2.3 Team Project Report Week 2

Christofer Weiss

Colton Nash

Joshua Deason

Ivan Fernandez

Vinicio Morillo Romero

Ensign College

IT 360: Cloud Integration

Brother Jackson Ivan Isiko

September 11th, 2025

Multicloud High Availability Website Project

Business Case

In the competitive digital landscape, a website's availability and reliability are critical for success. Service interruptions, whether from technical failures, unexpected traffic spikes, or natural disasters, can result in significant losses in revenue and credibility. This project addresses the availability challenge by creating a multicloud infrastructure architecture. By replicating the website environment on two different cloud providers, Amazon Web Services (AWS) and Google Cloud Platform (GCP), we ensure redundancy and failover capability. This guarantees the website remains accessible and functional, even if one of the providers experiences an outage.

Project Goals

The main objective of this project is to build and deploy a robust, scalable, and secure IT infrastructure for a static website. This environment will serve as a foundation for the development team to deploy their application. The specific goals include:

- Create Virtual Machines (VMs): Use Terraform to provision two virtual machines, one in AWS and one in GCP.
- Automate Configuration: Use Ansible to install and configure the Apache2 web server on both VMs.
- Deploy Content: Deploy the static website content to the web servers from a GitHub repository.

- Setup DNS Failover: Configure a DNS failover service to direct traffic to the active VM and manage failover between the clouds.
- Simulate High Availability: Simulate a failover and verify that traffic successfully moves to the VM in the other cloud.

Proposed Solution Architecture

Our solution is based on a multi-cloud hybrid approach to ensure redundancy and reliability. The Operations team will prepare this environment for the Development team.

Infrastructure: Two Ubuntu servers with Apache2 will be created, one on an AWS EC2 instance and the other on a Google Compute Engine (GCP) instance.

Infrastructure Automation: Terraform will be used for the programmatic creation and management of the virtual machines in both AWS and GCP.

Configuration Automation: Ansible will be used to automate the installation of Apache2 and the environment configuration on both VMs.

Content Deployment: The static website content will be pulled directly from a GitHub repository, and Apache2 will be configured to serve it.

Traffic Management: A DNS failover service, such as NO-IP.com, will be used to point users to the active VM's IP and manage the automatic switch if one of them fails.

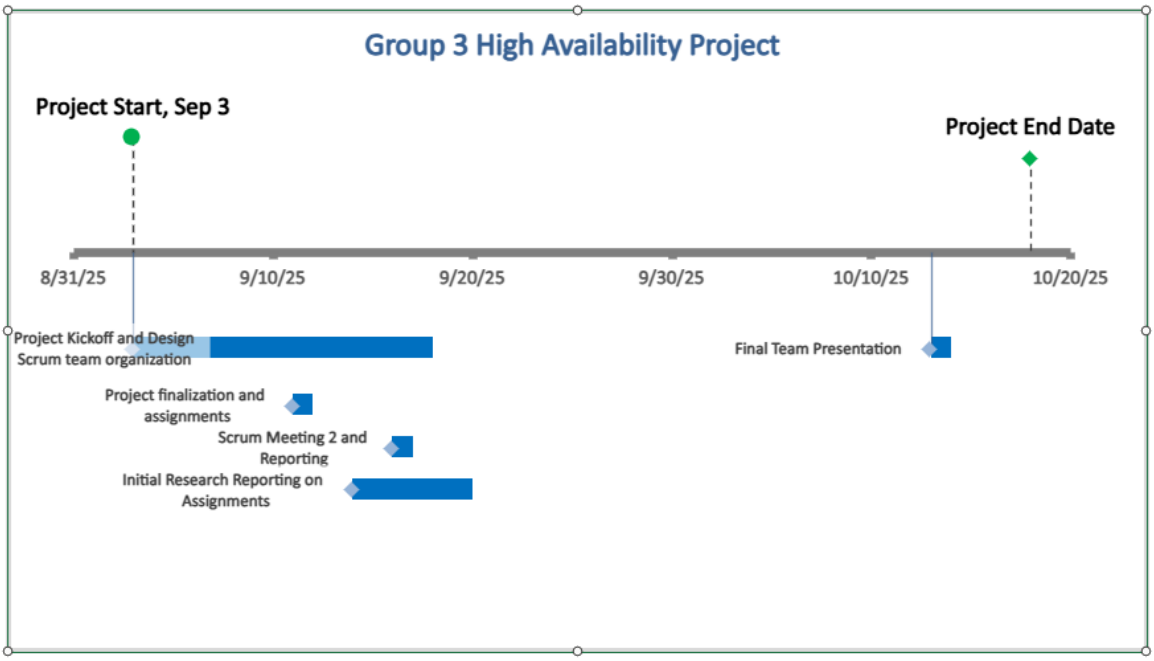
Tools and Collaboration Strategy

To manage the project efficiently, the team will use the following tools and practices:

Project Management: A Project Scrum timeline Excel document will be used for task management, progress tracking, and team communication. The team will list tasks, assign them to members, and update their status and progress within the document.

Version Control: The static website code and Ansible automation scripts will be managed through a centralized Git repository.

Communication: The team will hold weekly online meetings via Teams to report on completed assignments and set new ones. A group chat will be used for daily updates and to address any immediate issues or roadblocks.



Tasks

Start	End	Duration	Label	Vert. Position	Vert. Line
9/3/2025	9/8/2025	15	Project Kickoff and Design Scrum team organization	-25	-25
9/11/2025	9/11/2025	1	Project finalization and assignments	-40	
9/16/2025	9/16/2025	1	Scrum Meeting 2 and Reporting	-51	
9/14/2025	9/20/2025	6	Initial Research Reporting on Assignments	-62	
10/13/2025	10/13/2025	1	Final Team Presentation	-25	-25
			Insert new rows above this one		

Milestones

Date	Label	Position
9/3/2025	Project Start, Sep 3	30
10/18/2025	Project End Date	25
	Insert new rows above this one	