INSY 5399 - Section 001

GRADUATE INFORMATION SYSTEMS INTERNSHIP

Spring 2023

Internship report

Date of submission: 05/04/2023

**Submitted By:**

Ankita Phadke (1001963738)

Table of Contents

[Internship report: 3](#_Toc134048793)

[Executive Summary: 3](#_Toc134048794)

[Academic Application: 4](#_Toc134048795)

[Academic applications/concepts being used and why are those effective: 4](#_Toc134048796)

[Management principles and tools: 4](#_Toc134048797)

[Technology processes and tools: 4](#_Toc134048798)

[Academic principles not being used/why can those be beneficial to be used: 6](#_Toc134048799)

[Significant learning experience: 6](#_Toc134048800)

[Student Recommendation: 15](#_Toc134048801)

[Recommendations: 15](#_Toc134048802)

[Suggestions about changes to the program: 16](#_Toc134048803)

[Weekly reporting & status updates: 16](#_Toc134048804)

# **Internship report:**

## **Executive Summary:**

This report is about my Spring’23 internship/research work experience under the guidance of Prof. Dr. Santoso Budiman. I have summarized details about the areas of research work, my internship experience, learning curve, technology areas that I did the research on, & my overall learning experience in this report. My internship work is concentrated around research on AWS Cloud technology and its utilization in real-time work environment from the software development perspective & work profile. Main work responsibilities were around technologies/ tools like ‘AWS’ cloud computing services, programming & database, software IDE & code repositories. More details on the work areas are provided in the later sections.

I got a chance to work on multiple technologies and tasks during my internship work tenure. It involved research on software development platforms and tools available in the market, in-depth study on comparison of their features, finding the best suitable tool as per requirement, integrating the tool with code repositories & version control, AWS cloud services integration, writing Python scripts, database management, HTML & JavaScript code pieces and integrating all these together for serving multiple tasks. The research work involved weekly checkpoint & status update calls with Professor to discuss progress.

Overall, the research work & all the tasks I worked on added a very valuable experience in my professional career and expertise. I got a chance to work directly under the guidance of my Professor Dr. Budiman who has a wide variety of industry experience, and I got an opportunity to learn directly from him. I would like to thank my department, internship coordinator Dr. Budiman, and department professors for their continued assistance throughout the process.

## **Academic Application:**

### **Academic applications/concepts being used and why are those effective:**

The internship/research work tenure was overall a very enriching experience from the academic learning standpoint. I learnt a lot about how the theories and technologies learnt in academic settings are used and subject matters are applied in the industry. I also got an opportunity to work on many of those. This is minutely related to the courses taught during the master’s program curriculum, like Cloud computing, Python programming, Database management, Project management, etc. and helped me learn the practical applications of each.

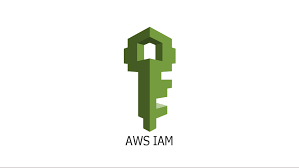
### **Management principles and tools:**

While working on different tasks of the research work, I learnt to apply the concepts of work, project management fundamentals. The importance of considering aspects like mutual comparison of available software options, their features, cost involved, support & feasibility of features in free versions of software, popularity & technical opinions from industry/users, integration with other software used at various stages of development, pros vs cons. One example of such task is study of Jupyter Notebook, Visual Studio, & Visual Studio Code IDE. Other than this, I learnt the importance of creating technical documentation of the tasks that I worked on and how the detailed stepwise description can be documented & is useful.

During the weekly status update call, I could get an opportunity to present my findings, discuss the impediments, receive feedback, and discuss the work items for subsequent week & obtain feedback on the action items. All these closely relate to the daily tasks & project management fundamentals.

### **Technology processes and tools:**

In the technology area, below are the tools and technologies that I researched on.

1. **AWS**: 

The main tool that I used in depth is AWS cloud computing platform. AWS is a commonly used & most popular cloud computing services provider, used by companies all around the world in data storage, backup-recovery, analytics, e-commerce, software development etc. The main types of cloud computing services are IaaS (Infrastructure-as-a-Service), PaaS (Platforms-as-a-Service), & SaaS (Software-as-a-Service). I researched multiple important services provided by AWS platform, which are listed in subsequent sections. I learnt more about the various services provided by AWS namely, EC2, VPC, RDS, DynamoDB, Amazon SageMaker, nginx services integration, AWS Command Line Interface (CLI), AWS Boto3 extension, AWS Software Development Kit (SDK), AWS Cloud9 platform, IAM policies & access control, and so on.

1. **Github**: 

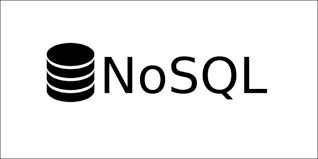
Github is an open-source tool that is widely used in software development & version-control. I studied specific use cases like resolving merge conflict, usability & popularity of github, its integration with other tools/AWS services.

1. **Software IDEs namely Visual Studio Visual Studio - Wikipedia, Visual Studio Code Microsoft Apps, Jupyter Notebook **:

The focus of this task was to compare & contrast which one of the three code editors/IDEs are the most popular & useful for code development.

1. **Software programming languages**:  

I used software languages like **Python** to facilitate use of AWS services through above software IDEs like DynamoDB, RDS, Cloud9, EC2, & languages **JavaScript**, **Node.js, HTML** to use & test AWS services and their outputs like, DynamoDB, nginx etc.

1. **Database Management** **Software**: 

This involved CRUD (create, read, update, delete) operations on the AWS database tables, writing queries, on relational & non-relational DB.

### **Academic principles not being used/why can those be beneficial to be used:**

One of the major initiatives that I implemented during my work and tasks items was preparing the technical documentations for all my work items. This involved making use of Microsoft Office & Lucid chart software. I successfully prepared word documents, presentation documents, & scripts that would be an easy reference to whoever would look at those activities later or try to modify any aspects of that work. It also included applying technical knowledge on scripting tools like Python which can be used for smaller and simpler AWS tasks.

## **Significant learning experience:**

The internship tenure contributed to my learning experience and expertise at multifold levels. In addition to helping me polish my technical skills, the internship experience helped me to learn a lot of technical and non-technical aspects during my research work experience.

Below is a detailed description of tasks that I worked on & summary of each.

1. **Task: Resolving merge conflicts in github**:

**Findings**: As part of this, I created a merge conflict situation in github code repository that typically occurs when multiple users are working on same lines of code in parallel & try to commit changes same time. Error is thrown by github as shown in the below figure 1, to prevent the user from committing changes.

A screenshot of a computer

Description automatically generated with medium confidence

**Fig. 1** Resolving merge conflicts in github

**Conclusion:** In such situations, github prevents from overwriting changes & forces the 2nd user to resolve the merge conflict so that no code changes are accidentally overwritten/removed.

1. **Task: Comparative analysis & study on the IDEs:**

**Findings:** As part of this, Visual Studio (VS), Visual Studio Code (VSCode), Jupyter Notebook were compared based on github, AWS, CLI, DynamoDB & Python compatibility, cost, pros/cons, support. Also, documented the steps involved in connecting these editors with AWS CLI, DynamoDB & github using Toolkit support.

|  |  |  |  |
| --- | --- | --- | --- |
| **Comparison based on** | **Visual Studio** | **Visual Studio Code** | **Jupyter Notebook** |
| **github/code editing** | Available | Available | Available |
| **AWS CLI/Python** | Not available on MAC,  AWS Toolkit extension using IAM user | All OS, AWS Toolkit extension using IAM user | SageMaker Studio using IAM user & relevant policy access |
| **AWS DynamoDB** | All DB operations | Read-only | All DB operations |
| **Industry usage** | Debugging/Application development | Data Science applications, Jupyter Notebook support | Data Science/Machine Learning operations |
| **OS Support** | Only Windows & MAC | Windows, Linux & MAC | Windows, Linux & MAC |
| **CI/CD application** | In-built Azure DevOps | Azure tools extensions | Using Amazon Sagemaker |
| **Cost involved** | Free for community only | Completely free of cost | Completely free of cost |

**Conclusion:** Based on the comparative study, we decided to continue our research on Visual Studio Code.

1. **Task: Launching new EC2, VPC using VSCode using Python & AWS CLI:**

**Findings:** This research was pursued from the standpoint of new EC2 & VPC using Python code & AWS CLI, to check the feasibility of using VSCode for software code development.

Using VSCode: Using CLI, AWS config setup needs to be done first. Either using Python or AWS CLI, ‘create vpc’ or ‘ec2 run-instances’ commands should be run for setting up new EC2 & VPC. For Python, AWS boto3 package needs to be installed. The steps involve logging in to the VSCode with IAM user, using access key & secret access key. There are 2 ways in which this setup can be done:

1. Install Python & pip, boto3 on Windows, and use those from VSC
2. Install pip & boto3 inside VSCode, this is a more convenient option since this allows flexibility to support different Python versions on Windows & inside VSCode as per the requirements of the system.

Using Jupyter Notebook: Using CLI, AWS config setup needs to be done first. Either using Python or AWS CLI, ‘create vpc’ or ‘ec2 run-instances’ commands should be run for setting up new EC2 & VPC. For Python, AWS boto3 package needs to be installed, and Amazon SageMaker service can be used. For using SageMaker, it is important to have an IAM user registered, which SageMakerFullAccess policy associated with the selected user/role.

**Conclusion:** Using VSCode, new EC2 & VPC can be launched using both Python & AWS CLI, it is a better option than using Jupyter Notebook through Amazon SageMaker studio.

1. **Task: Connecting to RDS from VSCode**:

**Findings**: For connecting to an existing RDS database, extensions like SQLTools, need to be installed in VSC via Marketplace. For connecting to the RDS through Python code, Python package: PyMySQL needs to be installed, which contains different methods to connect to the database, and allows users to perform CRUD operations on tables with ease.

CRUD operations include:

1. **C**reate: creation of table,
2. **R**ead: running select query to read the table contents,
3. **U**pdate: update, insert, modify the data inside the table,
4. **D**elete: deleting the created table

A screenshot of a computer

Description automatically generated

**Fig. 2** Connecting to RDS from VSCode

A screenshot of a computer

Description automatically generated

**Fig. 3** CRUD operations on RDS from VSCode

**Conclusion**: Since the option to perform CRUD operations on Amazon RDS is not available in AWS Toolkit for VSC, using Python script to connect to the database is a better option that suits all the requirements.

1. **Task: Accessing EC2 from VSCode:**

**Findings:** Along with the setup needed for using IAM user in VSCode, it is also important that the selected EC2 to be accessed using VSCode should have SSH access enabled in the Security group linked with EC2.

Inside VSCode, an extension named ‘Remote-SSH’ needs to be installed from the marketplace, which enables VSCode to connect to machines via SSH protocol. Once the SSH host is connected inside VSCode, created a text file for testing & ensured that it can be accessed from the EC2 directly from AWS management console.

A screenshot of a computer

Description automatically generated with medium confidence

**Fig. 4** Connecting to EC2 inside VSCode using SSH protocol

A screenshot of a computer program

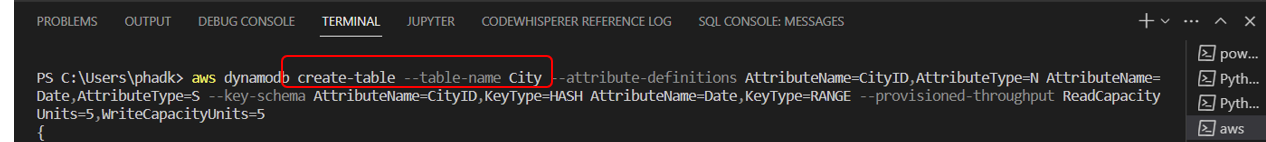
Description automatically generated with medium confidence

**Fig. 5** Files are synced when checked from AWS Management Console

**Conclusion:** VSCode can connect to AWS EC2 simply using SSH protocol.

1. **Task: Accessing Cloud9 from VSCode**:

**Findings**: AWS Cloud9 is an online IDE published for Open-source use. I tested the connection from VSCode to Cloud9 and could successfully connect using EC2, SSH and IAM user as used in previous steps. The Cloud9 environment is associated with an existing/new EC2 which can in turn be accessed from VSCode via SSH protocol. In similar way as the previous test case, SSH access from public IP addresses needs to be enabled for this EC2. Similarly, once Cloud9 environment is up & running, the same IAM user used to login to the VSCode needs to be associated with the Cloud9 and enable the user to use the same from VSCode.



**Fig. 6** Created table in dynamodb from VSCode, using above setup

A screenshot of a computer

Description automatically generated with medium confidence

**Fig. 7** Newly created table can be accessed from Cloud9

**Conclusion**: Using the EC2, SSH, IAM user, every service inside AWS Cloud9 environment can be accessed from VSCode.

1. **Task: Accessing DynamoDB from VSCode using Python via EC2**:

**Findings**: This task involved connecting from VSCode running Python script to the DynamoDB table inside AWS. This is facilitated through the IAM profile already logged in to the VSCode & SSH connection to the EC2. Other than these setups, we need to install Python, pip, boto3 & AWS CLI inside the ubuntu EC2 machine and run the Python script there, which contains the table CRUD operations scripts.

A picture containing text, software, multimedia software, graphics software

Description automatically generated

**Fig. 8** Installing pip inside EC2, using VSCode, ubuntu commands.

A screenshot of a computer

Description automatically generated with medium confidence

**Fig. 9** Installing boto3 inside EC2, using VSCode, ubuntu commands.

A picture containing text, screenshot, font, information

Description automatically generated

**Fig. 10** Installing AWS CLI inside EC2, using VSCode, ubuntu commands.

**Conclusion**: DynamoDB can be accessed using EC2, by connecting to the EC2 machine using SSH connection & running the Python scripts to create/modify tables there, on the ubuntu machine.

1. **Task: Accessing DynamoDB from VSCode without using EC2**:

**Findings**: This task involved steps to create, insert, query, create global secondary index, query global secondary for a DynamoDB using python scripts in laptop using VS Code, without using EC2. The important setup here is the IAM user should be logged in to the VSCode and simply creating a new Python script for the CRUD operations on the DynamoDB can suffice. All the operations can be executed from local VSCode copy & executing commands from the VSCode terminal.

**Conclusion**: DynamoDB table operations can be done successfully directly from VSCode, without using EC2 connectivity, only using the IAM user login.

1. **Task: AWS ReadyOnly test on DynamoDB:**

**Findings**: This task involved testing DynamoDB access permissions from VSCode, after IAM policy is changed to allow read-only access to DynamoDB instead of full access to DynamoDB. The IAM user is first logged into VSCode with full access to DynamoDB & created a table. Once done, changed the policy access from ‘DynamoDBFullAccess’ to ‘AmazonDynamoDBReadOnlyAccess’, also removed the ‘AdministratorAccess’ associated with the IAM user. After these changes were made, the IAM user was prevented from inserting data into the DynamoDB table, as expected. Below error was thrown.

A screenshot of a computer program

Description automatically generated with medium confidence

**Fig. 11** Dynamo access test after changing IAM policy

**Conclusion**: After restricting the DynamoDB access to IAM user, through policy usage, access could be restricted & same was tested successfully through VSCode.

1. **Task: Accessing DynamoDB using nginx on EC2**:

**Findings**: This task involved accessing DynamoDB table using nginx on EC2, calling queries from browser to query items in the DynamoDB. This involved installing ‘nginx’ on EC2 using the steps below. Using the index.html file called after opening the public IP address of EC2, via nginx, the browser loads the requested data from DynamoDB table: ‘books\_test’ for author = ‘John Grisham’. The complete operation involves below steps:

1. Import the AWS SDK Javascript file.
2. Get the request parameter from the browser URL (author name in our case)
3. Update AWS Config – enter Region, access & secret access Key.
4. Create DynamoDB API object.
5. Note the public IP address to run inside the browser.
6. Enter parameters: FilterExpression (author name), ExpressionAttributeValues, ProjectionExpression (columns data to be retrieved), TableName
7. Run the scan function and fetch the data.
8. Set the field values into corresponding HTML page values.
9. Inside the HTML body tag, set the field ids for each of the fields in table, so that those can be mapped with the script field names.

A picture containing text, screenshot, font

Description automatically generated

**Fig. 12** Installing nginx on EC2.

A close-up of a computer screen

Description automatically generated with low confidence

**Fig. 13** Browser shows output of the DynamoDB query call from HTML file.

**Conclusion**: Using the AWS SDK, node.js & HTML script, DynamoDB can be queried & data can be rendered inside HTML page on browser to show output of the database query.

## **Student Recommendation:**

### **Recommendations:**

Based on my overall experience and learning that I achieved through this research/internship program, I would recommend other students to take up internships and gather experience that would help solidify the foundations of learning and would be useful in subsequent professional career paths. As my report highlights, my learning through the internship experience has contributed to my overall development at multiple levels. The technical learning, tools hands-on experience was the most important take-away from this program.

The internship experience helped me to learn in-depth & apply the academic fundamentals to real-time situations and learn problem solving. The internship experience also helped me to look at the possible career paths that I can work in, after graduation and helped me to get a view of how the professional world would look like. In addition to these, the internship also helps one to develop professionalism as well as work with seniors and experts in professional fields. Getting direct feedback from my professor helped me to learn more about my strengths, improvement areas, made me think how I can do tasks with better efficiency and gave me an overall perspective & confidence about successfully operating in a professional field.

This internship research work experience has helped me to look closely and narrow down the possible career choices that I have after graduation, as well as what fields do I feel more interested in. Working under direct guidance of professor has added more experiences to my learnings curve. Overall, I would strongly recommend everyone to look for and take up internships as those are very helpful to one in their professional careers in different ways.

### **Suggestions about changes to the program:**

I would like to sincerely extend my thanks to our college faculty members and professors for being encouraging as well as helpful throughout the process of getting approvals. The professor quickly resolved all the queries and questions.

The internship program is currently designed in the best possible way. There is one thing which I feel could be added to the reporting is presentations about internship experiences by all the students registered in the course. This would help everyone to learn from other people’s experiences and feedback about the presentations can help others in direct or indirect ways.

## **Weekly reporting & status updates:**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Week/Dates** | **Tasks performed** |
| 1 | 02/20/23 – 02/26/23 | GitHub resolve Merge-Conflict |
| 2 | 02/27/23 – 03/05/23 | Comparative analysis on Visual Studio vs Visual Studio Code vs Jupyter |
| 3 | 03/06/23 – 03/12/23 | Launching new EC2 from Visual Studio Code & Jupyter Notebook |
| 4 | 03/13/23 – 03/19/23 | Launching new EC2 & VPC from Visual Studio Code using Python |
| 5 | 03/20/23 – 03/26/23 | Launching AWS RDS from Visual Code |
| 6 | 03/27/23 – 04/02/23 | Accessing AWS EC2 from VSCode |
| 7 | 04/03/23 – 04/09/23 | Accessing AWS Cloud9 from VSCode |
| 8 | 04/10/23 – 04/16/23 | * Accessing AWS DynamoDB from EC2 using Python script from VSCode * Documenting all the scripts used |
| 9 | 04/17/23 – 04/23/23 | * DynamoDB operations from VSCode & accessing DynamoDB tables using ‘nginx’ * AWS DynamoDB ReadOnly access test |
| 10 | 04/24/23 – 04/30/23 | DynamoDB operations from EC2 & nginx |
| 11 | 05/01/23 – 05/07/23 | Summary & report preparation |