Rundeck_Playgrounds

A repository that contains the IaC Code to Configure and Deploy an AWS AMI that contains Rundeck for testing purposes.

Important

First verify you have Terraform, Packer and the AWS CLI installed before running the project.

- Run the following command on your system: `packer --version`
- It should return the binary's version like such: `1.4.3`
- If you don't have Packer installed, please go to https://www.packer.io/downloads.html

---

- Run the following command on your system: `terraform --version`
- It should return the binary's version like such: `Terraform v0.12.16`
- If you don't have Terraform installed, please go to https://www.terraform.io/downloads.html

---

- Run the following command on your system: `aws`
- It should return an output similar to the following:

```
usage: aws [options] <command> <subcommand> [<subcommand> ...] [parameters]
To see help text, you can run:

  aws help
  aws <command> help
  aws <command> <subcommand> help
aws: error: the following arguments are required: command
```

- If you don't see the output, you don't have the `awscli` installed, please go to https://aws.amazon.com/cli/ and install the CLI.

Bootstraping the Project

- When `Packer`, `Terraform` and the `awscli` are available on your system please navigate directly into the desired playground's directory.
- `cd v3.2.0/` or `cd v1.4.4/`
- To run the whole project that includes building the AMI and the Infrastructure, run `./build-all` and it will take care of creating all the necessary files and steps, like creating the `.pem` file that gives you access to the running instance.

Rundeck v3 API

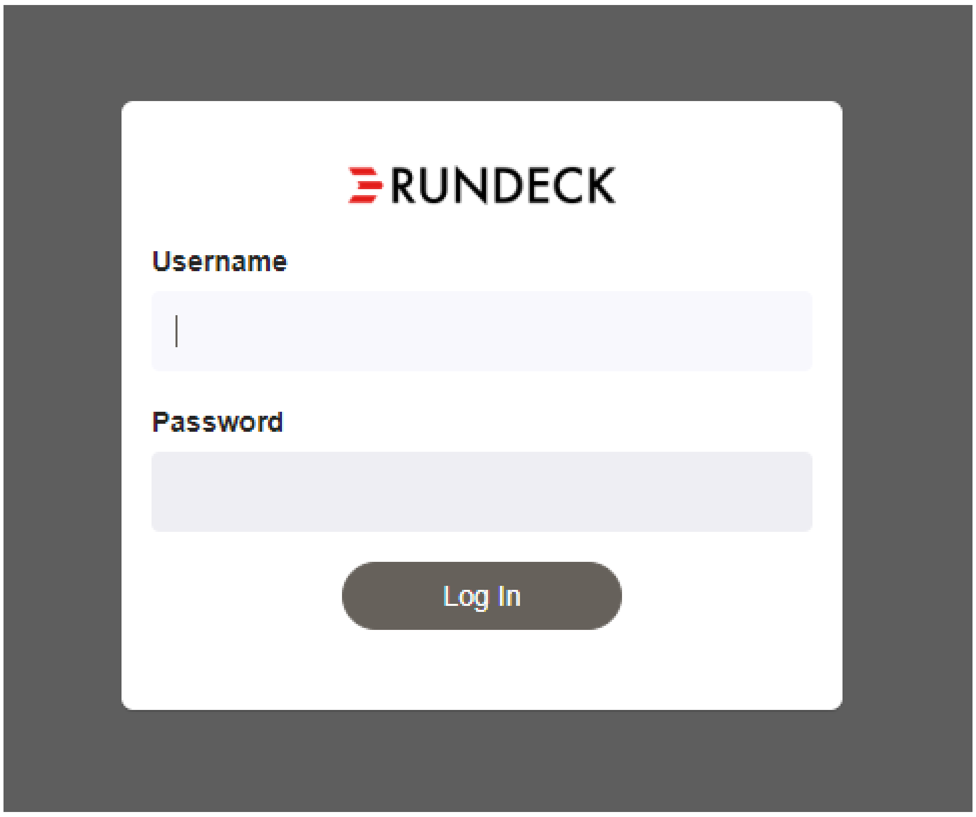
v3.2.0 - Obtaining a Rundeck Token (UI)

We need an **API** token, we can obtain it by following these steps:

v3.2.0 - Step 1

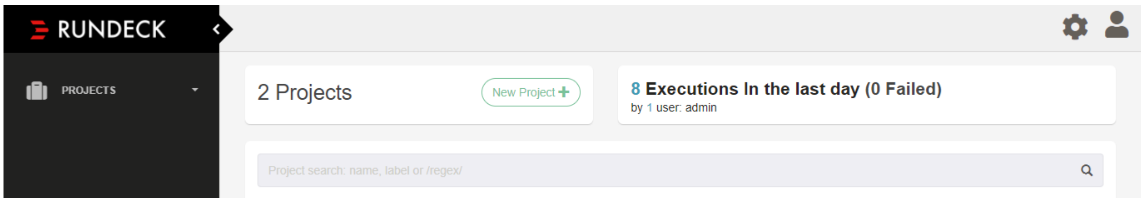
Login in your main page, the predeterminate credentials are:

```
Username: admin
Password: admin
```



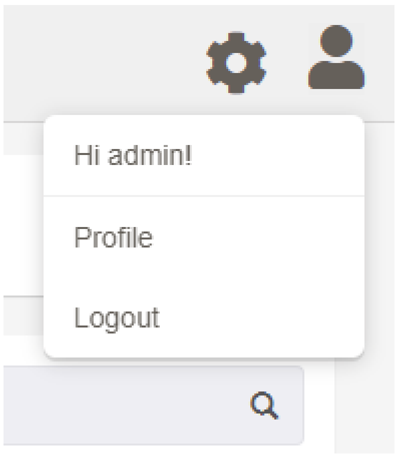
v3.2.0 - Step 2

Once logged in, we will go to the top right of the page and click on the user image.


RUNDECK
PROJECTS
2 Projects
New Project
8 Executions In the last day (0 Failed)
by 1 user: admin
Project search: name, label or /regex/

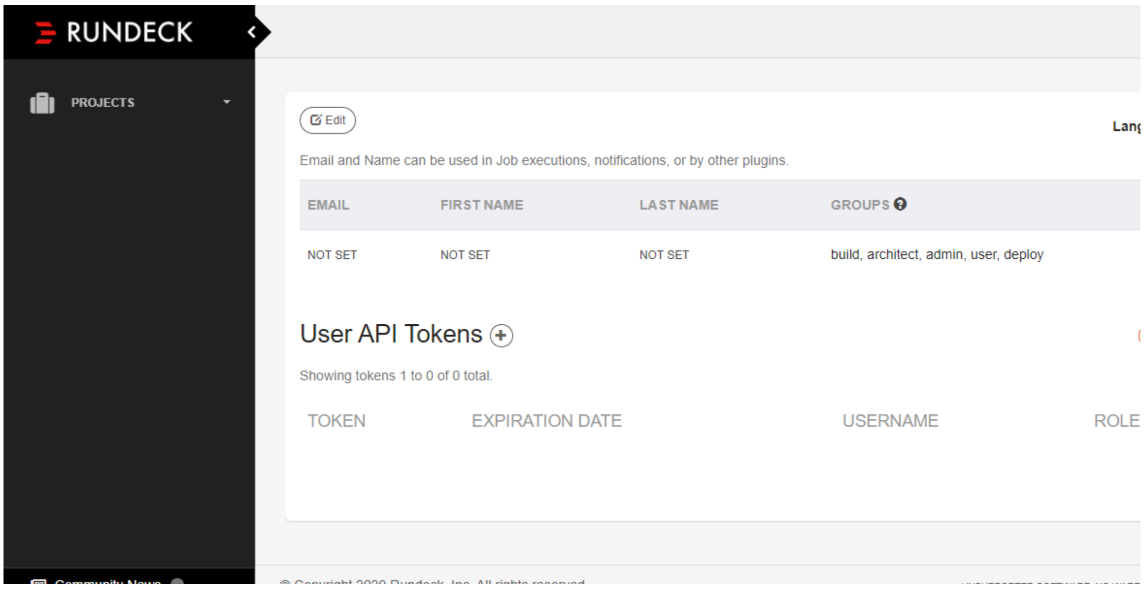
v3.2.0 - Step 3

Then we right click on the **"profile"** option.



v3.2.0 - Step 4

Once inside, we select the plus icon on the **"User API tokens"** section.



Email and Name can be used in Job executions, notifications, or by other plugins.

EMAIL	FIRST NAME	LAST NAME	GROUPS
NOT SET	NOT SET	NOT SET	build, architect, admin, user, deploy

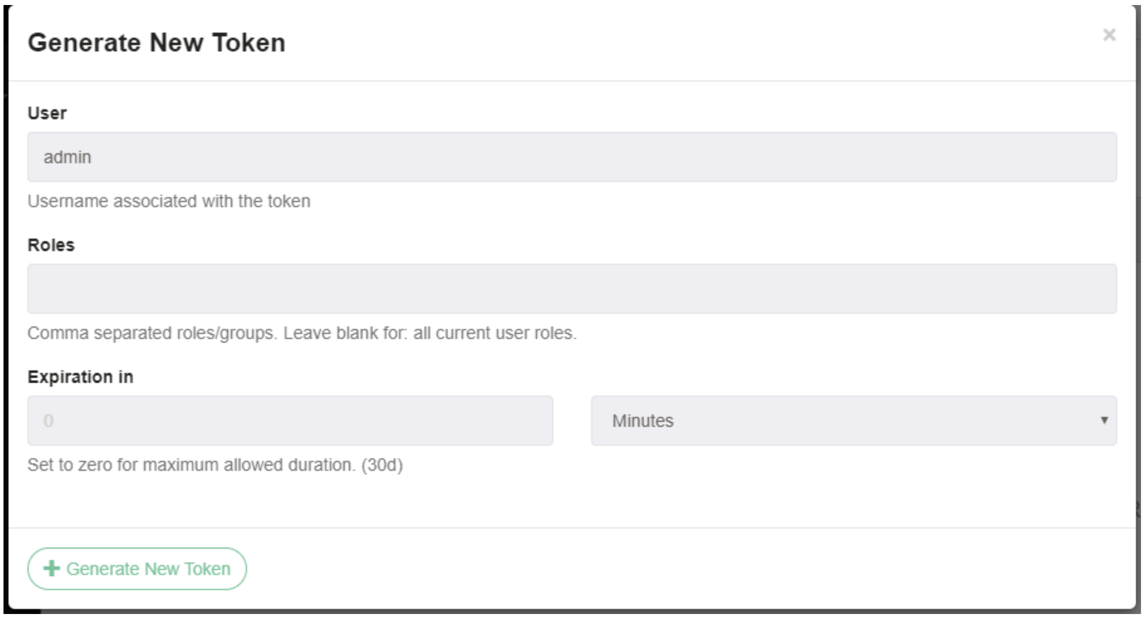
User API Tokens

Showing tokens 1 to 0 of 0 total.

TOKEN	EXPIRATION DATE	USERNAME	ROLE

v3.2.0 - Step 5

In this section we write the username to which we want to associate the token with, any roles or groups associated, and the expiration if it's needed. Once done, we click the **"Generate New Token"** button.



Generate New Token

User

admin

Username associated with the token

Roles

Comma separated roles/groups. Leave blank for: all current user roles.

Expiration in

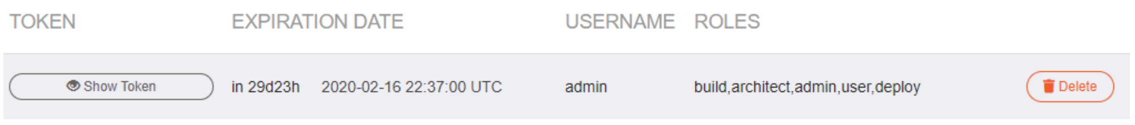
0 Minutes

Set to zero for maximum allowed duration. (30d)

v3.2.0 - Step 6

In the panel we click on the **"Show token"** button.

This will be our token to use de **API**.



TOKEN	EXPIRATION DATE		USERNAME	ROLES	
Show Token	in 29d23h	2020-02-16 22:37:00 UTC	admin	build,architect,admin,user,deploy	Delete

Create Project

In a terminal we can create a new project with the following command.

```
curl -vk -X POST $URL:4440/api/11/projects \
  --header "X-Rundeck-Auth-Token: $TOKEN" \
  --header "Content-Type: application/json" \
  --data-raw '{ "name": "$PROJECT_NAME" }'
```

Where the variables equate to the following values.

```
$URL="Your Rundeck URL."
$TOKEN="Your Rundeck Token obtained in the previous steps."
$PROJECT_NAME="The name of the project that you want to create."
```

Get information about our project

In a terminal we can obtain information about our project with the following command.

```
curl -vk $URL:4440/api/11/project/$PROJECT_NAME \
  --header "X-Rundeck-Auth-Token: $TOKEN" \
  --header "Content-Type: application/json"
```

Where the variables equate to the following values.

```
$URL="Your Rundeck URL."
$TOKEN="Your Rundeck Token."
$PROJECT_NAME="The name of the project that you want to create."
```

Importing a Job with an XML file

In a terminal we can create a Job with an XML file with the following command.

```
curl -vk http://$URL:4440/api/21/project/$PROJECT_NAME/jobs/import \
  --header "X-Rundeck-Auth-Token: $TOKEN" \
  -F xmlBatch=@"$FILENAME"
```

Where the variables equate to the following values.

```
$URL="Your Rundeck URL."
$TOKEN="Your Rundeck Token."
$PROJECT_NAME="The name of the project that you created."
$FILENAME="The name of your file with the configuration to create the Job."
```

Importing an example job

We can use this xml configuration to deploy an example job with the command "Hello World".

```
<joblist>
  <job>
    <defaultTab>nodes</defaultTab>
    <description>An example description</description>
    <executionEnabled>true</executionEnabled>
    <id></id>
    <loglevel>INFO</loglevel>
    <name>Example Job</name>
    <nodeFilterEditable>false</nodeFilterEditable>
    <scheduleEnabled>true</scheduleEnabled>
    <sequence keepgoing='false' strategy='node-first'>
      <command>
        <exec>echo "Hello World"</exec>
      </command>
    </sequence>
    <uuid></uuid>
  </job>
</joblist>
```

You can edit the Name of the Job, add more commands, or call scripts.

After importing the Job we will obtain an output that looks like this

```
<result success='true' apiversion='34'>
  <succeeded count='1'>
    <job index='1' href='http://$SERVER_IP:4440/api/34/job/$JOB_ID'>
      <id>f3e96a18-f516-4058-b24c-6293fc20c15d</id>
      <name>API</name>
      <group></group>
      <project>Test-Project</project>
      <permalink>http://$SERVER_IP:4440/project/Test-
Project/job/show/$JOB_ID</permalink>
    </job>
  </succeeded>
  <failed count='0' />
  <skipped count='0' />
</result>
```

In this output we can see an **"id"**, this is the ID of the job we just imported into Rundeck, we need it to run the Job after importing it.

Running the example Job

In a terminal we can run the example Job with the following command.

```
curl -vk -X POST $URL:4440/api/19/job/$JOB_ID/run \
  --header "X-Rundeck-Auth-Token: $TOKEN" \
  --header "Content-Type:text/xml"
```

Where the variables equate to the following values.

```
$URL="Your Rundeck URL."
$TOKEN="Your Rundeck Token."
$JOB_ID="The Job ID obtained when you imported the XML File into Rundeck"
```

Rundeck v1.4.4 API

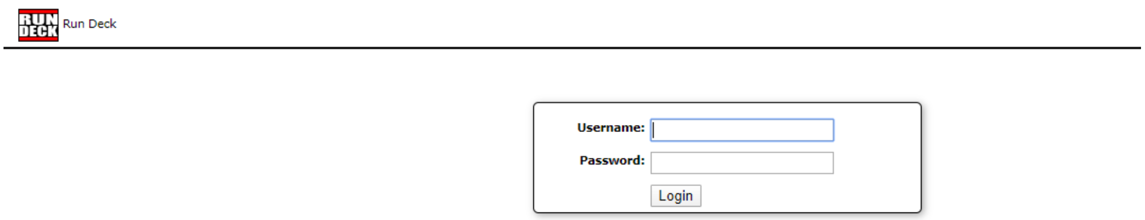
Obtaining a Rundeck Token (UI)

We need an **API** token, we can obtain it by following these steps:

Step 1

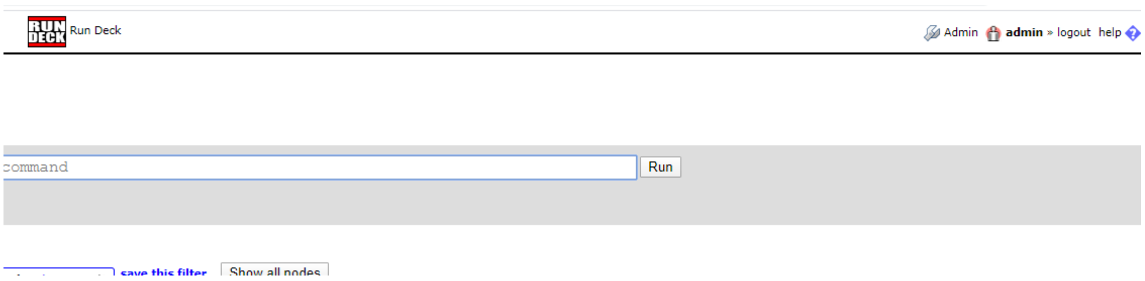
Login with the default credentials.

```
User:admin
Password:admin
```



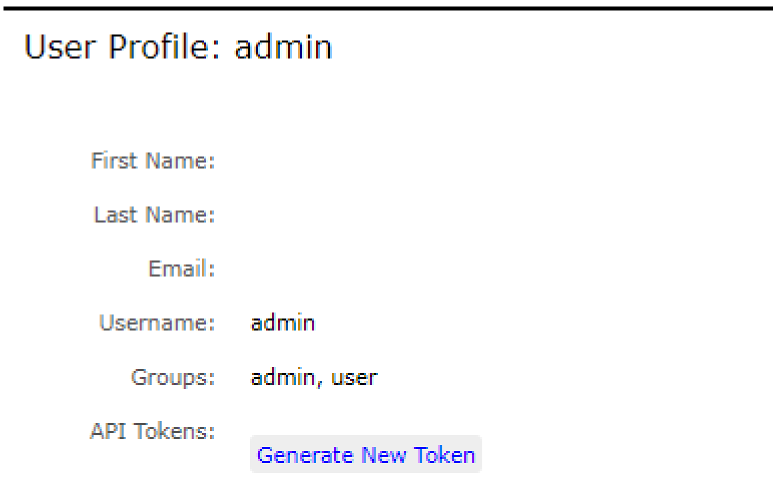
Step 2

Once logged in, we will go to the top right of the page and click on "admin".



Step 3

Once inside, we'll select the **"Generate New Token"** button.

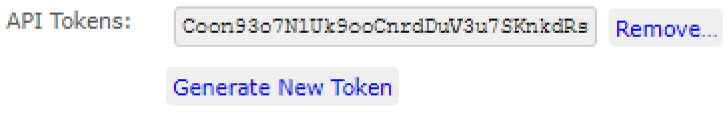


User Profile: admin

First Name:	
Last Name:	
Email:	
Username:	admin
Groups:	admin, user
API Tokens:	

Step 4

In the panel we can see the generated token, this will be the token we use to authenticate on the **API**.



API Tokens: Coon93o7N1Uk9ooCnrdDuV3u7SKnkdRs

Using de Rundeck v1.4.4 API

v1.4.4 - Create Project

In this version we can't create a project using the API.

v1.4.4 - List the projects

In a terminal we can list the projects with the following command:

```
curl -vk -X POST $URL:4440/api/5/projects \
  --header "X-Rundeck-Auth-Token: $TOKEN" \
  --header "Content-Type: application/json"
```

Where the variables equate to the following values.

```
$URL="Your Rundeck URL."
$TOKEN="Your Rundeck Token obtained using the previous steps."
```

v1.4.4 - Importing a Job with an XML file

In a terminal we can create a Job with an XML file with the following command.

```
curl -vk -L $URL:4440/api/5/jobs/import  \
  --header "X-Rundeck-Auth-Token: $TOKEN" \
  -F xmlBatch=@"$FILENAME"
```

With the following values:

Where the variables equate to the following values.

```
$URL="Your Rundeck URL."
$TOKEN="Your Rundeck Token."
$FILENAME="The name of your file with the configuration to create the Job."
```

We can use this xml configuration to deploy an example job with the command "Hello World"

```
<joblist>
  <job>
    <id></id>
    <loglevel>INFO</loglevel>
    <sequence keepgoing='false' strategy='node-first'>
      <command>
        <exec>echo "Hello World"</exec>
      </command>
    </sequence>
    <description></description>
    <name>job1</name>
    <context>
      <project>Test-Project</project>
    </context>
    <uuid></uuid>
  </job>
</joblist>
```

You can edit the Name of the Job, add more commands, or call scripts.

v1.4.4 - After importing the Job we will obtain an output that looks like this

```
<result success='true' apiversion='5'>
  <succeeded count='1'>
    <job index='1'>
      <id>2dfbea96-8eb9-46f3-a185-fd716eb9d6c8</id>
      <name>job1</name>
      <group></group>
      <project>ONE</project>
      <url>/job/show/2dfbea96-8eb9-46f3-a185-fd716eb9d6c8</url>
    </job>
  </succeeded>
  <failed count='0'></failed>
  <skipped count='0'></skipped>
</result>
```

In this output we can see an **"id"**, this is the ID of the job we just imported into Rundeck, we need it to run the Job after importing it.

Run the Job

v1.4.4 - Running the example Job

```
curl -vk -X POST $URL:4440/api/1/job/$JOB_ID/run \
  --header "X-Rundeck-Auth-Token: $TOKEN" \
  --header "Content-Type:text/xml"
```

Where the variables equate to the following values.

```
$URL="Your Rundeck URL."
$TOKEN="Your Rundeck Token."
$JOB_ID="The Job ID obtained when you imported the XML File into Rundeck"
```