Cloud Environment Manager

Design Document

Adis Osmankic - Team leader

Zane Seuser - Cloud Architect

Jet Jacobs - System Architect

Rishabh Bansal - Report Manager

Gavin Monroe - Meeting Scribe

Team Number: sdmay21-39

Client: PwC

Adviser: Lotfi Ben Othmane

Team Email: sdmay21-39@iastate.edu

Website: https://sdmay21-39.sd.ece.iastate.edu

Last Revised: Sept. 29, 2020

Executive Summary

Development Standards & Practices Used

- Agile Development
- Cloud IAC (Infrastructure As Code)
- Modular Code
- Well-Documented Code

Summary of Requirements

- Simple User Interface
- Easily Create & Manage Any Amount of Lab Environments
- Able To Provision Labs On Multiple Cloud Providers

Applicable Courses from Iowa State University Curriculum

- COM S 309 (Software Development Practices)
- COM S 319 (Construction of User Interfaces)
- COM S 362 (Object-Oriented Analysis & Design)
- COM S 363 (Database Systems)
- SE 329 (Software Project Management)
- SE 339 (Software Architecture & Design)

New Skills/Knowledge Acquired That Was Not Taught In Courses

- Knowledge Of Different Cloud Providers
- Knowledge Of Infrastructure As Code
- Python Development
- React Development

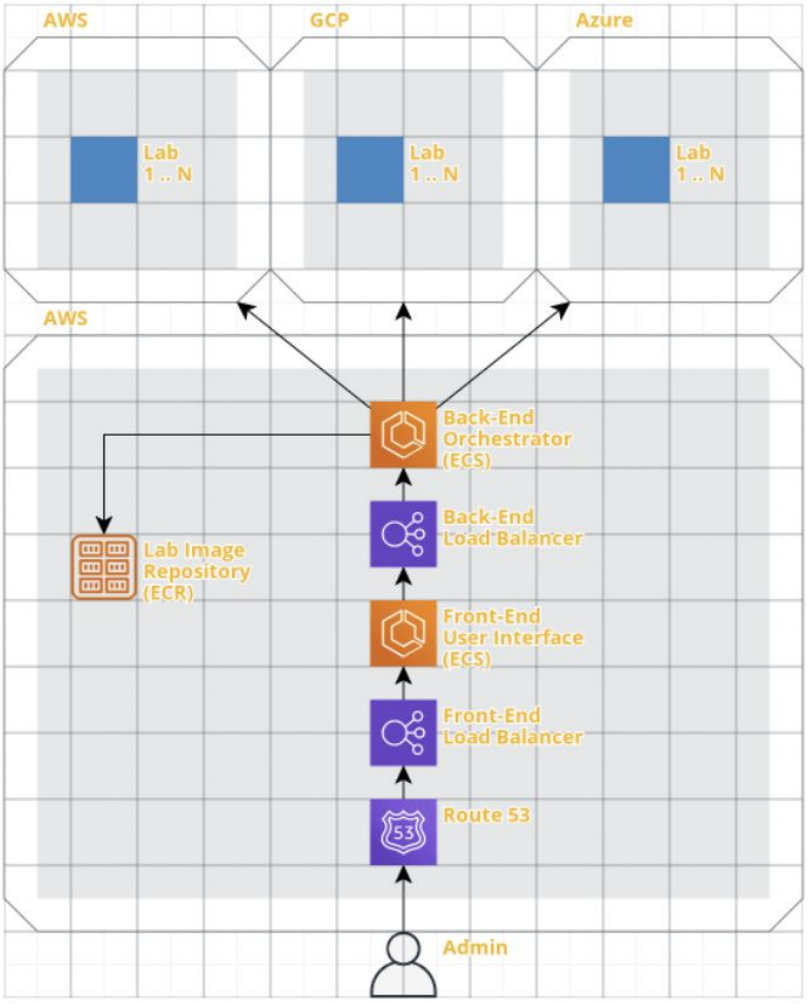
Table of Contents

List of Figures/Tables/Symbols/Definitions/Diagrams

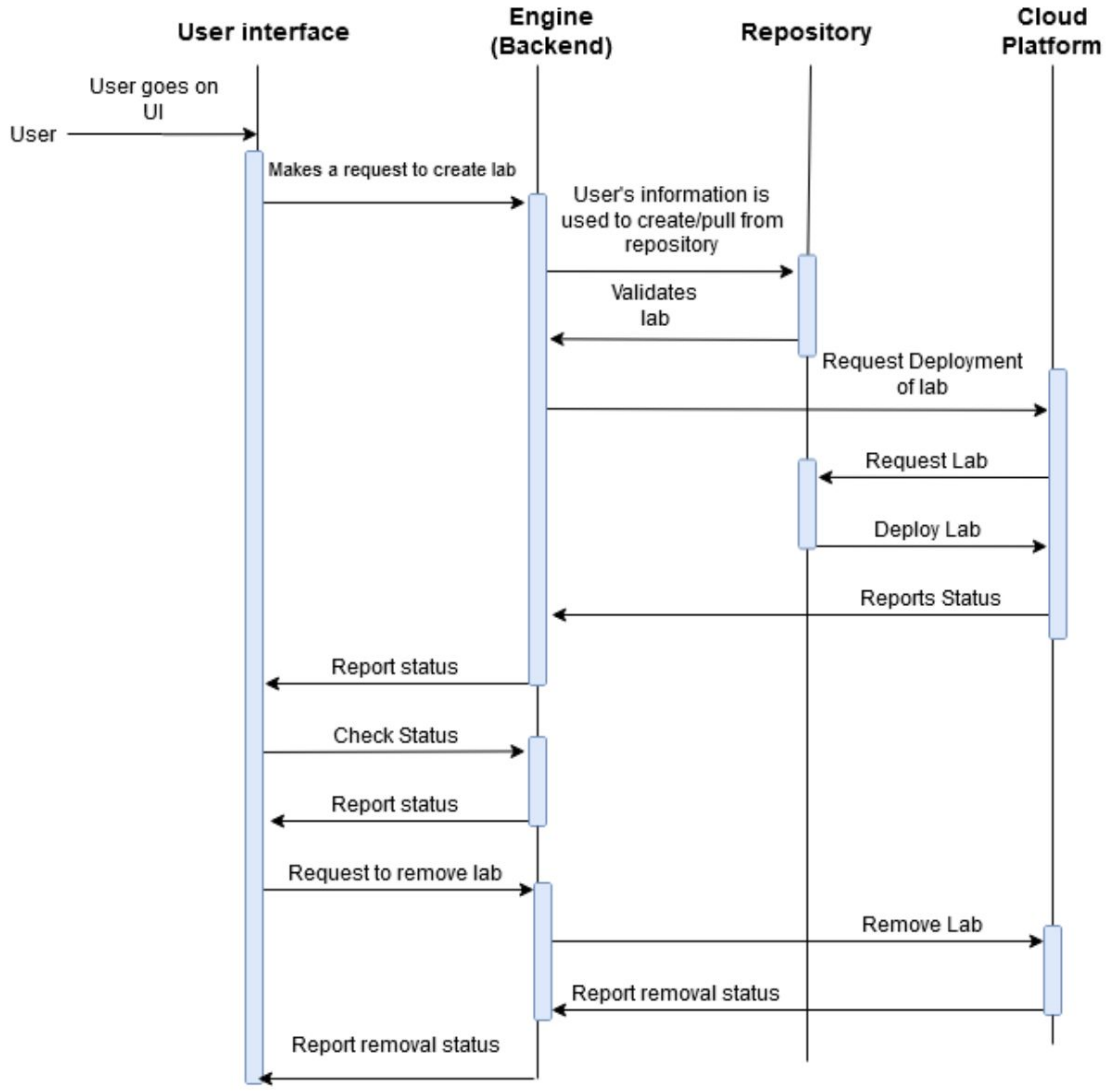
Definitions:

AWS: Amazon Web Services
ECS: Elastic Container Service
ECR: Elastic Container Repository
GCP: Google Cloud Platform
IAC: Infrastructure As Code
UI: User Interface

Component Diagram:



Interaction Overview Diagram:



List of Tables

1 Introduction

1.1 Acknowledgement

Thank you to our client PwC, and our contact, Matt Weidman, for the technical support and architectural guidance. We would also like to thank Lotfi Ben Othmane, our advisor for this project, for helping us through the planning and development process.

1.2 Problem and Project Statement

Currently, our sponsor, PwC, provisions lab environments for capture the flag events within a variety of cloud providers and on-premise resources by hand. This does not scale well because there is no way to quickly create and manage a large amount of lab environments. There is also no way to easily make sure that all these environments are the same.

By developing the Cloud Environment Manager application, we hope to provide PwC with a clean and simple user interface that allows them to quickly and easily deploy lab environments to their desired cloud providers. The application will also allow them to manage and destroy their existing lab environments once they are no longer needed.

1.3 Operational Environment

This main application will be hosted within Amazon Web Services. The front-end application will be accessible from a web browser and the back-end API it interfaces with will be able to deploy lab environments to different cloud providers such as AWS, Azure, GCP, etc.

1.4 Requirements

- **Functional Requirements**
    - A user shall not be able to have access without signing in via Google OAuth2
    - A user should be able to sign in to the application
    - A user should be able to view a list of existing lab environments
    - A user should be able to view the status of existing lab environments
    - A user should be able to view attributes of existing lab environments
    - A user should be able to create lab environments within AWS, GCP, and Azure
    - A user should be able to destroy existing lab environments
- **Non-Functional Requirements**
    - The application should be available at all times
    - The application should properly handle errors behind-the-scenes
    - The user interface should be visually appealing
    - The user interface should be easy to use

1.5 Intended Users and Uses

This project is designed for use by an organization that needs to deploy lab environments to different cloud platforms for a variety of different users. This project is focused on education and deploying capture the flag events to large groups of people for training. By using this product, organization admins will be able to deliver training resources to members of the organization quickly and easily.

1.6 Assumptions and Limitations

- **Assumptions**
    - The organization has a cloud platform
    - The organization has the resources to host the user interface and backend
    - The organization can configure the lab environments to accept programmatic access to the virtual machine resources in the cloud
- **Limitations**
    - There are some configurations that need to be manually changed by an organization
    - Only administrators can set up the application

1.7 Expected End Product and Deliverables

- Cloud-Hosted Cloud Environment Manager And Source Code
- Project Documentation
- May 2021 Hand-Off

2 Project Plan

2.1 Task Decomposition

Goal	Sub Tasks
Lab Environment Manager	• Orchestrator Flow • Deploy basic EC2 with basic playbook • Configure EC2 Through Playbook • Create VM From Scratch • Hook Ansible to Backend Call • Configure Automation Scripts • Create Full Demo
Cloud Readiness	• Setup group cloud environments for team use • Configure AWS for labs • Configure Azure for labs • Configure GCP for labs
Course Work	• Design Document v1 Draft • Design Document v1 Final • Design Document v2 • Final Design Document • System Block Diagram • UI description • Work plan • System Design • Detailed Design • Test Plan • Course Site

Table 1: Task breakdown

Task	Description	Deadline
Orchestrator Flow	Flow of the orchestrator	10/3
Deploy basic EC2	Need everyone to fully understand Ansible by running through some examples of deploying to EC2	10/2
Configure EC2	Need to customize the vm for deployment	10/7
Create VM From Scratch	Fully custom make a VM from scratch	10/13

Hook Ansible to Backend Call	Need the playbooks to be run from a backend service	10/20
Configure Automation Scripts	Configure the scripts to be run dynamically for the project	10/27
Create Full Demo	Need a demo of deployment for the client	11/15
Setup group cloud environments for team use	We need environments to be setup for group usage	10/2
Configure AWS for labs	Configure AWS to properly host the lab environment	10/20
Configure Azure for labs	Configure Azure to properly host the lab environment	11/3
Configure GCP for labs	Configure GCP to properly host the lab environment	11/15
Design Document v1 Final	Final version of the first design document version	10/4
Design Document v2	Final version of the second design document version	10/25
Final Design Document	Final version of the design document for the project	11/15
System Block Diagram	System Block diagram for the entire system	10/3
UI description	Description of the UI to give a basic path to how the UI should be built	10/6
Work plan	Plan of work for the project	11/1
System Design	Diagrams and description for the entire project	10/13
Detailed Design	An in depth review of the design of the project	11/4
Test Plan	Plan to test if the project meets the requirements	11/11
Course Site	Site for the project holding information and documentation	11/15

Table 2: Task Summary

2.2 Risks And Risk Management/Mitigation

All risks are evaluated based on the perceived impact that it might have on the completion of the project.

Risk	Evaluation	Mitigation
A cloud environment may not be provided	Low	This can be mitigated by using school resources to do the hosting of the application.This in combination with the free tier AWS should be enough for the initial development.
Cloud platforms may not have much overlap in functionality	Medium	This does pose a fairly considerable barrier to implementation, but we can mitigate using a few different techniques. Firstly, by prioritizing the most important cloud platforms. Second, by using Ansible we hope to minimize the feature set required by cloud platforms. **Note:** GCP and AWS both support Ansible
With all projects there is a possibility that members may leave	Low	While this is low likelihood, the best mitigation strategy is to not divide into clear roles, in which only one person knows how to fill any given role. We should all have a firm grasp of each area, so that we may spot fill in the event of a member leaving.

Table 3: Risk Table

2.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

Orchestrator Flow	Software flow, how will everything flow together
Deploy Basic EC2 with basic Playbook	Playbook on to deploy a EC2 on AWS cloud platform.
Configure EC2 through playbook	Playbook on to configure a EC2 on AWS cloud platform.
Create VM from scratch	Create a VM image from scratch that we can

	use on the EC2
Hook Ansible to some backend call	Hook up Ansible for backend software calling and usage.
Configure automation scripts	Configure scripts that will perform automatic commands, and processes.
Create Full Demo	Create a demo for the client showing everything working from backend to frontend.

Table 4: Milestones Table

2.4 Project Timeline/Schedule

Orchestrator Flow	October 3, 2020
Deploy Basic EC2 with basic Playbook	October 2, 2020
Configure EC2 through playbook	October 7, 2020
Create VM from scratch	October 13, 2020
Hook Ansible to some backend call	October 20, 2020
Configure automation scripts	October 27, 2020
Create Full Demo	November 15, 2020

Table 5: Project Schedule Table

2.5 Project Tracking Tools & Procedures

- **Team Communication**
  - Discord
- **Source Code Repository**
  - GitLab
- **Task Tracking**
  - Jira
- **Procedure (Agile)**
  - Determine most pressing tasks from project assignments and sponsor
    - Place into our Agile Board TODO
  - Volunteer for, assign, and commit to project tasks in weekly meetings
    - Move to Agile Board In-Progress
  - Give status update on completed tasks in weekly meetings
    - Move to Agile Board Done
  - Hold each other accountable for completing assigned tasks by deadlines

2.6 Personnel Effort Requirements

Task	Person hours	Explanation
Orchestrator Flow	**10**	To establish a cursory flow for the orchestrator is fine.
Deploy basic EC2	**40**	For each member to complete a deploy and be knowledgeable on the topic 8 or so hours each seems fair.
Configure EC2	**40**	Seems fitting as, it will likely take some research, but ultimately shouldn't be too time exhaustive.
Create VM From Scratch	**70**	This will likely be one of the most difficult steps moving from using premade VMs. Has multiple parts, but this may also end up being completed much faster that expected.
Hook Ansible to Backend Call	**20**	With a backend established, having Ansible send a message should be fine.
Configure Automation Scripts	**10**	This may break this prediction, but it appears to be rather straightforward so long as the other steps are complete.
Create Full Demo	**L**	
Setup group cloud environments for team use	**10**	Should be rather quick environments that we can use are pretty easy to set up.
Configure AWS for labs	**25**	Fairly major step, but should be fairly simple.
Configure Azure for labs	**30**	The cloud platform that we as a group know the least about.
Configure GCP for labs	**30**	The second least known cloud platform.
Design Document v1 Final	**15**	Not too bad about 3hrs per person seems reasonable
Design Document v2	**15**	Not too bad about 3hrs per person seems reasonable
Final Design Document	**20**	Not too bad about 3hrs per person seems reasonable

System Block Diagram	**10**	Diagram shouldn't be too time extensive
UI description	**15**	Fairly easy, for the most part would be working with our client to ensure that it is acceptable
Work plan	**10**	Pretty straight forward, and shouldn't require much effort beyond the few man hours to implement.
System Design	**25**	While typically simple, it may require some research, as well as maintenance if it changes.
Detailed Design	**40**	This will be quick for the initial draft, but effort to keep it current seems reasonable to push this upward in people hours.
Test Plan	**30**	This could get complex, since there doesn't seem to be a very clean way to test the configuration via some framework. It may take research, etc.
Course Site	**25**	Once all steps are done, it should be pretty easy to set up, and fill out, but may take some time.

Table 6: Effort Table

2.7 Other Resource Requirements

There will be no other resource requirements.

2.8 Financial Requirements

There will be no financial requirements. This application will be developed with free tools, and we are using the free tier of the various cloud providers.