

# Cloud Recovery

## Sprint 2

Carlton Duffett, Deema Kutbi, Emilio Teran  
Konstantino Sparakis, Minhan Xiang

# ReClo Overview

## Our Clients:

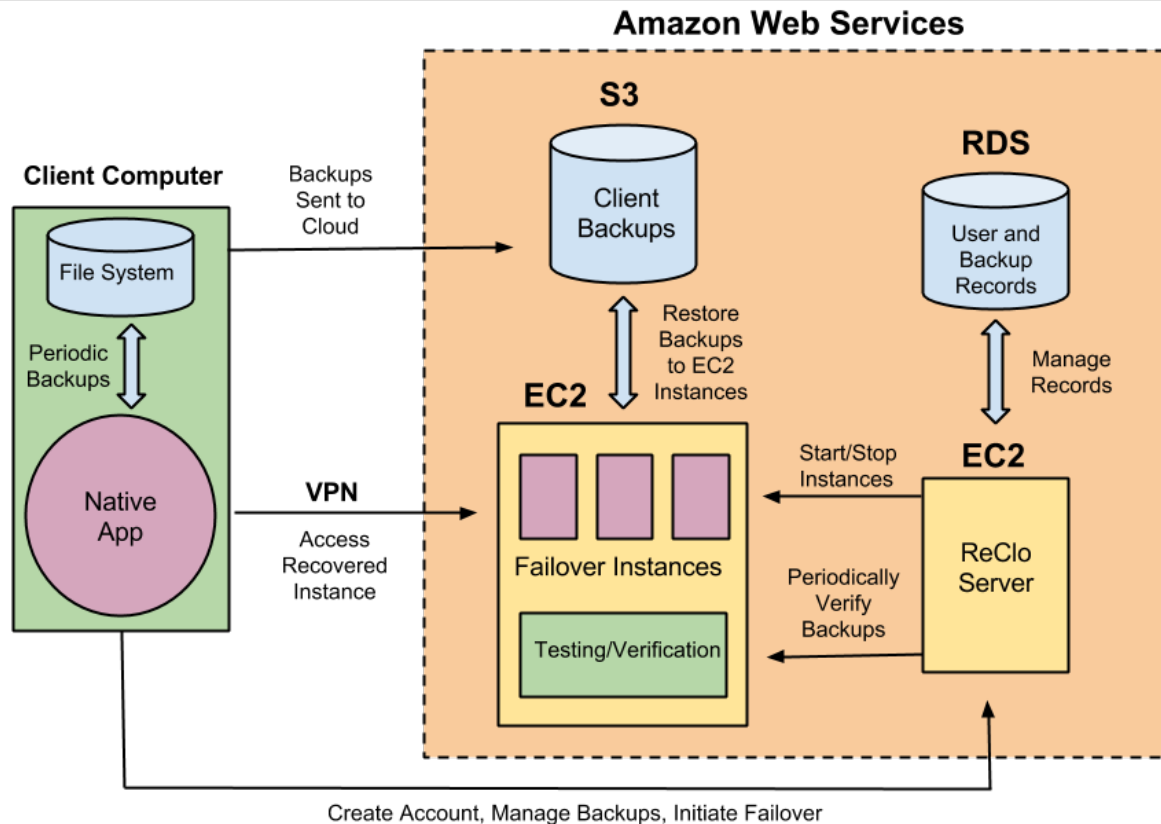
- Small to medium-sized businesses, e.g. franchise stores, doctors offices
- Generally non-technical users

## Key Features:

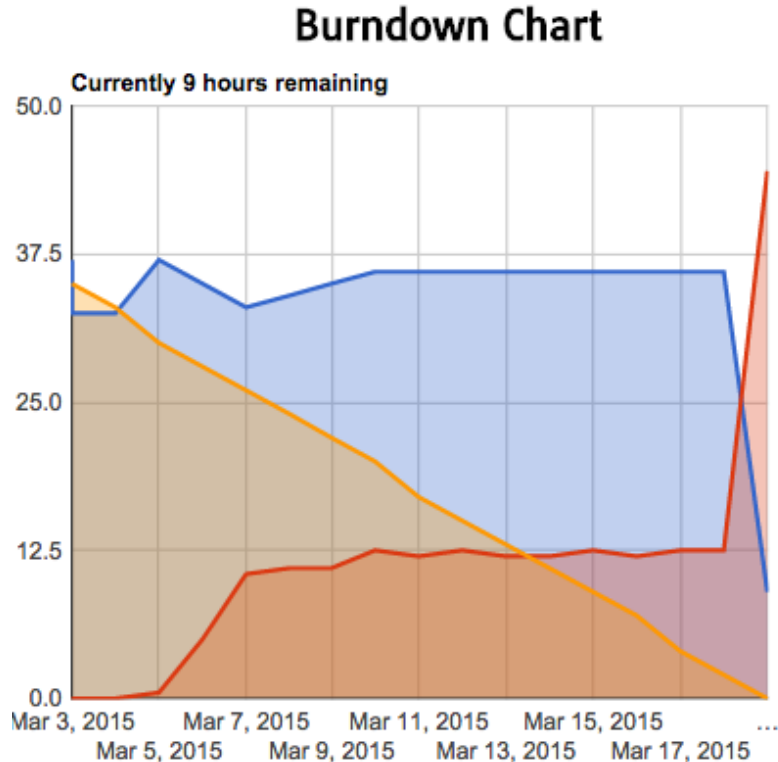
**Backup Manager** performs regular backups and pushed them to the cloud

**Recovery Manager** starts a new EC2 instance from a backup file and establishes a VPN connection, allowing users to access their data

# Architecture



# This Sprint: Burndown



“We bit off more than we could chew...”

Almost every task took longer than we expected.

Now getting to the “hard part” of building our application.

# This Sprint: Our Goals

- Client “Backup Manager” Application
- Client Installer
- Product Website
- RESTful API
- Incremental Backups

# This Sprint: Trello

**EC500 - Sprint 2** ☆ Private 🔥 [Burndown Chart](#)

**Sprint Backlog** 7

- Integrate PowerShell backups with client UI, client-side backend code  
2 MX
- Implement static product website with download links  
3
- Create a new AWS instance from a .vhd backup file  
2

Add a card...

**In Progress** 20

- Create ReClo AWS API  
4/8 13 KS
- Start new EC2 instance from Rest API call  
2 KS
- Incremental backups  
5 MX

Add a card...

**Done** 12 15

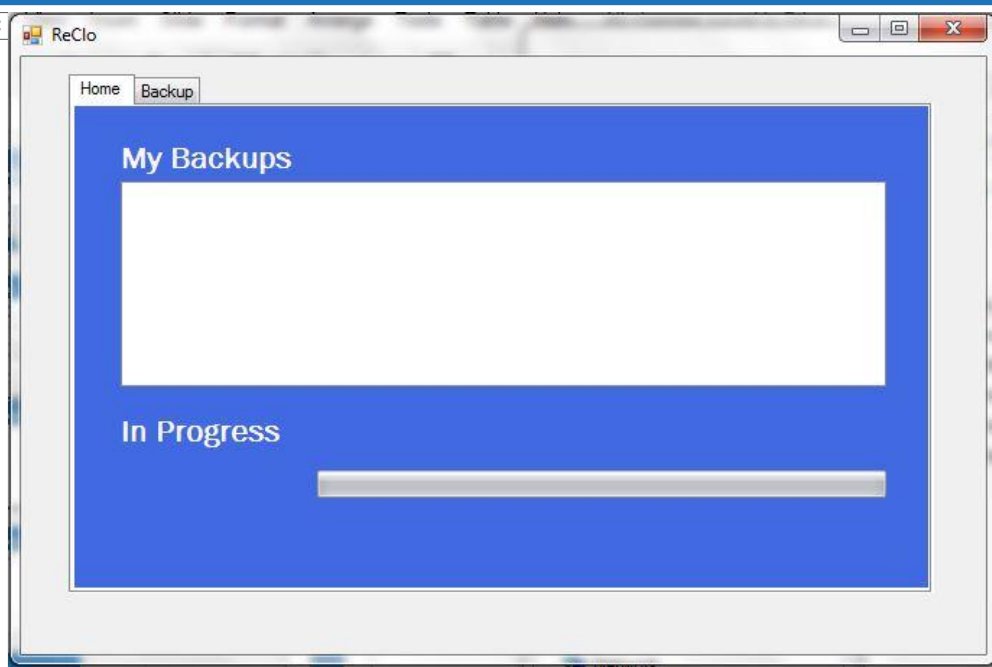
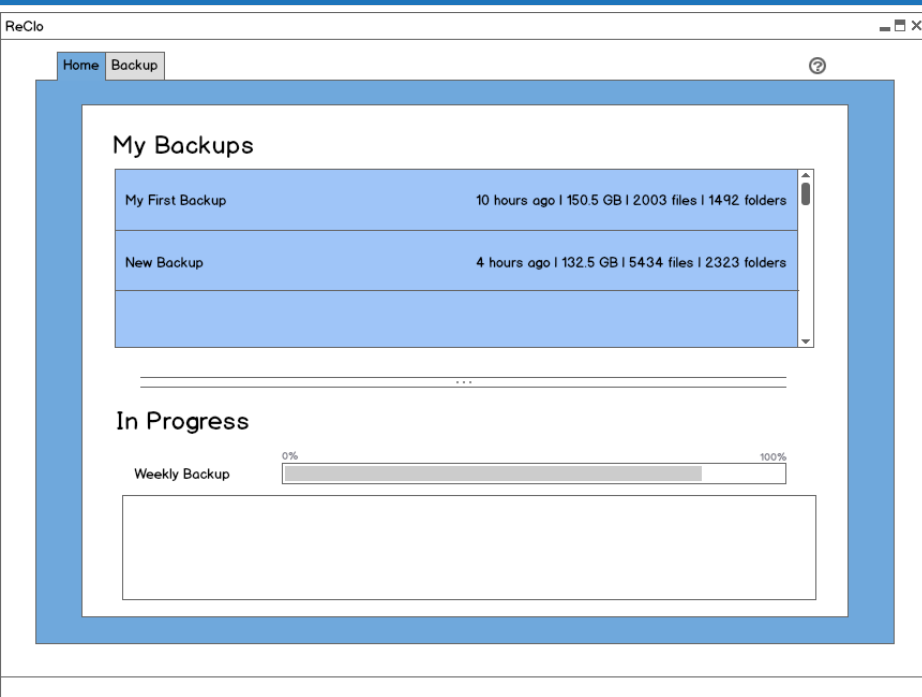
- Setup Amazon RDS using MySQL  
1 6/6 5 1
- Convert backend database to use Amazon RDS with MySQL  
4/4 5 3
- Design ReClo AWS API  
2 2 KS
- Design product website  
4/4 2 ET

Partial view of a card:

# What We Accomplished:

- Backup Manager GUI and Installer
- Async Class for connecting Backup Up manager to API
- RESTful API for Backup Manager client  
(no Recovery Manager API yet)
- Client website with product info and download links
- Converted back-end database to MySQL  
(took much longer than expected)
- Still working on incremental backups

# Backup Manager Application





# Backup Manager Application

ReClo

Home Backup

Backup source:

Backup destination:

Description:

---

What to backup:

When to backup:

ReClo

Home Backup

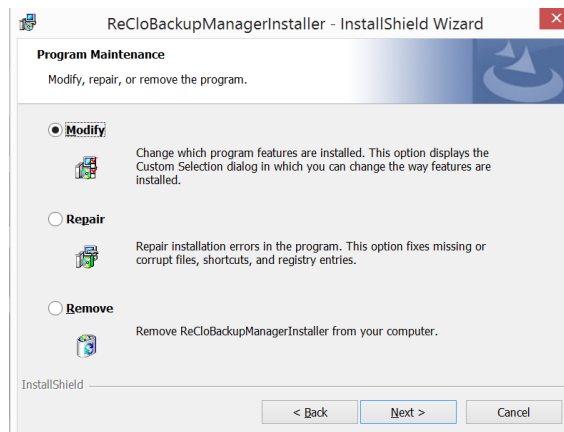
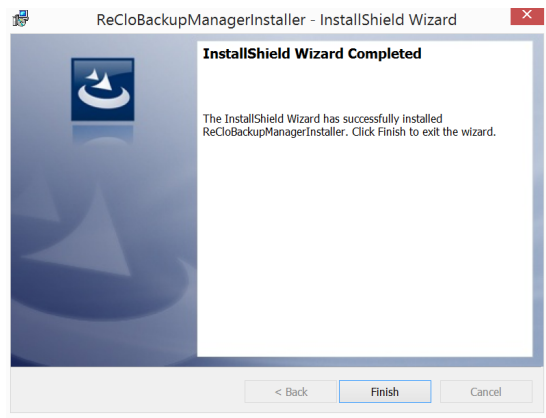
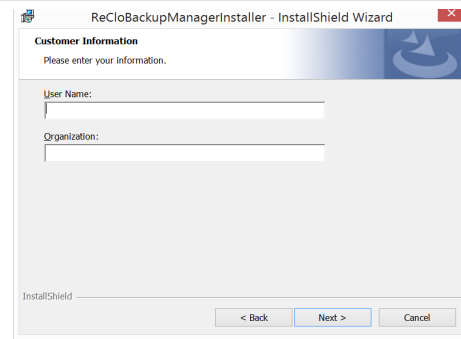
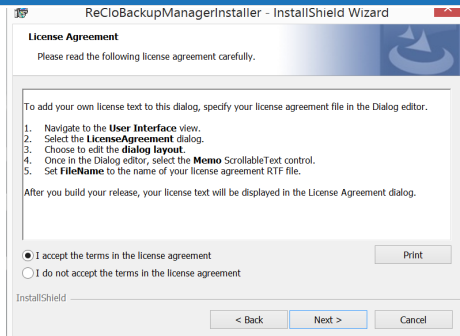
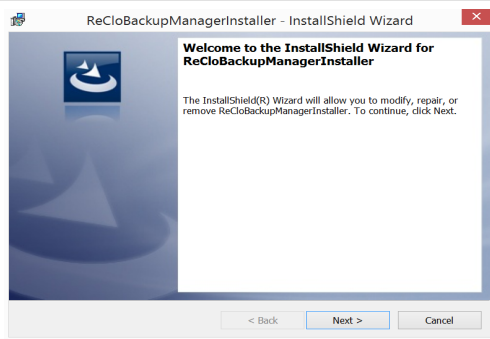
Backup Source:

Backup destination:

What to backup:

When to backup:

# Client Installer



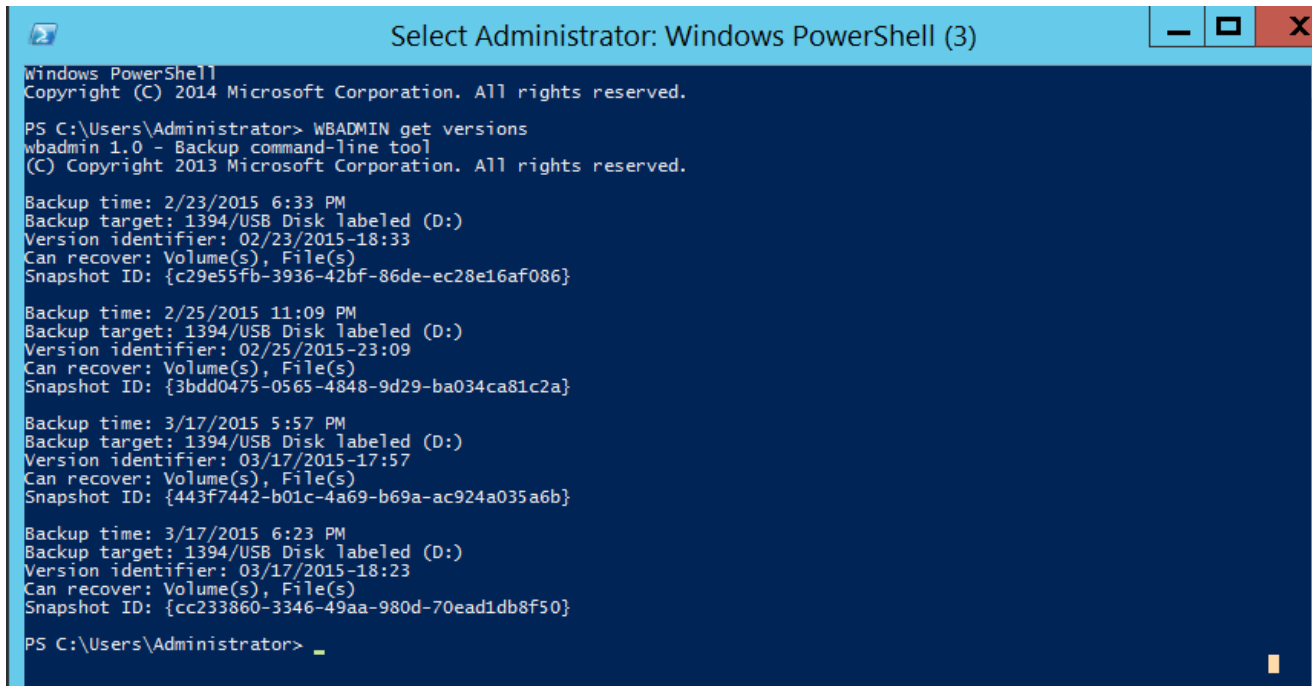
# Product Website

# Incremental Backups

The current version of WBADMIN in Windows Server 2012 compress incremental image with the original image together.

A log is kept for each backup operation  
Customers can choose the image to be recovered by Snapshot ID.

# Incremental Backups



```
Windows PowerShell
Copyright (C) 2014 Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> WBADMIN get versions
wbadmin 1.0 - Backup command-line tool
(C) Copyright 2013 Microsoft Corporation. All rights reserved.

Backup time: 2/23/2015 6:33 PM
Backup target: 1394/USB Disk labeled (D:)
Version identifier: 02/23/2015-18:33
Can recover: Volume(s), File(s)
Snapshot ID: {c29e55fb-3936-42bf-86de-ec28e16af086}

Backup time: 2/25/2015 11:09 PM
Backup target: 1394/USB Disk labeled (D:)
Version identifier: 02/25/2015-23:09
Can recover: Volume(s), File(s)
Snapshot ID: {3bdd0475-0565-4848-9d29-ba034ca81c2a}

Backup time: 3/17/2015 5:57 PM
Backup target: 1394/USB Disk labeled (D:)
Version identifier: 03/17/2015-17:57
Can recover: Volume(s), File(s)
Snapshot ID: {443f7442-b01c-4a69-b69a-ac924a035a6b}

Backup time: 3/17/2015 6:23 PM
Backup target: 1394/USB Disk labeled (D:)
Version identifier: 03/17/2015-18:23
Can recover: Volume(s), File(s)
Snapshot ID: {cc233860-3346-49aa-980d-70ead1db8f50}

PS C:\Users\Administrator>
```

# RESTful API

Added API calls to create, upload, and get information on client backups.

Uses node.js (lightweight, asynchronous JavaScript) for the server.

C# client library for use with the applications.

# API Design

## Several types of parameters:

request URL parameters	For globally unique IDs, e.g. user, backup
request body key-value pairs	For request data, e.g. username, email
request URL query	For session tokens

## HTTP request conventions:

POST	For user authentication, creating new objects
GET	For query the database for information about existing objects
PUT	For updating existing objects
DELETE	For removing/invalidating existing objects

# API Design

Example: getBackupList()

GET `host:port/backups/:user_id?token=`

`52.11.1.237:3000/backups/c2cf281d-061c-40e8-8732-7bedb9e763ec?token=31c4b1ba6b484654b256d452109762a1`



# API Client Library

- Asynchronous HTTP methods
- Used Mono for compilation
- Lots of debugging and errors
- Just basic functions for logging in, logging out, and registering

# RESTful API

# Next Sprint: Goals

Functional Backup Manager and Recovery Manager UIs

Complete API (node.js) and client library (C#)

Better estimation of tasks and better burndown pace