

# Frequently Asked Questions

## What Blueprints are available for the Regional Controller?

Presently there is only a Blueprint for the Radio Edge Cloud member of the Telco Appliance Blueprint Family. However, Blueprints are being developed for the Rover and Unicycle projects in the Network Cloud Blueprint Family.

Akraino Edge Stack Project	Blueprint Location
Radio Edge Cloud (Telco Appliance blueprint family)	<a href="https://gerrit.akraino.org/r/#/admin/projects/rec">https://gerrit.akraino.org/r/#/admin/projects/rec</a>
Rover (Network Cloud Blueprint Family)	TBD
Unicycle (Network Cloud Blueprint Family)	TBD

## What are the URL Endpoints of the Regional Controller?

URL	Purpose
<a href="https://&lt;RC_IP_address&gt;/api/v1/">https://&lt;RC_IP_address&gt;/api/v1/</a>	This is the base URL for all of the API calls to the Regional Controller.
<a href="https://&lt;RC_IP_address&gt;/docs/">https://&lt;RC_IP_address&gt;/docs/</a>	This is the address of an online copy of the documentation for the Regional Controller. Note that because this documentation is built along with the codebase, it is likely to be more up to date than this wiki.
<a href="http://&lt;RC_IP_address&gt;:8080/admin/">http://&lt;RC_IP_address&gt;:8080/admin/</a>	This is the console for the embedded copy of Apache® Airflow that is used as the workflow engine within the Regional Controller.

## Where is the full API documented?

As it is still being modified and in a state of flux, the full API is documented only in the online copy under [https://<RC\\_IP\\_address>/docs/api.html](https://<RC_IP_address>/docs/api.html)

Alternately, you can clone the gerrit repository for the Regional Controller, and look at the markup files under `/src/site/docs/`.

## What are the Predefined Users in the Regional Controller database?

The regional controller will be installed with the following pre-defined users. You are not required to use these usernames and can remove/replace them as you wish; however, the **workflow** user is used internally to log workflow events, so you would be well advised to leave that user alone.

Username	Password	Roles	Purpose
admin	admin123	create/read/update/delete all	This is the "super user" for RC; e.g. it has all roles and capabilities.
readonly	admin123	read all	This user has read only access; e.g. it can issue GET requests, but cannot change objects or make new objects.
workflow	admin123	read all, create pdevents	This user is used internally in order to create POD events from within the workflows.
noaccess	admin123	(none)	This user has no access at all.

## How are Workflows initiated?

All workflows are started as a result of operations on PODs:

- **POST** - creates a POD and causes the *create* workflow to run
- **PUT** - updates a POD and cause one of several update workflows to run
- **DELETE** - deletes the POD and causes the *delete* workflow to run

The details of the workflow are described in the Blueprint for the POD.

## How do I look at the Workflow Logs?

The workflow logs are stored within Airflow. As such, you need to access them via Airflow. There are two ways to do this:

1. Use the Airflow console (see above for the URL).
  - a. Click on DAGs
  - b. Click on the individual DAG you are interested in. The name should contain both the workflow name and the UUID of the associated POD.
  - c. Click on Graph View
  - d. Click on the box for *maintask*
  - e. In the popup sub-window, click on *View Log*

2. Look at the logfile directly by following these steps:

- a. Login to the host where the regional controller is running
- b. Search within the *arc-airflow-worker* container for the log, and then *cat* it.

```
c. $ UUID=<UUID of the POD>
   $ docker exec arc-airflow-worker find /usr/local/airflow/logs -type f | grep $UUID/maintask
```

## How can I determine the current state of a POD?

The current state of any POD can be determined by using the *rc\_cli* command to look at the current POD state. For example:

```
$ ./rc_cli -H rc -p admin123 pod show 18c3e63e-8847-4d97-a172-3a48b0ffa069
---
blueprint: 690450c0-776a-11e9-ae9b-f3cee2e49e42
description: CD of blueprint on OE
edgesite: 60ab1298-7769-11e9-92b3-373d9b2f2476
events:
- level: INFO
  message: Pod created.
  time: '2019-06-12 16:56:22.0'
- level: INFO
  message: 'Starting workflow: create'
  time: '2019-06-12 16:56:22.0'
- level: INFO
  message: 'Workflow directory created: $DROOT/workflow/create-0-18c3e63e-8847-4d97-a172-3a48b0ffa069'
  time: '2019-06-12 16:56:22.0'
- level: INFO
  message: 'Workflow fetched: http://www.example.org/blueprints/work-flow-v0.1.sh'
  time: '2019-06-12 16:56:22.0'
- level: INFO
  message: Workflow template created.
  time: '2019-06-12 16:56:22.0'
- level: INFO
  message: Starting create workflow for POD 18c3e63e-8847-4d97-a172-3a48b0ffa069
  time: '2019-06-12 17:00:18.0'
name: pod_on_oel
state: WORKFLOW
url: /api/v1/pod/18c3e63e-8847-4d97-a172-3a48b0ffa069
uuid: 18c3e63e-8847-4d97-a172-3a48b0ffa069
workflows:
- create
yaml:
  input_yaml: http://www.example.org/mtlab/aknode201-user_config.yaml
  iso_primary: http://www.example.org/iso/latest/install.iso
  iso_secondary: http://www.example.org/iso/latest/bootcd.iso
```

The status filed in the output shows the current status of the POD; **WORKFLOW** means that a workflow is currently running, and **ACTIVE** means the POD is active with no current workflow.

## How are User passwords stored in the Database?

They are not stored in clear text; rather a SHA-256 hash is stored. If you want to change the default passwords, or add your own users, you need to store the SHA-256 hash of the password in the PWHASH field in the database. One way to generate this hash is:

```
$ echo -n "$PASSWORD" | openssl sha256 | tr '[:lower:]' '[:upper:]'
(STDIN)= 240BE518FABD2724DDB6F04EEB1DA5967448D7E831C08C8FA822809F74C720A9
```