Reading Guide

Graduation Internship

Dennis Catharina Johannes Kuijs

June 17, 2025

openremote

Contents

1. Context

This document serves as a reading guide and provides an overview of the work I completed over the past few months during my graduation internship. During this period I created several deliverables that proof the HBO-i learning outcomes. This reading guide provides a brief summary for each deliverable. Detailed results can be found in the documents in my portfolio.

Each deliverable will have a reference visible where this document can be found in the portfolio.

Finally, I would like to express my thanks to a number of people including my internship supervisor Pierre Kil, Don Willems and Richard Turner. They provided me with excellent guidance during the internship period. I also want to thank my internship supervisor from Fontys Gertjan Schouten for the guidance during the internship period.

2. Project

2.1. Company

OpenRemote, founded in 2015, is an organization developing an Open-Source IoT platform. The goal of the platform is to simplify the integration of different communication protocols and data sources into one user-friendly system. The platform enables integration with different types of sensors, data sources, (IoT) devices, and allows them to be managed centrally through a user-friendly management portal. The data from these sources can then be visualized and deployed in customized applications and (mobile) apps.

The software is completely Open-Source, allowing anyone to use or contribute to it for free. Currently, OpenRemote is used in various industries, such as the management and monitoring of energy systems, crowd management and vehicle fleets.

2.2. Problem

In addition to the Open-Source product, companies and (government) organizations can choose to have the software managed and maintained (for a fee) by OpenRemote. This so-called 'managed' service is becoming increasingly popular, which brings new challenges. One of the main challenges is maintaining the virtual machines on which the OpenRemote software runs.

These machines regularly need to be updated, ranging from updates to the operating system to situations where multiple (software) packages on the virtual machine need to be updated. In most cases, these updates can be performed without problems and with minimal downtime, leaving the virtual machine intact. However, when too many changes are made to the AWS `CloudFormation` file, Amazon will rebuild the virtual machine, losing all the data.

Because updating is a risky action and there is no way to predict in advance when Amazon will opt for this drastic measure, the process for each customer and virtual machine is currently performed manually through Amazon's management portal, rather than through the `CLI` or the `CI/CD` workflow.

Several steps are performed prior to the update process, including taking a `snapshot` (backup) of the virtual machine. This additional backup, in addition to the automatic daily backups, ensures that customer data remains protected. After the update process, the customer data is then manually restored appropriately so that the customer can continue to use the software without interruption.

Performing these tasks manually makes the process time-consuming, vulnerable and error-prone. As the number of customers for the 'managed' service increases, updating will take more and more time from the team member responsible for this task.

2.3. Solution

OpenRemote is looking for a way to automate the update process for the virtual machines in a scalable, secure and above all reliable way. The solution must be able to be applied to multiple virtual machines simultaneously without further modifications, while ensuring the security of the virtual machines and persisting customer data at all times.

In addition, the solution must integrate seamlessly with existing tools and platforms, such as Amazon's Cloud platform (AWS), on which the virtual machines run, and the `CI/CD` workflow on `GitHub` Actions, which is used to set up customer environments on this cloud platform.

2.4. Goal

The goal of the project is to investigate how the update process of the virtual machines can be automated in a scalable, secure and, above all, reliable way.

2.5. Assignment

Investigate which possibilities exist to automate the update process of the virtual machines in a scalable, secure and most importantly reliable way and develop a `POC` (Proof Of Concept) where the possible solution(s) are tested on a development environment.

3. Process

3.1. Analyse & Design

3.1.1 Projectplan

I started my internship by writing a project plan. This document outlines topics like the assignment, company, scope, research questions etc. This document can be found on the following location within my portfolio: Analyse -> 1. Projectplan

3.1.2. Problem Analysis

To get a better understanding of the problem I'm trying to solve. I analysed the problem using the SPA and MoSCoW method. This document can be found on the following location within my portfolio: Analyse -> 2. Analysedocument

3.1.3. CI/CD Workflow (Decomposition)

After I discussed the projectplan and problemanalysis with my company and university mentor.

I started my research by creating a decomposition of the `CI/CD` workflows `provision_account` and `provision_host` to get a better understanding how their current (`AWS`) infrastructure is functioning. I visualized the whole process by creating a flowchart for each workflow. During this phase I also had meetings with several team members to get an introduction how the codebase is build-up and to get a better understanding of the problem and my assignment.

The diagrams including explanation can be found on the following location within my portfolio: Analyse -> 3. `CI/CD` Workflow (Decomposition)

3.1.4. Decoupling IoT data (Research)

After finishing the decomposition part, I started my research by investigating the possible solutions to decouple the `IoT` data from the virtual machines. I investigated how `EBS` is working for data storage, `DLM` for creating/managing snapshots, the options for integrating the solution in the `CI/CD` workflow and the best way to store the `IoT` data on this `EBS` volume. I created several prototypes, each using a different method, to verify which one is the best for OpenRemote's use case. I discussed my findings with the team and got some valuable feedback to improve my prototypes.

The (technical) results/explanation from this research can be found on the following location within my portfolio: Analyse -> 4. Decoupling IoT data (Research)

3.1.5. User Stories

When I finished my research, I start describing the user stories as `GitHub` issues. At OpenRemote we follow the `GitHub` Flow by defining `issues`, creating `pull-requests`, using `feature branching` and `reviewing` via `Github`. All my (sub)issues can be found in the `GitHub` repository via the links in the assignment.

The assignment can be found on the following location: Analyse -> User stories. I also uploaded pictures for each individual user story in case the links are not available anymore.

3.2. Realise

3.2.1. Separate EBS (data) volume for storing/decoupling the IoT Data (Implementation)

After completing my research and identifying the best prototype, I began the implementation. During this phase I used my own `AWS` account to test and provision the required resources. After the initial implementation, it got reviewed by Richard and I received some feedback to improve the implementation. I responded on every review (comment) with my thoughts about the topic and why I implemented it in the way I did.

The (technical) results/explanation from this implementation can be found on the following location within my portfolio: Realise -> 5. Separate `EBS` (data) volume for storing/decoupling the IoT Data (Implementation)

3.2.2. SSM Automations (Implementation)

When the `EBS` data volume functionality was created, I enhanced the implementation by introducing several automations using `AWS Systems Manager (SSM)`. These automations handle volume `attachment`, `detachment`, `mounting`, `umounting`, and even replacing an existing data volume with a `snapshot`. This setup is particularly valuable during updates or maintenance if an issue occurs, the instance can be quickly reverted to a previous state. Instead of relying on time-consuming manual steps that can increase customer downtime, the recovery process is now streamlined to a single button press. Within approximately `30` seconds, the instance is restored and back online using a `snapshot`.

The (technical) results/explanation from this implementation can be found on the following location within my portfolio: Realise -> 6. `SSM` Automations (Implementation)

3.3. Tests

3.3.1. AWS Production Account (Tests)

The final part of this assignment was all about testing. I developed the implementation using my own AWS account which has a different setup compared to the production account of OpenRemote.

I triggered the `CI/CD` workflow in GitHub using my branch to create a new host (instance) in the `AWS` account from OpenRemote which uses the new `EBS` data volume implementation. When the instance was successfully provisioned I deployed OpenRemote on it and tested all the `SSM` automations, `CloudWatch` metrics, `instance` configuration etc.

When the tests passed successfully, my implementation got merged in the main codebase. This happend on Thursday May 22, 2025 at 5:24 PM

The tests results/explanation can be found on the following location within my portfolio: Realise -> 7. AWS Production Account (Tests)

3.4. Migration

3.4.1. AWS Production Instance (Migration)

After finishing the tests, I worked on migrating an existing production instance to use the new `EBS` data volume implementation.

The (technical) results/explanation from this migration can be found on the following location within my portfolio: Realise -> 8. `AWS` Production Instance (Migration)

4. Conclusion

The project I worked on has been completed, integrated into the main codebase, and is already being used in the first production instances. I'm proud to have achieved this milestone during my internship.

While the core implementation is finished, there are still some potential improvements that could be explored in the future. These include adjusting the number of snapshots retained before deletion, investigating the possibility of attaching an `EBS` data volume to multiple instances simultaneously, and implementing rolling updates using `blue/green deployments` with `EC2` instances. These features aren't critical to the current functionality, but would be valuable enhancements to further automate and optimize the update process.

5. Reflection

I'm very satisfied with the final results of my internship project. During this period, I had the opportunity to learn and work with several new technologies and tools, including `bash`, `Linux`, and the `AWS` cloud platform. I find `AWS` particularly interesting and I'm happy to extend my knowledge of it further. Although I already had some experience with `AWS` from previous semesters and personal projects, this internship gave me valuable insight into how `AWS` is used in a professional environment, especially with real clients who depend on its infrastructure.

I also learned more about `CI/CD` workflows and how they can be created using `bash`. I wrote for the first time `bash` scripts to extend the current `CI/CD` workflow with the changes for the `EBS` data volume implementation. Furthermore, I worked with `linux` on the virtual machines, I learned several new commands to interact with the filesystem and configure block devices.

This knowledge is very useful, as I'm thrilled to continue exploring the world of `Cloud Computing` and `DevOps`.

Of course, there is always room for improvement. Initially, I outlined several research questions, but as the project progressed, I discovered that only the first question was relevant. Based on feedback and new insights from my team members, I explored different topics to enhance and expand my assignment. Spending more time refining the research questions at the beginning would have helped create a more structured and focused approach.

I realize that I should have taken more days off during my internship. My contract included 12.5 spare days, but I only used one of them. Looking back, I definitely overworked myself at times. It's something I'll keep monitoring when I start full-time at OpenRemote.

Lastly, I noticed that sometimes during my research, I initially believed a specific implementation wasn't possible to later discover that it was not only feasible but actually the best option. I realize I may have drawn conclusions too quickly. With a bit more research, such situations can be avoided. This is definitely something I want to improve in the next projects.

The team is happy with the final results. The project was completed within the internship period, is fully functional, and is already being used in a production environment. Additionally, I received an offer to join OpenRemote after graduating, an opportunity I was excited to accept!

6. Learning Outcomes

During my internship I created various `deliverables` that prove the different learning outcomes. In this section I will give you an overview of the learning outcomes and which `deliverables` proving them.

6.1. Analyse

- `2. Analysisdocument` This document describes the problem analysis I created in the first weeks of the internship.

- `User stories (GitHub)` In this assignment you can find links to the different `GitHub` issues I created for this assignment. All the issues are written as user stories with OpenRemote issue template.

- `4. Decoupling IoT data (Research)` In this document I explained the research about decoupling the `IoT` data from the virtual machines and the different prototypes I've created.

- `1. Projectplan` This document describes the inital projectplan I've created in the first weeks of my internship. It outlines topics like the assignment, company, scope, research questions etc.

- `3. CI/CD Workflow (Decomposition)` In this document you can see the flowcharts I've made to visualize and decomposite the working of the different `CI/CD` workflows.

- `User Stories (Pictures)` In this assignment you can find pictures from the GitHub issues I created for this assignment. All the issues are written as user stories with OpenRemote issue template.

6.2. Advise

- `4. Decoupling IoT data (Research)` I ended my research with some advice, which can be found at the bottom of this document.

6.3. Design

- `Architecturediagram - Current Situation - Virtual machines` This diagram illustrates the current architecture based on the AWS CloudFormation template. To better understand its structure and functionality, I have decomposed the CloudFormation template and visualized it accordingly.

- `Diagram - New situation - Virtual machines` This diagram illustrates the proposed new situation where the IoT data is stored on a separate EBS volume.

- `Flowchart - Provision Account - CI/CD Workflow` To gain a better understanding of how the `provision account` `CI/CD` workflow works, I decomposed the bash script and created a flowchart to visualize the process. This `CI/CD` workflow provisions an `AWS` account for the customer and connects it to the primary `AWS` account within the OpenRemote organization.

- `Flowchart - Provision Host - CI/CD Workflow`   To gain a better understanding of how the `provision host` `CI/CD` workflow works, I decomposed the bash script and created a flowchart to visualize the process. This `CI/CD` workflow provisions an `EC2` virtual machine for the customer and configures it with the OpenRemote software.

- `3. CI/CD Workflow (Decomposition)`   In this document you can see the flowcharts I've made to visualize and decomposite the working of the different `CI/CD` workflows.

6.4. Realise

- `Testcases - Prototype - EBS Volume - Video`   During the research phsase I tested my prototypes. In this section you can find a `.zip` file with the video's that demonstrate different test case's/scenario's.

- `Personal GitHub Repository` I used my own `GitHub` repository for storing a copy of the work I've made. A link to the repository can be found here. Some parts of this repository is being merged into the main codebase of OpenRemote.

- `5. Separate EBS (data) volume for storing/decoupling the IoT Data (Implementation)`   In this document, I explained how I tested and implemented the prototype in my own `AWS` account.

- `6. SSM Automations (Implementation)`   This document describes the implementation of the `SSM` automations I've created for automating different tasks regarding the `EBS` data volume implementation.

- `7. AWS Production Account (Tests)`   In this document you can read about the tests I've done in the AWS account from OpenRemote.

- `8. AWS Production Instance (Migration)`   In this document, I explained how I've migrated an production instance using the new `EBS` data volume.

- `Technical Products (.zip)` An export of the technical products I've created during this internship.

6.5. Manage & Control

- `Testcases - Prototype - EBS Volume - Video` During the research phsase I tested my proto-types. In this section you can find a `.zip` file with the video's that demonstrate different test case's/scenario's.

- `Personal GitHub Repository` I used my own `GitHub` repository for storing a copy of the work I've made. A link to the repository can be found here. Some parts of this repository is being merged into the main codebase of OpenRemote.

- `5. Separate EBS (data) volume for storing/decoupling the IoT Data (Implementation)` In this document, I explained how I tested and implemented the prototype in my own `AWS` account.

- `6. SSM Automations (Implementation)` This document describes the implementation of the `SSM` automations I've created for automating different tasks regarding the `EBS` data volume implementation.

- `7. AWS Production Account (Tests)` In this document you can read about the tests I've done in the AWS account from OpenRemote.

- `8. AWS Production Instance (Migration)` In this document, I explained how I've migrated an production instance using the new `EBS` data volume.

- `Technical Products (.zip)` An export of the technical products I've created during this intern-ship.

6.6. Personal leadership

- `Logbook` Every week on Friday, I have written a summary of the tasks I've completed in that week. In this section you can find all the summaries.

- `FeedPulse` An export of my FeedPulse can be found here.

- `Communication` The e-mail communication with my first assessor can be found here.

- `Midterm review - Company Mentor` I asked my company mentor to fill in the reviewdocument for the midterm review, the results are visible in this document.

- `Midterm review - Student` I also filled in the review form myself, it can be found in this docu-ment.

6.7. Professional standard

- `Logbook` Every week on Friday, I have written a summary of the tasks I've completed in that week. In this section you can find all the summaries.

- `FeedPulse` An export of my FeedPulse can be found here.

- `GitHub Projects/Sprint Board` An overview of the `GitHub` projects/sprint board is visible in this section.

- `Midterm Presentation - OpenRemote` This is the PowerPoint presentation I've used for the midterm meetup on April 27

- `User stories (GitHub)` In this assignment you can find links to the different `GitHub` issues I created for this assignment. All the issues are written as user stories with OpenRemote issue template.

- `User Stories (Pictures)` In this assignment you can find pictures from the GitHub issues I created for this assignment. All the issues are written as user stories with OpenRemote issue template.