A Project Management Framework for Cloud and HPC Resource Providers

Jefferson Ridgeway[2], Ifeanyi Rowland Onyenweaku[3], Gregor von Laszewski[1*], Fugang Wang[1]
[1*] Indiana University, Bloomington, IN 47408, U.S.A., laszewski@gmail.com, kevinwangfg@gmail.com
[2] Elizabeth City State University, jdridgeway4@gmail.com
[3] Mississippi Valley State University, rowlandifeanyi17@gmail.com

Abstract

Cloudmesh is a project that allows the management of virtual machines in a federated fashion. It can be run in two modes. One is a standalone mode where the users run cloudmesh on the local machines. The second mode is a hosted mode where multiple users share a web server through which the virtual machines are managed. One of the important functions for cloudmesh is to provide a sophisticated user management. This user management is currently conducted in drupal through the FutureGrid portal via an integration to the FutureGrid LDAP server. However, as the rest of cloudmesh is developed in python, hence in order to increase sustainability, we benefit from transitioning the user management also to python. This will also allow us to add more advanced user and project management functionality into cloudmesh.

The implementation leverages a data model design provided in python via mongoengine to represent users projects and project committees that approve projects. As part of the management functionality, we need to implement a queue in which users are queued for approval, and a project queue whereby projects are queued and approved by a committee. An Application Interface written in python will support this task and provide an abstraction that is outside the web interface.

Introduction

Ever since the inception of clouds and their functionality in maintaining data, the field of cloud computing has grown immensely. An important academic project is FutureGrid lead by Indiana University. FutureGrid provides an experimental testbed for clouds, HPC, and Grids. It enables researchers to experiment in difficult research challenges in the computer science field that are related to the applicability of grids and clouds [1]. The testbed aids virtual machine based environments, and native operating systems for experiments aimed at minimizing overhead and maximizing performance [1]. This testbed has been the motivating driver for Cloudmesh. Cloudmesh allows for federated resource management of virtual machines , bare metal provisioning, and access to a rich set of interfaces including REST, shell, and a python api of its services. The goal is to provide a Software Defined Distributed System (SDDSaas)[2]. Currently, Cloudmesh uses flask, a web development framework. While there is no issue with using flask as the main web development framework, the cloud computing community uses django as web development framework. Django operates in a similar fashion as flask, such as displaying views, using certain templates, and other components, but mainly it is more widely used and accepted within the community.

The goals of Cloudmesh include to develop a role based user, a project management framework, and to evaluate if Django can be used instead of flask as the web development framework for accessing Cloudmesh databases and much of the logic in Cloudmesh can be easily moved from flask to django. All the while, developing sample use cases for using certain django features, so that the transition form flask to django an be facilitated easily. This will include creating proper and appropriate documentation on how to install and manage a Django server. An additional goal to this research is to see if we can reuse the MongoDB that we used as part of the flask based framework within the django based framework [3].

Design

Users and project information must be verified before they can be activated. The user is verified by validation of the information entered. Include the username, email, institution, country, and much more

Screenshots and Diagrams

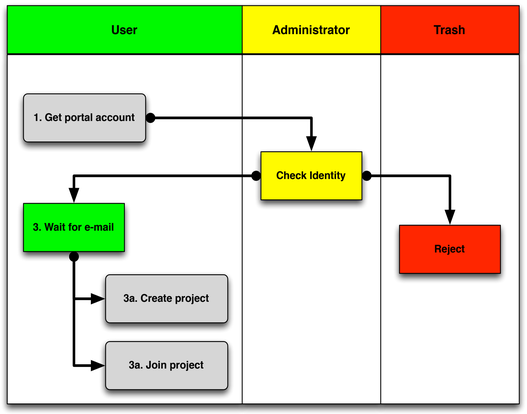

Figure 1: User Management Framework

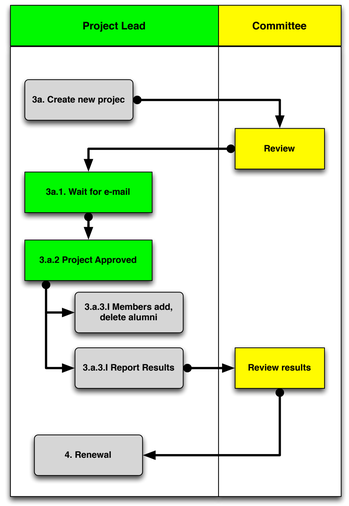

Figure 2: Project and Committee Framework

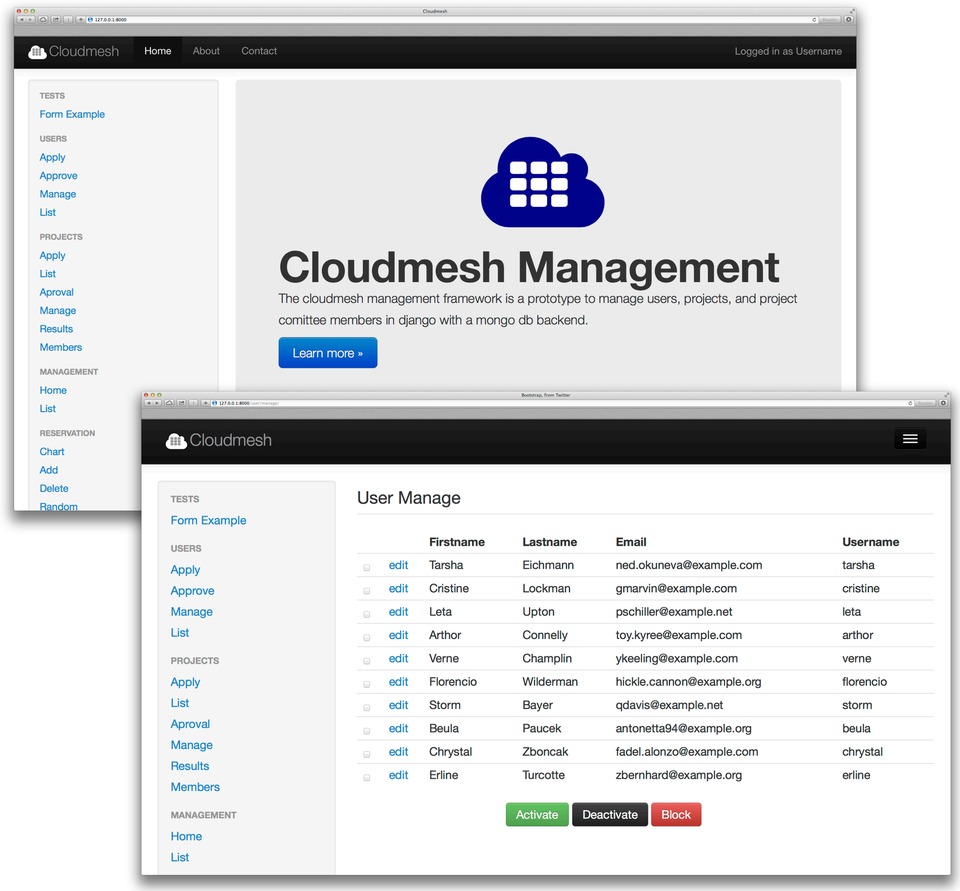

Figure 3: Web interface for the Cloudmesh Management

Implementation

Cloudmesh Management is implemented with frameworks such as python Django and MongoDB (with access through mongoengine).

Using the frameworks mentioned above, an API that performs the addition of users and projects to the database was implemented. In this API, the user is added to the database after being verified. We were able to display all the users and projects that has been created, and perform certain functions like activate, deactivate, block, find, delete a user and many more with the database. In the creation of the web Framework of Cloudmesh Management, we used classes that contains attributes that represents fields in the database, to connect with mongodb using the form API to display the forms on the django development framework.

Status

We have developed a prototype web service for the User Interface displaying links to management, administration, cloudmesh and projects via the django web devlopment framework on the browser. Currently, we are working on the approval mechanism and a mixed database model in order to connect the mongoDB database with the Django web framework to display users, projects, committees, and approvals/disapprovals.

Future work to improve the Cloudmesh management framework includes finishing the implementation of the approval mechanism for both users and projects registration through web interface, completion of the functions of the committee roles, authentication and authorization framework, improving workflows of management and to display reservation data and list virtual machines on various clouds accessing the cloudmesh database.

Acknowledgments

We like to thank Dr. Geoffrey Fox for his support, We also would like to thank the School of Informatics at Indiana University Bloomington and the IU-SROC director Dr. Lamara Warren. This material is based upon work supported in part by the National Science Foundation under Grant No. 0910812.

References

1. Laszewski Von G., Cloudmesh:Overiew, *Cloudmesh*. Retrieved June 28, 2014, from Indiana University, Bloomington, 2013: http://cloudmesh.futuregrid.org/cloudmesh/about.html
2. von Laszewski, G.; Fox, G. C.; Wang, F.; Younge, A. J.; Kulshrestha; Pike, G. G.; Smith, W.; Voeckler, J.; Figueiredo, R. J.; Fortes, J.; Keahey, K. & Deelman, E. Design of the FutureGrid Experiment Management Framework, Proceedings of Gateway Computing Environments 2010 (GCE2010) at SC10, IEEE, 2010

*Corresponding Contact

Gregor von Laszewski, Indiana University, laszewski@gmail.com