MAT JOURNALS

Journal of Signal Processing
Volume 5 Issue 2

Airflow Directed Acyclic Graph

Shubha B G[1], A.M.Prasad[2]
[1]PG Student, [2]Associate Professor
[1,2]Department of CSE, Dayananda Sagar College of Engineering, Bengaluru, Karnataka, India
Email: *shubhagopal27@gmail.com*
DOI: *http://doi.org/10.5281/zenodo.3247274*

Abstract

The paper "Airflow Dags" is based on the generation of directed acyclic graphs during the creation of profiles and roles in the Enterprise management software. Profiles are facilities that are provided to the clients in organization, they are seen in the configuration view of enterprise software. Roles are the permissions and the access that is given to a client. Automation is necessary to make repetitive and difficult procedures simpler. In this case the automation is done so that the onboarding of new clients in organization is done in an easier way. Previously this was done manually which was time consuming, the development of this project has made the process simpler. It parses the data from the pod directory json file to generate client list, pod configurations. These are then stored in a config which is made for each of the pods. The details are then taken from this repository as the details cannot be hard coded into the code. This makes the program seem more discrete.The project work recorded below shows the implementation using an open source software called airflow which is used to generate directed acyclic graph.

Keywords: *Automation, airflow, enterprise management software, profiles, roles*

INTRODUCTION

The success of a novel emerging technology could be proven as it is used in a pervasive way by common people having no scientific or technical knowledge about how it works under the hood.Airflow is a platform to author, schedule and monitor workflows. It is used to author workflows as a directed acyclic graph for a particular task. This helps us get an idea of the workflow of the code and all the dependencies that follow it. The command line utilities are rich which make running complex directed acyclic graph simple.In arithmetic, especially diagram hypothesis, and software engineering, a coordinated non-cyclic is a limited coordinated chart with no coordinated cycles. That is, it comprises of limitedly numerous vertices and edges (likewise called arcs), with each edge guided starting with one vertex then onto the next, to such an extent that there is no real way to begin at any vertex x and pursue a reliably coordinated grouping of edges that in the end circles back to x once more.

Proportionally, a DAG is a coordinated diagram that has a topological requesting, an arrangement of the vertices withend goal that each edge is coordinated from before to later in the grouping.

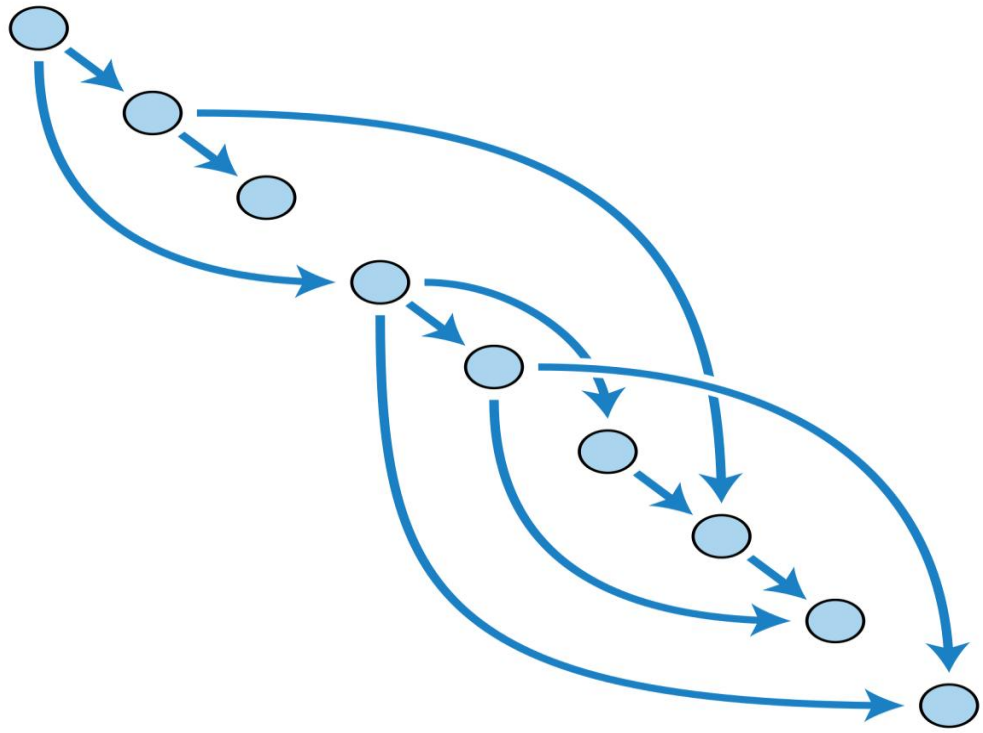

Figure 1: *Airflow.*

The ascent of AI methods, over all the profound neural system, unchained the intensity of huge information examination and, in the meantime, accelerated the need to oversee registering control in a direct manner.The pods have to be authenticated each time the enterprise needs to be created. Hence the details have to be parsed. Once all the values have been generated, we need to store them in data structures. The LDAP information is generally stored in a domain. All the authentication is done via these domains in which the pods are present. The pods are pulled out from the json file by generating the list of all clients which in turn has the key to pull out the pods they belong to.Once the parsing is done, we are passing the values to the command that is to be passed in the python code to run the jar files.We pass the information in the command which in turn creates the profiles and roles based on which version we use. Profiles are the facilities provided by the organization to client. Roles are the permission or service levels allocated to the client based on the requirement.The code is typed in python and stored in a "yaml Ain't Markup Language" file which will be executed by Jetstream.The entire task of execution takes place with the help of AIRFLOW by 'Airbnb' with a layer of our own customized automation software called Jetstream which is used to schedule and run tasks.AIRFLOW is an open source software under Apache which is widely used to execute and keep track of the performance of a task that is automated, that is monitoring workflow.Here it is used to author workflows as a directed acyclic graph for a particular task.

RELATED WORK

There is a plenty of accessible work process motors committing their highlights to empower the client to run complex applications including up to a large number of jobs on various computational assets [1]. Among the latest writing there are various Python-based devices and libraries intended to help the work processes the board as PaPy [2] a parallel

and circulated information preparing pipelines, PyCOMPSs [3] a structure to encourages the improvement of parallel computational work processes, Fireworks a dynamic work process framework intended for high throughput applications and DASK [4] a parallel registering library intended for parallel examination. The administration procedure of every work process motor is entirely comparable [5].A structure for planning calculations having intertask conditions—displayed through coordinated non-cyclic diagrams (DAGs)— for Internet-based registering [6]. Directed cyclic graph (DAGs) applies to an enormous class of (transient) design acknowledgment issues and other acknowledgment issues where the information has a direct requesting.

The information streams are coded (DAG-coded) into DAGs for vigorous division. The similitude of two streams can be showed as the way coordinating score of the two relating DAGs. This work likewise shows a proficient and powerful unique programming calculation for their correlations (DAG-look at). Since the DAG-coding strategy straightforwardly gives a strong division process, it tends to be connected recursively to make a novel framework engineering. The DAG structure additionally permits versatile rebuilding, prompting a novel way to deal with neural data handling. DAG-coding may likewise be connected to discourse acknowledgment or whatever other nonstop streams where a powerful multipath division helps the acknowledgment procedure [7].The project is done to generate the profiles and roles in management enterprise software. Enterprise management software helps us keep track of all the data collections from all the servers. It also acts as a tool to add new nodes to client profile and make changes to their profiles based on their requests. Enterprise management software also helps in monitoring the systems and add new infrastructure if necessary.Roles are mostly accessing and permissions that are given to a particular client. The roles will be set to the default levels until the client requests for further access on validation. We can't give the clients more permissions than what we have for ourselves. These are the default settings put up on each client.Work process motors assume an essential job in computational sciences in light of the fact that the accessibility of cloud gave flexible and for all intents and purposes vast computational assets. In this situation, the calculation orchestrator goes about as foundation looking for execution, moderateness, dependability, accessibility and, most importantly,reproducibility of any computational analysis.We utilize an open source Python library and related segments empowering the execution of directed acyclic graph (DAG) occupations on anything, extending from the neighborhood machine to virtual HPC bunches facilitated on private, open or cross breed mists. Wewill allude to a hub of the DAG (which speaks to the work process) as assignment. Eacherrand must bewrapped by an execution envelope, guaranteeing the meaning of a sandbox in which the product must be executed. The assignment devours input information and produces yield information. The undertakings are amassed utilizing a particular scripting language.The democratization of computational assets, on account of the approach of open, private, and half and half mists, changed the standards in numerous science fields. For a considerable length of time, one of the exertions of PC researchers and PC engineers was the advancement of devices ready to rearrange access to top of the line computational assets by computational researchers. Be that as it may, these days any science field can be considered "computational" as the accessibility of incredible, however simple to oversee work process motors, is crucial. In this work, they present DagOn* (Direct non-cyclic chart on anything), a lightweight

Python library actualizing a work process motor ready to execute parallel occupations spoken to by direct acyclic graph on any of neighborhood machines, on-premise elite figuring groups, compartments, and cloud-based virtual foundations. We utilize a certifiable creation level application for climate and marine estimates to show the utilization of this new work process motor [8]. In software engineering and science, a directed acyclic graph (DAG) is a diagram that is coordinated and without cycles associating different edges. This implies it is difficult to navigate the whole diagram beginning at one edge. The edges of the coordinated chart just go one way. The chart is a topological arranging, where every hub is in a specific request.For the assessment of the system requirements with relation to the technical expertise will be considered with respect to the technical feasibility study where all aspects of handling the technical requirements and problems based on a detailed examination will be considered. We will include all the factors related to the authentication and the hierarchy that has to be defined for the usage of different assets provided by the system even into the consideration of multiple logins through multiple client structure at the same time on individual basis. The human factor will be associated for the required technical requirements as multi functionalities will be integrated with the system to promote overall system where all different business ventures can be managed, and graph outline can be produced. The respective outline information that is required in automated way is required to be arranged in theform of model information making it better for the visualization. All the related opportunities required for the development and for the management will be outlined with respect to the operations as we want that all the processes that have been designed can be organized properly either with respect to the development of with respect to the implementation. The scenarios that can be important for the effective corporate culture and client management time of implementation and working is required to be organized properly.At the time of operations individual stakeholder that will provide the resources will be considered for the eventual real time help that will be required by theclients and sometimes it can be confusing to maintain the understandability in respect to the resource that has been Integrated so we have to define all the related plan to counter and manage the requirements.

PROPOSED WORK

The proposed system is an automation of profiles and roles creation. Automation is necessary to make repetitive and difficult procedures simpler. Automation is the technology by which a process or procedure is performed with minimal human assistance. There are multiple modules to play in this work which are configurations, result export task, management console execute task, pod filter, JSON parsing, enterprise filter task.

Configurations contain all the information to get authenticated to a pod which is stored in these modules. Result export task contains the output of the json parsing and the output of the program is in this module. Management console execute task is where the parsed values from the configs and the pod directory json are used inside command that is run using subprocess command. Pod filter is where the list of pods that are generated match the once in the config. Thus, getting rid of the execution of unnecessary tasks.

Json parsing is done from the json file. Some of the details from the pod directory are stored in the configurations.

Enterprise filter task is used to filter out all the client mnemonics from the Json file.
The workflow is shown below.

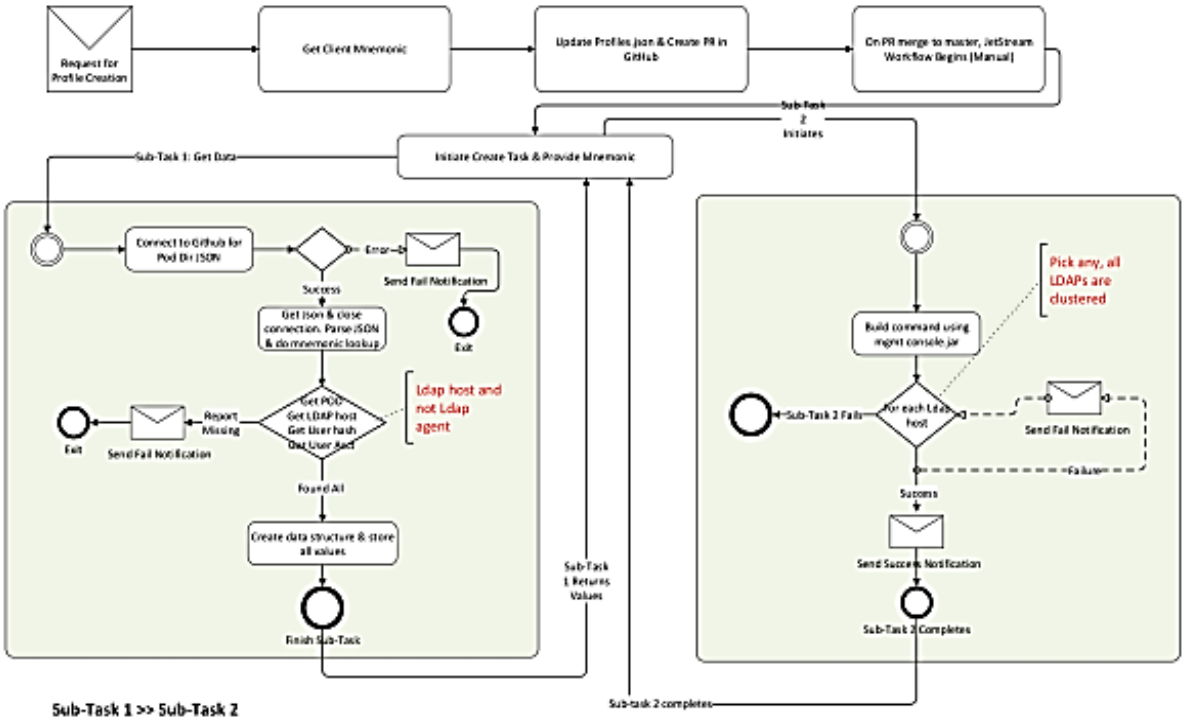

Figure 2: *Profile creation workflow.*

In the process of new client profile creation, we follow the below procedure:
When the request for the profile creation arrives to the organization, first get the client mnemonic where the client mnemonic has a particular pattern it has to be generated based on the lightweight directory active protocol and the region it belongs to. Once we client mnemonic is obtained update the client mnemonic in json file which is present in a particular repository in GitHub and alsocreate the pull request inGitHub.Once the pull request is created wait for it to get it merged into master repository and once it gets merged the Jetstream workflow begins. At first, initiate the task creation and provide the client mnemonic. Here the subtask is to get the above data and then connect to json file which is in GitHub. If it fails to get the json from GitHub it sends fail notification and exits. If it is successful, then it gets thejson file and closes the connection to GitHub. It will parse the json and do the client mnemonic lookup. Then it gets the information like lightweight direct access protocol host, pod name, user hash, user account from the json file. If it fails to get the following information or the report is missing, then it sends the fail notification and exits. If it finds all the information, then it creates the data structure and stores the values. The first subtask is completed here which outputs the information like ldap, user hash, user account and the pod name. This information is used as an input to the second subtask. The second subtask is initiated when all this information is received. Then create a command to generate the profile for a particular client using management console.jar file.When run this command if it is not able to fetch the ldap host then it sends a fail notification and exits. For each ldap host if the enterprise is not created then also it triggers the failure and sends a fail notification and exists. If it is successful, then the profile is successfully created for particular client and it sends the success

notification and exits.

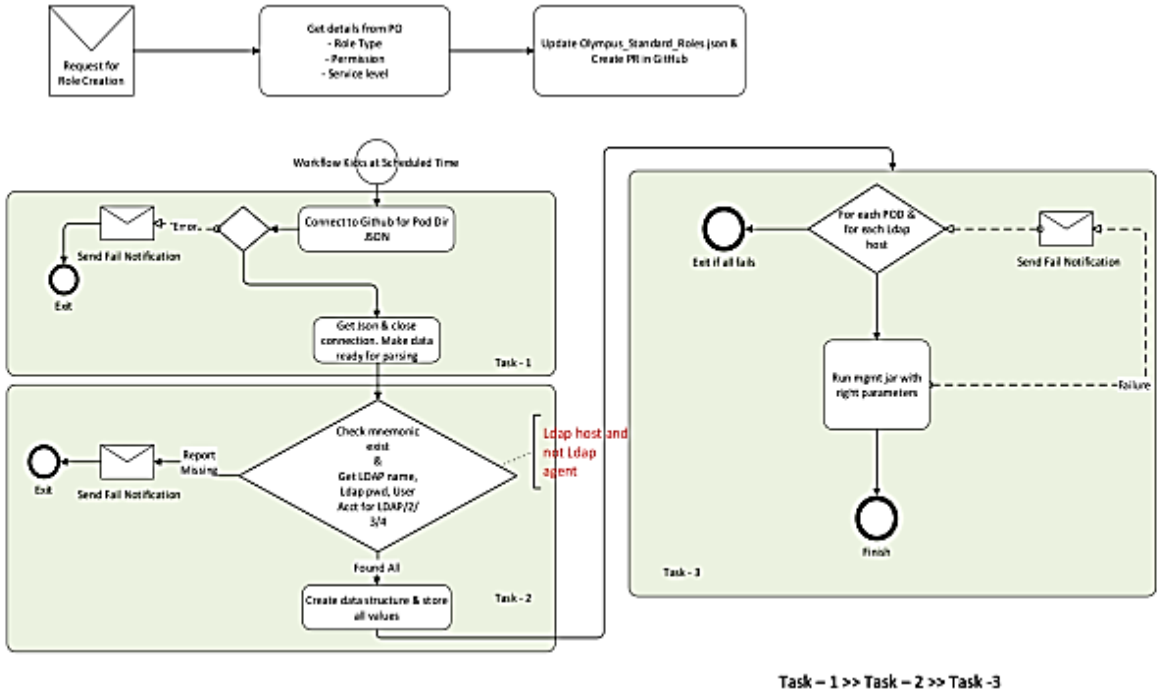

Figure 3: *Roles creation workflow.*

In the process of new role creation, we follow the below procedure:

When the request for role creation arrives to the organization, fetch the details from json file by connecting to GitHub. From json file get details such as the role type, permission and the service level for a particular client. The permission level and the service level for each client differs. After obtaining the above information, update the client mnemonic in json file which is present in a particular repository in GitHub and also, will create the pull request in GitHub. This process runs as a cron job. The product utility cron is a time sensitive activity scheduler in Unix-like PC working frameworks. Individuals who set up and keep up programming conditions use cron to plan occupations to run occasionally at fixed occasions, dates, or interims. At first, try to connect to GitHub to fetch json file. If it fails to get the json from GitHub it sends fail notification and exits. If it is successful, then it gets the json file and closes the connection to GitHub. It makes the data ready for parsing. Then it gets the information like ldap host, pod name, user hash and user account from the json file. If it fails to get the followinginformation or the report is missing, then it sends the fail notification and exits. If it finds all the information, then it creates the data structure and stores the values. This information is used as an input to the second subtask. The second subtask is initiated when all this information is received. Then create a command to generate the rolefor a particular client using management console.jar file.When run this command if it is not able to fetch the ldap host then it sends a fail notification and exits. For each ldap host if the enterprise is not created then also it triggers the failure and sends a fail notification and exists. If itis successful, then the role is successfully created forparticular client and exits.

RESULTS

In software engineering and arithmetic, a Directed acyclic graph (DAG) is a chart

that is coordinated and without cycles associating different edges. Thisimplies it is difficult to cross the whole chart beginning at one edge. The edges of the coordinated diagram just go one way. The diagram is a topological arranging.

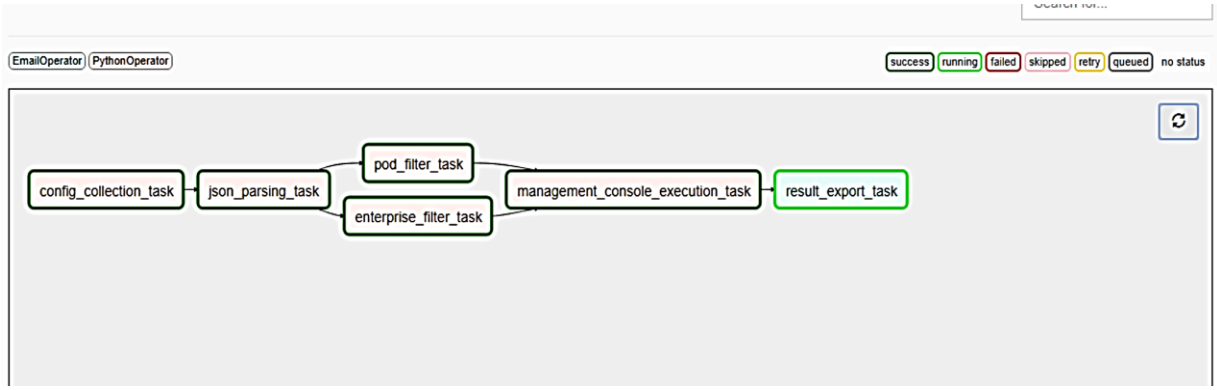

Figure 4: *Screenshot of running DAG.*

As shown in Fig.4 the directed acyclic graph is created for running task. It is decided as a running DAG based on the colors. As we can see different stages are represented with different colors in airflow which is an apache open software. By using airflow DAG's we can also see the dependencies between the modules we are executing. As shown in figure parsing task is dependent on config collection task, pod filter task and enterprise filter task isdependent on parsing task. Management console execution task is dependent on pod filter task and enterprise filter task whereas the result export task is dependent on management console execution task.

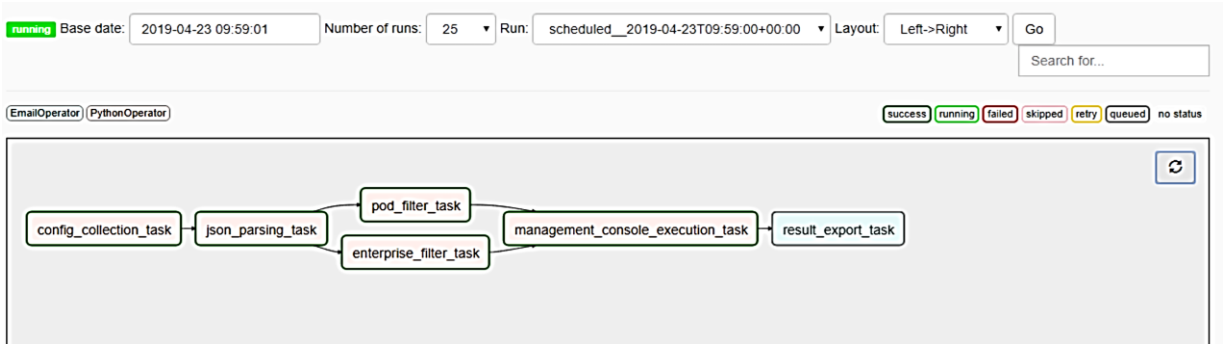

Figure 5: *Screenshot of queued DAG.*

The above figure shows the DAG created for queued job. The result export task is queued. With the help of DAG, it is simpler to identify the dependencies and the status of execution.

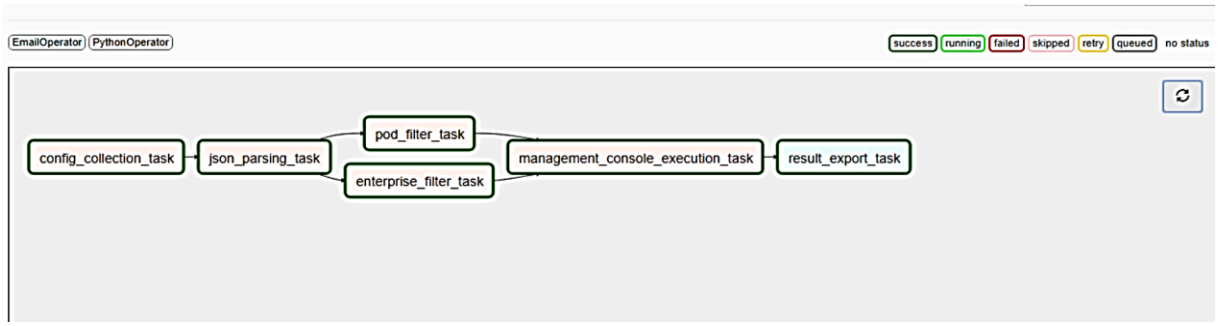

Figure 6: *Screenshot of success DAG.*

The above Fig. 6 shows the DAG created for success job. All the modules are executed without any error.

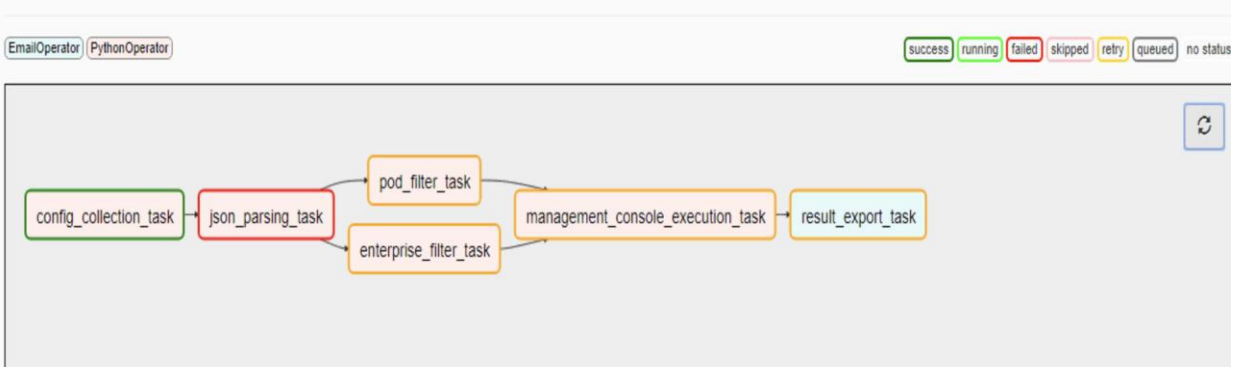

Figure 7: *Screenshot of retry DAG.*

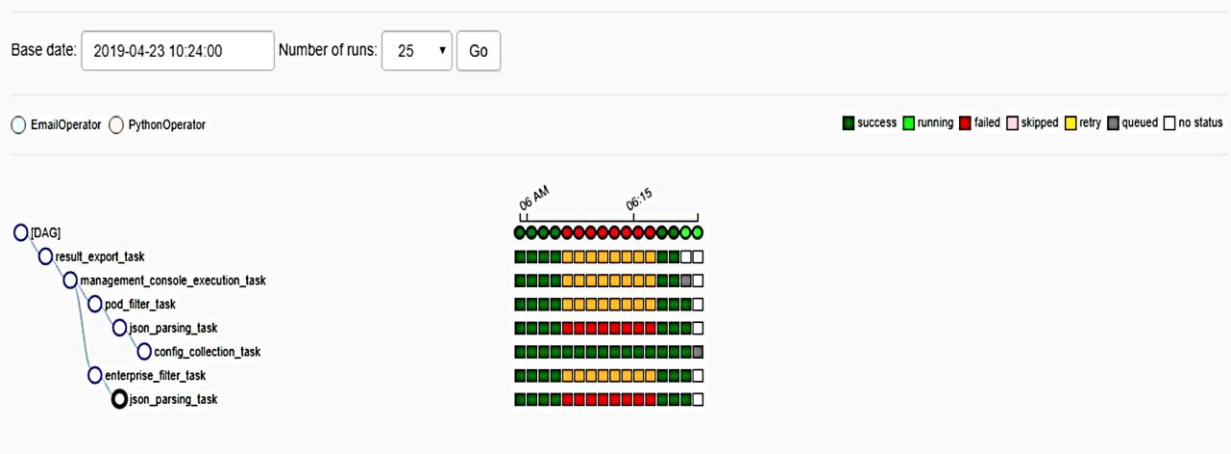

Figure 8: *Tree view of DAG.*

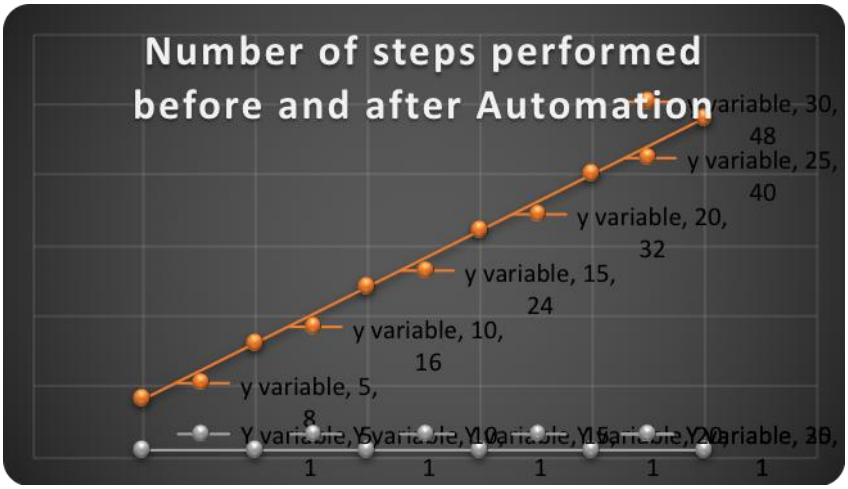

Figure 9: *Number of steps performed before and after automation.*
X axis- Number of clients
Y axis – Number of steps performed

CONCLUSION

AIRFLOW is used to know the workflow path taken by a particular task that is being run by representing it as a DAG. This project involves creation of profiles and roles by using airflow and Jetstream. The

automation of Olympus profiles and roles helps us to reduce the client on boarding time whenever a new client has been added. This helps in speeding up the set-up of the infrastructure for the clients. The added benefit of executing the client list every day and adding of new pods in config makes sure that the code remains the same, but the functionality can be upgraded at any given point of time.

REFERENCES

1. A. Barker, J. Van Hemert (2007), "Scientific workflow: a survey and research directions", *International Conference on Parallel Processing and Applied Mathematics. Springer*,pp. 746–753.
2. M. Cieslik, C. Mura (2014), "Papy: Parallel and distributed data-processing pipelines in python", arXiv preprint arXiv: 1407.4378.
3. E. Tejedor, Y. Becerra, G. Alomar, A. Queralt, R. M. Badia, J. Torres, T. Cortes, J. Labarta (2017), "Pycompss: Parallel computational workflows in python," *The International Journal of High-Performance Computing Applications*, Volume 31, Issue 1, pp. 66–82.
4. M. Rocklin (2015), "Dask: Parallel computation with blocked algorithms and task scheduling",*Proceedings of the 14th Python in Science Conference*, pp. 130−136.
5. L. Liu, M. Zhang, Y. Lin, L. Qin (2014), "A survey on workflow management and scheduling in cloud computing", *Cluster, Cloud and Grid Computing (CCGrid), 2014 14th IEEE/ACM International Symposium on*. IEEE, pp. 837–846.
6. Gennaro Cordasco, Grzegorz Maleroicz (November 2007), "Advances in IC-scheduling expansive and reductive DAGs and scheduling DAGs via Duality on IEEE transactions on parallel and distributed systems, Volume 18, Issue 11.
7. I-Jong lin, Sun-yuan kung, fellow (November 1997), "IEEE coding and comparison of dag's as a novel neural structure with applications to on-line handwriting recognition on ieee transactions on signal processing", Volume 45, Issue 11.
8. Raffaele Montella, Diana Di Luccio, Sokol KostaDagOn*: Executing direct acyclic graphs as parallel jobs on anything 2018 IEEE/ACM Workflows in Support of Large-Scale Science (WORKS).

Cite this article as: *Shubha B G, & A.M.Prasad. (2019). Airflow Directed Acyclic Graph. Journal of Signal Processing, 5(2), 12–20. http://doi.org/10.5281/zenodo.3247274*