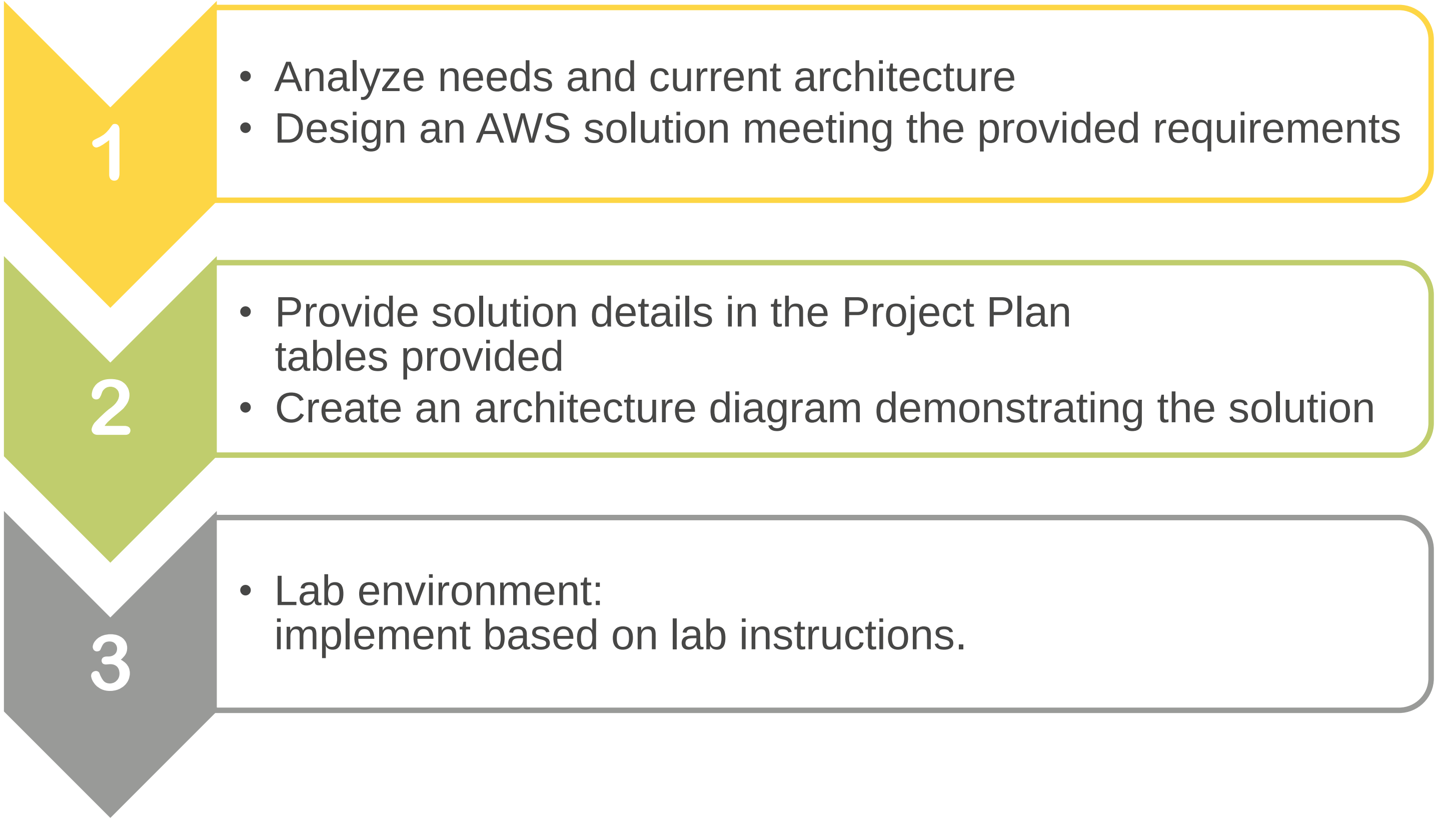
Final Project

GoGreen Insurance Company

Final Project Overview



1

- Analyze needs and current architecture
- Design an AWS solution meeting the provided requirements

2

- Provide solution details in the Project Plan tables provided
- Create an architecture diagram demonstrating the solution

3

- Lab environment:
  implement based on lab instructions.

Background

Company:	GoGreen Insurance Company
Locations:	Europe, South America Southern California (headquarters)
Application:	CRM web application allows sales personnel to input and edit customer data.
Technical:	3-tier web app stores customer data and documents. Converts the documents into multiple formats (e.g. images for web/mobile)
Goal:	Go *paperless* for all user data, documents and pictures

Issues

Problems with the existing application include:

- On-premises performance and reliability issues occur.
- The architecture is continuously overprovisioned to handle growth.
  This process has occurred three times in the last year.
- Growing the architecture has become too expensive.
- The procurement process takes 20 days.
- Deployment takes a week.
- The entire process costs in excess of $100,000.

The plan is to move the application into the cloud.

Current Architecture

Web Tier

- Six virtual machines (Two vCPUs/4-GB memory)
- SUSE Linux Enterprise Server 12
- Apache web server
- PHP server and PHP files

Application Tier

- Five virtual machines (Four vCPUs/32-GB memory)
- SUSE Linux Enterprise Server 12
- Java SRE 7/Java application files

Database Tier

- Two virtual machines (Eight vCPUs/48-GB memory /5.5-TB storage)
- SUSE Linux Enterprise Server 12
- MySQL 5.6.22 database cluster

Requirements (1 of 4)

Environment

- Infrastructure will be managed by members of the new Cloud Team.
- All data must be encrypted in transit and at rest.
- Infrastructure should be secured using a defense-in-depth approach.
- Users should connect to stateless web servers.
- A baseline for the number and type of instances needed should be established.
- Recovery Point Objective for the application is four hours.
- A user base that is expected to grow 90% in the next three years must be supported.
- Documents and pictures must be kept for five years. However, these files are rarely requested after three months.
- To enhance availability and lower cost, managed services must be leveraged whenever possible.

Requirements (2 of 4)

Web Tier

- Architecture must be flexible and handle any peak in traffic or performance
- Servers are currently at 75% of memory capacity all the time. This number must decrease to between 50% and 60% when moved to AWS.
- Application administrators want to be notified by email if there are more than 100 “400 HTTP errors” per minute in the application.
- All instances in Web Tier should be tagged as “Key=Name” and “Value=web-tier”.

Requirements (3 of 4)

Application Tier

- Architecture must be flexible and handle any peak in performance
- Servers are currently at 90% of memory and CPU capacity all the time. This number must decrease to between 50% and 60% when moved to AWS.
- Overall memory and CPU utilization should not go above 80% and 75% respectively, or below 30% for each.
- Internet access for patching and updates must be available without exposing the servers.
- All instances in Application Tier should be tagged as “Key=Name” and “Value=app-tier”.

Requirements (4 of 4)

Database Tier

- The database needs consistent storage performance at 21,000 IOPS.
- Internet access for patching and updates must be available without exposing the servers.
- High availability is a requirement.
- No change to the database schema can be made at this time.

Cost Considerations (optional)

The proposed solution must take into consideration all the technical requirements as well as the most cost-conscious financial options.

Typical cost considerations include:

- Type of instances and payment models
- Number of instances
- Estimated monthly cost for the solution

Student Solution Template

Project Objectives

Project Plan

- Determine the region, VPCs, subnets, and Availability Zone requirements.
- Document encryption and security details.
- Design a plan for storage and backups.
- Using what you learned in class, determine how to resolve the issues concerning the Web, App, and Database Tiers.
- Use this document as your implementation plan.
- Build the infrastructure in the lab based on this document.

VPC Details

VPC	Region	Purpose	Subnets	AZs

Subnet Name	VPC	Subnet type (Public/private)	AZ

Security Details

Security Group	SG Name	Rule	Source
ELB load balancer			
Web Tier			
App Tier			
Database Tier			

Other Security Options	Justification

Encryption Options

Based on the requirements, list your encryption options:

Requirement	Solution
Encryption option for **data at rest**	
Encryption option for **data in transit**	

Instance Details

Describe the type, size, and justification for the instances you will use for each tier.

Tier	AMI	Tag	Type	Size	Justification	# of instances
Web		**Key: Name** **Value: app-tier**				
App		**Key: Name** **Value: web-tier**				
DB		**N/A**				

RPO Options

Q. How would you achieve a Recovery Point Objective (RPO) of four hours?

A.

Document Storage

How would you design document storage based on the requirements?

Storage/Archive Option	Detail

Web Tier Requirements

Requirement	Solution
Architecture must be flexible and handle any peak in traffic or performance.	
The overall acceptable incoming network bandwidth is between 300 Mbps and 750 Mbps.	
Application administrators want to be notified by email if there are more than 100 “400 HTTP errors” per minute in the application.	

App Tier Requirements

Requirement	Solution
Architecture must be flexible and handle any peak in traffic or performance.	
Overall memory and CPU utilization should not go above 80% and 75% respectively or below 30% for either.	
Internet access is required for patching and updates without exposing the servers.	

Database Tier Requirements

Requirement	Solution
Database needs consistent storage performance at 21,000 IOPS.	
High availability is a requirement.	
No change to the database schema can be made at this time.	

Additional Services

List any additional AWS services that you would use for your solution and why?

Architecture Diagram

AWS Simple Icons for Architecture Diagrams can be downloaded from:

http://aws.amazon.com/architecture/icons/

Proposed Architecture Diagram