

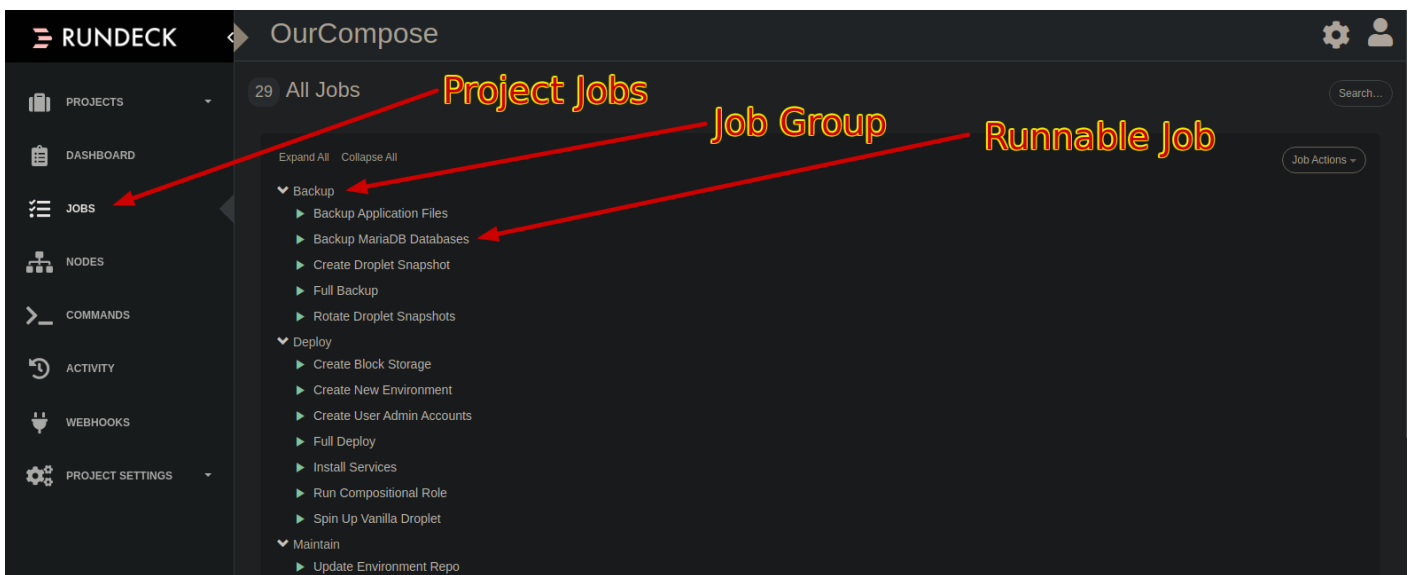
Everyday Usage

- [Jobs](#)
- [Activity](#)
- [Nodes and Commands](#)

Jobs

Main Jobs Page

<https://docs.rundeck.com/docs/manual/04-jobs.html>

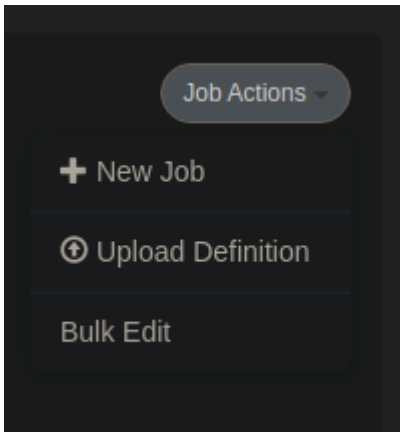


The "Jobs" link on the left-hand column will take us to a listing of all the jobs in this project. From there, all runnable jobs are displayed in alphabetical order in their Job Group.

NOTE: Jobs do not have to have a Job Group. If they don't, they will be displayed at the bottom underneath all of the job groups.

New Job

The top right "Job Actions" button will provide an option to create a new job:



There are only two required fields:

1. **Details > Job Name**
2. **Workflow > Step**

Everything else is optional

Details

- Job Name: Shows up on main jobs page
- Job Group: Section on main jobs page
- Description: Shows up as small text under job on main jobs page

Workflow

<https://docs.rundeck.com/docs/manual/job-workflows.html>

This spells out the steps and variables to use for the job.

NOTE: Every change in each of the following sections has to be saved separately before the workflow itself can be saved.

Options

<https://docs.rundeck.com/docs/manual/job-options.html>

Arguments/variables that are set when the job is run. Notable options are:

- Option Name: The name you will use to pass it to the steps
- Option Label: The prompt you will see when you go to run the job
- Input Type: One of Text/Date/Secure (passwords)
- Allowed Values: comma-separated list

- Restrictions: Can enforce from the Allowed Values list

Workflow Control

Several options pertaining to the workflow itself are present separate from the steps:

- If a step fails either stop immediately or run the rest before failing
- Strategy: Node First/Parallel/Sequential
 - See the "Explain" menu for more details
- Global Log Filters: <https://docs.rundeck.com/docs/manual/log-filters/>
 - Mask Passwords
 - Key/Value Data

Steps

Here are where the actual executions are structured. There are a lot of options and ways to construct this, so I'll only lay out what I'm familiar with:

- Node Steps: Runs once for every node
 - Command: Run a command from the command line
 - Script: Run a shell script
 - Job Reference: Run another job
- Workflow Steps: Runs once per workflow
 - Job Reference: Run another job
 - Global Variable: Set global variables to be used in other steps or notifications
 - Flow control: *Halt* indicates that the execution should halt. Enter a *Status* to halt with a custom status string. Otherwise, enable *Fail* to exit with failure, or not to exit with success.

NOTE: This is where plugins will usually be adding functionality.

Nodes

Remote nodes or locally. We always execute locally.

Schedule

- Run Repeatedly: Actually schedule when you want it to run
- Enable Scheduling: Turning the repeat runs on/off
- Enable Execution: Allow/Disallow ANYONE to run it at all

Notifications

<https://docs.rundeck.com/docs/manual/jobs/job-notifications.html>

- On Start
- On Success
- On Failure
- On Retryable Failure
- Average Duration Exceeded

Other

- Log Level: Debug level produces more output
- Multiple Executions: Allow this Job to be executed more than one time simultaneously
- Limit Multiple Executions: Max number of multiple executions
- Timeout
- Retry
- Default Tab: Nodes/Log Output/HTML

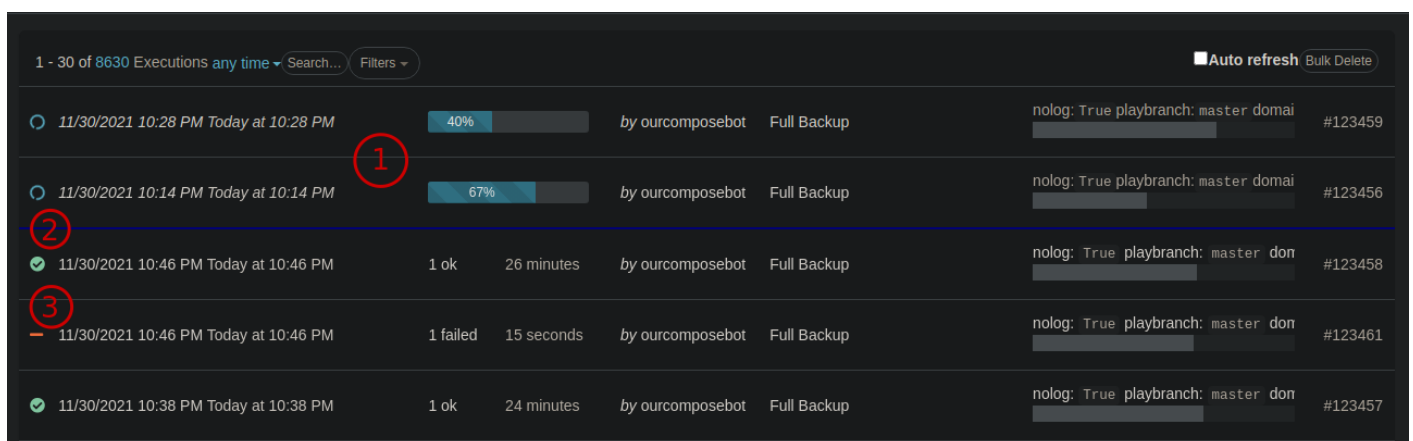
FWIW, the Log Output is very handy when executing locally to simply jump directly to the output.

Activity

<https://docs.rundeck.com/docs/manual/08-activity.html>

Execution history for commands and Jobs is stored by the Rundeck server. Execution history can be filtered and viewed inside the "Activity" page.

Execution State



1 - 30 of 8630 Executions any time Search... Filters							Auto refresh Bulk Delete
🔄	11/30/2021 10:28 PM Today at 10:28 PM	40%		by ourcomposebot	Full Backup	nolog: True playbranch: master domai	#123459
🔄	11/30/2021 10:14 PM Today at 10:14 PM	67%		by ourcomposebot	Full Backup	nolog: True playbranch: master domai	#123456
✅	11/30/2021 10:46 PM Today at 10:46 PM	1 ok	26 minutes	by ourcomposebot	Full Backup	nolog: True playbranch: master don	#123458
❌	11/30/2021 10:46 PM Today at 10:46 PM	1 failed	15 seconds	by ourcomposebot	Full Backup	nolog: True playbranch: master don	#123461
✅	11/30/2021 10:38 PM Today at 10:38 PM	1 ok	24 minutes	by ourcomposebot	Full Backup	nolog: True playbranch: master don	#123457

1. Currently Running Jobs
2. Successful Job (green check mark and "1 ok" text)
3. Failed Job (red dash, and "1 failed" text)

NOTE: The currently running jobs needs to be refreshed to get the latest update on this page, or the 'Auto refresh' box in the top right selected.

Selecting any of these executions will take you to their Log Outputs.

Searching for Prior Executions

Searching is most successful when using the "Name" field to search the name of the job you would like to find. The rest of the filters likely aren't going to be of much help. From there, you can look through the job executions to find the one you were looking for.

NOTE: There is currently no way to search by options (variables) passed.

Nodes and Commands

Nodes

<https://docs.rundeck.com/docs/manual/05-nodes.html>

A *Node* is a resource that is either a physical or virtual instance of a network accessible host. Nodes have a few basic attributes but a Node's attributes can be extended to include arbitrary named key/value pairs. Attributes typically describe the properties of a node or reflect the state of the node. One of a Node's built in attributes is called "tags" which is a list of classifications or categories about that Node.

NOTE: To get a listing of all nodes, place `. *` into the filter on top.

In the docker container, Rundeck will always list itself as a node. The executable directory is the home directory of the user that runs the Rundeck service, in docker's case, the `rundeck` user.

Node actions

The Node Actions menu contains links to

- Run a command. Choosing this menu item will forward your browser to the Commands page.
- Create a job. Choosing this menu item will forward you to the Job create page and enter the filter expression in the edit form.

Commands

<https://docs.rundeck.com/docs/manual/06-commands.html>

A *command* is a single executable string executed on a Node. Rundeck invokes commands on nodes via a *node executor* which evaluates the command string and executes it. Node executors evaluate the command string in a data context containing information about the Node resource. Command strings can reference this data and thus avoid hard coding node or environment specific values.

NOTE: The default *node executor* is `/bin/sh`

Simply type into the Command field what you would like to be ran in order to execute it on the nodes specified. This is going to be typically one-off commands or status retrieving commands. This is formatted in the same style as Job steps, and can use the same tricks.