# Recovery in the Cloud Project Proposal

## 1. Vision and Goals Of The Project:

The goal of this project is to create a simple backup and recovery solution for small businesses in times of server failure. The vision is to create a client app that regularly backs-up a Windows 2012 server into the cloud (AWS S3 storage space). If the Windows server fails, the user can start up an instance of his/her backed-up server on the cloud (hosted on AWS EC2) and connect to it using a lightweight client app supporting an automated VPN connection.

Users/Personas Of The Project:

- Small to medium-sized businesses, i.e. franchise stores, doctors offices
- Generally non-technical users
- Administration by Carbonite employees

## 2. Scope and Features Of The Project:

Scope:
Backup and recovery of a client's server to the AWS cloud

Features:

- Lightweight native Windows app allows user to initiate the backup process and establish a VPN connection to backups in the event of failure
- Server manages a database of user accounts and backups in AWS
- VPN connection between the client and the restored server in AWS

## 3. Solution Concept:

Global Architectural Structure Of the Project:

1. Image a Windows Server
2. Format and save periodic backups (full and incremental) to cloud on AWS S3 (.vhd, .vhda formats)
3. Regularly save lightweight, incremental backups to S3
4. Sandbox the saved system images to validate backups
5. For recovery, use the Windows UI to create an EC2 instance of a saved server image from S3
6. Establish VPN connection from client to the restored server

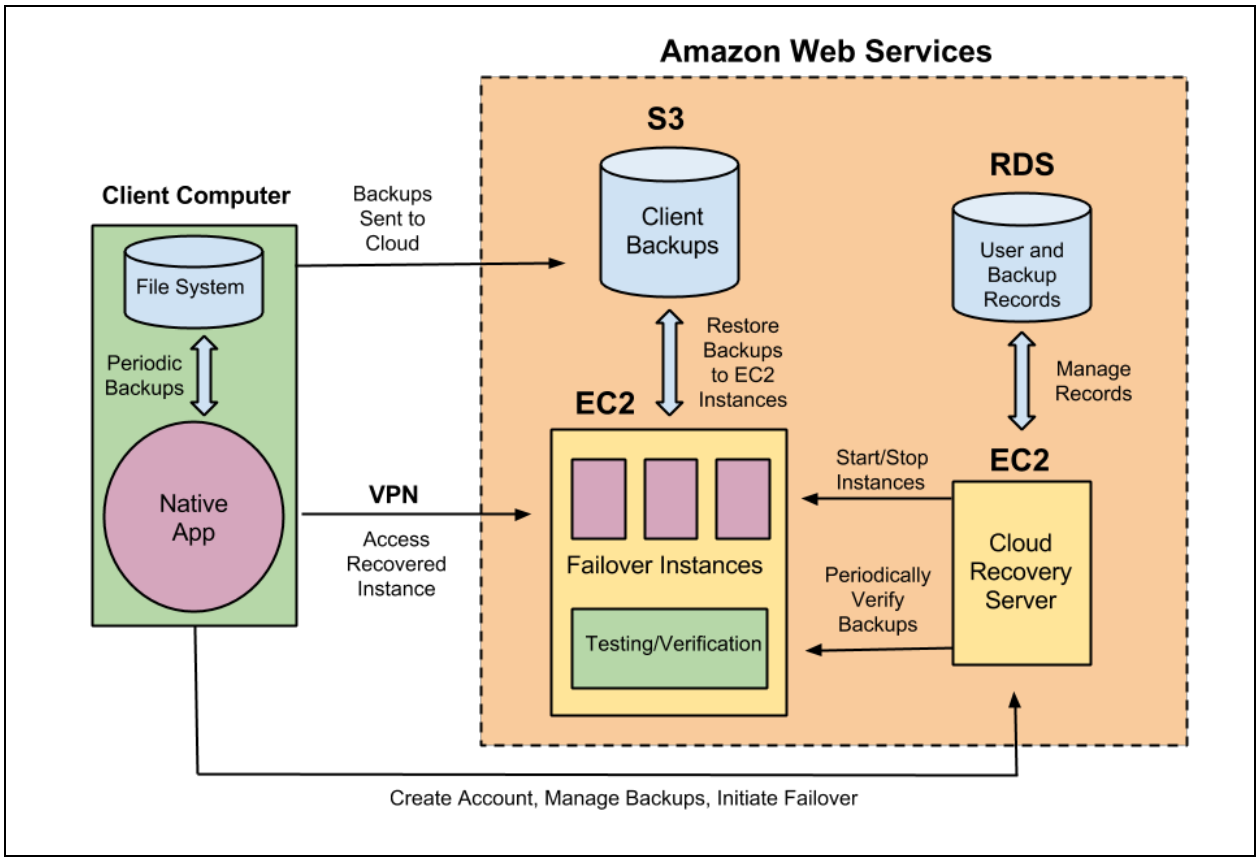


Figure 1: System Architecture

Design Implications and Discussion:

- Present the user interface as a native Windows application as opposed to a web application. This will make interacting with Windows PowerShell and other native processes significantly simpler. This will also simplify the VPN connection process.
- Use AWS (as opposed to other cloud providers) for its detailed documentation and robust API. Also AWS is already used extensively by Carbonite.
- Simple user authentication. The focus of this project is not user management but on the cloud recovery process. This may make our application less secure than a commercial product.
- Have at least one full backup of each client's server. Subsequent backups will be incremental to save time and space.

## 4. Acceptance criteria:

Basic failover of the client's system, including one full backup and the ability to access this recovered backup through a VPN connection. Deliverables will include: a Windows application that can provide support for backups, VPN connection to the recovered server, and tools to interface with AWS; the orchestration server running in the cloud to manage users and backups.

Reach Goals:

- Incremental backups on a user-defined schedule
- Automated testing/verification of backups
- Allow users to manage individual backups and select a particular backup to restore
- Failback, namely bare metal restoration of the client's new physical server and files from the cloud

## 5. Release Planning:

*Release #1 (due by Week 5):*

- Product name
- Polished design of client UI
- Setup AWS environments (S3, EC2, RDS)
- Powershell script for creating a VHD backup
- User authentication and management system

*Release #2 (due by Week 7):*

- Substantial client UI that can initiate a full backup to S3
- Product website for clients to download the desktop application

*Release #3 (due by Week 9):*

- Client to be able to start new EC2 instance from backup in S3
- Expand server software to handle multiple restored EC2 instances
- Windows installer for client download package

*Release #4 (due by Week 11):*

- VPN connection between client and recovered instance
- Incremental backups to S3
- Customization of backup settings, i.e. user-define backup schedule, scope and size of backups

*Release #5 (due by Week 13):*

- Automated testing and verification of backups
- Create tool for users to manage individual backups
- Failback