|  |  |  |
| --- | --- | --- |
|  | | |
|  | | |
|  | | |
|  | | |
|  | | |
|  | | |
| **SCHOOL OF INFOCOMM** | | |
|  | | |
| **Diploma in Information Technology** | | |
|  | | |
| **C936 INDUSTRY IMMERSION PROGRAMME II (IIP2) REPORT** | | |
|  | | |
| RP Logo - Full Colour | | |
|  | | |
|  | | |
|  | | |
|  | | |
|  | | |
|  | | |
|  | | |
|  | | |
|  | | |
|  | | |
|  | | |
| Student Name | Darrel Dominic Lim De Xiang |  |
|  | |  |
| Student ID | 22030316 |  |
|  | |  |
| Internship Period | 13th March 2024 to 10th Jan 2025 |  |
|  | | |
| Company Name | NCS |  |
|  | | |
| Supervisor Name | Xander Lumapac |  |
|  | |  |
| Supervisor Email Address | xander.lumapac@ncs.com.sg |  |
|  | | |
|  | |  |

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
|  | **PAGE** |
| ACKNOWLEDGMENTS | **TODO** |
| EXECUTIVE SUMMARY |  |
| 1. INTRODUCTION |  |
| 1. BACKGROUND INFORMATION OF ORGANISATION |  |
| 1. MAIN TASKS DONE AND SKILLS ACQUIRED IN C935 IIP1 AND C936 IIP2 |  |
| 1. ELABORATION ON TRAINING AND WORK ASSIGNMENTS FOR C936 IIP2 |  |
| 1. REFLECTIONS |  |
| 1. CONCLUSIONS |  |
| REFERENCES |  |
| APPENDIXES |  |
| 1. Internship Report Clearance Form |  |

**Guidelines on writing the IIP2 Report**

1. **All guidelines provided in the document (identified with a red font) must be removed from the final report.**
2. Students are required to submit a minimum 1000-words report on their industry immersion experience by the submission deadline. Extensions for late submissions must be approved in advance. Additionally, students must obtain clearance for their report from their company supervisor using the ***Internship Report Clearance Form*** and should factor this into their timeline to meet the submission deadline.
3. Section requirements are given below. Guidelines for each section are provided at each section where necessary.

* Cover Page and Table of Contents
* Acknowledgements
* Executive Summary
* Main Text - Introduction, Background Information of Organisation, Main Tasks Done and Skills Acquired in C935 IIP1 and C936 IIP2, Elaboration on Training and Work Assignments for C936 IIP2, Reflections and Conclusions
* References
* Appendixes

1. Font and formatting requirements:

* Font - **Times New Roman**, size **12**, color **Black**.
* Formatting - line spacing of **1.5 lines** between lines and double-spaced between paragraphs.

1. The report must be presented with the following in mind:

* **Adherence to the requirements** (required sections, font and formatting).
* **Clarity and conciseness** - Adopt a straightforward and functional writing style; describe all relevant facts, experiences, and observations without including extraneous information.
* **Structure** - report is well-organized and content is presented coherently.
* **Readability** - Ensure sentences are grammatically correct, words are spelled correctly, and punctuation marks are properly placed. The report should not be written in point form.

1. Other requirements:

* Figures and tables may be used to enhance clarity, structure and / or readability.
* Additional appendixes, if required, should be brief and contain information that supports the main text but is too detailed for inclusion in the body of the report. This could be graphs, charts, tables or additional data that was referenced in the report. References to the appendixes should be made in the main text where relevant.
* Students are required to save their report as a Microsoft Word Document. The filename of the document must follow the format given below:

*C936-IIP2-Student ID-Student Name (e.g. C936-IIP2-22001234-Alfred Tan)*

**ACKNOWLEDGEMENTS**

Throughout my 1 Year Internship, I have received many helps and guidance in my IRAS IRIN3 project and would like to thank the following people:  
From NCS TA Team: Sithu, Velan and Bryan

From NCS DevOps Team: Naveen, Rajesh, Kiran, Surya, Fernando, Shafeeq, Kai Xiong, Maha and Xander

From ACN: Yee May, Rakesh and Pirathepan  
From IRAS: Jonathan, Jun Feng, Mervyn and Xiao Jing

All of the people listed above have played a significant role in guiding me and imparting knowledge in various areas. Without their support, I would not be where I am today.

Amongst those listed, I would like to specifically thank Xander, Fernando, Jonathan, Jun Feng, and Sithu. These individuals have consistently offered their help, answered all my questions, encouraged my curiosity, and provided thought-provoking insights that have broadened my perspective. In return, they have helped me to think more critically.

Lastly, I would like to thank Republic Polytechnic and NCS for offering me this internship opportunity.

NCS has kindly accepted my request to continue this internship, which extends till 28 February.

**EXECUTIVE SUMMARY**

< Provide a summary (one page or less) that captures the key points of your report. Mention the main aspects of your internship, including the organization, your role, main responsibilities, major learnings, and your overall evaluation of the experience.

You may make improvements and updates to the text you submitted for your C935 IIP1 report.

This section should be done last as it summarises the completed report. >

**I INTRODUCTION**

The purpose of this report is to consolidate and reflect on my experiences during my internship at NCS, where I was contracted to IRAS to work on the IRIN3 project. This project went live on August 6th, 2 days before national day. Afterwards, I was transferred to the second phase of the IRIN3 project, titled CDE.

So, what is NCS and what do they do? NCS (Formerly known as National Computer System) provides a wide range of technological services to its partners, including cloud solutions, cybersecurity, engineering, and infrastructure.

IRAS oversees tax collection and regulation, ensuring compliance and efficient tax management in Singapore.

As a DSO Engineer, my team was responsible for ensuring that the pipelines and build agents were operational and available for application team to use for code deployment and scanning. We maintain and upgrade these systems, ensuring that the pipeline is always up to date and able to deploy the application code free of vulnerabilities to deploy to various environments. We also work with other departments in IRAS to ensure that the tools, extensions, and software that we use are free of vulnerabilities and are patched if there are any before incorporating it into our project, as well as troubleshooting services when there are deployment failures or raising service request.

**II BACKGROUND INFORMATION OF ORGANISATION**

NCS was founded in 1981 and has a strong presence in the Asia-Pacific region, delivering end-to-end ICT solutions to various sectors. Its mission is to advance communities and business by harnessing the power of technology to improve lives and drive business success, by bring people together to make the extraordinary happen. NCS has grown significantly, establishing itself as a trusted partner for digital transformation initiatives. Its market position over the years is reinforced by strategic partnerships, continuous innovation, and a commitment to customer satisfaction.

Currently, it has over 13,000 talents across Singapore, Australia, Hong Kong, China, and India, with more than 4,000 active projects, 57 areas of specialization, and operations in more than 20 countries within the Asia-Pacific region.

In 2023, NCS was the Market Leader in IT services in Singapore with a 6.4% in Market Shares, and 16% Market Shares in Southeast Asia.

NCS company organisational structure is broken down into 8 different sections:

Product & Engineering, Finance, HR, Legal, Marketing, Health, Operations and Sales

The Inland Revenue Authority of Singapore (IRAS) was established in 1960 as a statutory board under the Ministry of Finance (MOF) and plays a critical role in ensuring the efficient and effective collection of taxes to fund the government in developing Singapore. Their mission is to act as an agent of the Government and provide services in the administration of taxes and enterprise disbursements, as well as to represent Singapore internationally on matters related to taxation.

IRAS has evolved to become a modern tax administration, leveraging technology and data analytics to enhance its services and ensure a high level of compliance. Its proactive approach and focus on innovation have positioned IRAS as a model tax authority globally, with a vision to be the leading revenue authority in the world. Provided below is the organizational structure taken from their webpage.A diagram of a company's company's company

Description automatically generated

**III MAIN TASKS DONE AND SKILLS ACQUIRED IN C935 IIP1 AND C936 IIP2**

|  |  |  |
| --- | --- | --- |
| **C935 IIP1** | | |
| **Main tasks** | **Skills acquired** | **Brief description of training** |
| * Set up pipeline for Azure Datafactory (ADF) D365 | * Bash Scripting to replace values. * Troubleshooting pipeline for error * Learn how to code in YAML | * Analysis ADF pipeline and add D365 to it * Collaborate with App Team to create the pipeline and test it * Work with App team to troubleshoot deployment and check if values are updated |
| * Debug Agent memory issue when running Mend, commonly seen for DNP Notification Pipeline | * Run Kusto query to check agent logs. * Run Kubectl to enter the agent * General debugging in the agent as well as in the pipeline | * Monitor the DNP Notification pipeline during Mend Scan to find any memory issues. * Retrieve the logs to see the issue. * Try various method to fix the issue |
| * Set up Pipeline for D365 SaaS | * PowerShell scripting * Reverse Engineering the pipeline to understand | * Reverse engineer the pipeline to understand end to end flow * Collaborate with apps team * Update PowerShell values as they are deprecated * Test pipeline deployment |
| * Set up SonarQube (SQ) | * Set up SQ UI and assign the relevant permissions | * Learn how to navigate the SQ Portal from an admin perspective * Create group and permission template * Update Quality Gate rules |
|  | | |
| **C936 IIP2** | | |
| **Main tasks** | **Skills acquired** | **Description of training** |
| * Set up Microservice (MS) Pipeline | * Run kubectl commands (cmd) to check the application pod * Use pipeline to upload to Azure Container Registry * Use pipeline to deploy API Policies to API Management (APIM) Service | * Update the MS pipeline to support the 2nd phase (CDE) of the project * Ensure pipeline is able to deploy to APIM * Able to update the L4 Repo with HelmRelease file, and that Azure Kubernetes Service (AKS) is able to read the deploy the configuration |
| * Set up ADF Pipelines | * **Same as C935** | * Update the ADF Pipeline to support CDE * Work with the different ADF team for deployment |
| * Set up D365 pipeline | * **Same as C935** | * Continue troubleshooting and developing D365 pipeline for CDE |
| * Set up Database/Storedproc pipeline | * Set up pipeline to validate the schema script and deploy * Learnt how to fill up an Access Control Matrix | * Update Database/Storedproc pipelines for CDE * Ensure schema can be deployed |
| * Set up Powerbi SaaS | * Set up pipeline to connect to PowerBi workspace using SPN and deploy | * Check that agent can use SPN to connect to Synapse workspace, which is a SaaS endpoint. * Check if able to update the tables |
| * Update Warrant of Fitness (WOF) for our build agents | * Leant various ways to harden our agent | * Update the WOF for CDE type 1 and 2 build agents |
| * Create network diagram for ADF and D365 SaaS | * Learn how to visualise and create an architecture network diagram | * Create a network diagram for D365 and ADF in our DevOps network |

**IV ELABORATION ON TRAINING AND WORK ASSIGNMENTS FOR C936 IIP2**

< Your report must address all the guidelines given below.

Provide a summary of the work assignments for C935 IIP1 **as context** for the subsequent elaboration on training and work assignments for C936 IIP2.

**With respect to the main tasks done and skills acquired in C936 IIP2**:

* Identification of prior knowledge - describe the knowledge and skills that you acquired in RP. Use broad descriptive categories where possible.
* Identification of knowledge gaps - describe the knowledge and skills gap that you had. Use specific examples where possible.
* Description of learning - report on the training and learning processes you underwent to acquire the necessary skills and knowledge.
* Application of learning - report on how you applied your prior knowledge and new learning to your job role, work responsibilities and tasks. This includes describing your major contributions. For each major contribution, explain why:
  1. the contribution is of value to the company
  2. the contribution is of value to you as an individual >

Throughout my internship, I worked on the IRAS IRIN3 project, which is divided into different phases. AB is the phase before the project went live, and CDE is the phase for continued development after it went live. In C935, I was in the AB phase, and in C936, I was in the CDE phase.

Before I get started on my tasks, please refer to the appendix section, where there is a PowerPoint slide with diagrams and screenshots to help elaborate and visualize what is described here. Refer to it as you go through this section.

Let’s start with the repositories (Repos) that I used: Controller, L4, and Pipeline Template (refer to slides 1-3). These three Repos are the core of our pipeline; without any of them, our pipeline would not run.

During AB phase, I mainly worked on four different tasks: ADF, D365 SaaS, Mend memory issue and setting up SonarQube for C5 UDP (different project)

**ADF** (slides 10-13):

I was tasked to incorporate a new Data Factory for D365 into the pipeline, where it would deploy the ARM Template. This contains the application team pipeline and configuration, stored as a JSON file. There was a previous issue where the ARM Template file size was above the 4MB limit, causing deployment to fail. Originally, D365 was part of Data Migration, but it was split off into another Data Factory as a workaround. In AB, I managed to deploy the ARM Template and minify the JSON file, making it smaller.

**D365 SaaS** (slides 14-17):

I was tasked with onboarding it to Government on Commercial Cloud (GCC 2.0). Since the pipeline had not been used for quite some time and was outdated, I had to reverse engineer it to understand its overall functionality and code structure. Due to the project environment, the previous DSO person who worked on this was no longer in our department, leaving me to figure this out by myself. Since D365 uses Windows Type 3 VM, our agent had to be upgraded to support GCC 2.0 as well. Eventually, I was able to get the pipeline to work, but not without issues. Since our pipeline does code scanning and the last run of the pipeline was in 2023, the new application code had many vulnerabilities, which caused a blocker for the pipeline as it could not progress. This issue only allowed me to deploy to the SIT environment, but not the upper environments. Unfortunately, the application code was not fixed in time to test further deployment to upper environments in AB as it was near the go-live period, so the CD pipeline was incomplete.

**Mend** (slide 24):

There was an issue where the pipeline build agent would time out and abandon the pipeline task. This occurred commonly for the DNP Notification Pipeline during the Mend CI job, which would take more than an hour to run. Using Kusto Query to check the build agent log and resources, it was observed that the memory usage was quite high when running the task, and some instances had out-of-memory errors. I also found a part of the code where it did repeated scanning of the files, scanning twice instead of once.   
Hence, the fix I introduced was to:

1) Remove the duplicate scan,

2) Set a memory limit on the task to prevent it from exceeding the threshold,

3) Set a retry of three times if the task fails.

**SonarQube Portal:**I was given a temporary admin account to the C5 UDP SonarQube Portal, where I created different groups and group template, as well as assigning the respective permission to the different group to limit their access base on their roles. I also modified the SonarQube Quality gate policies to match IRIN3, and worked with the different project team to ensure they are able to view the SonarQube and have their pipeline upload the results there.

Moving onto C936, I was transferred to the CDE phase of the project, where most of the resources and environments are new. During this time, I worked on pipelines that I am familiar with, and those that are new to me. I will be covering Microservice, Database/Storedproc and PowerBI. I will also be mentioning a bit of D365 and ADF

**Microservice Pipeline** (Slide 7-9):

The Microservice Pipeline deploys three key components: Application pod to AKS, API and Swagger files to the APIM Service.

Before I could start on the pipeline, since all the resources were new, I had to raise several service requests (called AMP Requests, similar to what was covered in C381 when we simulate the IT HelpDesk) for our DevOps CDE Build Agent to have network connectivity to the different Azure resources, and for our Azure Resource Group to be tied to it. This was necessary to allow our build agent to have connectivity and access to the Azure resources for deployment. While waiting for the requests to be fulfilled, I worked with my supervisor, Xander, on getting the new CDE values. From there, I went through the pipeline files and updated the values.

Once that was done and the AMP Requests were fulfilled, I tested the pipeline CICD, where we encountered multiple issues. The first issue was missing Swagger and API policy files from the application team’s side in L4. Since the application team needed time to port over and update the files from AB, I created a dummy API policy file to test. The pipeline uses bash scripting (learned in C330/C226) to target the different files. For testing purposes, I updated the path file to point to my dummy file. After a few more errors and tries, I was able to update the APIM Service with my dummy values.

Next was the application container image. In their repo, it uses Docker files (briefly mentioned in C381/C330), where it pulls the base container image from the staging ACR. It then builds, scans, and signs the application container image. The pipeline uses the “az acr import” CLI to import the container image into the CDE ACR, where it is used in the CD Pipeline. It also adds the related certificate into the images to authenticate access to the necessary website.

How the image gets deployed into AKS:

In L4, under app-config/{env}/backend/{app}, there is a Helm Release (HR) file. The HR file contains all the configuration to be deployed into AKS. During the CD process, the pipeline deployment job bakes the app-config configuration, updates it with the actual container and controller configuration, and places it into the release/{env}/backend/{app} folder. In AKS, we use a GitOps tool called Flux. In summary, Flux reads the .release folder every 10 minutes and checks for updates. If there is a new update, it triggers the kustomization controller to deploy the new HR file, triggering a “Helm Reconciliation,” meaning AKS is deploying the new HR configurations.

However, there were issues with the Helm Release at first. Since there is a kustomization file in L4 that specifies which path to look for updates, if the application folder is not commented in the kustomization file, it will try to deploy a non-existing file, causing Flux to have an error and stop reconciliation.

Additionally, all the application pods were failing when deployed, and they did not have access to the AKS Role-Based Access Control (RBAC) to enter AKS and check the logs. To help with these issues, Jun Feng and I worked on some improvements to enhance the pipeline. I added a Kubernetes function to summarize the total number of running pods and display the error logs of those having issues in the pipeline. I worked with Jun Feng and Yee May, the Application Lead, to test if we could add a Helm hook to allow the migration pod to self-recover when there is an error, which, after testing, worked. After these two implementations and a few more adjustments, the pipeline was operational for deployment. We shared the enhancements with the application teams, enabling them to self-help and check errors, and most importantly, deploy their product in CDE.

Overall, working on the microservice pipeline was a massive headache in a good way, as there were many concepts I did not know about and had to ask my colleagues for help to troubleshoot the issues. It was also the most impactful task for my character growth throughout the project, as it wasn’t as stressful and didn’t require rushing to meet deadlines by go-live in AB. It allowed me to revisit the fundamentals, rebuild concepts, and reflect on what I could have done better in my previous tasks. I also started to question the pipeline more, in the sense of why it is structured that way. The more I questioned and understood, the more confident I became in taking charge of making changes to it, such as the function I made. I improved my collaboration skills, checking in with the application team and helping when I could.

**Database/Storedproc** (Slide 18-20):

The Database (originally known as Storedproc) pipeline deploys app’s team SQL file to their databases.

Originally there was no timeline and no progression for the pipeline to be set up yet, but I saw there were a few app’s member started to ask when will the pipeline be ready to the Dev Env. After checking with my supervisor that no one has taken it yet since there was no discussion, I took the initiative to start working on the database pipeline and planned a deadline for the pipeline to be set up.

Similar to Microservice, the pipeline and database resource was new which needed AMP request to be raised for the proper permissions and connectivity. Additionally, I needed to fill up an Access Control Matrix (ACM), something that was new to me. This ACM was used to track what permission is needed for the user to access the resources. In this case our MSI to access the different app database and have the necessary permissions to deploy the SQL files to it.

After updating the pipeline values, I worked with apps team to use a repo to test the pipeline CICD, where I created my own dummy SQL script to test deployment while waiting for the AMP to be fulfilled. While doing so, I started to question the permission that we are giving to the MSI since it was mainly to edit the schema. No creation or deletion of schema tables which can be included as from what I learn in C207. After checking around, the pipeline is only to deploy, and update exist table. If tables are needed be to create or removed, app team will manually go in and remove them.  
There was some trouble with the implementation of the AMP but eventually the DBA Team managed to implement it and we could test the pipeline CICD. After further testing, pipeline was able to deploy to the target database successfully, in which I broadcast an announcement to the Help Desk that the database pipeline to DEV env was ready.

This

**V REFLECTIONS**

< Your report must address these reflections:

* With respect to your personal, education and professional career goals, reflect on the value and impact of the internship experience gained in C936 **compared to** the internship experience gained in C935
* Evaluate if your performance in C936 IIP2 was better / unchanged / worse than your performance in C935 IIP1, and elaborate on your evaluation
* Discuss 3 things you could have done to extract more value from the internship experience

You may provide any other meaningful insights and reflections here. >

**VI CONCLUSIONS**

< This section is to provide your conclusions on the overall internship experience with respect to your:

* work responsibilities
* work experience gained
* training received
* learning achieved >

**REFERENCES**

<References in your report to manuals and other pertinent literature, if any, should be listed here.>

**https://learn.microsoft.com/en-us/azure/container-registry/container-registry-import-images?tabs=azure-cli**

https://helm.sh/docs/topics/charts\_hooks/

**APPENDIXES**

<At a minimum, you are required to attach the following documents:

* Internship Report Clearance Form

Additional appendixes, if required, should be brief and contain information that supports the main text but is too detailed for inclusion in the body of the report. This could be graphs, charts, tables or additional data that was referenced in the report. References to the appendixes should be made in the main text where relevant.

>

Internship Report Clearance Form

<Attach file here>