RBMS Configuration Store

Version 21.9.1, 07 October 2021

Registered Address	Support	Sales
26, Kingston Terrace, Princeton, New Jersey 08540, United States		
		+91 80 4850 5445
http://www.rtbrick.com	support@rtbrick.com	sales@rtbrick.com

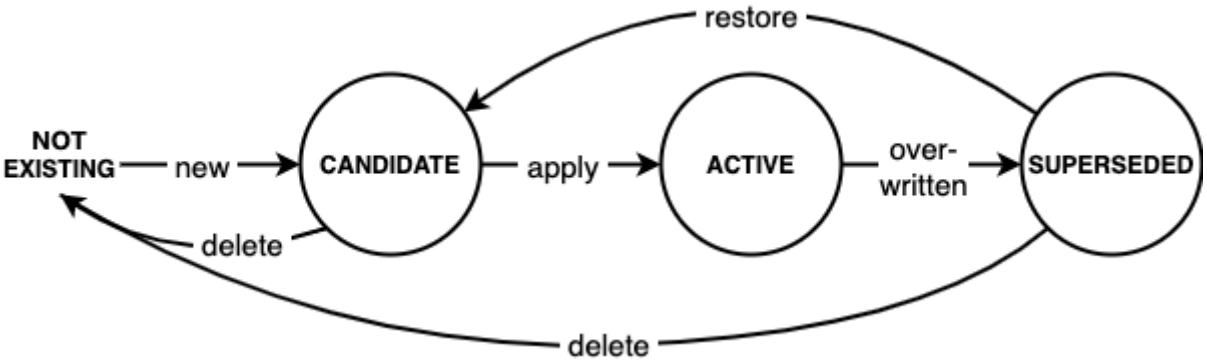
Table of Contents

1. RBMS Configuration Store

The RBMS resource inventory includes a configuration store to maintain switch configurations. The configuration store can store an arbitrary number of configurations per switch and provides a history of up to 50 revisions for each configuration.

RBFS stores the active configuration in RBMS after every configuration change.

The figure below shows the configuration lifecycle.



A new configuration is considered a *candidate* configuration. A candidate configuration turns into the *active* configuration when being applied to the switch. The previously activate configuration is marked as *superseded* at the same time. A superseded configuration can be restored by creating a new candidate configuration from it and applying the candidate configuration again. Candidate and superseded configurations can be removed from the configuration store. The active configuration cannot be deleted.

RBMS provides means to create or upload new configurations and apply them to the switch. In addition, RBMS UI allows inspecting configuration changes in the configuration history.

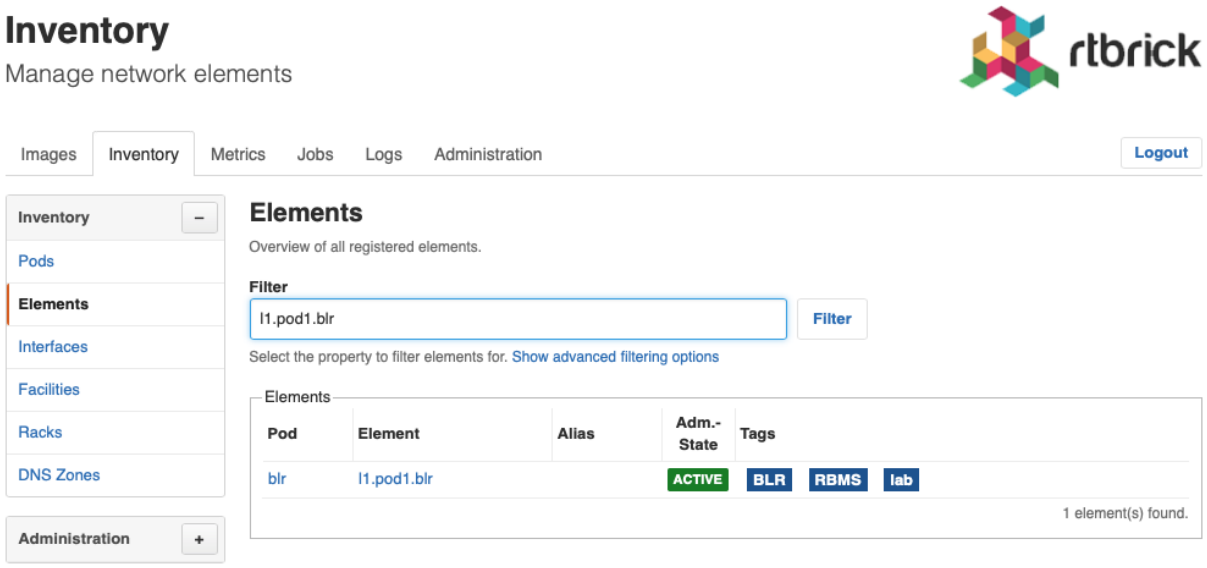
1.1. Creating a new candidate configuration

1.1.1. Uploading a candidate configuration

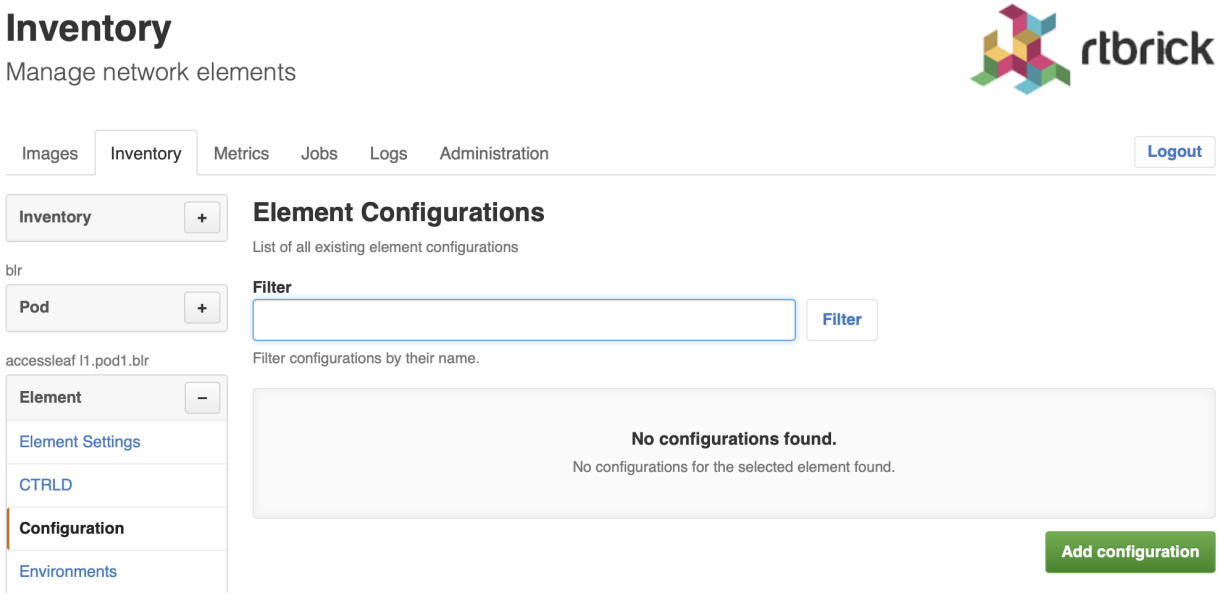
The simplest way to add a new candidate configuration is to upload a new configuration to the configuration store.

To upload a new candidate configuration

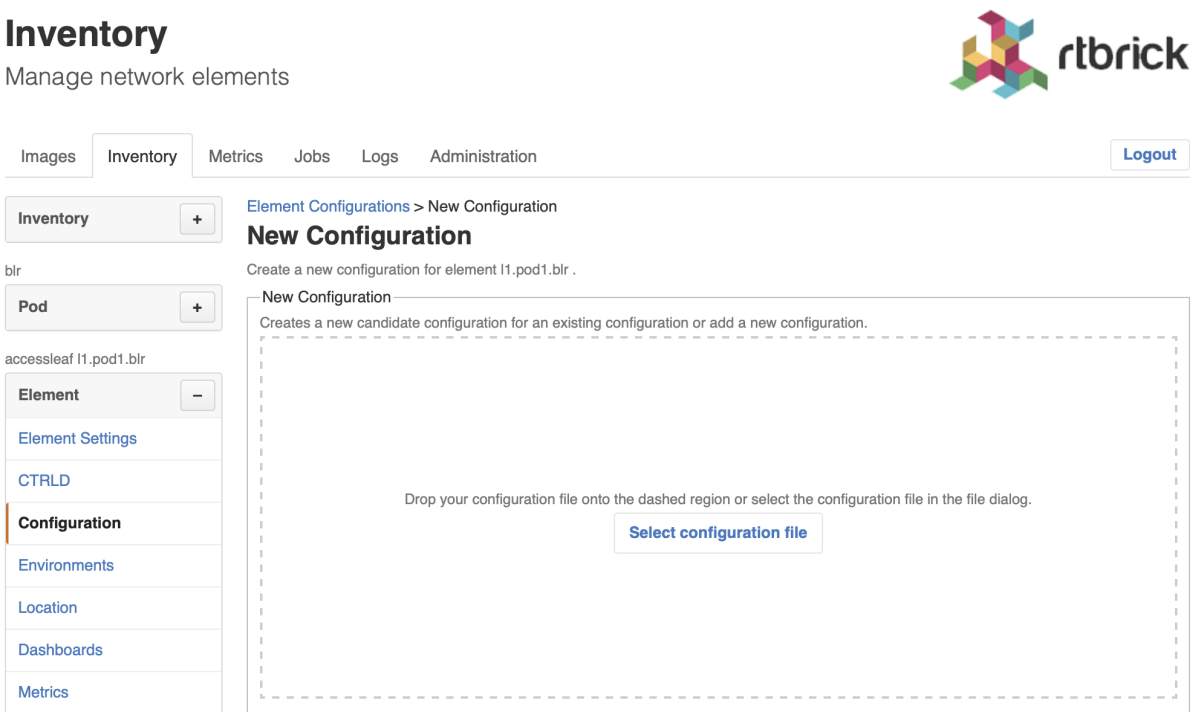
1. Click the **Inventory** tab.
2. Click **Elements** in the left navigation pane. The element list appears.



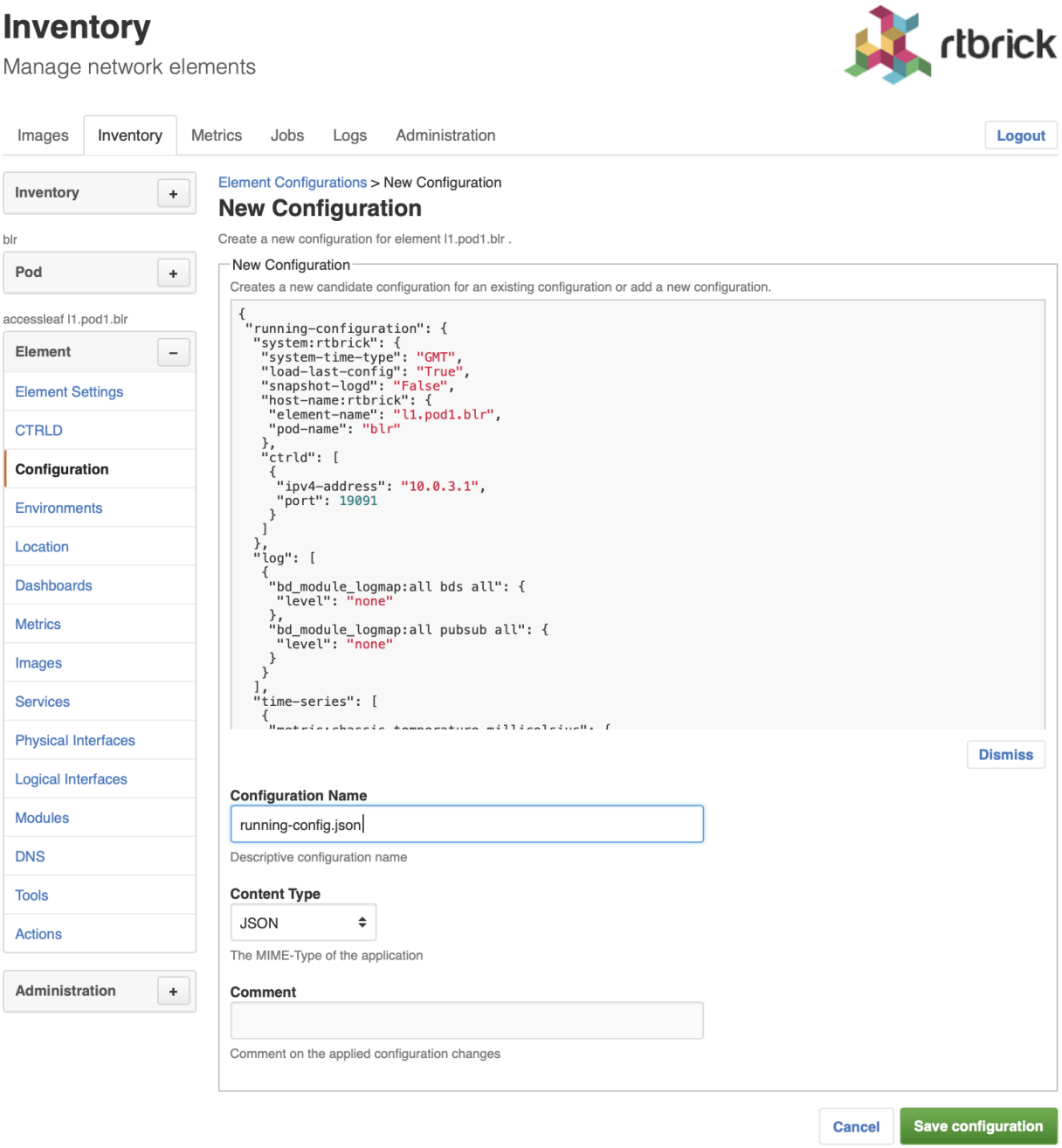
3. Click the element name for which you want to upload a new configuration.
4. Click **Configuration** in the left navigation pane. The configurations list appears.



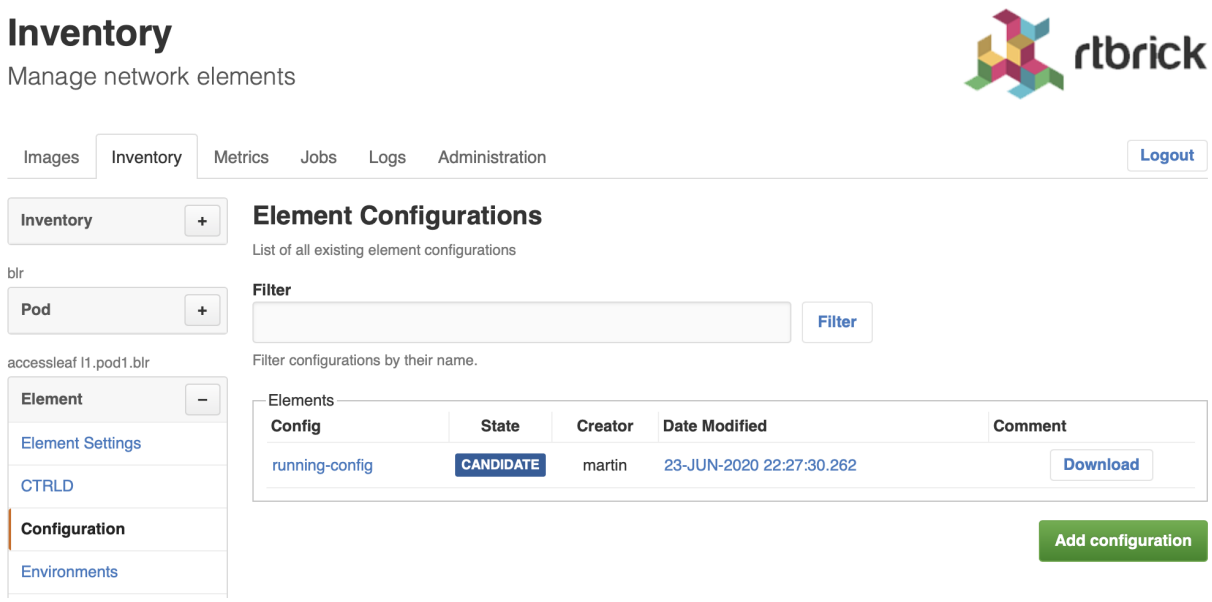
5. Click **Add configuration**. The New Configuration page appears.



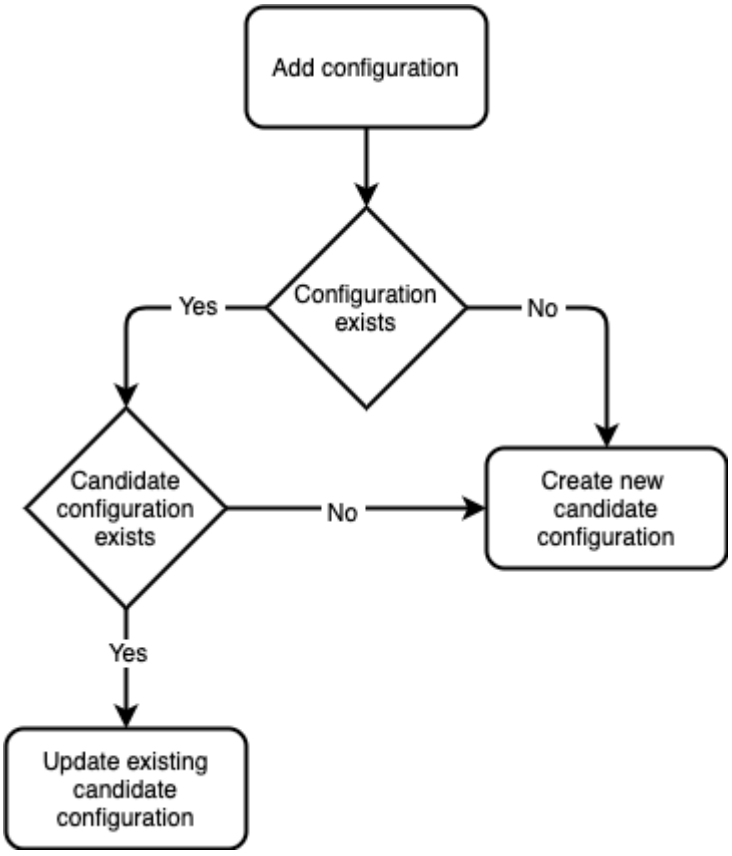
6. Drop a configuration file onto the dashed area or click **Select configuration file** to open the file dialog to select a file.
7. Review the configuration in the preview.



8. In **Configuration Name**, enter the switch configuration name. By default the configuration name is taken from the file name.
9. Check whether the selected **Content Type** is correct.
10. Optionally comment the configuration.
11. Click **Save configuration** to add the candidate configuration.



RBMS either creates a new candidate configuration or updates an existing candidate configuration if a candidate configuration for the specified configuration name exists. The complete flow is illustrated below:



1.1.2. Generating a candidate configuration

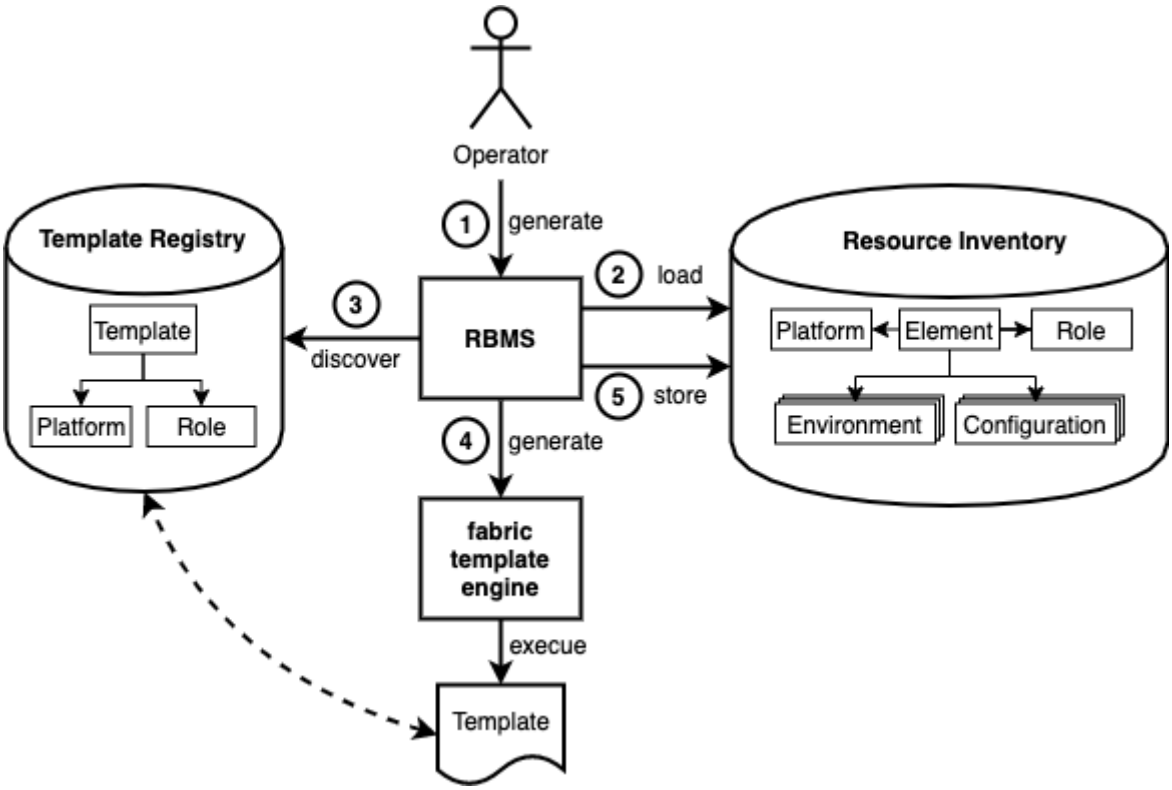
RBMS includes a template engine to generate switch configurations. The configuration templates are designed based on your conventions for building a fabric and then added to the template engine. The templates need to be registered in RBMS to make them eligible for execution. RBMS passes the following

information to the template engine:

- the general element settings
- the software image to be installed on the switch and
- all registered environments

RBMS can store an arbitrary set of *environments* per element. An environment is a JSON object containing parameters processed in the template. The structure and number of environments is defined by the template author.

The figure below shows the configuration generation flow:



1. The operator triggers the switch configuration generation in RBMS.
2. RBMS loads the element settings, environments and image information from the resource inventory.
3. RBMS discovers the templates eligible for the selected switch.
4. RBMS invokes the template engine for each discovered template.
5. RBMS stores the generated configuration as candidate configuration in the configuration store.

1.1.2.1. Template registration

The template registration maintains the list of templates available in the template engine.

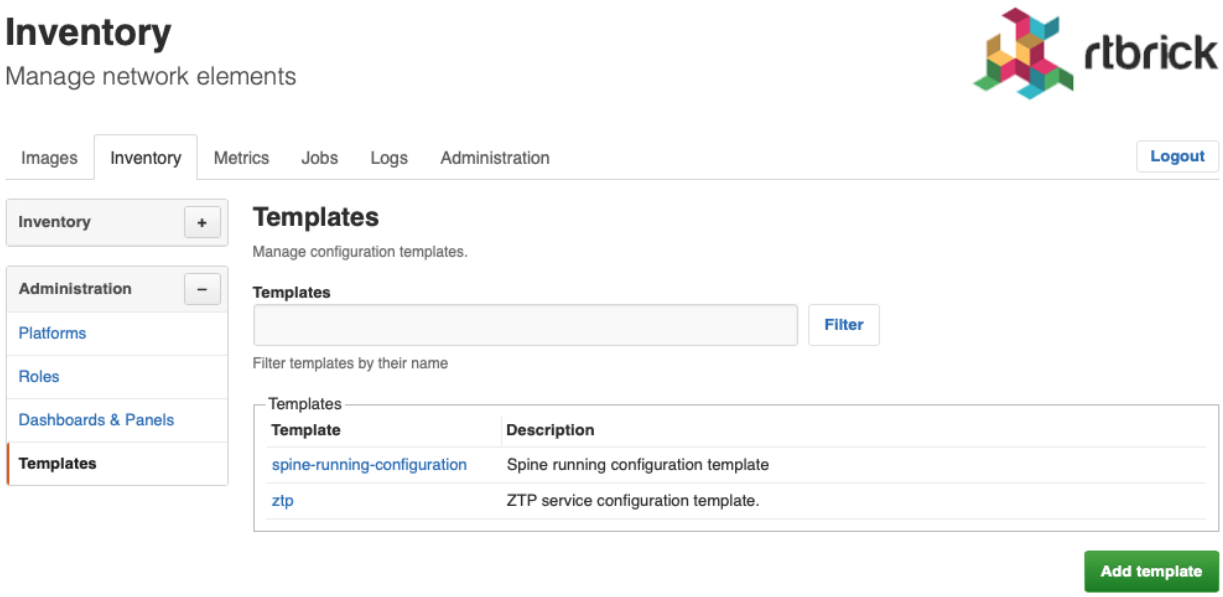
Say you have three different templates:

- ztp.gojson generates the ZTP configuration snippet for a switch.
- leaf-running-configuration.gojson generates the running-configuration of a leaf switch.
- spine-running-configuraton.gojson generates the running-configuration of a spine switch.

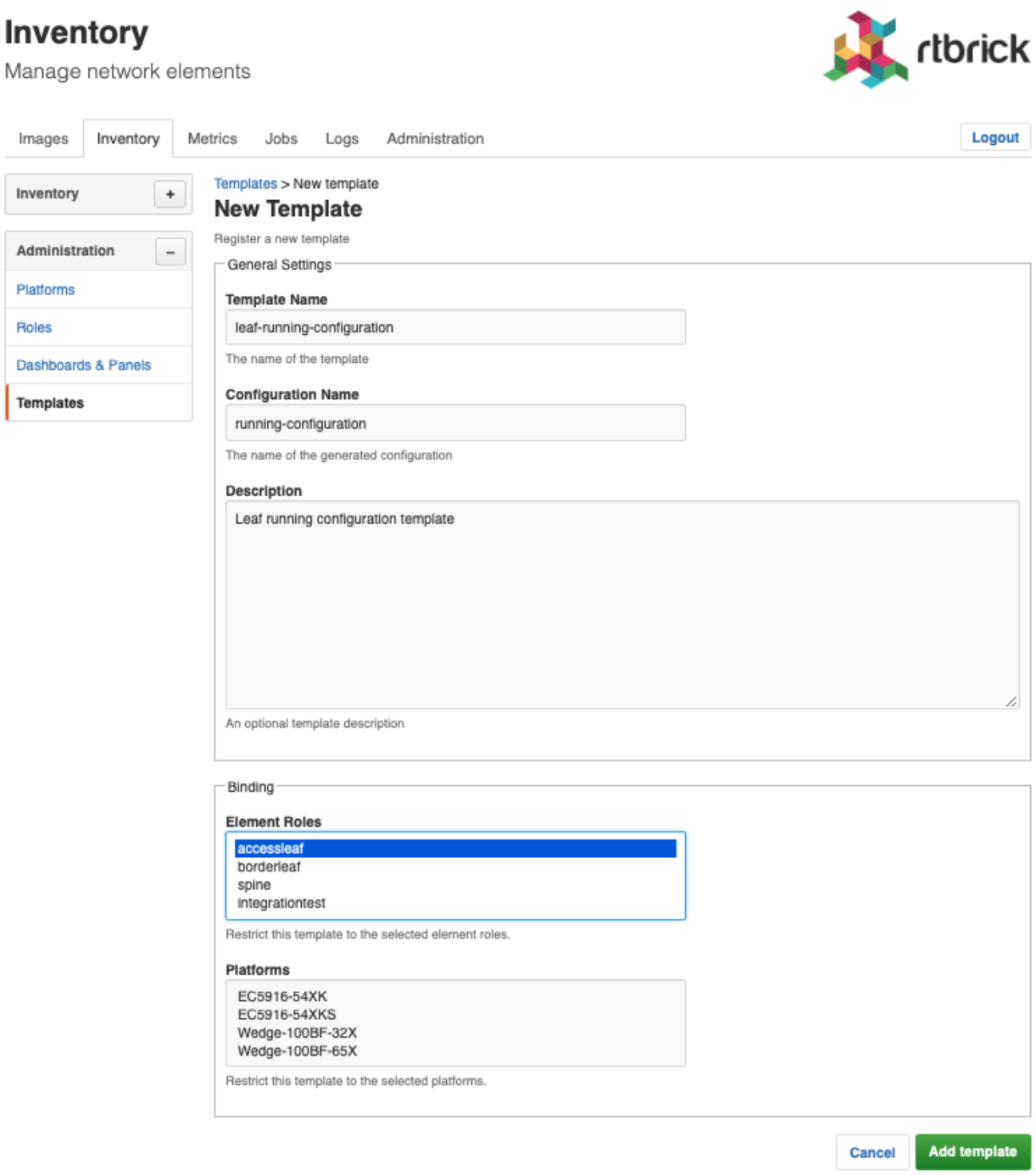
All templates need to be registered in RBMS to get executed when the switch configuration is genereated.

To register a new template

1. Click the **Inventory** tab.
2. Click **+** to expand the **Administration** menu.
3. Click **Templates** in the left navigatio pane. The template list appears. ZTP and spine templates are already registered.



4. Click **Add template** to register the leaf template.

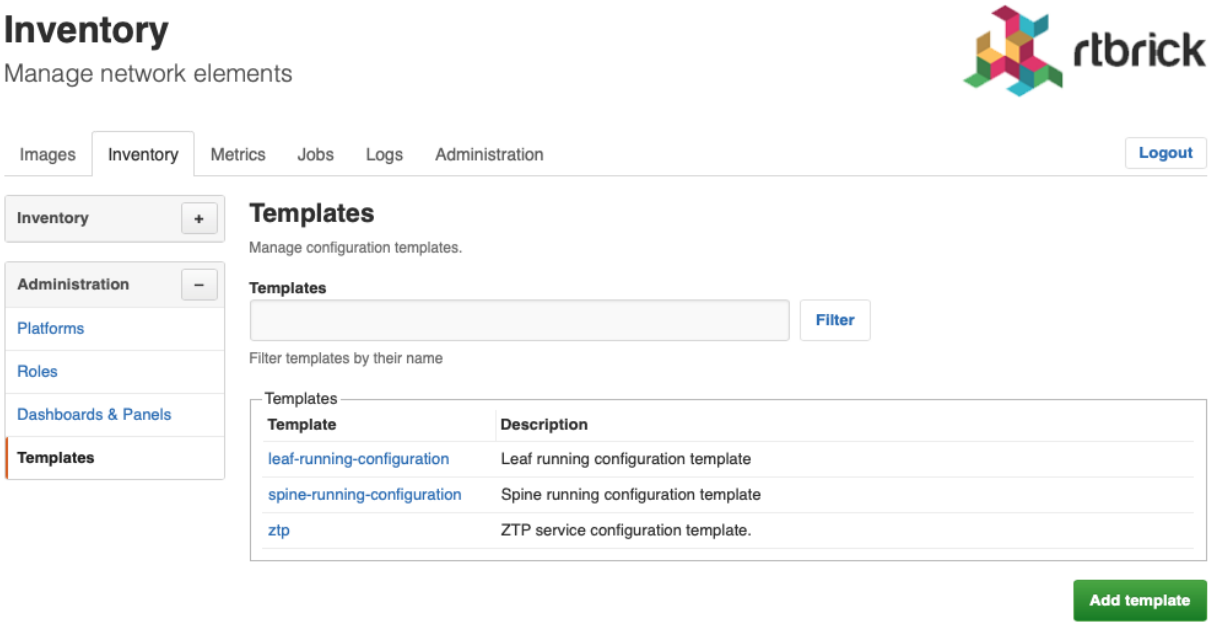


5. In **Template Name**, enter *leaf-running-configuration*.

The template name in RBMS matches the template folder name in the template engine. See the template engine guide for more details.

6. In **Configuration Name**, enter *running-configuration* since the template generates the *running-configuration* for a switch.
7. In **Element Roles**, select *accessleaf* to bind the template to accessleaf switches.
8. In **Description**, optionally describe the template and the generated configuration.

9. Click **Save template**. The template list appears. The new template is listed.

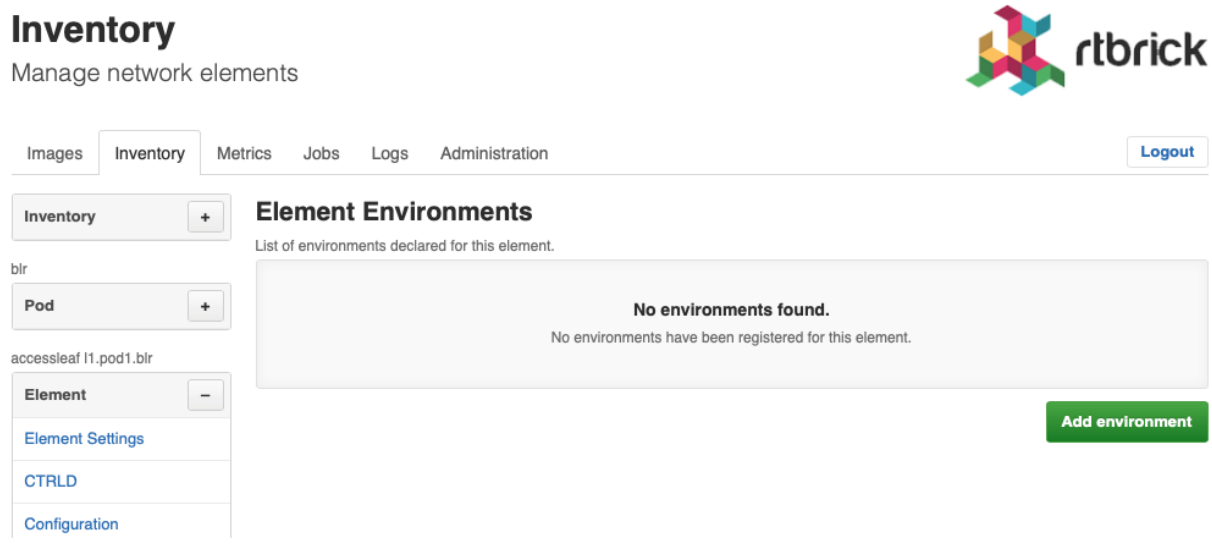


1.1.2.2. Environment management

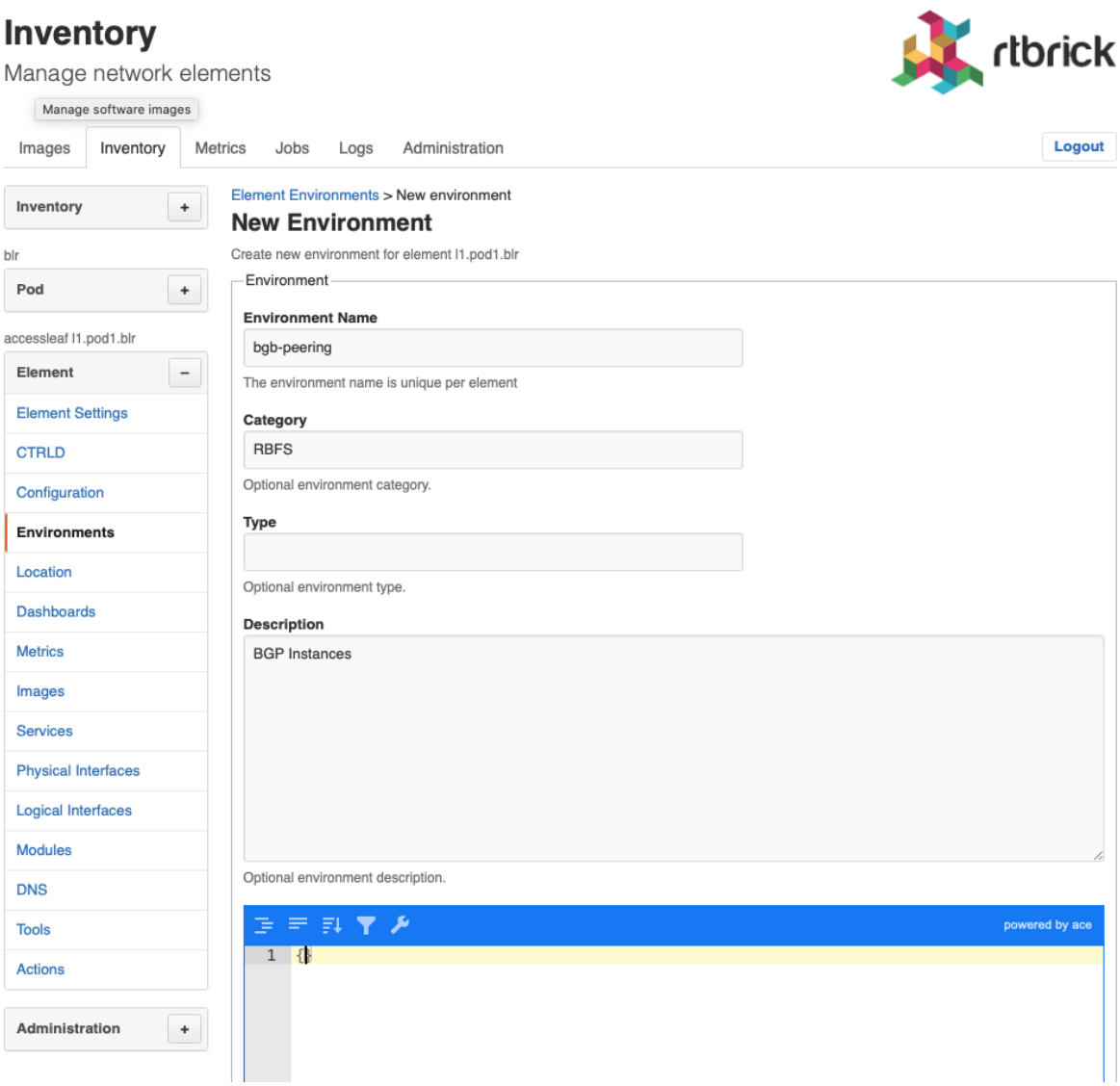
1.1.2.2.1. Adding a new environment

To add a new environment

1. Click **Environments** in the left navigation pane. The list of environment appears.



2. Click **Add environment**. The New Environment view appears.



3. In **Environment Name** enter the name of this environment.
4. In **Category** enter *RBFS*.

Only RBFS environments are processed by the template engine.

5. In **Type** enter an optional type or schema information of the environment.
6. In **Description** enter an optional environment description.
7. Enter the environment variables in the displayed JSON editor or click **Upload new environment** to upload a JSON file.
8. Click **Add environment** to add a new environment.

1.1.2.2.2. Editing an environment

To update an existing environment

1. Click **Environments** in the left navigation pane. The list of environment appears.

2. Click the name of the environment you want to edit.
3. Apply the modifications to the environment.
4. Click **Save environment** to save the environment.

1.1.2.2.3. Removing an environment

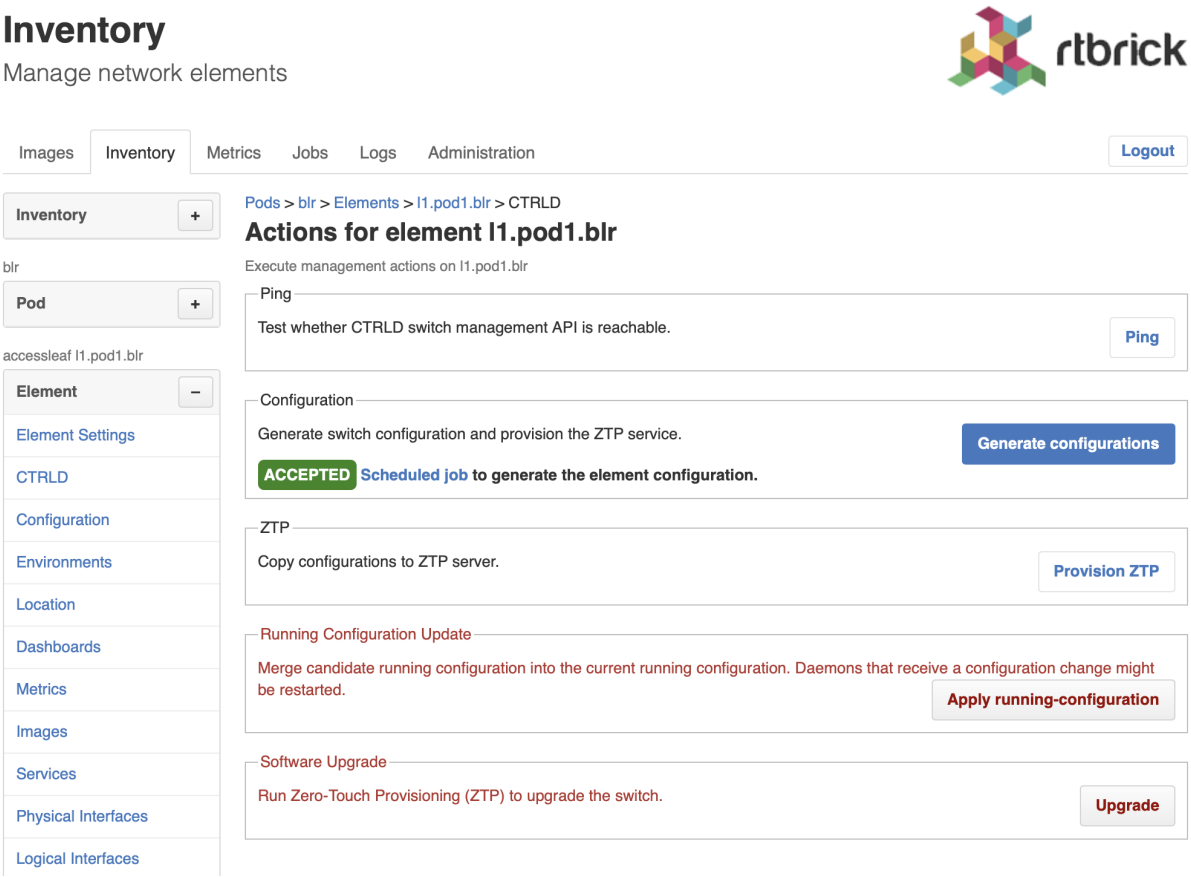
To remove an environment

1. Click **Environments** in the left navigation pane. The list of environment appears.
2. Click the name of the environment you want to remove.
3. Click **Remove environment**. A confirmation dialog is displayed.
4. Click **Confirm** to remove the environment.

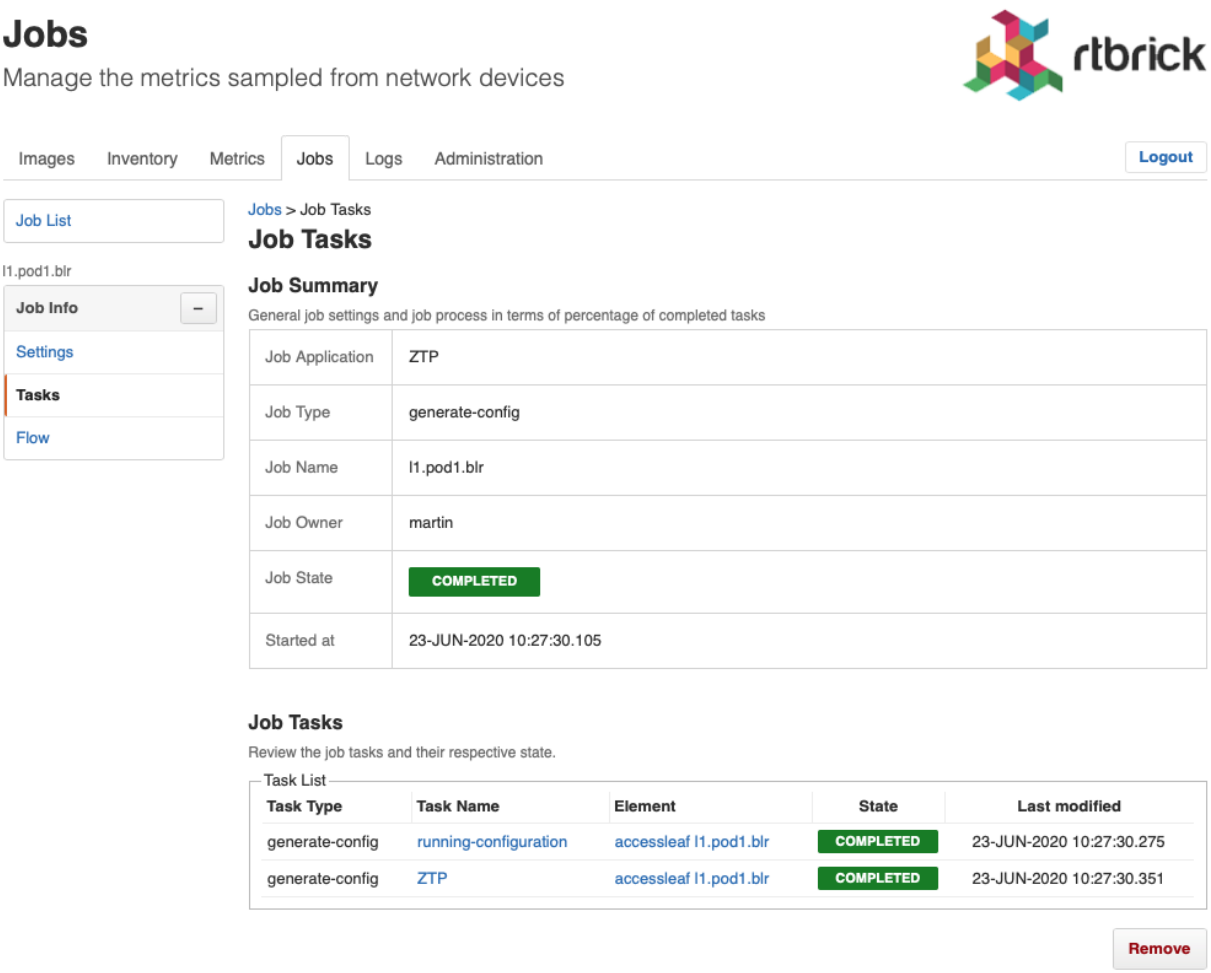
1.1.2.3. Configuration generation

To generate a configuration

1. Click **Actions** in the left navigation pane.



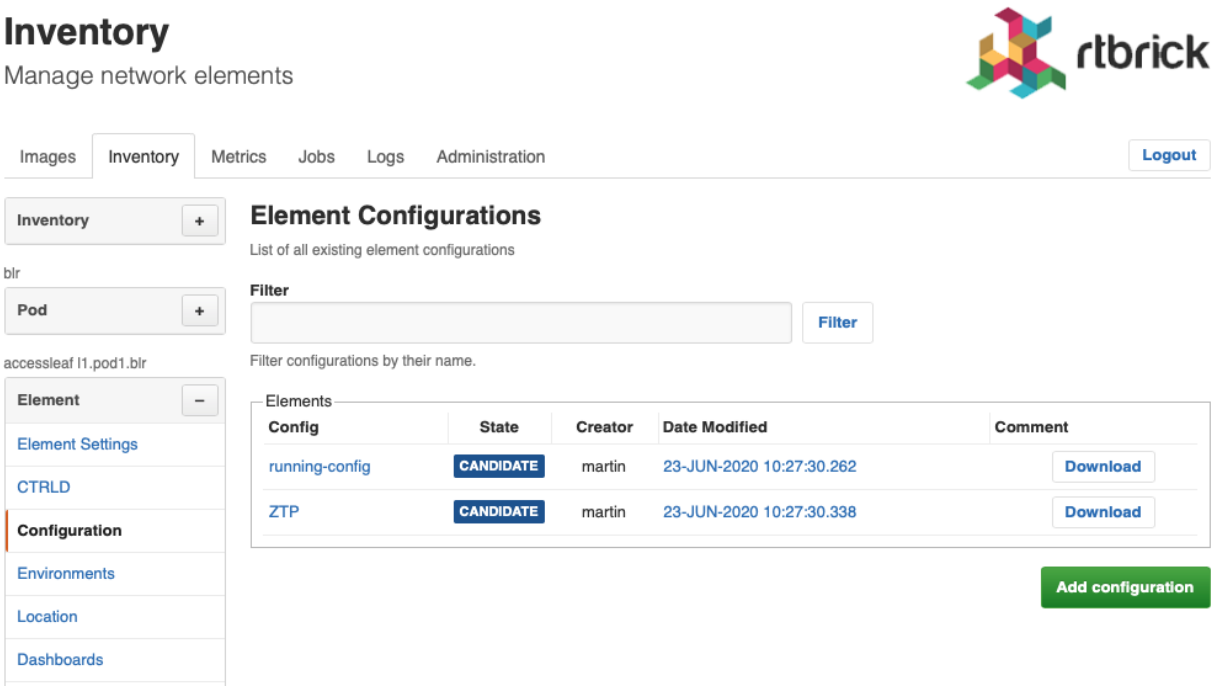
2. Click **Generate configurations**. RBMS schedules a configuration generation job. The job contains a task for each configuration and invokes the template engine to create the configuration.



The generated configurations are stored as new candidate configurations.

To review the generated configuration

1. Click **Configuration** in the left navigation pane. The configuration list shows new candidate configurations.

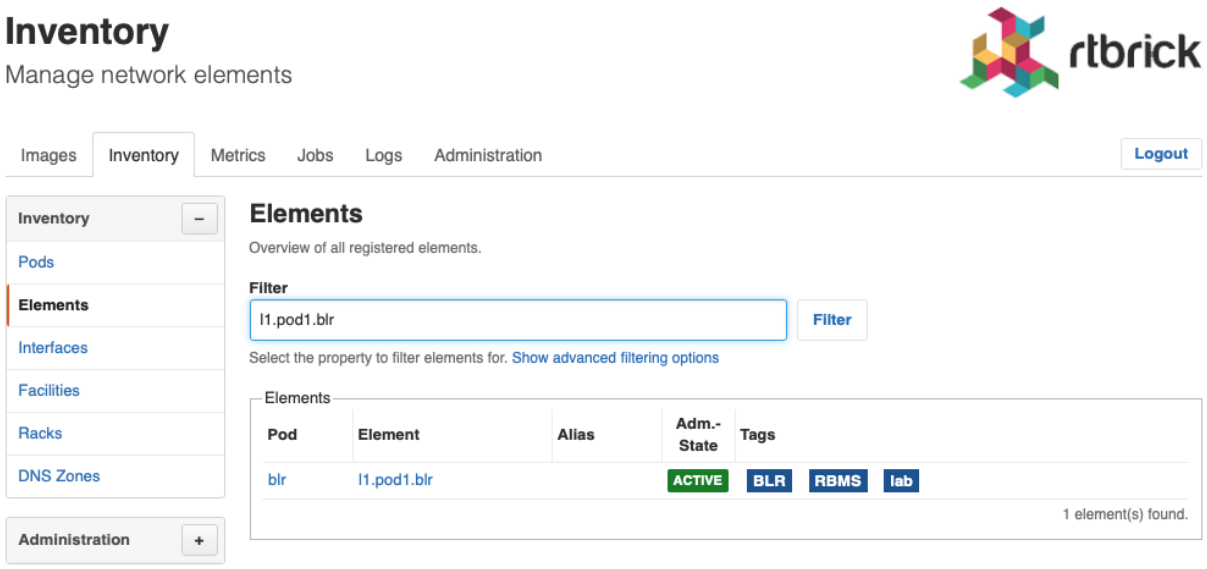


2. Click the timestamp of the generated configuration to show the configuration content. Click the configuration name to display the configuration history.

1.2. Review configuration history

To review configuration changes in the configuration history

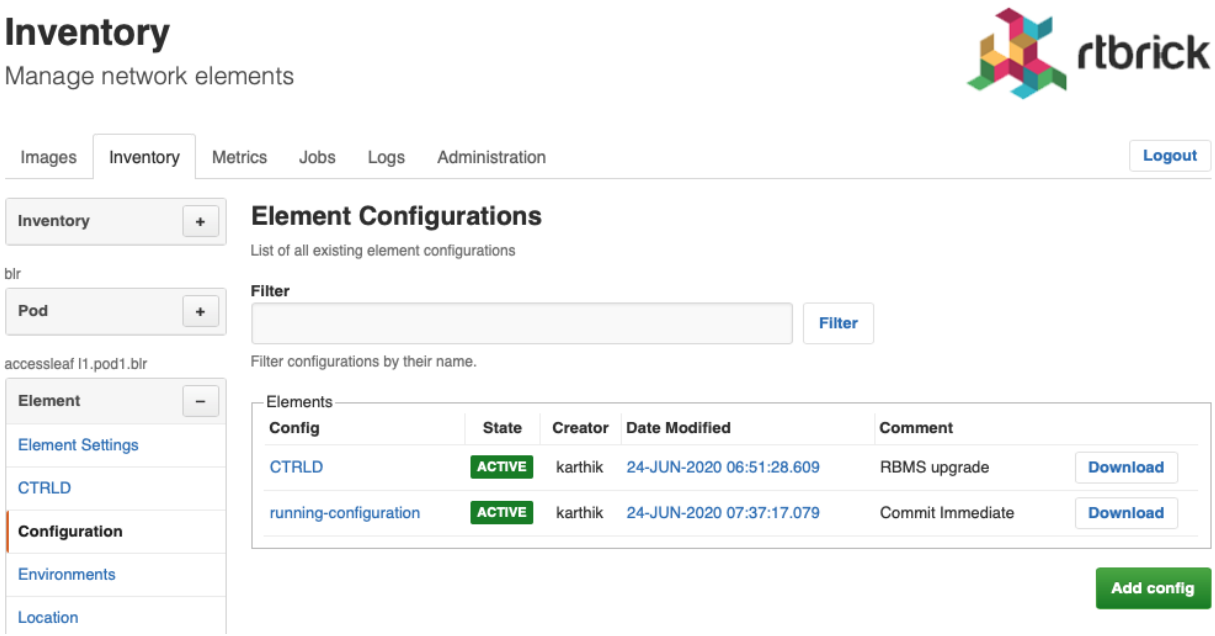
1. Click the **Inventory** tab. The list of pods appears.
2. Click **Element** in the left navigation pane. The element list appears.
3. In **Filter** enter the name of the switch and click **Filter**. The list of matching elements appears.



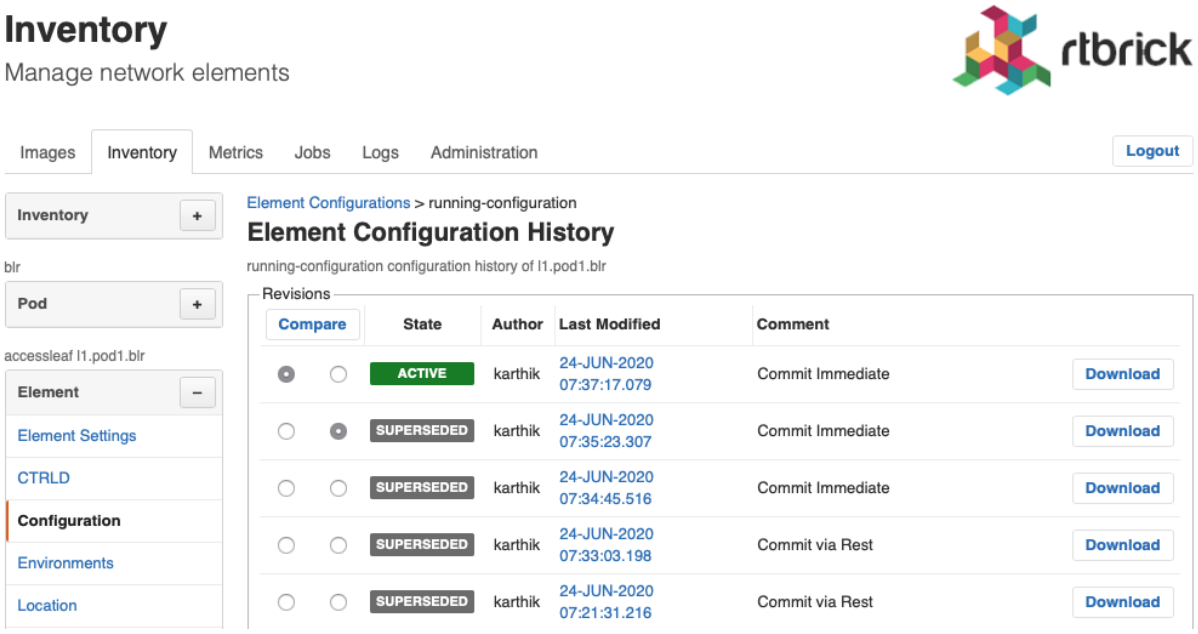
4. Click the element name for which you want to inspect configuration changes.

The element settings appears.

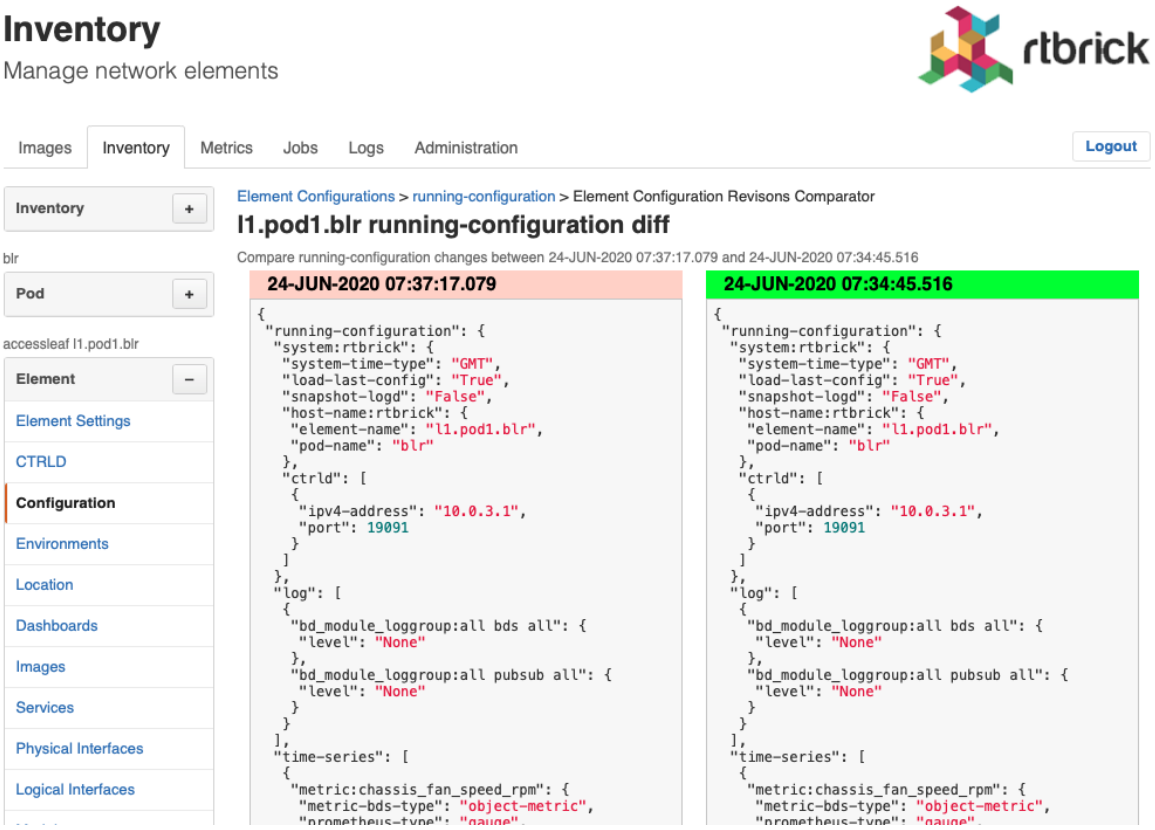
5. Click **Configurations**. The list of configurations appears.



6. Click the configuration name. The configuration history appears.



7. Select the two configurations you want to compare and click the **Compare** button. The diff viewer appears.



The diff viewer shows both configurations and the diff if you scroll down.

```
  {
   "physical:ifp-0/0/52": {
    "speed": "40G"
    "auto negotiation": true
   }
  }
```