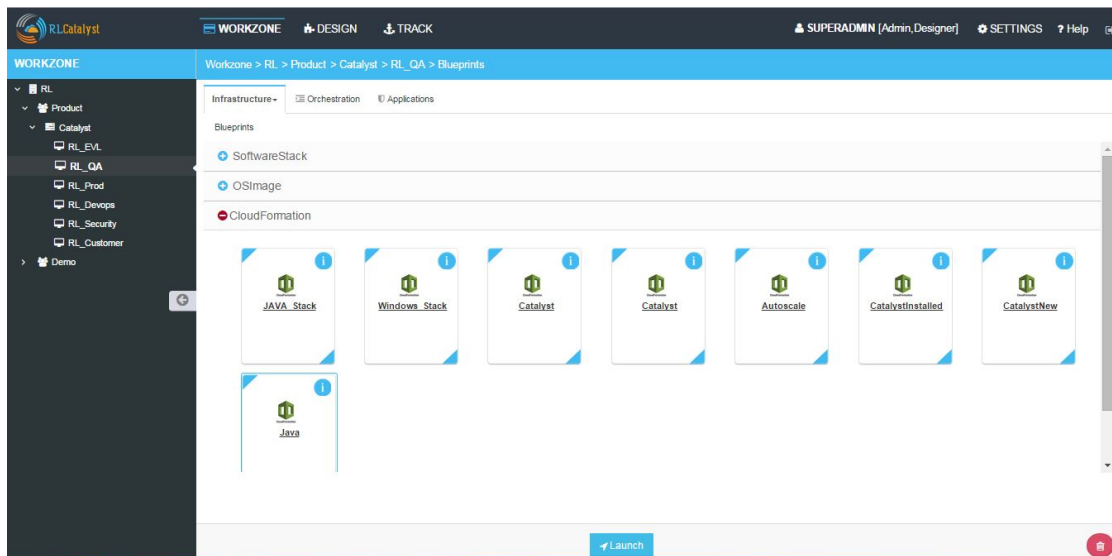


Following video demonstrates how to Import a Node by IP Address and connect to Instance from RLCatalyst:

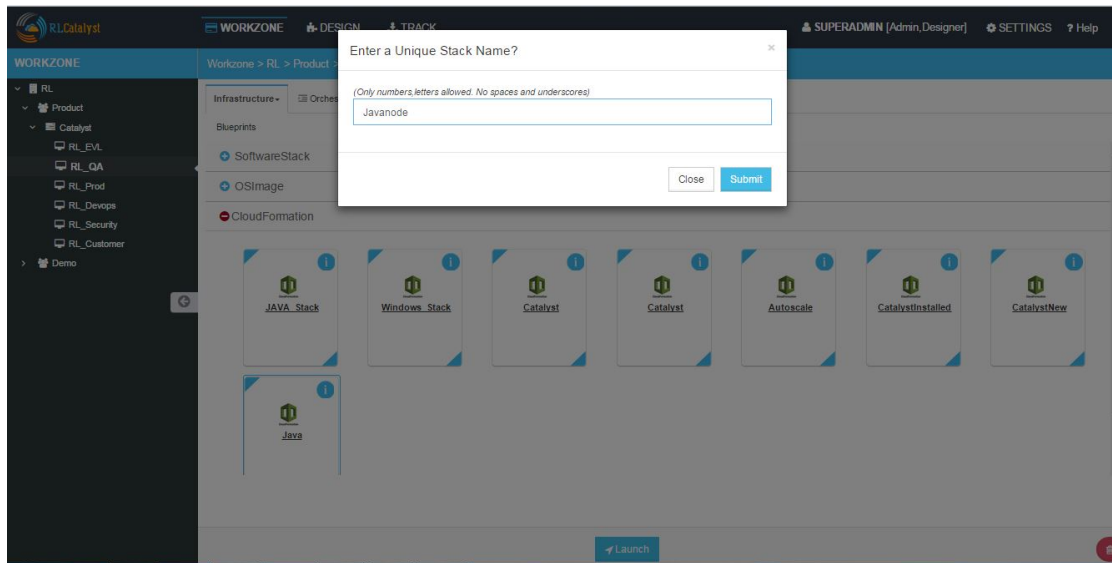
6.7 Cloud Formation Templates

Follow below steps to launch Cloud formation blueprints:

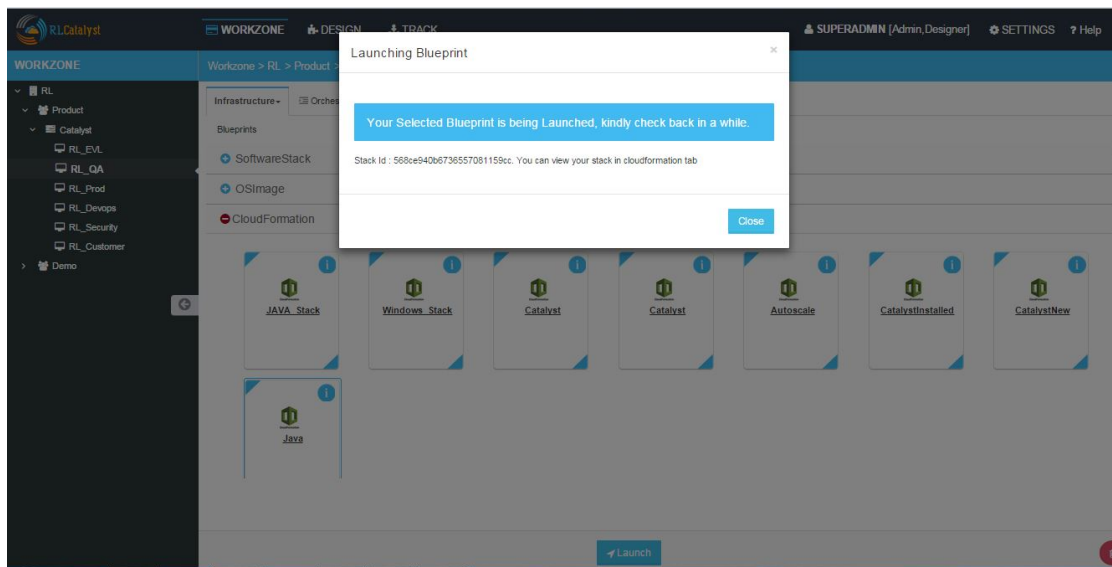
- Go to Workzone → Click on Infrastructure dropdown → Select Blueprints option → Click on ‘Cloud Formation’ template type



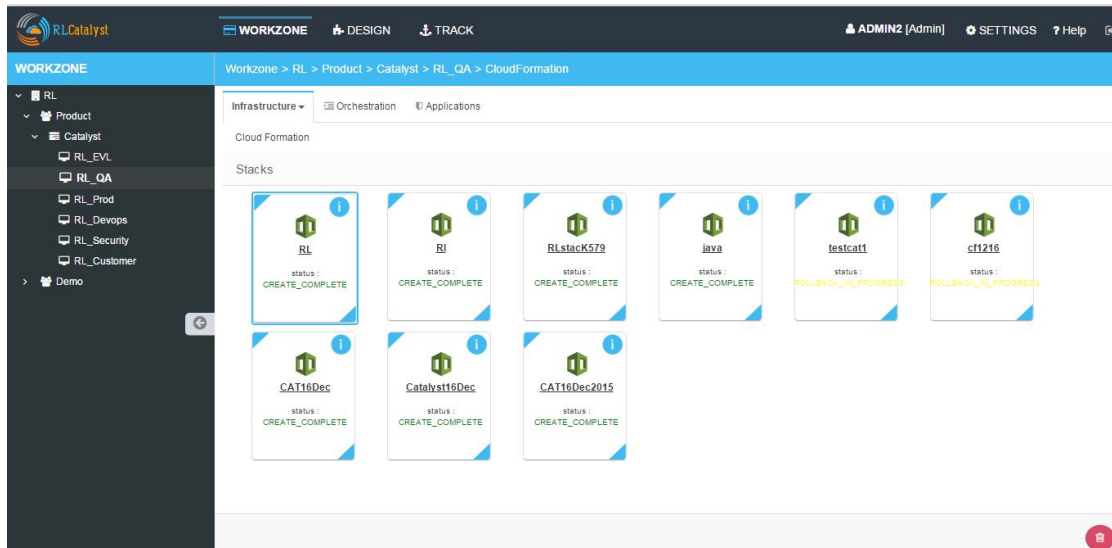
- Select the cloud formation blueprint and click on Launch button
- Enter the Unique Stack Name in the popup window



- Click on Submit button
- Confirmation pop will be displayed with Stack ID



- Close the popup
- Go to Infrastructure -> Cloud Formation , the CFT stack will be listed

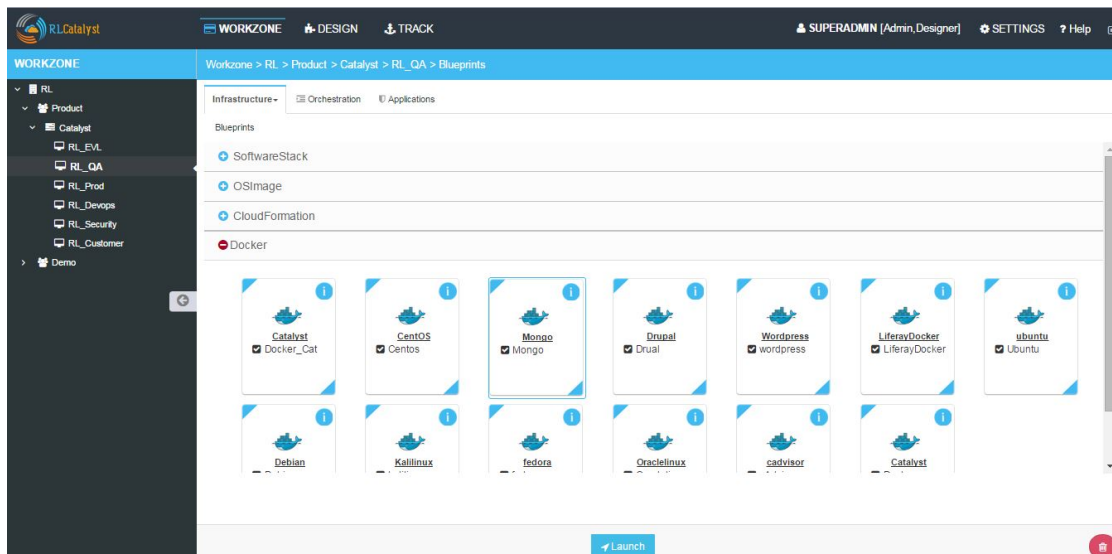


- Go to Instances tab to see the launched Instance

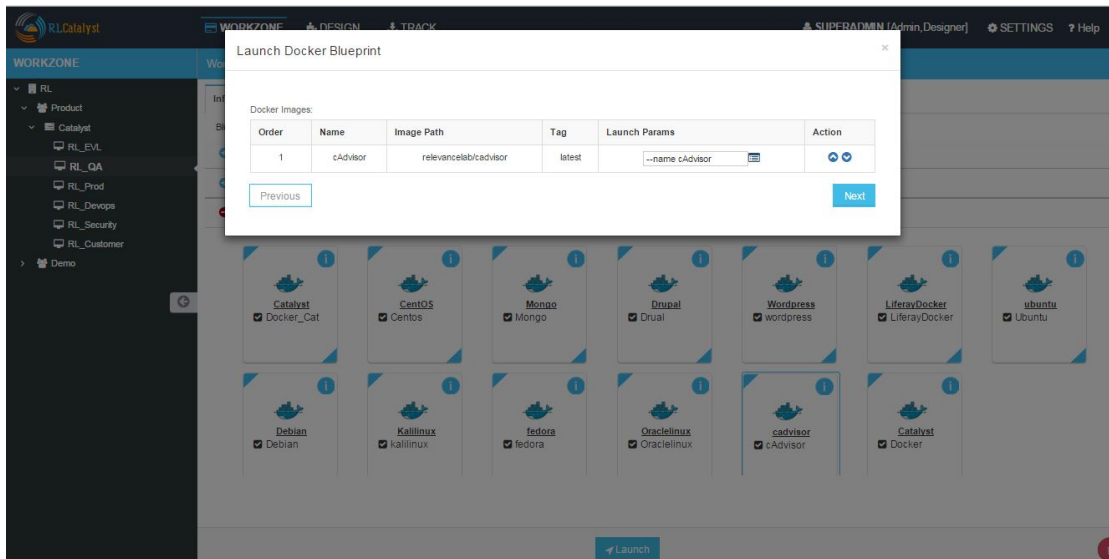
6.8 Docker Blueprints

Follow below steps to launch docker blueprints:

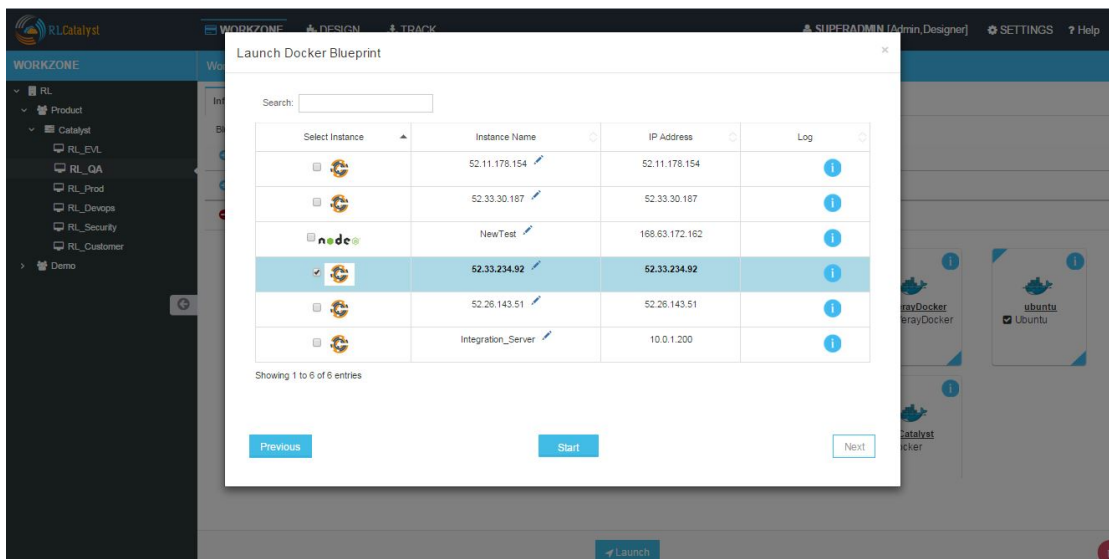
- Go to Workzone → Click on Infrastructure dropdown → Select Blueprints option → Click on ‘Docker’ template type



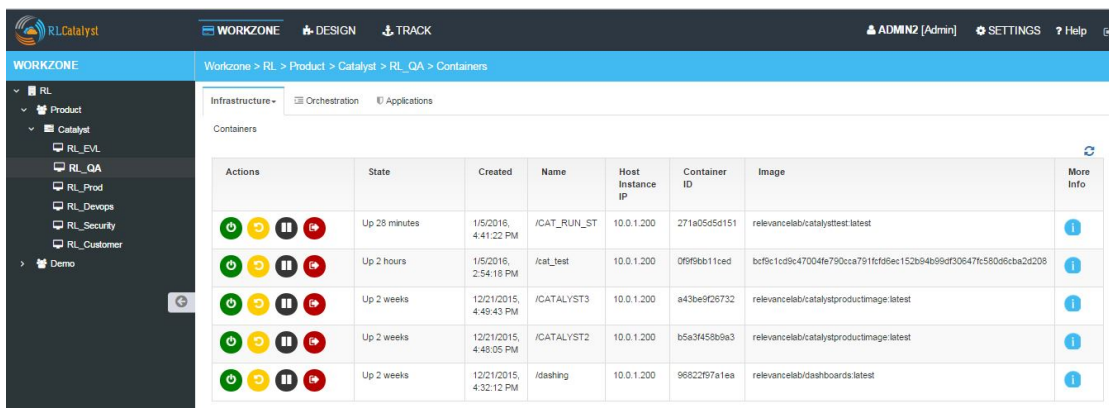
- Select the docker template which is listed and click on Launch button
- Click OK on the Confirmation popup
- Click Next button in the Launch docker blueprint window



- Select the node on which you are going to launch docker blueprint and click on **Start** button



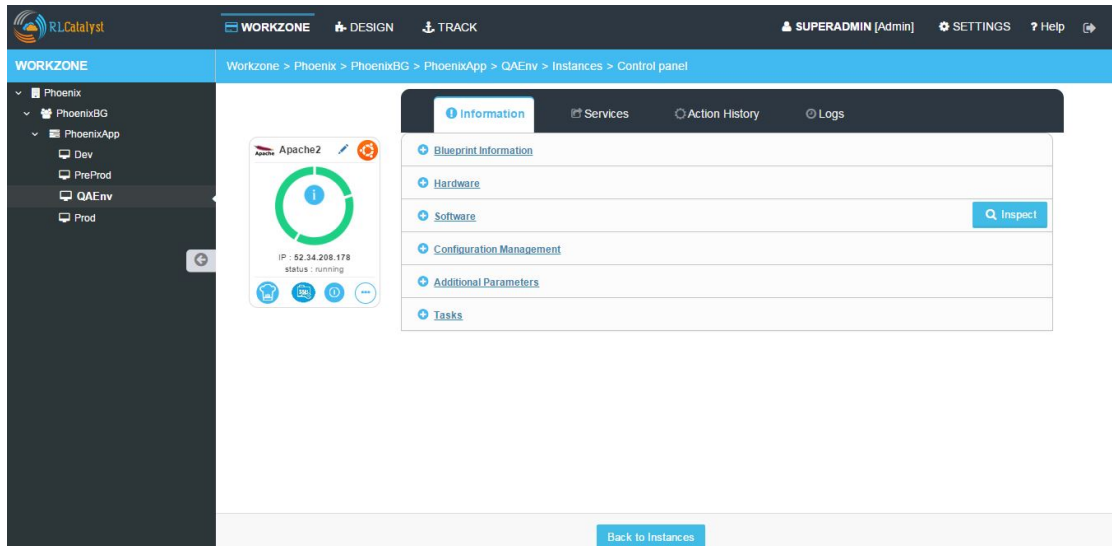
- Logs window will be displayed and wait until the installation successful
- Go to Infrastructure -> Containers tab, the container details will be listed



Following video demonstrates how to Configure Docker and Launch Containers in RLCatalyst:

6.9 Control Panel

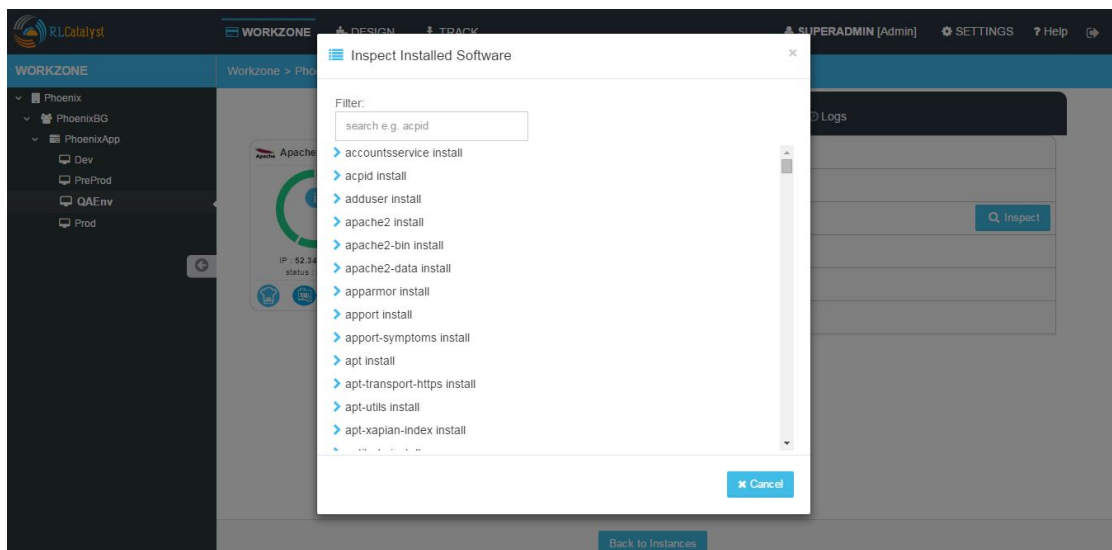
The **Control Panel** option displays the detailed information on the selected **Instance** . It displays information such as Blueprint Information , Hardware information, Software Information, Configuration Management, Additional Parameters, Services, Actions and Logs.



Inspect Software

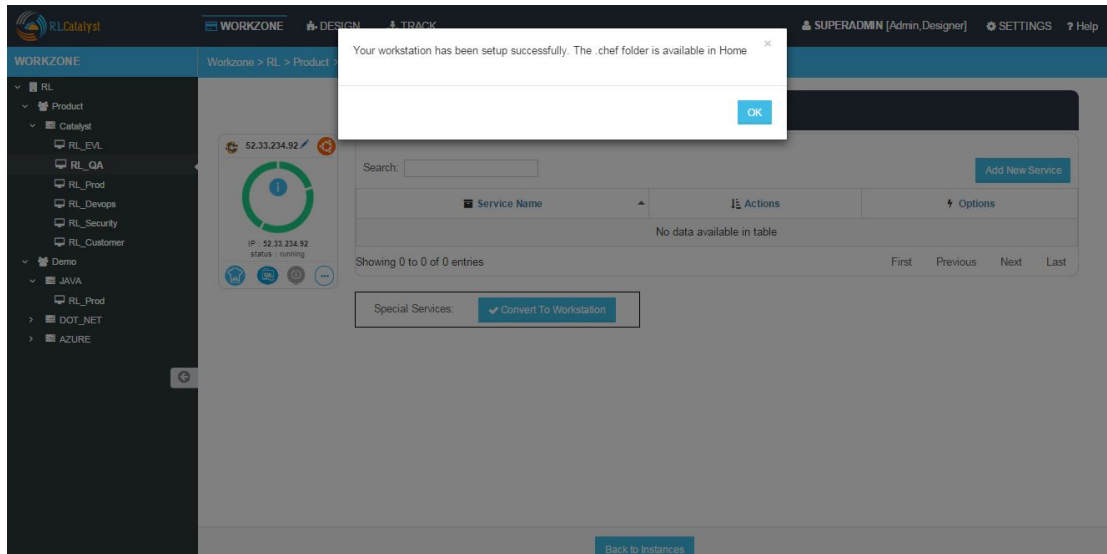
Inspect functionality allows user to know the installed software on the Instance.

- Go to Instance Control panel
- Click on Inspect Software button
- Popup is displayed to know the installed software on the instance



Convert to Workstation

- Go to Instance Control panel → Services tab
- Click on ‘Convert To Workstation’ button
- Click on ‘OK’ button
- Confirmation pop up is displayed saying ‘**Your workstation has been setup successfully. The .chef folder is available in Home.**’



- Click on **OK** button to close the popup

View Action History

Action history feature allows user to view the history of the actions performed on the Instances with complete details.

- Go to Instance Control panel
- Click on Action History tab

Workzone > Phoenix > PhoenixBG > PhoenixApp > QAEnv > Instances > Control panel

Information Services Action History Logs

Type	Status	Data	Timestamp	Users	MoreInfo
Bootstrap	●	runlist : apache2	1/6/2016, 2:27:34 PM	superadmin	i
Chef-Client-Run	●	runlist : apache2	1/6/2016, 2:45:43 PM	superadmin	i
Stop	●	-	1/6/2016, 3:07:57 PM	superadmin	i
Start	●	-	1/6/2016, 3:15:16 PM	superadmin	i

Showing 1 to 4 of 4 entries

First Previous **1** Next Last

Back to Instances

6.10 Orchestration

Orchestration option allows you to execute one or more tasks/actions on multiple nodes.

Chef Task

- Go to your respective **Environment**, click on **Orchestration**
- To add a new task click on the **New** button
- Select the task type from the **Select Task Type** drop down list (Chef)
- Enter a task name in the **Task Name** box
- Select the nodes from the **Select Nodes** list for which you want to assign task
- Click on **Edit Runlist** icon and add cookbooks to the runlist
- Click on **Update runlist** button
- You can also select the **Cookbook Attributes**

RLCatalyst WORKZONE DESIGN TRACK ADMIN2 (Admin) SETTINGS Help

Workzone > RL > Product > Catalyst > RL_QA > Orchestration

WORKZONE

- RL
- Product
- Catalyst
- RL_EVL
- RL_QA
- RL_Prod
- RL_Develop
- RL_Security
- RL_Customer
- Demo

Create New Job

Job Details

Select Job Type: Chef Job Name: Deploy-APP

Job Description:

Chef Server Details for - RLChefServer

Select Nodes:

- ☒ 52.11.178.154
- ☒ 52.33.30.187
- ☐ NewTest
- ☐ 52.26.143.51
- ☐ Integration_Server

Cookbook Attributes:

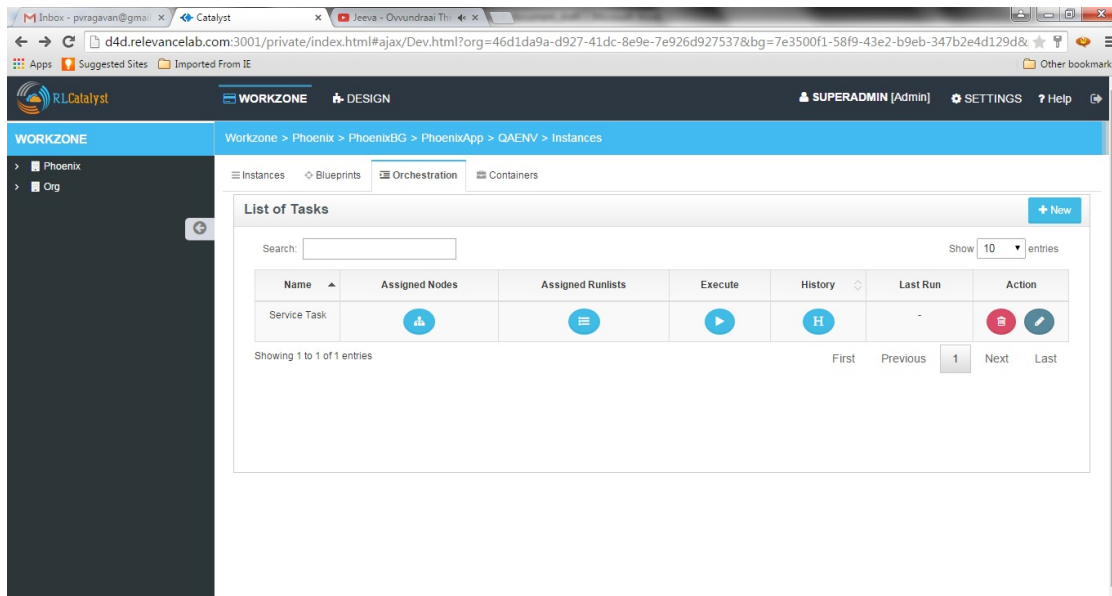
Attributes	Values

Add Runlist View Runlist

- deploy_petstore_app
- deploy_ricatalyst

Cancel Save

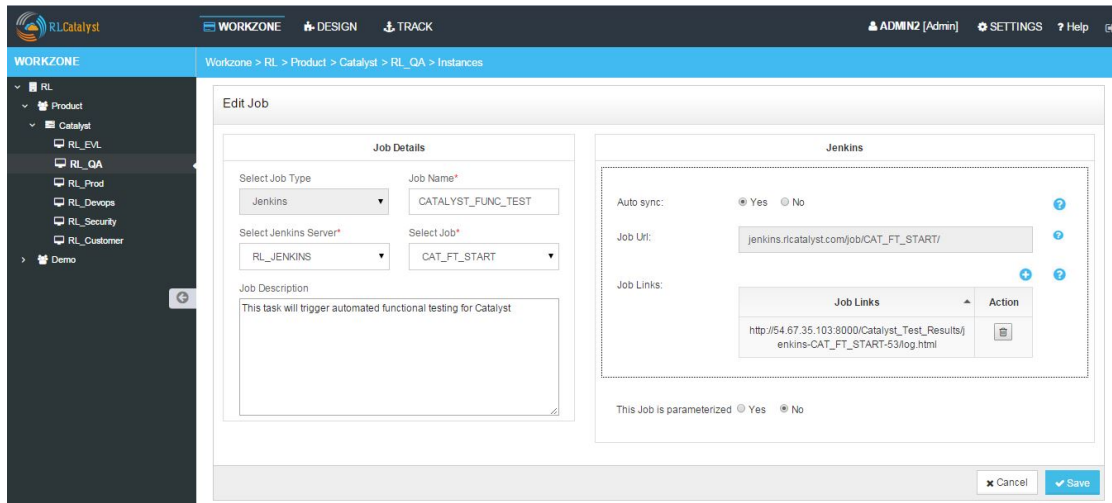
- Click Save button to save the task
- The task is added to the **Orchestration** list



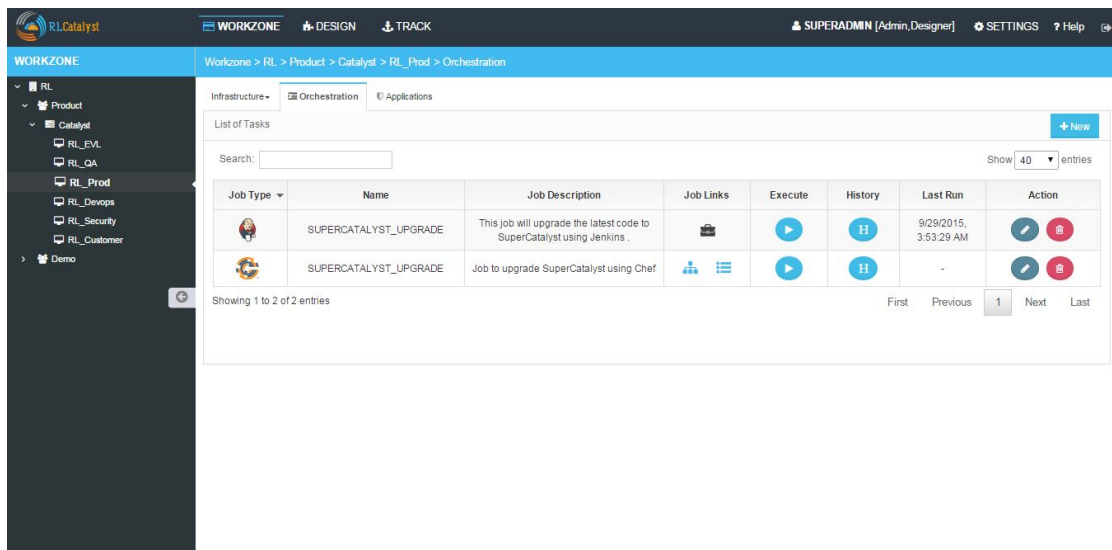
Following video demonstrates how to create and run Chef Task in RLCatalyst:

Jenkins Task

- Go to your respective **Environment**, click on **Orchestration**
- To add a new task click on the **New** button
- Select the task type from the **Select Task Type** drop down list (Jenkins)
- Enter a task name in the **Task Name** box
- Select the server from the **Select Jenkins Server** drop down list
- Select the job from the **Select Job** drop down list
- Select the Auto synch button to '**Yes**' [This will show previously existing task execution history]. If you set to '**NO**' previously existing task history will not be shown
- **Job URL** - The selected job url will be shown here
- Add **Job Links** for the Jenkins task - You can add external links to view the results



- Click Save button to save the task
- The task is added to the Orchestration list



Edit or Remove a Task You can edit or remove a task. Follow the steps below.

- Click on Edit button to edit a task from the Orchestration list
- Click on Delete button to remove a task from the Orchestration list

Execute Task

You can execute a task (Chef and Jenkins) by clicking Execute button in the list of tasks page.

Once you execute the task, Execute logs window will pop-up shows the status of the execution.

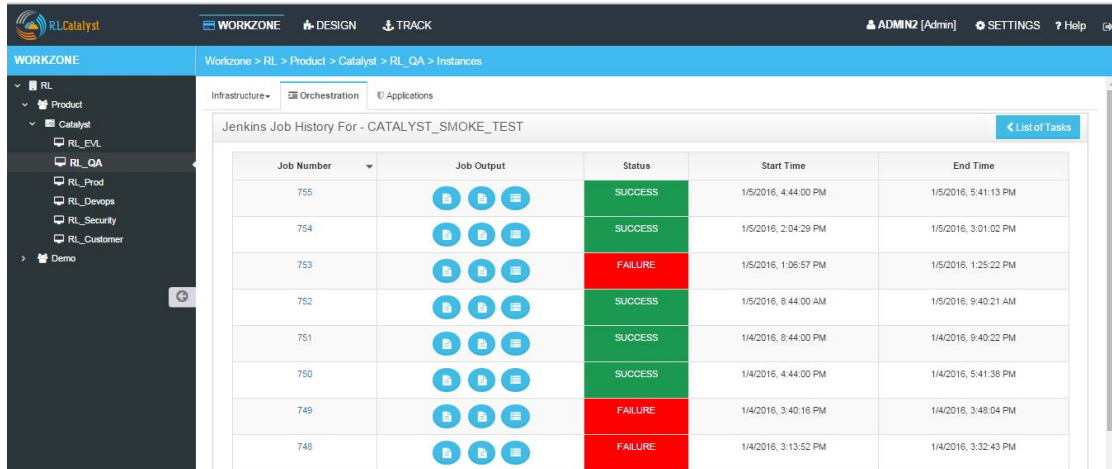
Task History

























You can view the task history by clicking the History button in the list of tasks page. Once you click on the history button, Task History window will pop-up and shows the history of the task.

The following information is shown in the history of task:

- Job number
- Job output links including logs info

- Status
- Start time
- Endtime
- Logs



Job Number	Job Output	Status	Start Time	End Time
755	  	SUCCESS	1/5/2016, 4:44:00 PM	1/5/2016, 5:41:13 PM
754	  	SUCCESS	1/5/2016, 2:04:29 PM	1/5/2016, 3:01:02 PM
753	  	FAILURE	1/5/2016, 1:06:57 PM	1/5/2016, 1:25:22 PM
752	  	SUCCESS	1/5/2016, 8:44:00 AM	1/5/2016, 9:40:21 AM
751	  	SUCCESS	1/4/2016, 8:44:00 PM	1/4/2016, 9:40:22 PM
750	  	SUCCESS	1/4/2016, 4:44:00 PM	1/4/2016, 5:41:38 PM
749	  	FAILURE	1/4/2016, 3:40:16 PM	1/4/2016, 3:48:04 PM
748	  	FAILURE	1/4/2016, 3:13:52 PM	1/4/2016, 3:32:43 PM

Following video demonstrates how to create and execute Jenkins Task in RLCatalyst:

6.11 Application Deployment

RLCatalyst makes your application deployments easy through its Orchestration feature. The artifacts or the build files can be sourced from Nexus or Docker repositories and you can deploy into single or multiple instances. The deployments happens through Jenkins or Chef based tasks , that can be configured from RLCatalyst.

Prerequisites:

1. A repository (Nexus/Docker) should be added from Settings
2. Repository should be attached to one or more projects.
3. There should be connectivity between the repository, the target instances and the RLCatalyst instance
4. There should be to & fro connectivity between RLCatalyst and the target instance

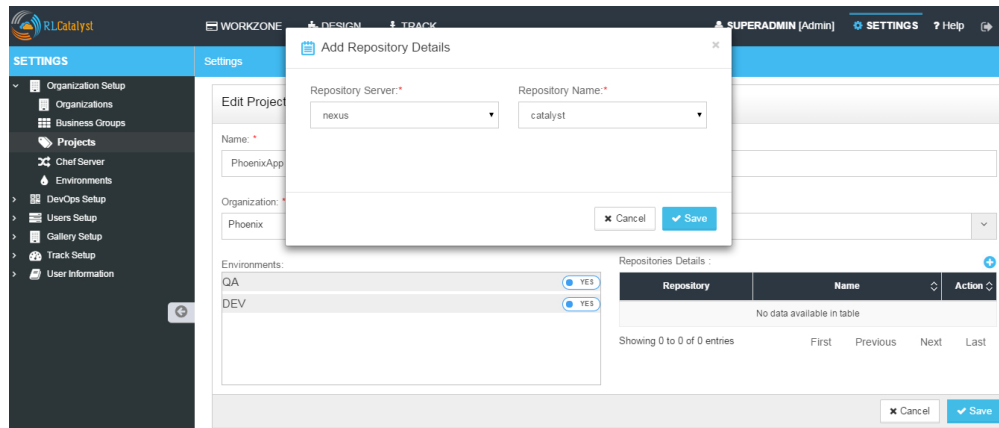
If you have not added a repository in Settings, follow the instructions at [Nexus Server](#).

Once Nexus Server is configured you have to associate Repository details to your Project.

CASE I: First time a new application has been deployed and it is deployed using Catalyst

Follow the below steps :

- Go to Projects Page
- Edit your Project
- Click on + icon present next to Repository Details
- Select your Repository Server and Repository Name

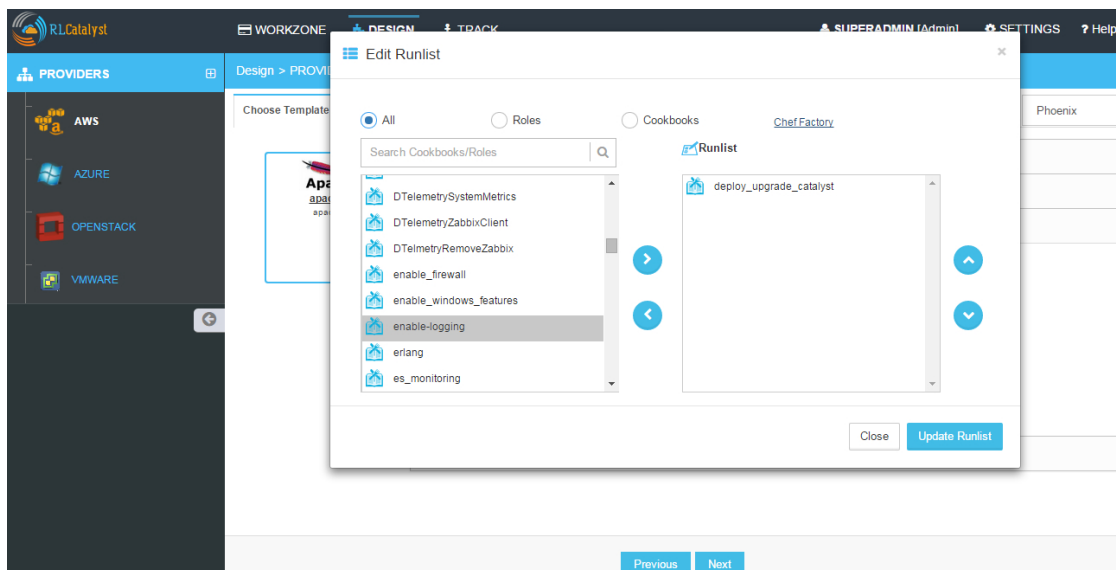


- Click on Save button on Add Repository Details page
- Click on Save button on Edit Project Page

6.12 App Blueprint

Once you associate repository details to your project now start creating blueprint. Follow the below steps:

- Go to Design
- Select Software Stack Template Type and click Next
- Select any Template and click Next
- Configure Provider Parameters by selecting all provider parameters
- Configure Organization Parameters by selecting
- Configure Runlist Parameters by adding **deploy_upgrade_catalyst** cookbook



- Click on Update Runlist
- Expand Configure Application
- Select **Deploy app during Bootstrap** checkbox

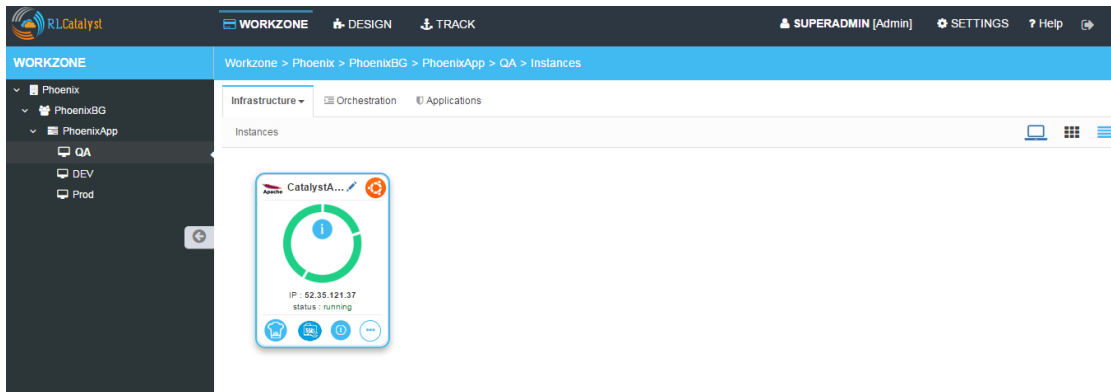
- On selecting checkbox all Repository details will autopopulate and the latest version will be always selected.
[Note: If you select previous version also by default it will take latest version]

- If you want to specify the URL at which the application is running, specify the URL in the format **http://\$host:3001**
- Click on Next button
- Click OK button in Confirm popup window
- Blueprint Saved Successfully message is displayed

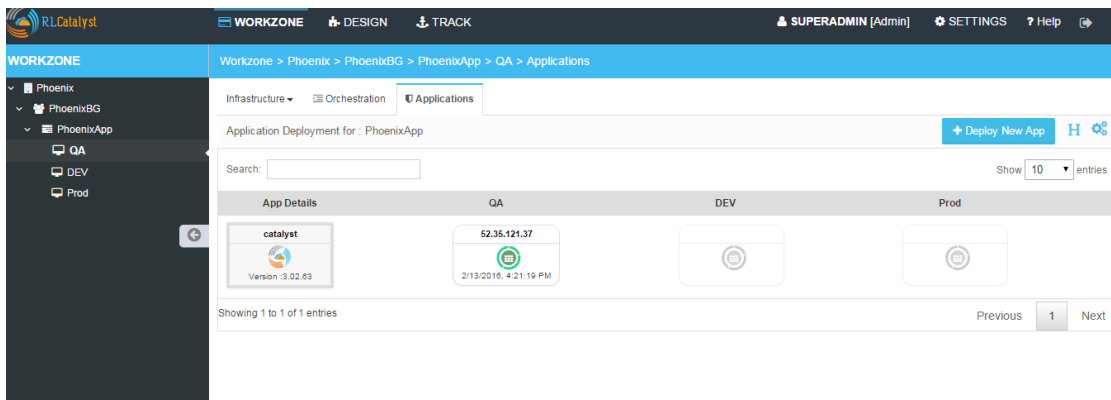
6.13 Launching Blueprint

- Go to Workzone
- Click Infrastructure dropdown and Select Blueprints tab
- Expand Software Stack

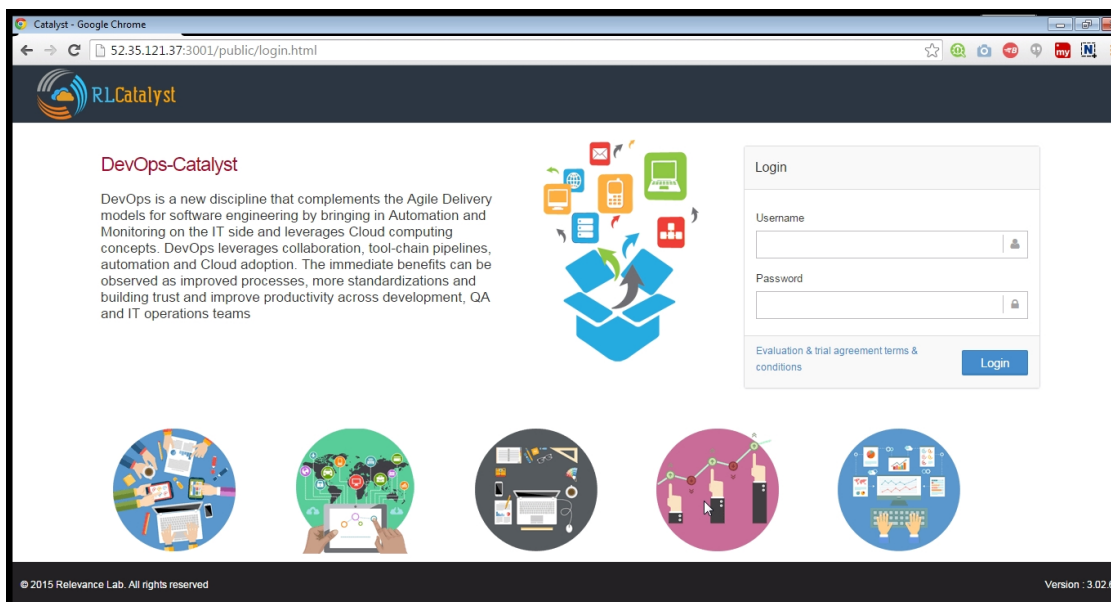
- Select the Instance and Click on Launch button
- Go to Instances tab and you can see node will be launched and wait until bootstrap is successful



- Go to Applications tab
- You will see the Application details with Name, Version, IP Address of the node and Time



- Now copy the IP address where application is deployed and open new tab and paste IP address with port number. [Eg: 52.35.121.37:3001]

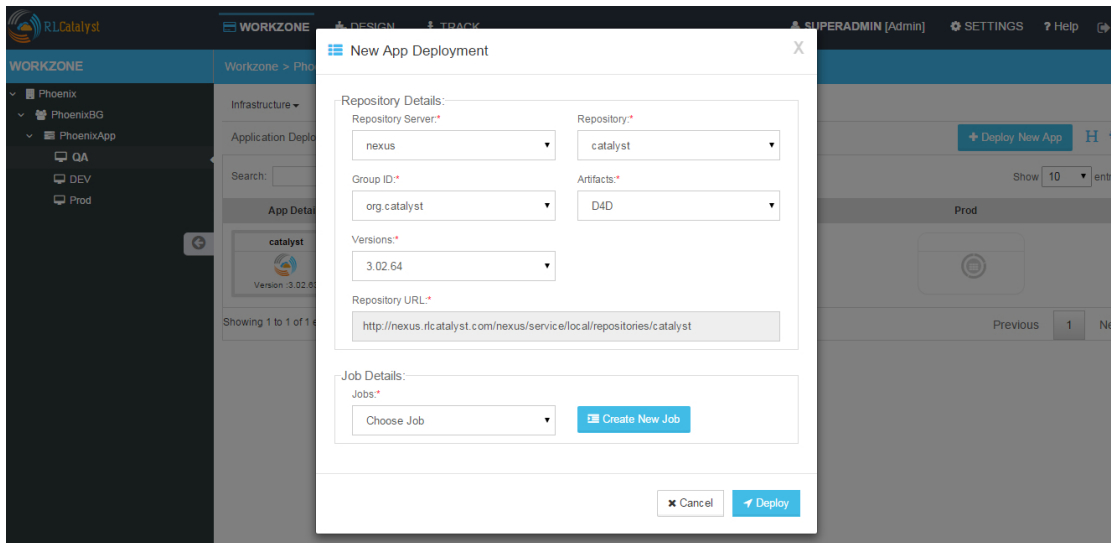


- Now Catalyst application is installed with the version 3.02.63 on the launched node. [See the version at bottom right corner of the window]

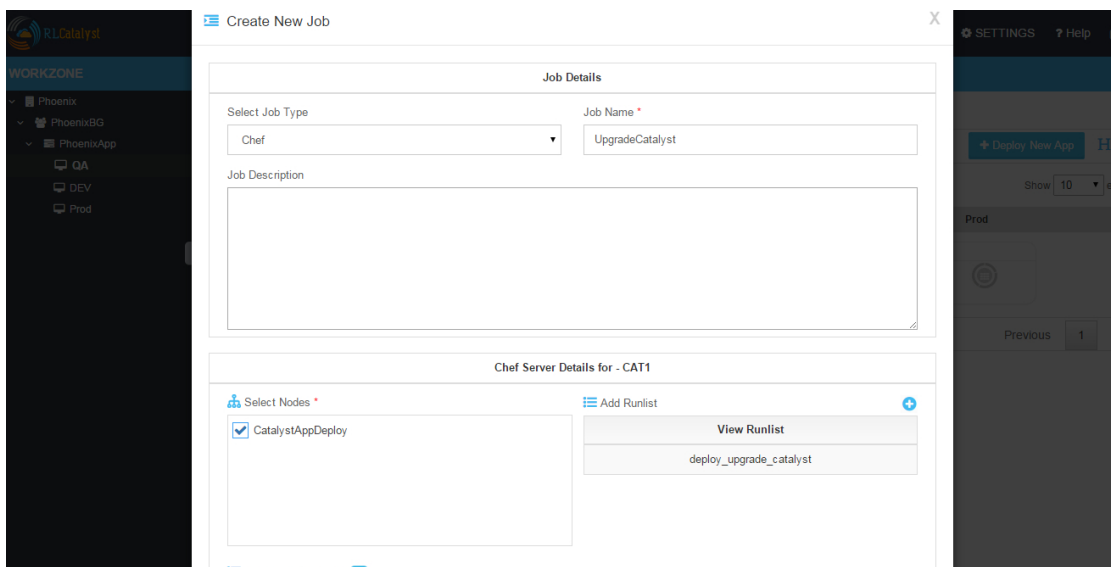
6.14 Deploy New App

Now I will show you how to upgrade latest version of catalyst application on the same node.

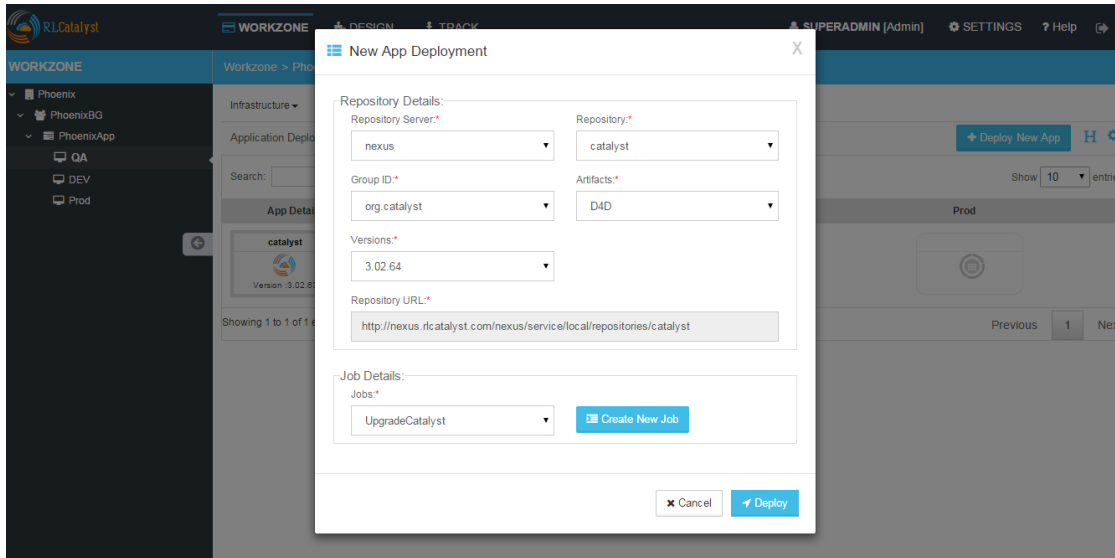
- Go to Applications tab
- Click on Deploy New App button
- Enter the Repository details by selecting latest version [Here latest is 3.02.64]



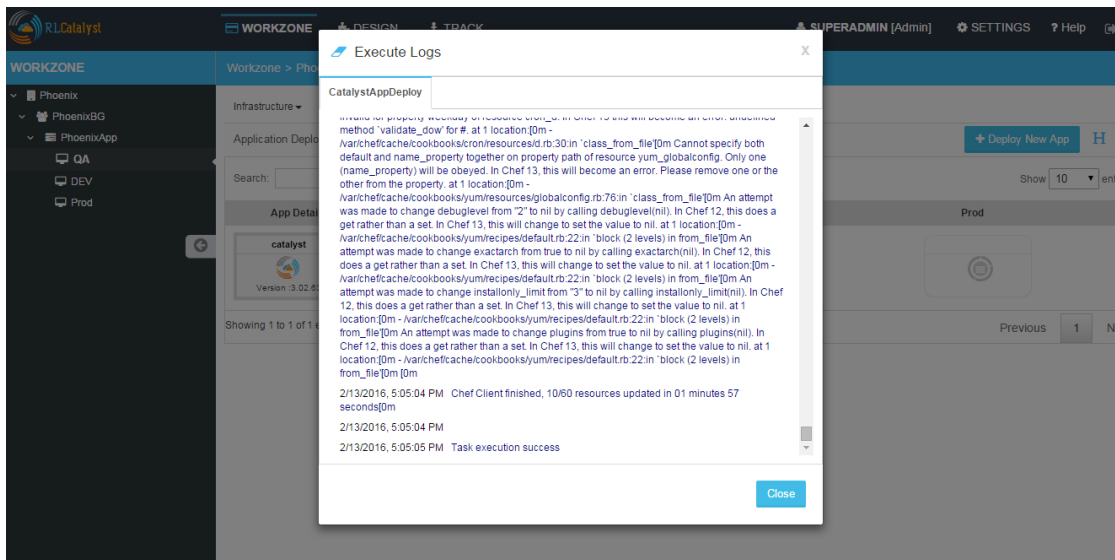
- Click on Create New Job button
- Enter the Job name
- Select the Node on which you are going to upgrade latest version
- Add the cookbook **deploy_upgrade_catalyst** to the runlist



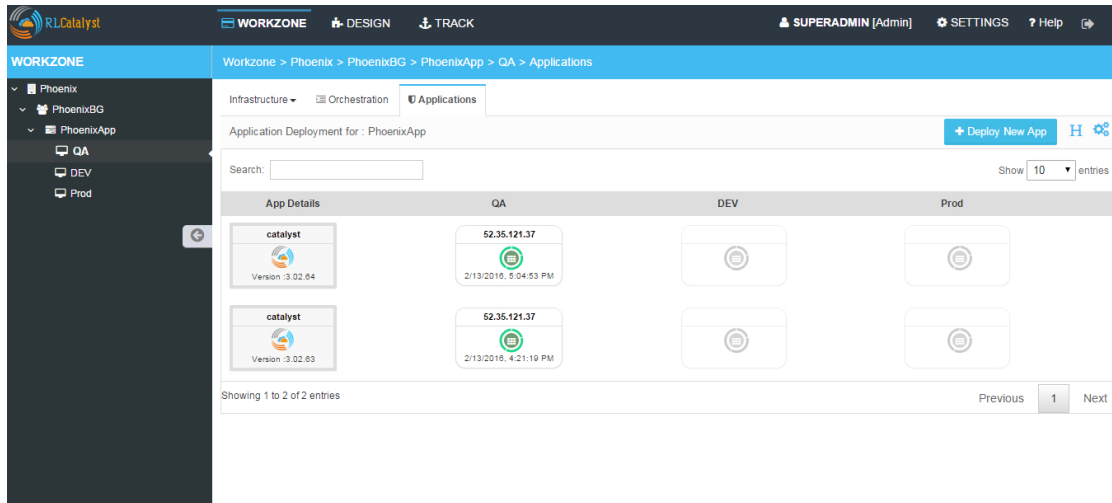
- Click on Save button
- Click OK button on Task Success popup window
- Click on Jobs dropdown
- Select the Job which is created in previous step



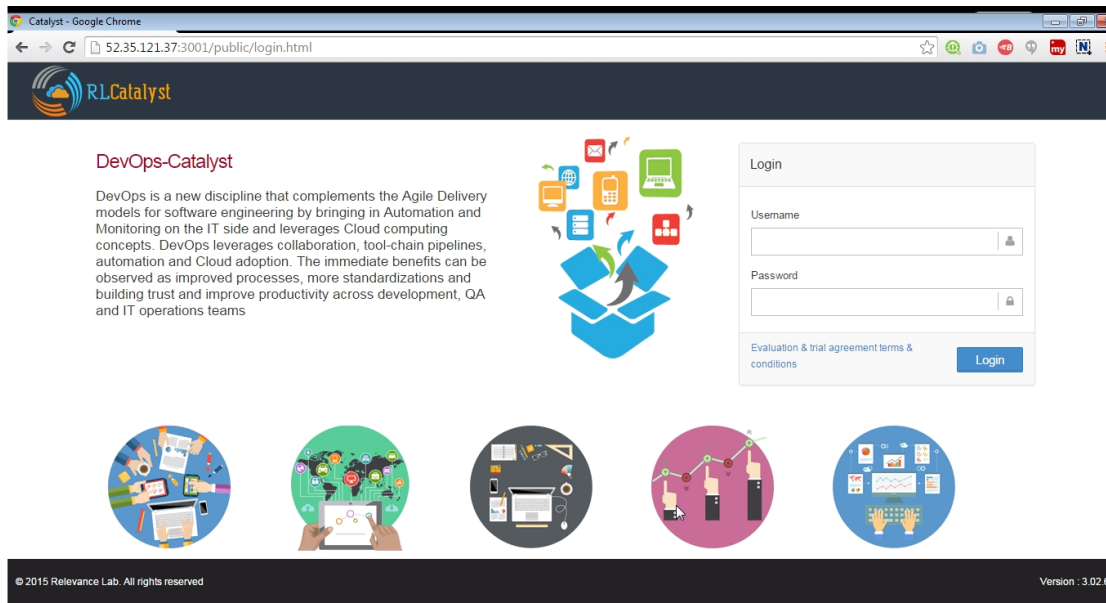
- Click on Deploy button
- Click OK button on Confirmation popup window
- Execute Logs window will open and wait until Task execution is successful



- Close Execute Logs window
- Now you can see Application card is displayed with Application details with Name, Version [3.02.64], IP address of the node and Time



- Now copy the Ip Address where application is deployed and open new tab and paste Ip Address with port number. [Eg: 52.35.121.37:3001] and verify the latest version [3.02.64] of the application is deployed on the node in right bottom corner of the window



CASE II: User has the application(s) running over several existing environments and all the application details must be imported to Catalyst

Pre-requisite: 1) The http/https request ports need to be open from the server for catalyst to get the information 2) Target instance(Application instance) must be part of Catalyst env that means that particular machine has been imported to the particular environment 3) Jenkins job must be associated for that particular ip/instance, the job parameters must be application name, nodeip, env, version, apptype, containerid, containerport etc.

Please specify the following piece of code in the jenkins Job for deployment to see the Application cards in Catalyst.

```
# Code for App Deployment history information via jenkins
exitStatus=$? export APPSTATUS if [ $exitStatus -eq 0 ] then
    echo "Successful" APPSTATUS="Successful"
else echo "Failure" APPSTATUS="Failure"
```

```
fi echo $APPSTATUS export APPVERSION="###<specify the major version>" echo $APPVERSION
export LASTDEPLOY="$(date +%y-%m-%d %r)" echo $LASTDEPLOY export IP="$(hostname -
I)" export THISHOST="$(hostname)" export APPINSNAME="Application name" export applicationN-
odeIP="XXX.XXX.XXX.XX" e.g. 192.168.105.22
```

#Send the information for Catalyst Application tab

```
curl -X POST -H "Content-Type: application/json" -d '{"appDeployData": {"applicationNodeIP" :
"192.168.105.22","applicationName": " Application name ","applicationInstanceName": ""$APPIN-
SNAME""","applicationVersion": ""$APPVERSION""","applicationNodeIP": ""$IP""","applicationLastDeploy":
""$LASTDEPLOY""","applicationStatus": ""$APPSTATUS""","applicationType": "Package","containerId":
""","hostName": ""$THISHOST""","appLogs": ""http://jenkinsip:port/job/\protect\T1\textdollarJOB_NAME/
\protect\T1\textdollarBUILD_NUMBER/console""","envId": "QA"}}' http://catalyst:ipaddress:port/app/deploy
```

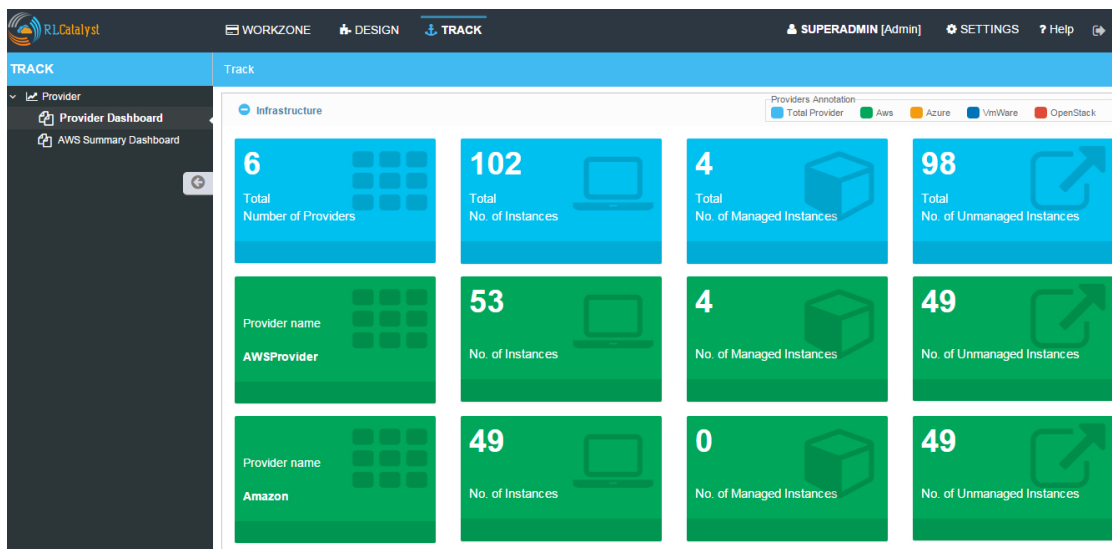
Following video demonstrates how to do AppDeploy in RLCatalyst:

CHAPTER 7

Track

- If you have not configured your Track setup, follow the instructions at *Track Setup*.

Click on **Track** link in the header. By default first link i.e, Provider Dashboard will be selected. Here you will be able to see total number of provider you are added in the catalyst with **Total number of instances**, **Total number of managed instances**, and **Total number of unmanaged instances**.

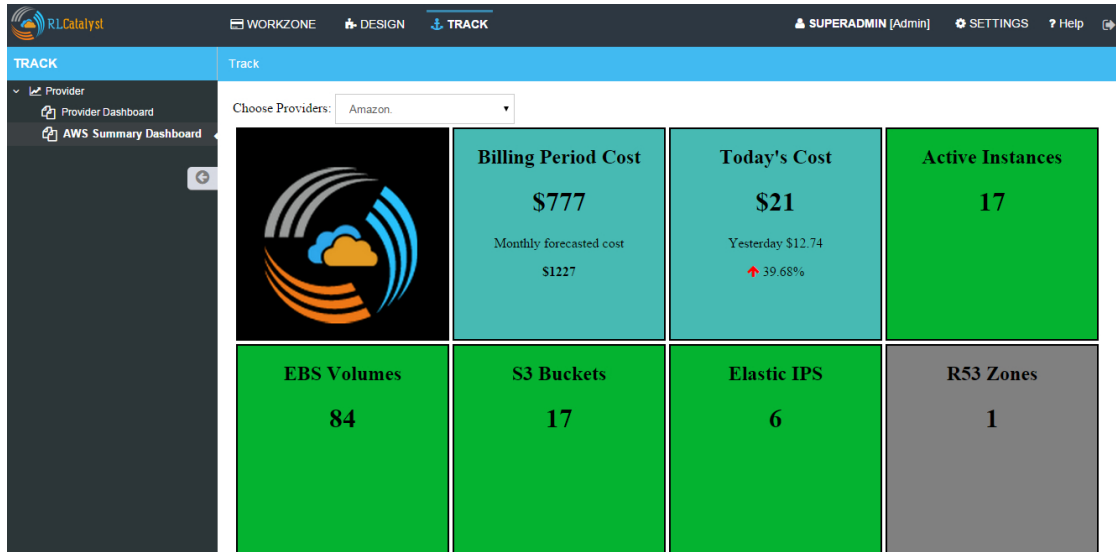


Click on AWS Summary Dashboard link in Tree Structure

Here you will be able to see

- Billing Period Cost
- Monthly Cost
- Todays cost
- Yesterday Cost

- Active Instances
- EBS Volume
- S3 Buckets
- Elastic IPS
- R53 Zones



Cloud is the cost, usage and capacity of cloud. It unlocks more insightful questions and produces more accurate outcomes by mapping relationships among high volumes of connected data

Prerequisites: A cloud account should be added in RLCatalyst settings.

How it is working:

- Aggregate and graphically represent statistical AWS resource cost data at different granularities across time intervals to assist end user to easily analyse and optimize resource usage.
- Collect and graphically represent usage trends data of provisioned AWS resources across time intervals to assist end user in identifying resource utilizations patterns and optimize deployment strategies.
- Provide APIs to access usage and cost analytics data at different granularities across time intervals to enable automation of usage analysis to gain insights and to develop prediction algorithms based on these insights to optimize resource utilization and reduce cost.

How to use:

Navigate to **Cloud** tab in the header. There is Menu at left handside and under the analytics we can see below options:

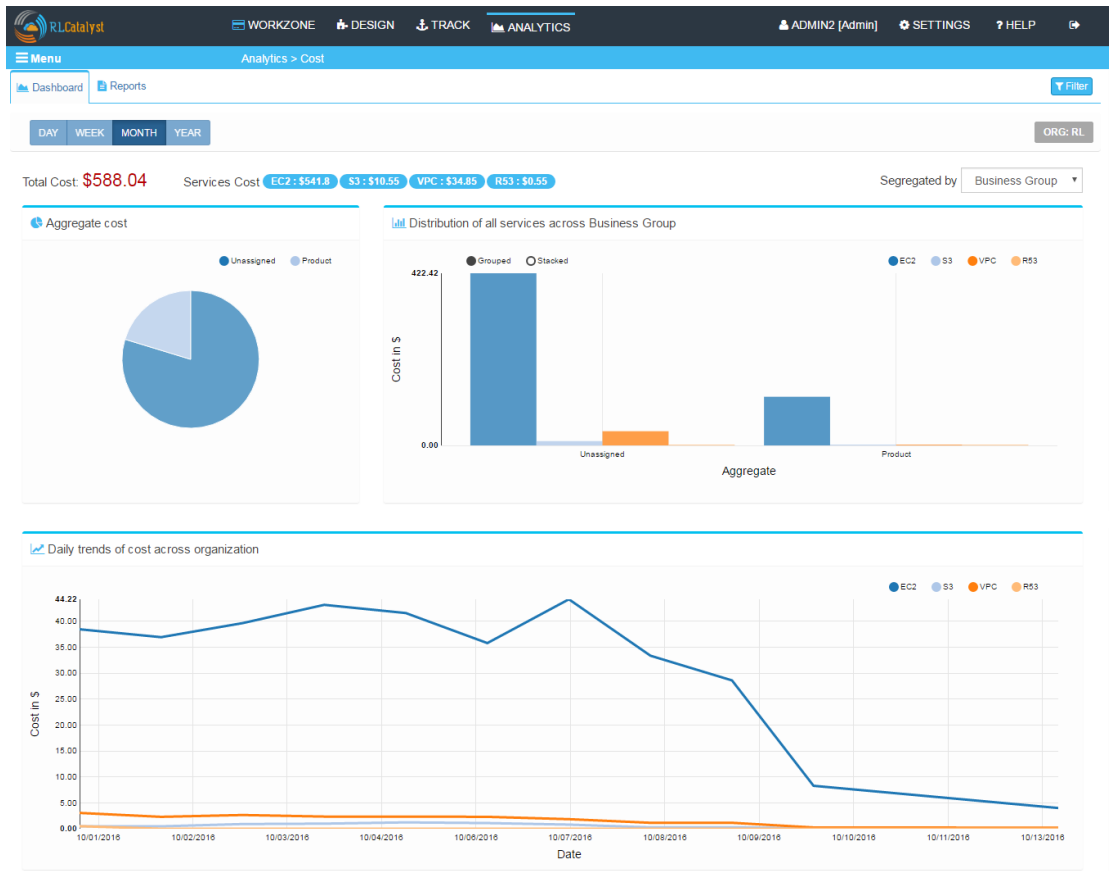
1. Capacity
2. Cost
3. Usage

8.1 Cost

Cost is **Cloud Cost on data**.

Page has two tabs Dashboard and Reports.

1. Dashboard: By default it will show Dashboard details in the form of graph. Dashboards let you monitor many data at once, so you can quickly check the health of your accounts or see correlations between different reports.



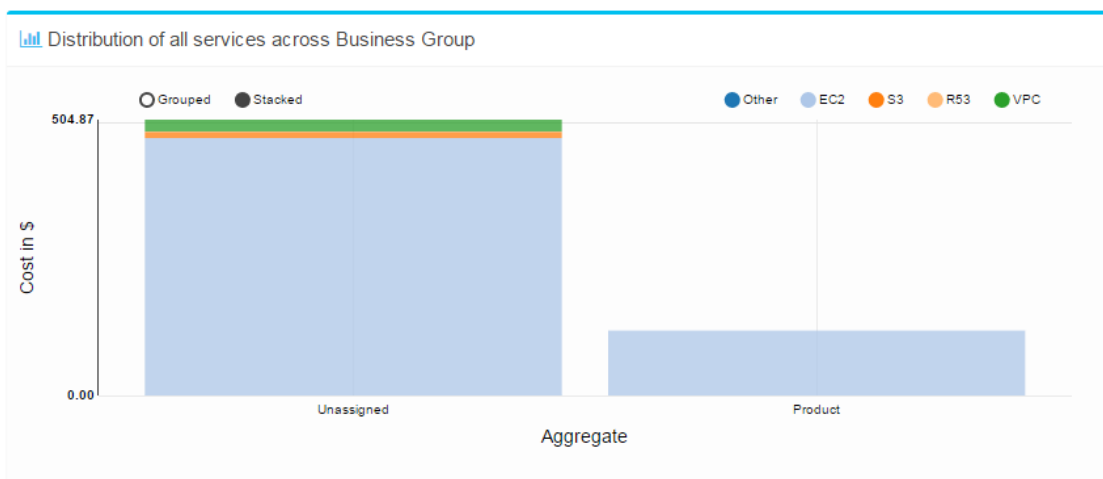
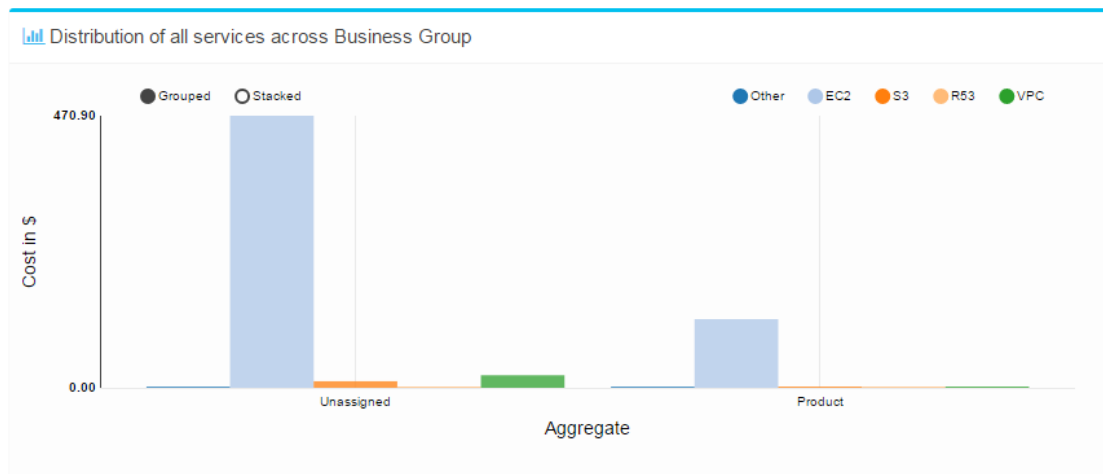
AWS cost analytics:

- Collect hourly costs for all AWS resources for all added providers.
- Aggregate hourly, daily, monthly and yearly costs for catalyst entities and provider specific entities based on resource allocation.
 - Catalyst entities: organization, business unit, project, environment, provider type, provider, resource
 - Provider specific entities: region, resource types (EC2, S3, RDS etc)
- Provide API to query for aggregated data based on catalyst and provider specific entities across time periods. Start date for aggregation shall be decided based on specified period. End date shall be specified by the consumer of the API
 - Periods: hour, month, day, year, 5 years, 10 years
- Provide APIs to query for resource cost trend for specific catalyst and provider specific entities across time periods, for specified intervals. Intervals shall be limited based on the specified period, limited by the cap of number of data points in the response.
 - Intervals: hourly, daily, weekly, monthly, yearly
- Time series data will be stored with UTC timestamps

We have total cost, and aggregated cost for different services EC2, S3, R53, other etc for the month segregated by Business Group, Region, Environment and Provider. To show the graph we have to do the mapping in provider sync, mapping tag values with Business Group, Project and Environment respectively. Graphs are in three forms:

1. Aggregate Cost: It is a Pie Chart, which display all aggregated total cost details.

2. Distribution of all services across Business Group: Here we have Bar chart, it is drill down of pie chart - grouped and stacked for services, in grouped the services would be available in single bar chart and in stacked the services would be in multiple bar charts.



3. Daily trends of cost across organization: It is a Line chart, the data would be on daily basis accross organization.

2. Reports: In Reports tab the data will be in tabular form Name, Total Cost, Services.

RL Catalyst WORKZONE DESIGN TRACK ANALYTICS SUPERADMIN [Admin] SETTINGS HELP

Menu Analytics > Cost

Dashboard Reports Filter

DAY WEEK MONTH YEAR ORIG: RL

Total Cost: **\$623.69** Services Cost: Other: \$0.3 EC2: \$589.72 S3: \$11.36 R53: \$0.56 VPC: \$21.75 Segregated by: Environment

Name	Total Cost	Other	E C2	S3	R53	Vpc
Unassigned	554.9	0.3	520.93	11.36	0.56	21.75
RL_EVL	68.79		68.79			

We have few options available for the table, on the right side of the table

RL Catalyst WORKZONE DESIGN TRACK ANALYTICS SUPERADMIN [Admin] SETTINGS HELP

Menu Analytics > Cost

Dashboard Reports Filter

DAY WEEK MONTH YEAR ORIG: RL

Total Cost: **\$623.69** Services Cost: Other: \$0.3 EC2: \$589.72 S3: \$11.36 R53: \$0.56 VPC: \$21.75 Segregated by: Environment

Name	Total Cost	Other	E C2	S3	R53	Vpc
Unassigned	554.9	0.3	520.93	11.36	0.56	21.75
RL_EVL	68.79		68.79			

Export all data as csv
Export visible data as csv
Columns:
✓ Name
✓ Total Cost
✓ Other

We can remove any column from the table by just clicking on the field name, let say remove “Total Cost”

RL Catalyst WORKZONE DESIGN TRACK ANALYTICS SUPERADMIN [Admin] SETTINGS HELP

Menu Analytics > Cost

Dashboard Reports Filter

DAY WEEK MONTH YEAR ORIG: RL

Total Cost: **\$623.69** Services Cost: Other: \$0.3 EC2: \$589.72 S3: \$11.36 R53: \$0.56 VPC: \$21.75 Segregated by: Environment

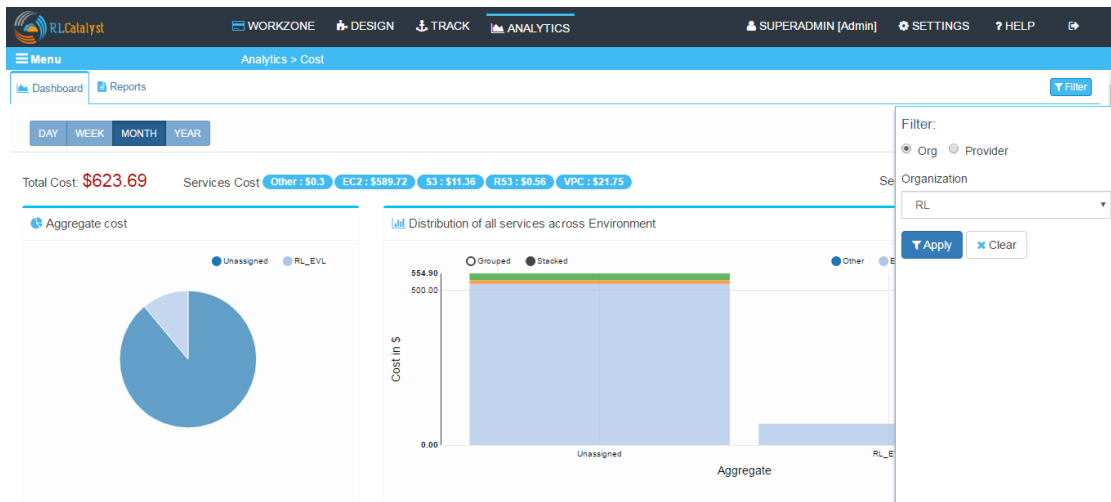
Name	Other	E C2	S3	R53	Vpc
Unassigned	0.3	520.93	11.36	0.56	
RL_EVL		68.79			

Export all data as csv
Export visible data as csv
Columns:
✓ Name
✗ Total Cost
✓ Other

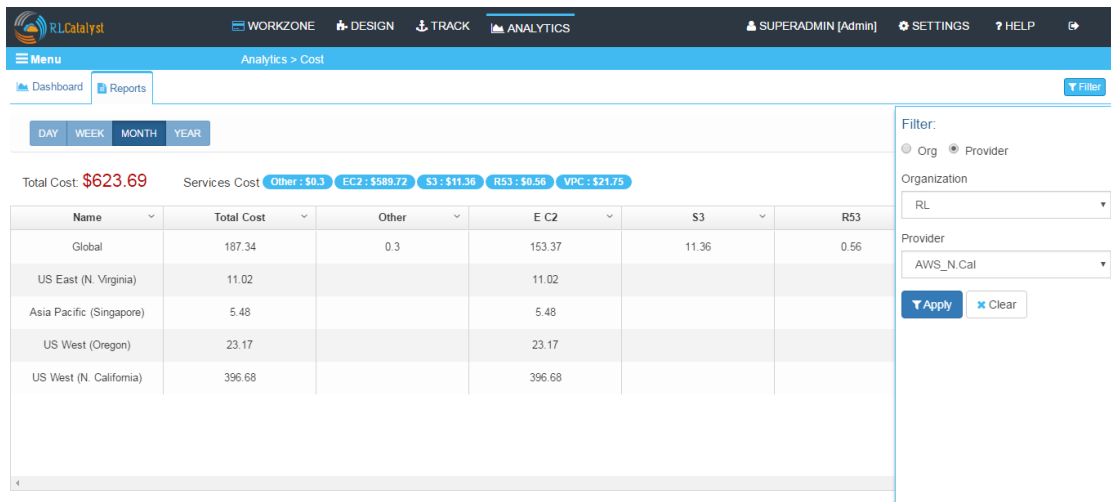
Now, “Total Cost” field is not available in the table.

We have two more options “Export all data as cvs”, it will export all data from the table in cvs format. And “Export visible data as cvs”, it will save only the visible data from the table, it will not save the removed data from the table.

- Filter: We can filter the data on the basis of Organization and Provider.

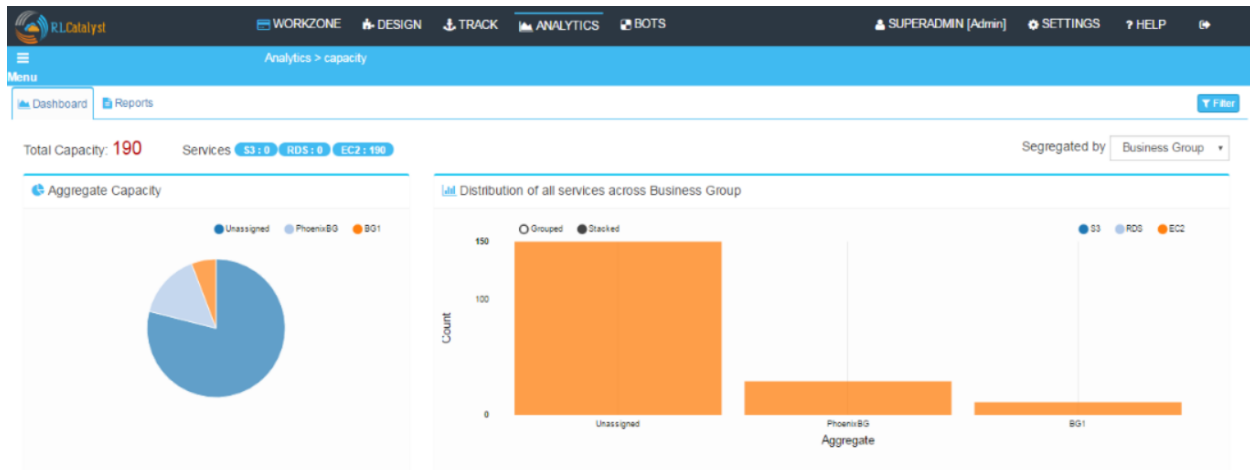


If we want to filter the data on the basis of provider, we can select any provider from the drop down



8.2 Capacity

Capacity is **number of Instances**



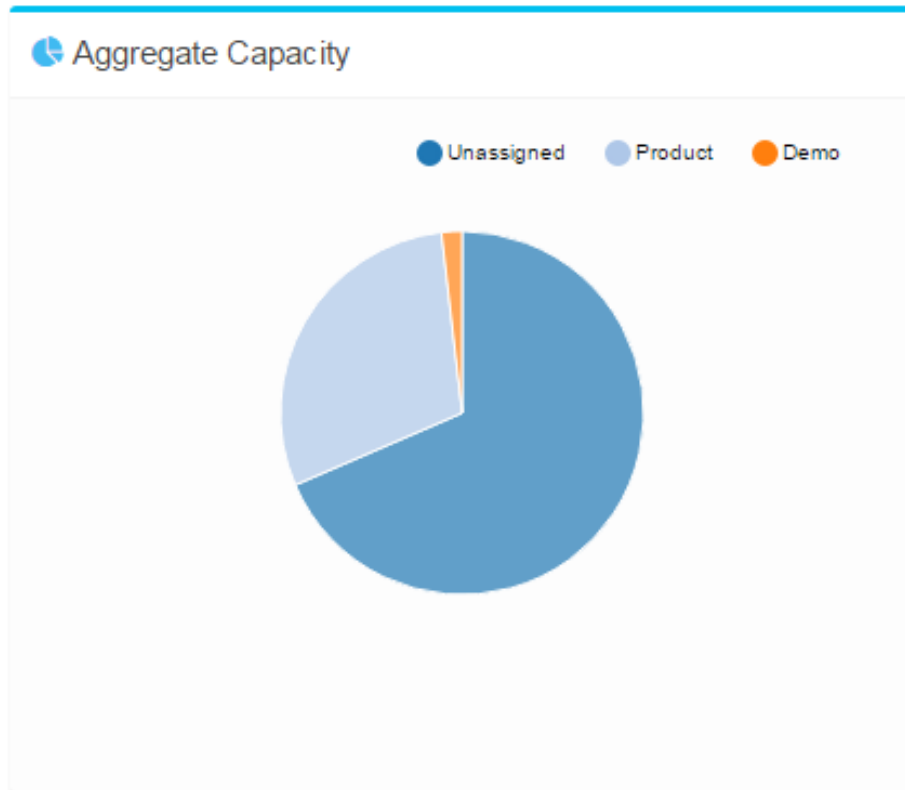
Page has two tabs Dashboard and Reports.

1. Dashboard: Dashboard shows you the capacity of the Instances.

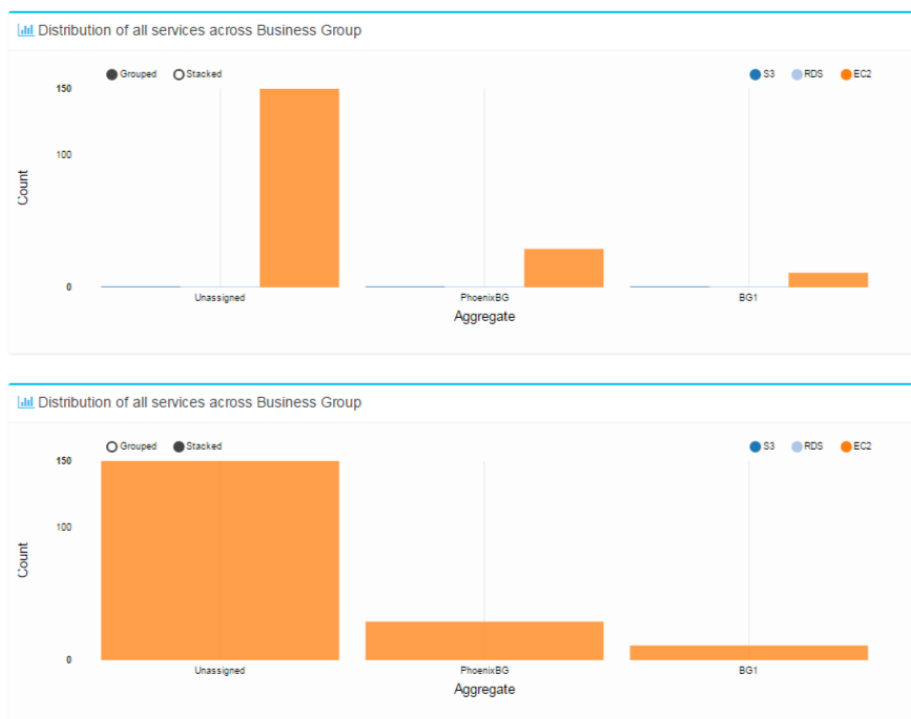
- Total Capacity, it shows you the available number of instances
- Services, it shows the capacity for services like s3, RDS etc.
- Segregated by, the capacity would be Segregated by Business Group, Provider, Environment and Region

We have two graphs

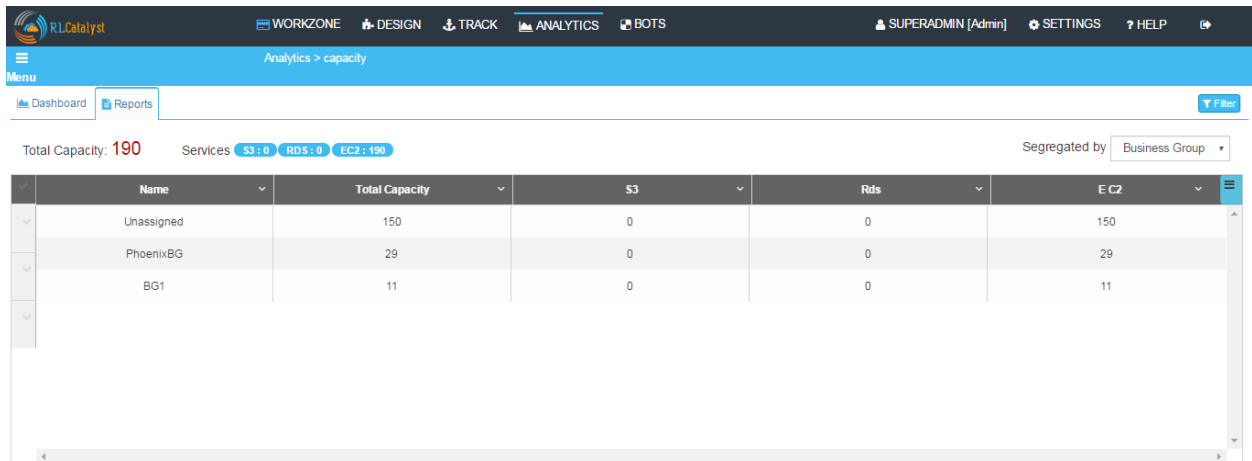
1. Aggregate Capacity: It is a Pie Chart, which display all aggregated total capacity details.



2. Distribution of all services across Business Group: Here we have Bar chart, it is drill down of pie chart



2. Reports: In Reports tab the data will be in tabular form Name, Total Capacity, Services.



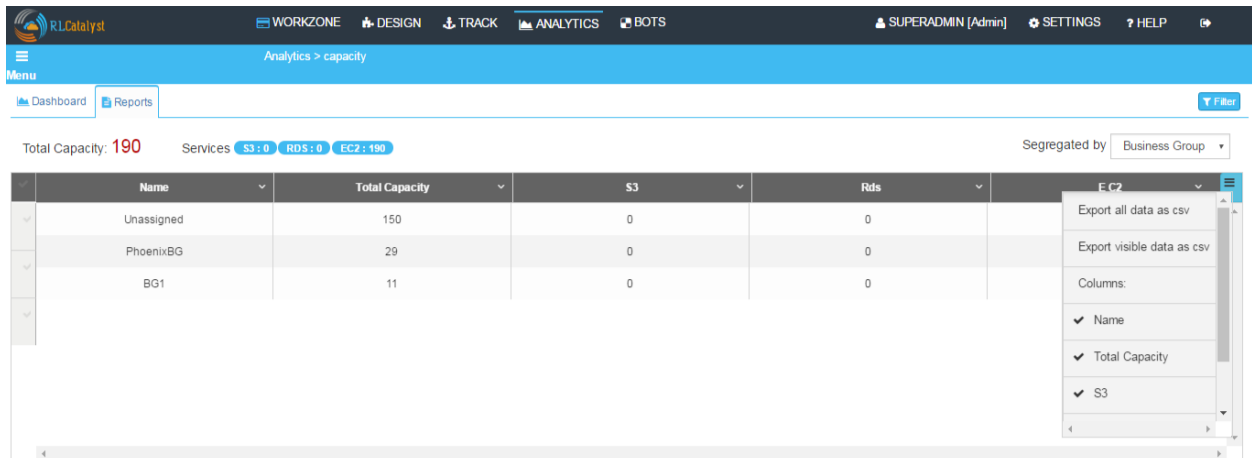
Analytics > capacity

Menu Dashboard Reports Filter

Total Capacity: 190 Services S3: 0 RDS: 0 EC2: 190 Segregated by Business Group

Name	Total Capacity	S3	Rds	EC2
Unassigned	150	0	0	150
PhoenixBG	29	0	0	29
BG1	11	0	0	11

We have few options available for the table, on the right side of the table



Analytics > capacity

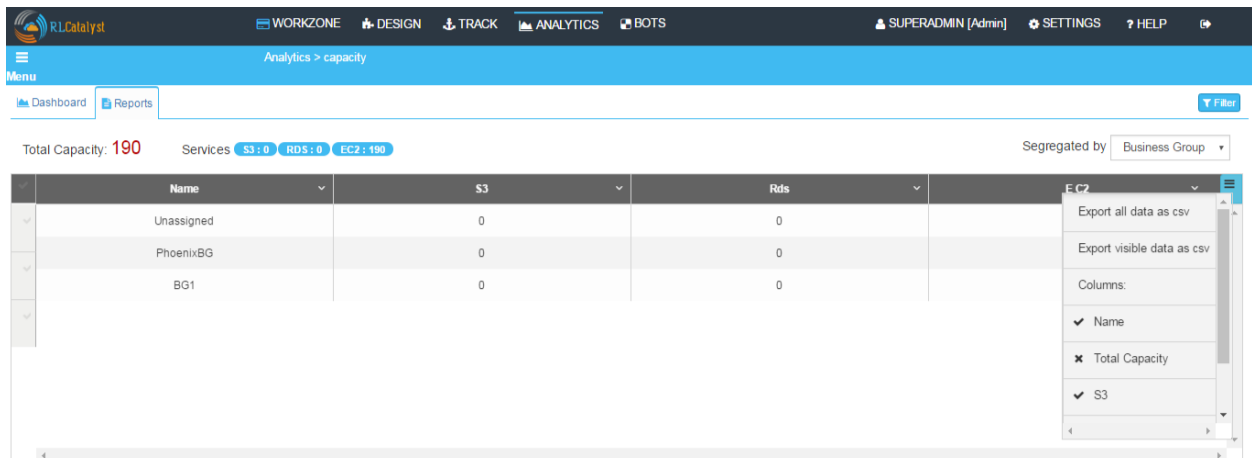
Menu Dashboard Reports Filter

Total Capacity: 190 Services S3: 0 RDS: 0 EC2: 190 Segregated by Business Group

Name	Total Capacity	S3	Rds	EC2
Unassigned	150	0	0	
PhoenixBG	29	0	0	
BG1	11	0	0	

Export all data as csv
Export visible data as csv
Columns:
✓ Name
✓ Total Capacity
✓ S3

We can remove any column from the table by just clicking on the field name, let say remove “Total Capacity”



Analytics > capacity

Menu Dashboard Reports Filter

Total Capacity: 190 Services S3: 0 RDS: 0 EC2: 190 Segregated by Business Group

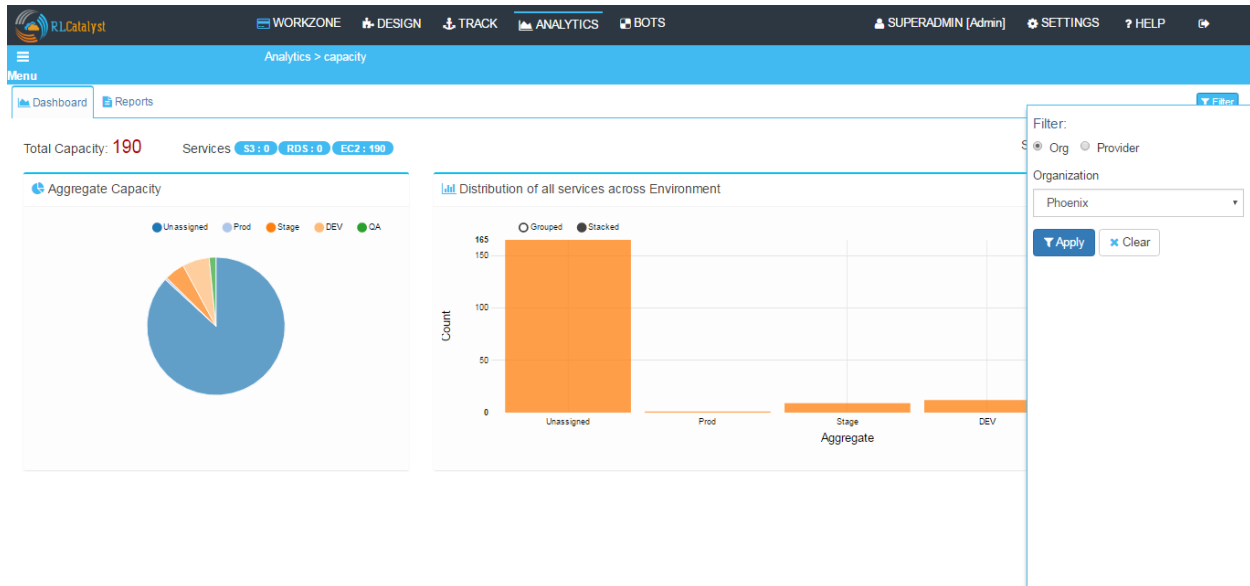
Name	S3	Rds	EC2
Unassigned	0	0	
PhoenixBG	0	0	
BG1	0	0	

Export all data as csv
Export visible data as csv
Columns:
✓ Name
✗ Total Capacity
✓ S3

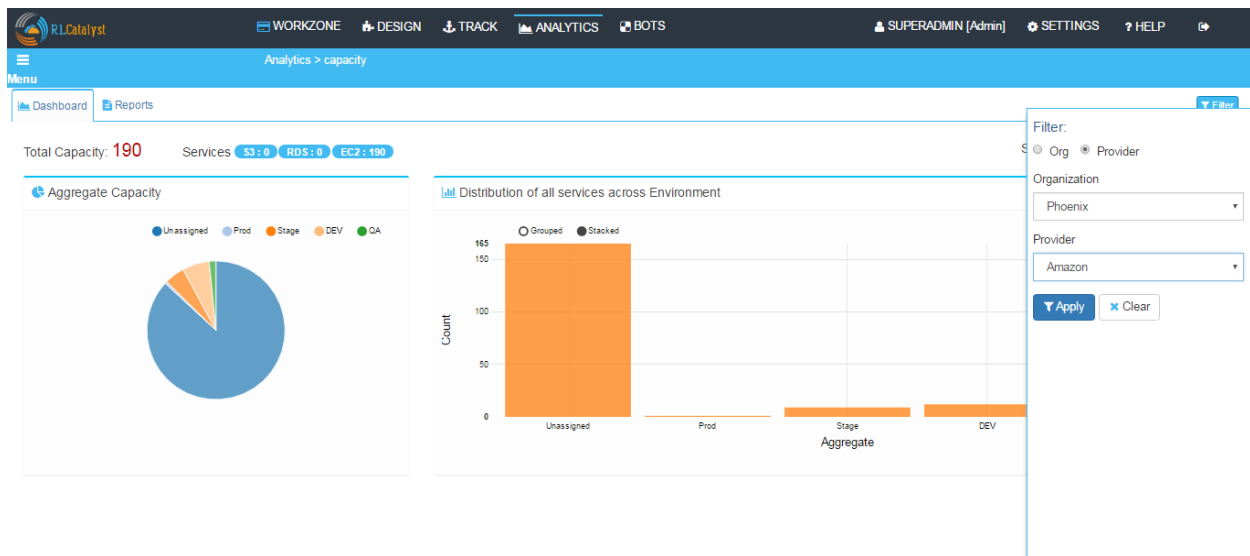
Now, “Total Capacity” field is not available in the table.

We have two more options “Export all data as cvs”, it will export all data from the table in cvs format. And “Export visible data as cvs”, it will save only the visible data from the table, it will not save the removed data from the table.

- Filter: We can filter the data on the basis of Organization and Provider.



If we want to filter the data on the basis of provider, we can select any provider from the drop down

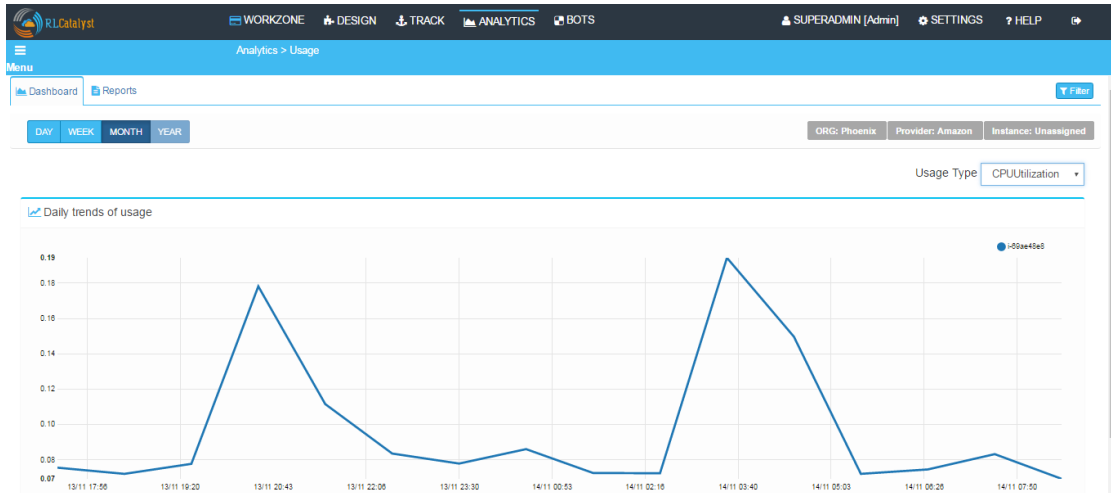


8.3 Usage

Usage is **machine level usage matrix**.

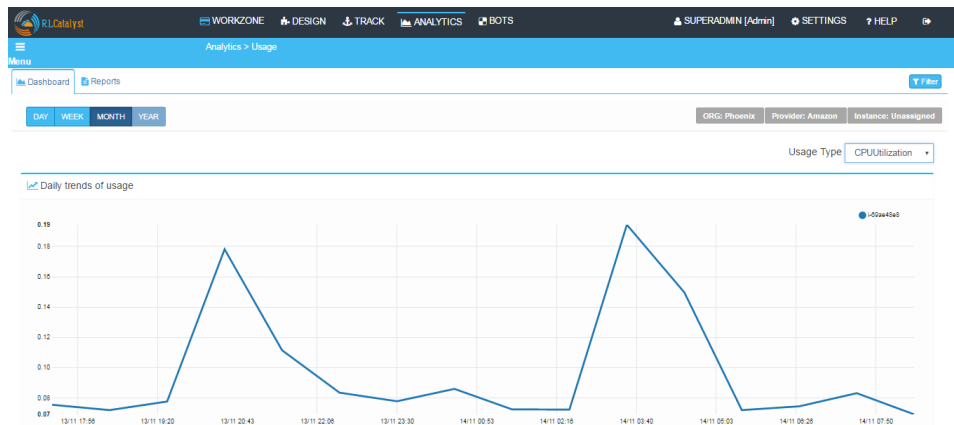
Page has two tabs Dashboard and Reports.

1. Dashboard:



We have 5 Usage Type:

- **CPUUtilization:** The percentage of allocated EC2 compute units that are currently in use on the instance. This metric identifies the processing power required to run an application upon a selected instance.



- **DiskReadBytes:** This metric is used to determine the volume of the data the application reads from the hard disk of the instance. This can be used to determine the speed of the application.
- **DiskWriteBytes:** This metric is used to determine the volume of the data the application writes onto the hard disk of the instance. This can be used to determine the speed of the application.
- **NetworkIn:** The number of bytes received on all network interfaces by the instance. This metric identifies the volume of incoming network traffic to an application on a single instance.
- **NetworkOut:** The number of bytes sent out on all network interfaces by the instance. This metric identifies the volume of outgoing network traffic to an application on a single instance.

2. Reports: In Reports tab the data will be in tabular form

The screenshot shows the 'Analytics > Usage' page in the RL Catalyst interface. The table displays CPU utilization data for various instances. The columns are Name, From Time, To Time, Maximum, Minimum, and Average. The data is filtered by Usage Type: CPUUtilization.

Name	From Time	To Time	Maximum	Minimum	Average
i-69ae48e8	2016-11-14T16:00:00.000Z	2016-11-14T17:00:00.000Z	0.17	0	0.06366666666666666
i-69ae48e8	2016-11-14T17:00:00.000Z	2016-11-14T18:00:00.000Z	0.17	0	0.0775
i-69ae48e8	2016-11-14T18:00:00.000Z	2016-11-14T19:00:00.000Z	0.17	0	0.07283333333333333
i-69ae48e8	2016-11-14T19:00:00.000Z	2016-11-14T20:00:00.000Z	0.34	0	0.07766666666666666
i-69ae48e8	2016-11-14T20:00:00.000Z	2016-11-14T21:00:00.000Z	0.33	0	0.08833333333333333
i-69ae48e8	2016-11-14T21:00:00.000Z	2016-11-14T22:00:00.000Z	0.17	0	0.06666666666666667
i-69ae48e8	2016-11-14T22:00:00.000Z	2016-11-14T23:00:00.000Z	0.17	0	0.07266666666666666

We have few options available for the table, on the right side of the table

The screenshot shows the 'Analytics > Usage' page with a context menu open on the right side of the table. The menu options are: Export all data as csv, Export visible data as csv, and Columns. The Columns section shows a list of columns with checkboxes: Name (checked), From Time (checked), and To Time (checked).

Name	From Time	To Time	Maximum	Minimum	Average
i-69ae48e8	2016-11-14T16:00:00.000Z	2016-11-14T17:00:00.000Z	0.17	0	0.06366666666666666
i-69ae48e8	2016-11-14T17:00:00.000Z	2016-11-14T18:00:00.000Z	0.17	0	0.0775
i-69ae48e8	2016-11-14T18:00:00.000Z	2016-11-14T19:00:00.000Z	0.17	0	0.07283333333333333
i-69ae48e8	2016-11-14T19:00:00.000Z	2016-11-14T20:00:00.000Z	0.34	0	0.07766666666666666
i-69ae48e8	2016-11-14T20:00:00.000Z	2016-11-14T21:00:00.000Z	0.33	0	0.08833333333333333
i-69ae48e8	2016-11-14T21:00:00.000Z	2016-11-14T22:00:00.000Z	0.17	0	0.06666666666666667
i-69ae48e8	2016-11-14T22:00:00.000Z	2016-11-14T23:00:00.000Z	0.17	0	0.07266666666666666

We can remove any column from the table by just clicking on the field name, let say remove “From Time”

The screenshot shows the 'Analytics > Usage' page with the 'From Time' column removed from the table. The context menu is still open, showing the 'Columns' section with 'Name' (checked), 'From Time' (unchecked), and 'To Time' (checked).

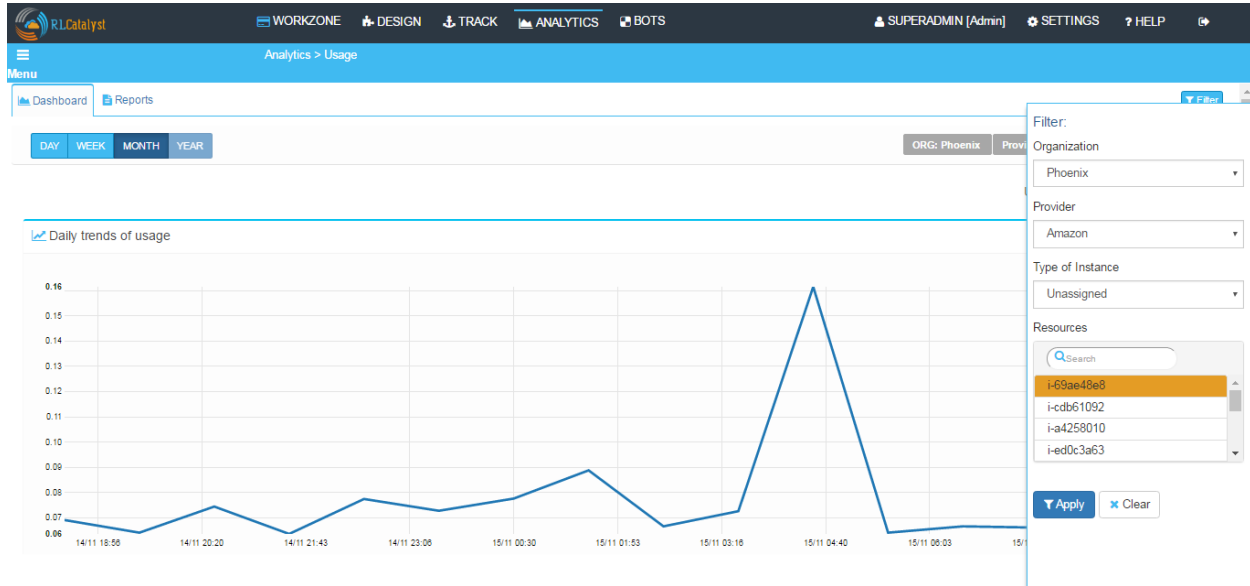
Name	To Time	Maximum	Minimum	Average
i-69ae48e8	2016-11-14T17:00:00.000Z	0.17	0	0.0
i-69ae48e8	2016-11-14T18:00:00.000Z	0.17	0	0.0
i-69ae48e8	2016-11-14T19:00:00.000Z	0.17	0	0.0
i-69ae48e8	2016-11-14T20:00:00.000Z	0.34	0	0.0
i-69ae48e8	2016-11-14T21:00:00.000Z	0.33	0	0.0
i-69ae48e8	2016-11-14T22:00:00.000Z	0.17	0	0.0
i-69ae48e8	2016-11-14T23:00:00.000Z	0.17	0	0.0

Now, “From Time” field is not available in the table.

We have two more options “Export all data as cvs”, it will export all data from the table in cvs format. And “Export visible data as cvs”, it will save only the visible data from the table, it will not save the removed data from the table.

- filter: We can filter the data on
 - Organization: We can select any organization from the drop down

- Provider: Select any provider from available list of providers
- Type of Instance: Types of instances are Managed, Assigned and Unassigned. We can select any type of instance from the drop down
- Resources: Here we can search for some node and Apply for the filter



BOTs are readymade automated scripts.

If you have not configured your BOTs setup, ask your admin to setup BOTs.

- BOTs
 - BOT Library
 - Audit Trail
 - BOT Report
- Runbooks

We have 3 types of BOTs:

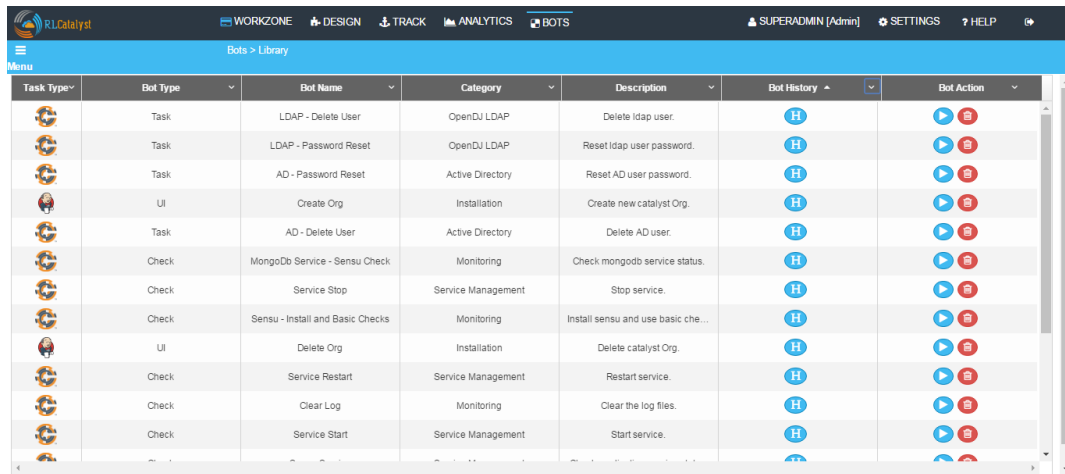
- Check: These type of BOTs used to provide service management(start, stop, restart), monitoring services and chef provider is used to deliver these services
- Run: These type of BOTs used to launch any blueprint, upgrade any application and chef provider is used to deliver these services
- UI: These type of BOTs are used to provide different kind of services using UI automation(selenium) and jenkins is used to deliver these services

These BOTs are used to provide following services:

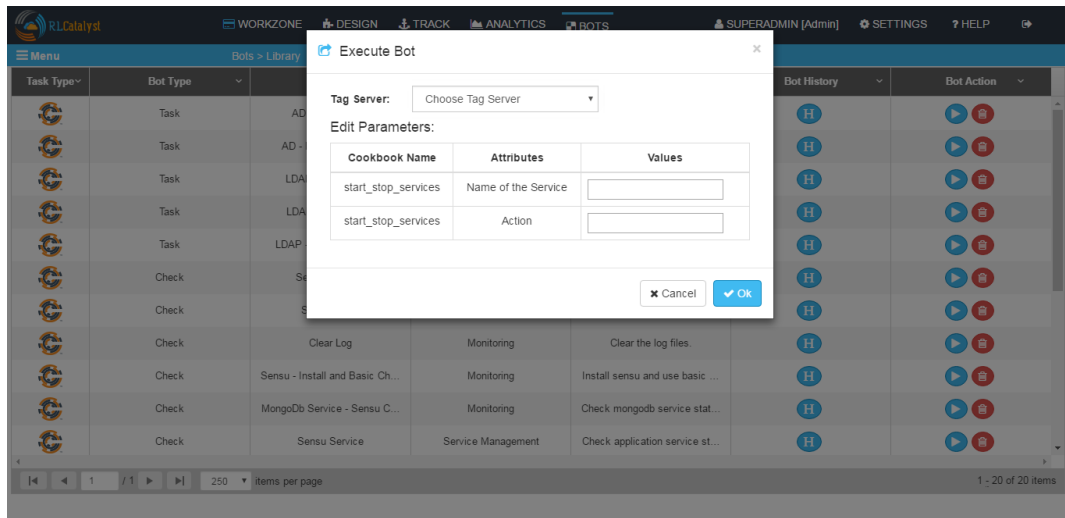
- Active Directory: We can create user, delete user and reset password using Run type BOTs
- OpenDJ LDAP: We can create user, delete user and reset password using Run type BOTs
- Monitoring: We can use sensu to monitor any service using check BOTs
- Application Deployment: We can deploy new application on any machine using Run BOTs
- Service Management: We can start, stop, restart service using check BOTs
- Upgrade: We can use Run BOTs to upgrade any service running on any machine
- Installation: We can install new services on any machine using Run BOTs

How to execute BOTs

- User can select any BOTs for execution



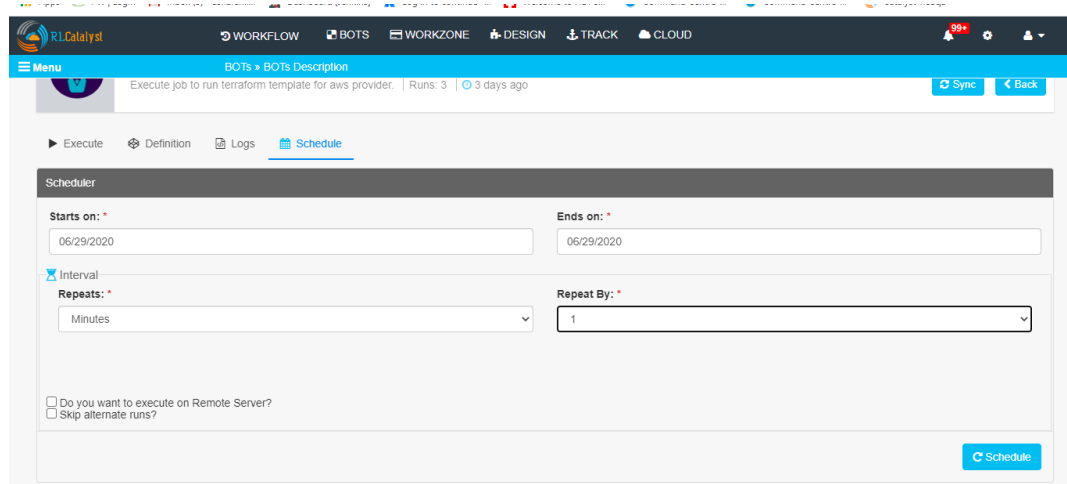
- click on execute button one popup will appear



- Select any Tag Server from the drop down
- Enter the parameters
- Now click on ok button
- Execution will start. We can see logs through logs tab.
- We can also schedule the BOTs.

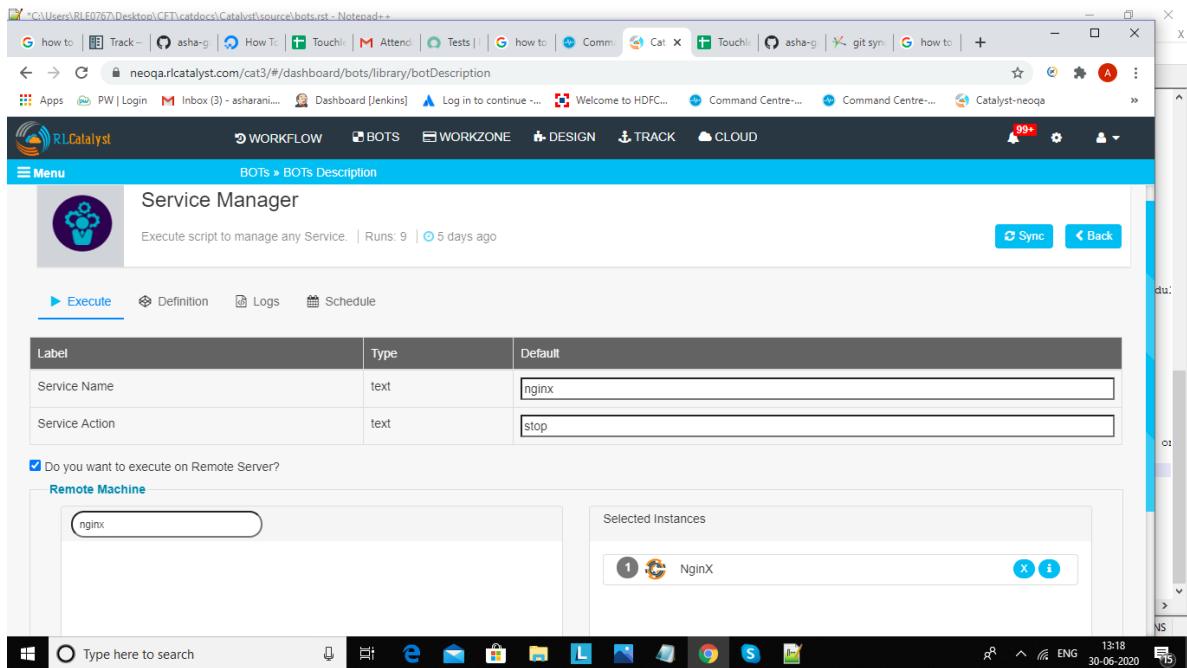
How to schedule the BOTs

- Select any BOT in BOTs library then navigate to schedule tab in the BOT.
- Enter the details like start date and end date and choose minutes/hourly/daily/weekly/monthly and how many times we want execute the BOT.
- If you want to execute the bot with remote server enable the checkbox .If you want skip alternate runs in the execution enable the checkbox. Click on “schedule” button.



Remote Execution of BOTs

- Select Service Manager BOT in BOTs library
- Enter the input parameters : 1. Service Name(nginx) 2. Service Action (stop/start)
- select the checkbox : Do you want to execute on Remote Server? - and select the remote instance that you want to stop the service nginx -192.1.1.2 and click on execute



CHAPTER 10

Installation

This is a documentation that gives the developer and the user an overview of how to install RLCatalyst on a local instance

Supported Platforms

RLCatalyst is currently supported on

- Ubuntu 16.04
- Ubuntu 14.04
- CentOS 7

Software Requirements

Softwares	Versions
MongoDB	3.2.7 and above
Node.JS	4.4.4 and above
Npm	3.5.2 and above
Chef Server	12.3 Open Source or Latest Enterprise Chef Server version.
Chef Client	12.6
LDAP Server	OpenLDAP 2.4.39 or an existing corporate LDAP server.

SCM – Any version of GIT or SVN

10.1 Install Using QuickInstaller

RLCatalyst can be installed quickly using a chef-based installer. This installer will be available for installing on Centos, Ubuntu and Windows. The catalyst installer will install RLCatalyst and Open Source Chef. Basic seed data to start the application will also be taken care by the installer

Ubuntu 14.04 and Centos

You can install RLCatalyst using the installer script

```
wget https://raw.githubusercontent.com/RLOpenCatalyst/installer/master/installer.sh
chmod +x installer.sh
./installer.sh
```

10.2 Install using Docker

You can install RLCatalyst on Containers. Use below commands to pull docker images for catalyst:

```
docker pull relevancelab/catalyst-db:latest
docker pull relevancelab/rlcatalyst:4.0
```

and run below commands.

```
docker run --name rlcdb -d relevancelab/catalyst-db:latest
docker run --link rlcdb:rlcdb -e DB_HOST=rlcdb --name rlcat -d -p 3001:3001_
↪relevancelab/rlcatalyst:4.0
```

Now you can access RLCatalyst using <http://<hostip>:3001>:

Login Credentials

```
UN:  superadmin
PWD: superadmin@123
```

10.3 Install using Vagrant

Setup RLCatalyst on your dektop/laptop using vagrant. You need to have vagrant installed in your machine:

RL Catalyst Installation on Vagrant with chef:

```
git clone https://github.com/RLOpenCatalyst/installer.git
cd vagrantwithchef
vagrant up
```

RL Catalyst Installation on Vagrant without chef:

```
git clone https://github.com/RLOpenCatalyst/installer.git
cd vagrant
vagrant up
```

To install the entire devops roles along with catalyst please follow the below mentioned instructions:

```
git clone https://github.com/RLOpenCatalyst/installer.git
cd RLCatalyst3.0.4
vagrant up
```

The devops roles contain the following third party softwares:

```
Jenkins
Nexus
```


10.4 Install from AWS AMI

If you have an AWS account, you can bring up RLCatalyst using the public AMI available. The public image is currently available for US east(N.Virginia) region. This comes with some basic configurations required by RLCatalyst

1. From your EC2 dashboard, select N.Virginia region . In the Images/AMI link, choose “Public Images” in the dropdown . Search for image with AMI ID `ami-08eab9dd32295134f` and AMI Name `RLCatalyst4.3.2.b59`
2. Select the image and hit Launch
3. On the “Instance Type” page ,choose the instance size as `t2.medium` or bigger . We recommend atleast 4 GB RAM
4. On the “Configure Instance Details” page, choose your preferred Network and Subnets. If you want to assign a public IP to RLCatalyst, then enable “Auto-assign Public IP”
5. On “Tag Instance” , name your instance
6. On “Security Group” , add rule to open ports 443, 8080,8081 and 3001.
7. Review and launch . Once the instance is in “Running” state , you can access RLCatalyst at `http://<ip>:3001` . Login using `superadmin/superadmin@123`
8. This image includes the devops role like jenkins at `http://<ip>:8080` (default credentials)
9. This image includes the devops role Nexus at `http://<ip>:8081` (default credentials)

Following video demonstrates how to Quick install using RLCatalyst public AWS Image:

10.5 Install Directly From Source

You can install RLCatalyst locally by downloading the code from our Github Code Repository. The following section gives the instructions to do the RLCatalyst Installation from source

10.5.1 Installation Steps for Ubuntu 16.04

Here is a step by step plan on how to install RLCatalyst on Ubuntu machine.

Start by updating the System:

```
sudo apt-get update
```

Install MongoDB

Import the public key used by the package management system.The Ubuntu package management tools (i.e. `dpkg` and `apt`) ensure package consistency and authenticity by requiring that distributors sign packages with GPG keys.

Issue the following command to import the MongoDB public GPG Key:

```
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv EA312927
```

Create a list file for MongoDB:

```
echo "deb http://repo.mongodb.org/apt/ubuntu trusty/mongodb-org/3.2 multiverse" | \
↪ sudo tee /etc/apt/sources.list.d/mongodb-org-3.2.list
```

Reload local package database & install MongoDB:

```
sudo apt-get update
sudo apt-get install -y mongodb-org
```

Check the Mongo version:

```
mongo -version
3.2.x
```

Start the mongodb:

In order to properly launch MongoDB as a service on Ubuntu 16.04, we additionally need to create a unit file describing the service. A unit file tells system how to manage a resource. The most common unit type is a service, which determines how to start or stop the service, when should it be automatically started at boot, and whether it is dependent on other software to run. We'll create a unit file to manage the MongoDB service.

Create a configuration file named `mongodb.service` in the `/etc/systemd/system` directory using nano or your favorite text editor.

```
sudo nano /etc/systemd/system/mongodb.service
```

Paste in the following contents, then save and close the file.

```
[Unit]
Description=High-performance, schema-free document-oriented database
After=network.target

[Service]
User=mongodb
ExecStart=/usr/bin/mongod --quiet --config /etc/mongod.conf

[Install]
WantedBy=multi-user.target
```

Next, start the newly created service with `systemctl`.

```
sudo systemctl start mongodb
```

Use `systemctl` to check that the service has started properly.

```
sudo systemctl status mongodb
```

Output:

```
mongodb.service - High-performance, schema-free document-oriented database
   Loaded: loaded (/etc/systemd/system/mongodb.service; enabled; vendor preset: _
→enabled)
   Active: active (running) since Mon 2016-04-25 14:57:20 EDT; 1min 30s ago
     Main PID: 4093 (mongod)
        Tasks: 16 (limit: 512)
       Memory: 47.1M
          CPU: 1.224s
      CGroup: /system.slice/mongodb.service
              └─4093 /usr/bin/mongod --quiet --config /etc/mongod.conf
```

Enable automatically starting MongoDB when the system starts.

```
sudo systemctl enable mongod
```

Install NodeJs 4.x & Curl:

```
sudo apt-get install curl
```

To Install Nodejs version 4.x

```
curl -sL https://deb.nodesource.com/setup_4.x | sudo -E bash -
sudo apt-get install -y nodejs
```

Check the version of node after installation. It should be 4.2.2 and above

```
node -v
v4.2.2
```

Check the version of npm

```
npm -v

NOTE - The npm version required is 3.x . If an older version got installed, upgrade
↳ the npm version
    sudo apt-get update
    sudo npm install npm -g

Now check the npm version and make sure it is 3.5.x and above
npm -v
```

Install Git(1.9.x)

```
sudo apt-get install git
```

NOTE:

```
Node Version - 4.2.2 and above
npm version - 3.6.x and above
monogo version - 3.2.x
```

Clone the repository to get the RLCatalyst code:

```
sudo git clone https://github.com/ROpenCatalyst/core.git
```

Create a Mongodb path:

```
sudo mkdir -p /data/db/
```

Install ChefClient:

```
sudo curl -L https://www.opscode.com/chef/install.sh | sudo bash

To Check the chef client version
knife -v
It should be 12.6 or above
```

Install the dependencies- make , g++ , Kerberos & library:

```
sudo apt-get install make
sudo apt-get install g++
sudo apt-get install libkrb5-dev
sudo apt-get install build-essential checkinstall
sudo apt-get install libreadline-gplv2-dev libncursesw5-dev libssl-dev libsqlite3-dev
tk-dev libgdbm-dev libc6-dev libbz2-dev
cd /usr/src
sudo wget https://www.python.org/ftp/python/2.7.13/Python-2.7.13.tgz
sudo tar xzf Python-2.7.13.tgz
cd Python-2.7.13
sudo ./configure
sudo make altinstall

To Check the python version
python2.7 -V

To set python environment variable run below command
export PYTHON=/usr/local/bin/python2.7

sudo ln -s /usr/local/bin/python2.7 /usr/local/bin/python
sudo npm install -g kerberos
sudo apt-get install ruby
```

To run the application we need to do a client side build as well:

Pre-requisites:

```
1. Grunt-cli
   sudo npm install -g grunt-cli

2. sass gem
   sudo gem install sass
```

Process to build the client side:

```
cd ~
cd core/client/cat3
sudo npm install --production
sudo npm run-script build-prod
```

Install Node Packages:

```
cd ~
cd core/server
sudo npm install
```

To Install seed data:

```
sudo node install --seed-data
```

To Install forever & start the RLCatalyst Application:

```
sudo npm install -g forever
cd ~
cd core/server/app
sudo forever start app.js
```

Now you can access RLCatalyst at <http://localhost:3001>

```
Login Credentials
superadmin/superadmin@123
```

You are ready to start using RLCatalyst now. Please see *Getting Started* for next steps.

10.5.2 Installation Steps for Ubuntu 14.04

Here is a step by step plan on how to install RLCatalyst on Ubuntu machine.

Start by updating the System:

```
sudo apt-get update
```

Install MongoDB

Import the public key used by the package management system. The Ubuntu package management tools (i.e. dpkg and apt) ensure package consistency and authenticity by requiring that distributors sign packages with GPG keys.

Issue the following command to import the MongoDB public GPG Key:

```
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv EA312927
```

Create a list file for MongoDB:

```
echo "deb http://repo.mongodb.org/apt/ubuntu trusty/mongodb-org/3.2 multiverse" | \
sudo tee /etc/apt/sources.list.d/mongodb-org-3.2.list
```

Reload local package database & install MongoDB:

```
sudo apt-get update
sudo apt-get install -y mongodb-org
```

Check the Mongo version:

```
mongo -version
3.2.x
```

Start the mongod:

```
sudo service mongod start
```

Install NodeJs 4.x & Curl:

```
sudo apt-get install curl
```

To Install Nodejs version 4.x:

```
curl -sL https://deb.nodesource.com/setup_4.x | sudo -E bash -
sudo apt-get install -y nodejs
```

Check the version of node after installation. It should be 4.2.2 and above

```
node -v
v4.2.2
```

Check the version of npm

```
npm -v
```

NOTE - The npm version required **is 3.x** . If an older version got installed, upgrade **↪** the npm version

```
sudo apt-get update
sudo npm install npm -g
```

Now check the npm version **and** make sure it **is 3.5.x and** above

```
npm -v
```

Install Git(1.9.x)

```
sudo apt-get install git
```

NOTE:

```
Node Version - 4.2.2 and above
npm version - 3.6.x and above
monogo version - 3.2.x
```

Clone the repository to get the RLCatalyst code:

```
sudo git clone https://github.com/RLOpenCatalyst/core.git
```

Create a Mongodb path:

```
sudo mkdir -p /data/db/
```

Install ChefClient:

```
sudo curl -L https://www.opscode.com/chef/install.sh | sudo bash
To Check the chef client version
knife -v
It should be 12.6 or above
```

Install the dependencies- make , g++ , Kerberos & library:

```
sudo apt-get install make
sudo apt-get install g++
sudo apt-get install libkrb5-dev
sudo npm install -g kerberos
```

To run the application we need to do a client side build as well:

Pre-requisites:

```
1. Grunt-cli
  sudo npm install -g grunt-cli

2. sass gem
  sudo apt-get install ruby
  sudo gem install sass
```

Process to build the client side:

```
cd core/client/cat3
sudo npm install --production
sudo npm run-script build-prod
```

Install Node Packages:

```
cd ~
cd core/server
sudo npm install
```

To Install seed data:

```
sudo node install --seed-data
```

To Install forever & start the RLCatalyst Application:

```
sudo npm install forever --global
cd ~
cd core/server/app
sudo forever start app.js
```

Now you can access RLCatalyst at <http://localhost:3001>

```
Login Credentials
superadmin/superadmin@123
```

You are ready to start using RLCatalyst now. Please see *Getting Started* for next steps .

10.5.3 Installation Steps for Centos7

Here is a step by step plan on how to install RLCatalyst on Centos7 machine.

Update your System with yum:

```
yum update
```

To Install node.js & npm:

```
# Install the repository
rpm -Uvh https://rpm.nodesource.com/pub_4.x/el/7/x86_64/nodesource-release-el7-1.
↪noarch.rpm
```

```
# Install NodeJS
yum install nodejs
```

```
checking the node version
node -v
4.2.2
```

```
Check the npm version
npm -v
```

NOTE - The npm version required is 3.5.x . If an older version got installed, upgrade ↪
↪the npm version.

(continues on next page)

(continued from previous page)

```
npm install npm -g

Now check the npm version
npm -v
3.5.3
```

To Install MongoDB (version 3.x):

```
Go to directory /etc/yum.repos.d/

Create a file mongodb-enterprise.repo
cat > mongodb-enterprise.repo
Edit the above file and add the contents

[MongoDB]
name=MongoDB Repository
baseurl=http://repo.mongodb.org/yum/redhat/$releasever/mongodb-org/3.2/x86_64/
gpgcheck=0
enabled=1

Save the file

Run the Command
yum install mongodb-org

check the mongo version
mongod --version
3.2.1
```

NOTE:

```
npm version 3.5.3
node version 4.2.5
monogd verison 3.2.1
```

To Install Chef-Client (version 12.6.0):

```
curl -L https://www.opscode.com/chef/install.sh | sudo bash
To check the chef client version
knife -v
Chef:12.6.0
```

To Install git:

```
yum install git
To check the git version
git -version
1.7.x
```

To Install RLCatalyst and to create a db path folder:

```
To pull the catalyst code
git clone https://github.com/RLOpenCatalyst/core.git
Check the current directory for the presence of catalyst code i.e core folder.
```

(continues on next page)

(continued from previous page)

```
Create a db path folder
mongo db path - mkdir -p /data/db/

Go to cd core/server
npm install
```

Start the mongod:

```
sudo service mongod start
```

To Install gcc library:

```
yum install gcc-c++
```

To Install the seed data:

```
node install --seed-data
```

To Start the Application:

```
Run (node app) to start your application.
npm install forever -g
cd core/server/app
node app.js
```

To run the application forever:

```
forever start app.js
```

Access RLCatalyst:

```
http://localhost:3001
username- superadmin
pass - superadmin@123
```

Now you are ready to start using RLCatalyst . Please see [Getting Started](#) for next steps

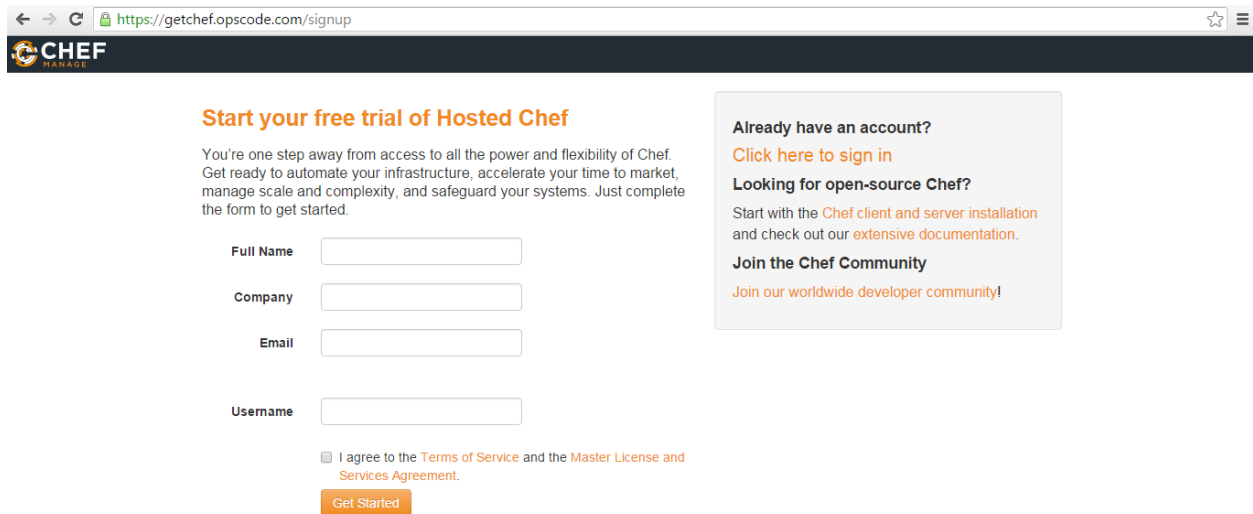
10.6 Working With Chef

Chef helps you express your infrastructure policy – how your software is delivered and maintained on your servers – as code. When infrastructure is code, it becomes more maintainable, versionable, testable, and collaborative. It allows you to install, configure and manage the packages required by your application without the complication of any client and server configuration. RL Catalyst allows you to use either chef or puppet configuration management to manage your Infrastructure.

To begin with you need to create your organization and an account in chef server to automate your infrastructure.

- Go to the below link and create an account in Chef

<https://getchef.opscode.com/signup>



The screenshot shows the Chef signup page at <https://getchef.opscode.com/signup>. The page features the Chef logo and a heading "Start your free trial of Hosted Chef". Below the heading is a paragraph explaining the benefits of Chef. There are four input fields: "Full Name", "Company", "Email", and "Username". Below these fields is a checkbox for agreeing to the Terms of Service and Master License and Services Agreement. A "Get Started" button is at the bottom. On the right side, there is a grey box with links for "Already have an account?", "Looking for open-source Chef?", and "Join the Chef Community".

Start your free trial of Hosted Chef

You're one step away from access to all the power and flexibility of Chef. Get ready to automate your infrastructure, accelerate your time to market, manage scale and complexity, and safeguard your systems. Just complete the form to get started.

Full Name

Company

Email

Username

☐ I agree to the [Terms of Service](#) and the [Master License and Services Agreement](#).

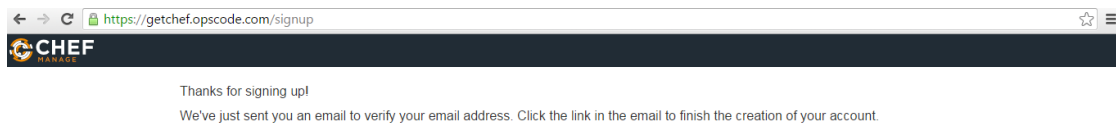
[Get Started](#)

Already have an account?
[Click here to sign in](#)

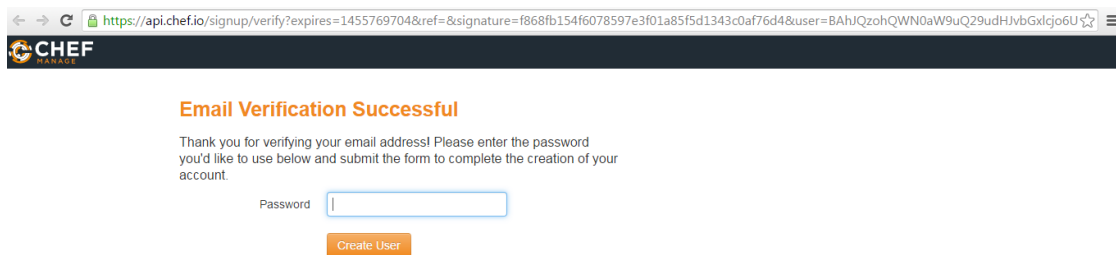
Looking for open-source Chef?
Start with the [Chef client and server installation](#) and check out our [extensive documentation](#).

Join the Chef Community
[Join our worldwide developer community!](#)

- Once you provide the details you will get the below message and email in your mailbox



- Check the email and go to the link provided in the email and provide the password



The screenshot shows the Chef email verification page at <https://api.chef.io/signup/verify?expires=1455769704&ref=&signature=f868fb154f6078597e3f01a85f5d1343c0af76d4&user=BAhJQzohQWN0aW9uQ29udHJvbGxlco6U>. The page features the Chef logo and a heading "Email Verification Successful". Below the heading is a paragraph explaining the next steps. There is a "Password" input field and a "Create User" button.

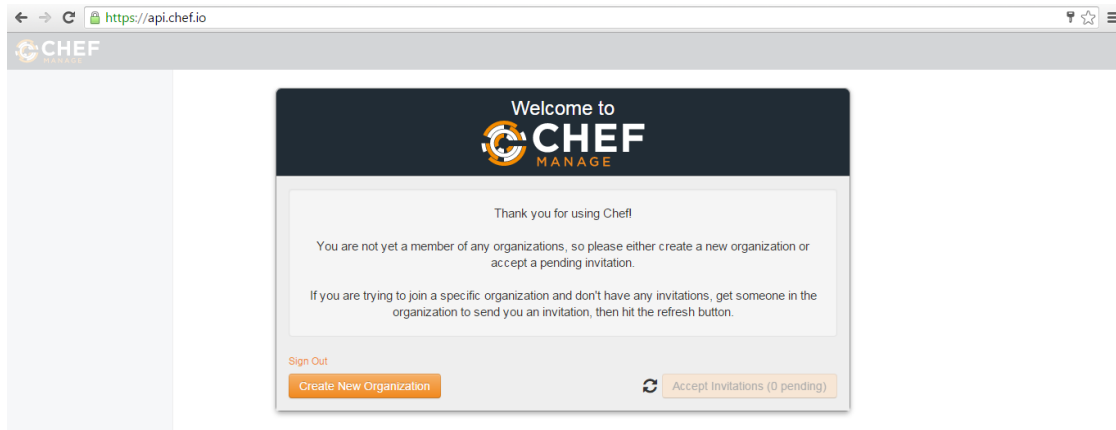
Email Verification Successful

Thank you for verifying your email address! Please enter the password you'd like to use below and submit the form to complete the creation of your account.

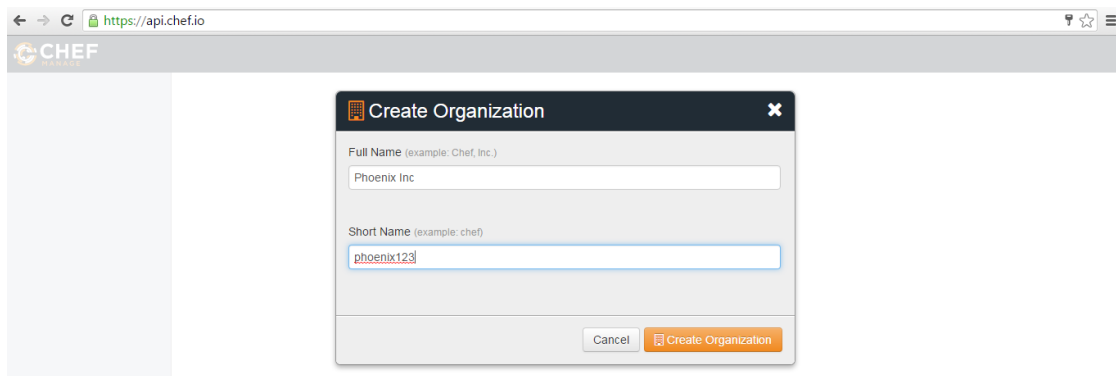
Password

[Create User](#)

- Once you provided the password, you will get the below message

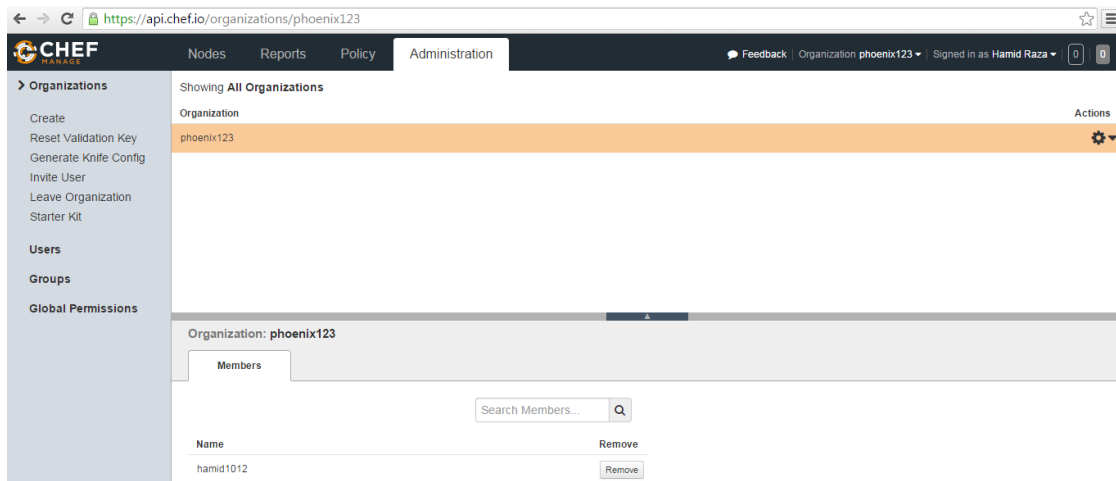


- Create an Organization

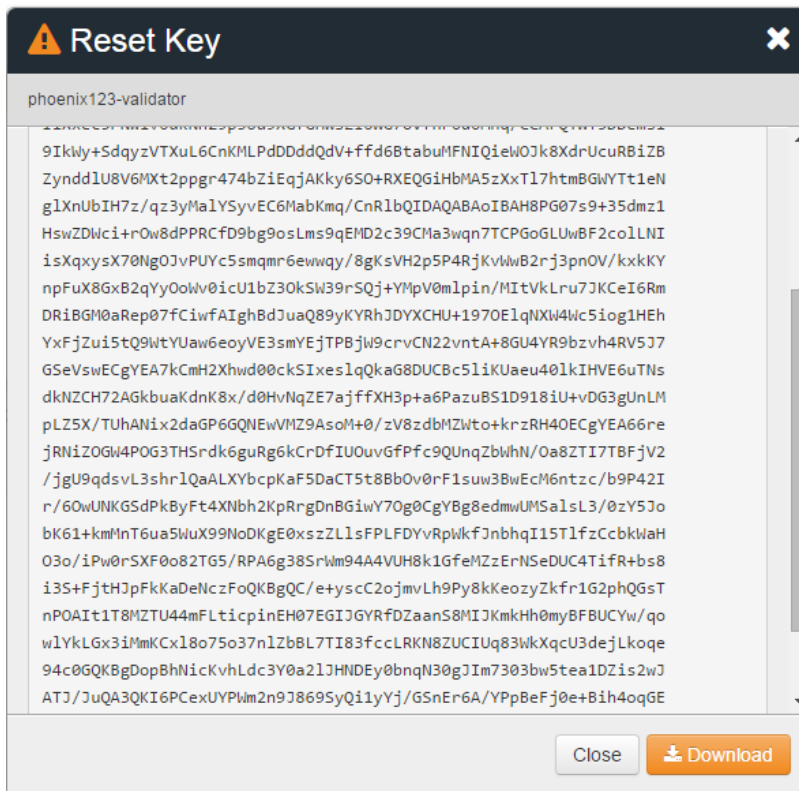
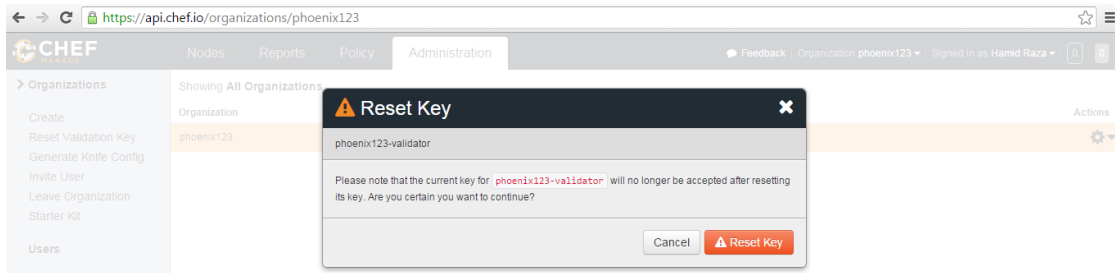


Done Now you have a Hosted Chef account.

- Goto The Administrator Tab

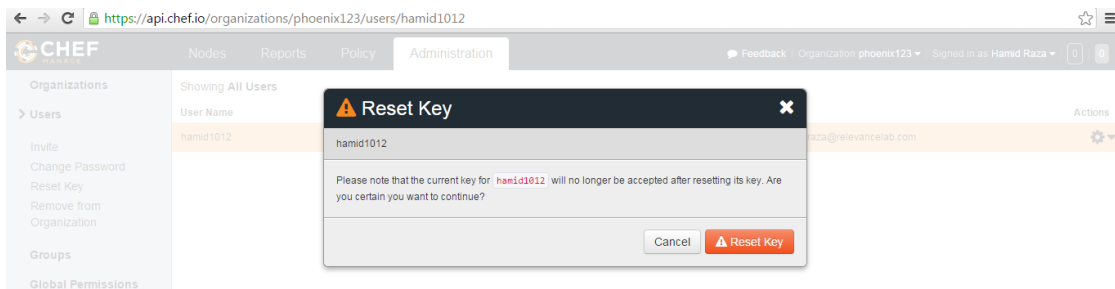


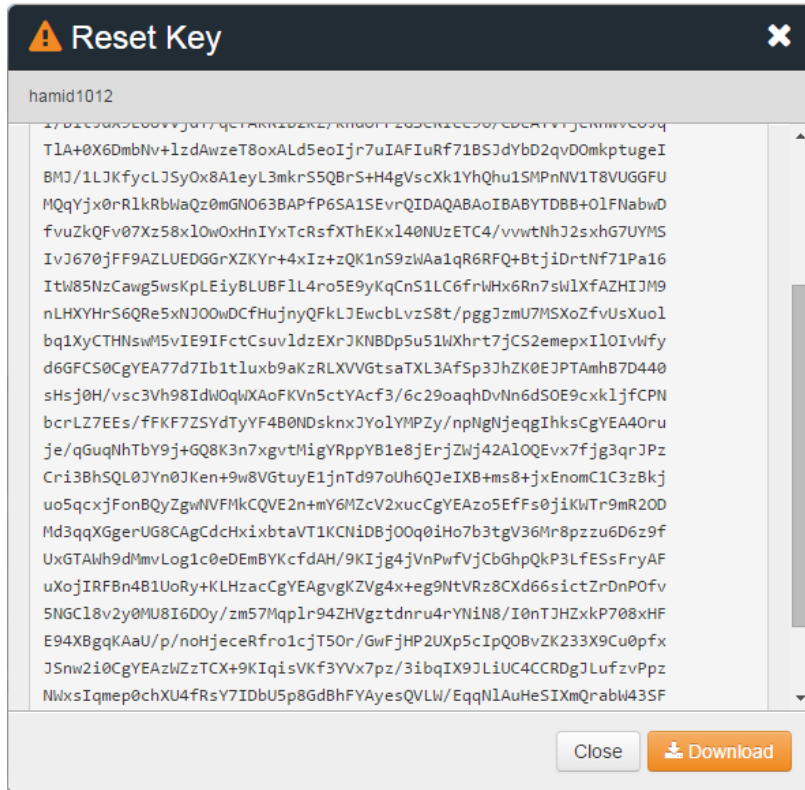
- Reset the Validation Key



Download the key in your desktop

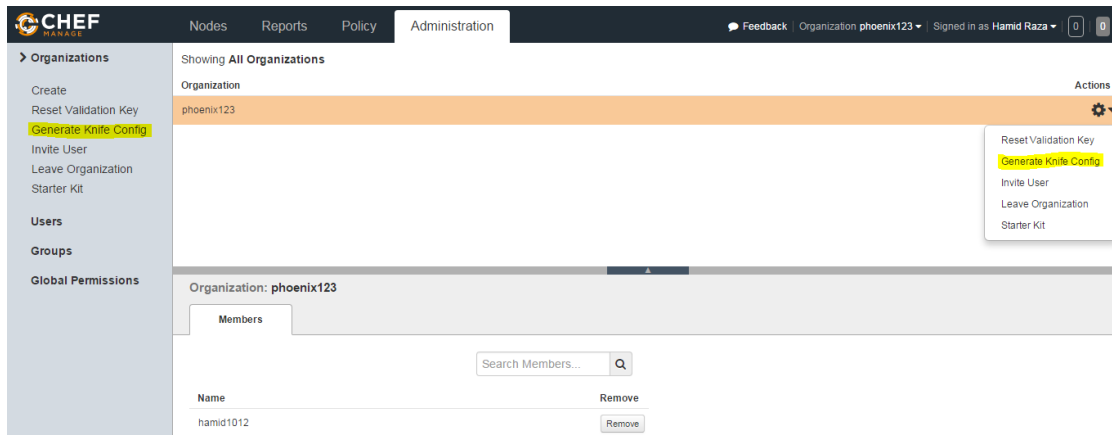
- Reset the User key





Download the key

- Generate the Knife Configuration



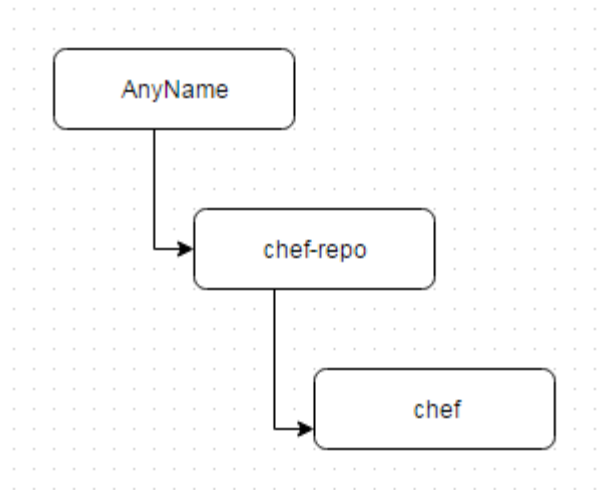
So Now you have the user key, organization key and knife configuration

- Install Chef-Client on your desktop (Windows or Linux)

<https://downloads.chef.io/chef-client/>

- Create a folder in any location and then under that create chef-repo . Create a folder chef(windows) or .chef(linux)

The directory structure will be:



Put all the files in the chef folder

- To check the connectivity from your workstation to the hosted chef server

Just run any command

```
RLE0261@RLE0261 MINGW64 /d/MyTestChef/chef-repo/chef
$ knife node list
```

10.6.1 Upload cookbooks into Chef

Once you setup Chef, the automation libraries need to be uploaded into Chef account. RL automation libraries are available at <https://github.com/RLOpenCatalyst/automationlibrary.git>

- clone the git repository to above directory
git clone <https://github.com/RLOpenCatalyst/automationlibrary.git>
- Run the ruby program (cookbooks_upload.rb) available in the git repo.
ruby cookbooks_upload.rb

It will upload all the cookbooks to the hosted chef server along with the dependencies.

Contributor License Agreement

If you are contributing to this repository you must agree the following :

- The code I'm contributing is mine, and I have the right to license it.
- I'm granting you a license to distribute said code under the terms of this agreement (typically "as you see fit" or "under an OSI-approved license" or whatever).

Branching Strategy

The central repository will have two branches with infinite lifetime

- master
- dev

Supported Branches For contributions to RLCatalyst create a feature or hot fix branch and make changes. Usage of the branches shall be explained in the following section.

Code Contributions

Step 1: Fork

```
$ git clone git@github.com:username/core.git
$ cd core
$ git remote add upstream git://github.com/RLOpenCatalyst/core.git
```

Step 2: Branch

Create a feature branch and start hacking:

```
$ git checkout -b my-feature-branch -t origin/master
```

Step 3: Commit

Make sure git knows your name and email address:

```
$ git config --global user.name "J. Random User"
```

```
$ git config --global user.email "j.random.user@example.com"
```

Writing good commit logs is important. A commit log should describe what changed and why. Follow these guidelines when writing one:

The first line should be 50 characters or less and contain a short description of the change prefixed with the name of the changed subsystem (e.g. "net: add localAddress and localPort to Socket"). Keep the second line blank. Wrap all other lines at 72 columns. A good commit log can look something like this:

subsystem: explaining the commit in one line

Body of commit message is a few lines of text, explaining things in more detail, possibly giving some background about the issue being fixed, etc. etc.

The body of the commit message can be several paragraphs, and please do proper word-wrap and keep columns shorter than about 72 characters or so. That way *git log* will show things nicely even when it is indented. The header line should be meaningful; it is what other people see when they run *git shortlog* or *git log --oneline*.

Check the output of *git log --oneline files_that_you_changed* to find out what subsystem (or subsystems) your changes touch.

Step 4: Rebase

Use *git rebase* (not *git merge*) to sync your work from time to time.

```
$ git fetch upstream
```

```
$ git rebase upstream/master
```

Step 5: Tests

Bug fixes and features should come with tests. Add these tests in the tests directory of the repository.

Step 6: Push

\$ git push origin my-feature-branch Pull requests will usually be reviewed within a few days. If there are comments to address, apply your changes in a separate commit and push that to your feature branch. Post a comment in the pull request afterwards; GitHub does not send out notifications when you add commits.

Code Review

Each Github pull request will go through 3 step before merge:

1. We will execute our automated test cases against the pull request. If the tests failed the pull request will be rejected with comments provided on the failures.
2. If tests pass, the RLCatalyst engineering team member will do the review of the changes. Technical communication possible via github.com pull request page. When ready, your pull request will be tagged with label Ready For Merge.
3. Your patch will be merged into master including necessary documentation updates.
4. After merge the feature branch will be deleted.

Release Strategy

Release from master branch will be tagged with `release_<X.X>`

Issue Contributions When opening new issues or commenting on existing issues please make sure discussions are related to concrete technical issues with the RLCatalyst software

RLCatalyst Issue Tracking is handled using Github Issues.

If you are familiar with RLCatalyst and know the repository that is causing you a problem or if you have a feature request on a specific component, you can file an issue in the corresponding GitHub project. All of our Open Source Software can be found in our GitHub organization.

Otherwise you can file your issue in the RLCatalyst project and we will make sure it gets filed against the appropriate project.

To decrease the back and forth in issues, and to help us get to the bottom of them quickly, we use the issue template below. You can copy/paste this template into the issue you are opening and edit it accordingly:

```
Version:[Version of the project installed]

Environment:[Details about the environment such as the Operating System, cookbook_
↳details, etc.]

Scenario:[What you are trying to achieve and you can't?]

Steps to Reproduce:[If you are filing an issue, what are the things we need to do to_
↳reproduce your problem?]

Expected Result:[What are you expecting to happen as the consequence of the_
↳reproduction steps above?]

Actual Result:[What actually happens after the reproduction steps?]
```