**Controlled Simulation of Targeted Automation**

Objective

This section demonstrates a practical proof-of-concept for AI-powered workflow automation in the client onboarding process. The goal is to show how a Robotic Process Automation (RPA) bot can reduce manual effort and time for repetitive tasks, using the *Secrets Manager Bot* as a representative example.

Target Task: **Secrets Manager Bot**

Example scenario:

Storing and verifying sensitive onboarding credentials (e.g., API keys, tokens, encryption keys) for bank connectivity.

Manual Baseline Scenario

A human operator would normally:

1. Identify required secrets:  
   Cross-check the onboarding checklist or config files.
2. Log into AWS Console:  
   Open AWS Secrets Manager in a web browser.
3. Verify each secret:
   * Search by name.
   * Open each secret’s details.
   * Compare stored values with current requirements.
4. Create missing secrets:
   * Click *“Store new secret”*, select type.
   * Paste JSON or string.
   * Name the secret and store.
5. Update outdated secrets:
   * Click *“Retrieve secret value”*, edit and save.
6. Repeat for all:  
   For each secret, repeat steps 3–5.

In the test scenario, the manual baseline to verify/update three secrets took ~17 minutes on average (timed by the researcher).

Automated Scenario: **Bot Procedure**

Implementation steps:

1. Input:  
   A secrets\_test.json file lists required secrets and their expected values:

|  |  |  |
| --- | --- | --- |
| **Secret Name** | **Initial Value** | **Purpose** |
| Bot-Testing-1 | { "key": "old-value-1" } | Exists, should be updated |
| Bot-Testing-2 | { "key": "old-value-2" } | Exists, should be updated |
| Bot-Testing-3 | Do not create | Should be created by the bot |

A screen shot of a computer

AI-generated content may be incorrect.

1. Python environment:  
   Local dev environment (VS Code) with boto3 installed and AWS CLI configured.

A screenshot of a computer program

AI-generated content may be incorrect.

1. Script logic:
   * Loop through each secret.
   * Use describe\_secret to check existence.
   * If missing, run create\_secret; if exists, run put\_secret\_value.
2. Run & test:  
   Secrets Bot-Testing-1 and Bot-Testing-2 were pre-created to be updated.  
   Bot-Testing-3 did not exist and was created by the bot.
3. Time it:  
   Bot execution took ~4 seconds for all three secrets.

Result & Impact

* The RPA bot achieved a >95% time saving for this simple scenario.
* The test confirms that even small automation scripts can save significant effort when scaled to dozens of secrets.

|  |  |  |
| --- | --- | --- |
| **Task** | **Manual (avg)** | **Automated Bot** |
| Secrets verification & update | ~17 min | ~4 sec |

A screenshot of a graph

AI-generated content may be incorrect.

Interpretation:

The bar chart illustrates the significant time saving (over 95%) achieved by the RPA bot, highlighting the potential impact of AI-powered automation on repetitive onboarding tasks.

The result demonstrates that even simple Python scripts combined with AWS services can meaningfully reduce repetitive manual work. When scaled across multiple secrets, APIs, or clients, the efficiency gains can be substantial, supporting the thesis objective of evaluating the feasibility and benefits of AI-powered workflow automation.

This experiment supports the potential value of automating repetitive tasks in onboarding workflows. While this is not a full causal estimation, it provides empirical evidence that AI-powered automation can meaningfully reduce manual time.