

INTERNSHIP: PROJECT REPORT

Name Of the Student	Aravinthkrishna A
Internship Project Title	TNSDC RIO-125: Project -1 Developing Web Application on AWS
Name of the Company	TCS ion
Name of the Industry Mentor	Krishna Sharma
Name of the Institute	Government College of Engineering, Salem

Start Date	End Date	Total Effort (hrs.)	Project Environment	Tools used
26/04/2023	30/05/2023	125	Web browser (Brave), Operating System (Windows, Ubuntu)	Amazon S3, AWS Console, AWS VPC, EC2 console, IDE.

TABLE OF CONTENT

- Acknowledgements
- Objective
- Introduction / Description of Internship
- Internship Activities
- Approach / Methodology
- Assumptions
- Exceptions / Exclusions
- Charts, Table, Diagrams
- Algorithms
- Challenges & Opportunities
- Risk Vs Reward
- Reflections on the Internship
- Recommendations
- Outcome / Conclusion
- Enhancement Scope
- Link to code and executable file
- Research questions and responses

Acknowledgements

I would like to express my sincere gratitude to my industry mentor and the entire team at TCS ION for providing me with the invaluable opportunity to learn and work on this project. Their guidance, support, and expertise were instrumental in the successful completion of this internship. I would like to extend a special thank you to my industry mentor for their continuous guidance and encouragement throughout the project. Their wealth of knowledge and expertise in the field of web development and AWS have been invaluable in shaping my understanding and skills. I am grateful to the entire team at TCS ION for their cooperation and assistance during this internship. Their willingness to share their expertise and provide necessary information regarding the project greatly contributed to its successful implementation. I am truly grateful for the opportunity to work with such a talented and supportive team, and I am confident that the knowledge and skills gained during this internship will have a lasting impact on my professional development.

Objective

Traditional on-premises web hosting architectures often face challenges related to complex solutions, reserved capacity forecasting, and low utilization rates. These challenges result in high operating costs and inefficient resource utilization. In contrast, AWS offers a reliable, scalable, secure, and high-performing infrastructure that can address these issues effectively. By leveraging AWS services, organizations can align IT costs with customer traffic patterns in near-real time, ensuring optimal resource utilization and cost efficiency.

The main objective of the project was to develop a web application using AWS services to create a more efficient and scalable hosting environment. Specific objectives included:

- Creating a static web hosting environment using AWS S3 for storing objects and configuring the bucket permissions and properties.
- Setting up a dynamic web hosting environment using AWS EC2 infrastructure-as-a-service (IaaS) to manually host and configure the web application.
- Configuring networking using AWS Virtual Private Cloud (VPC) to ensure secure communication and connectivity between components. Gain hands-on experience with AWS infrastructure: Gain practical experience in deploying and managing applications on AWS infrastructure, understanding the scalability and cost-effectiveness benefits provided by AWS.
- Overcome challenges and learn problem-solving skills: Encounter and overcome challenges related to infrastructure setup, package installation, and configuration. Develop problem-solving skills and troubleshooting abilities in the context of AWS services.
- Document the project activities and provide recommendations: Systematically document the project activities, challenges faced, solutions implemented, and lessons learned. Provide recommendations for further enhancements or improvements based on the experiences gained during the internship.

Introduction / Description of Internship

The internship provided an opportunity to gain hands-on experience in developing a web application on the Amazon Web Services (AWS) platform. The project focused on leveraging AWS services to overcome the challenges of traditional on-premises web hosting architectures. The internship aimed to explore and implement the benefits of AWS for web applications, including scalability, reliability, and cost-effectiveness. Throughout the internship, the main objective was to develop a web application infrastructure using a combination of AWS services. The project involved creating a static web hosting environment using AWS S3, configuring a dynamic web hosting environment using AWS EC2, and setting up networking using AWS VPC. The internship activities encompassed a range of tasks, including setting up AWS accounts, creating and configuring AWS resources, deploying and managing web applications, and conducting testing and verification. The project required a solid understanding of web development principles, AWS services, and networking concepts. Under the guidance of the industry mentor and the support of the TCS ION team, the internship provided an opportunity to apply theoretical knowledge to real-world scenarios, gain practical skills in working with AWS, and enhance problem-solving and critical thinking abilities. Overall, the internship offered a comprehensive learning experience in developing web applications on AWS, allowing for the acquisition of valuable skills and insights into cloud computing and modern web hosting architectures.

Internship Activities

During the internship, I actively participated in various activities related to the development of a web application on AWS. The following are the key activities I undertook:

Research and Planning:

- Conducted in-depth research on AWS services, specifically focusing on Amazon S3, EC2, and VPC.
- Explored best practices for web hosting and identified the benefits of using AWS for web applications.
- Gathered information on static web hosting, dynamic web hosting, and networking configurations.

Milestone 1: Create Bucket:

- Created an S3 bucket using the AWS Management Console.
- Configured the bucket settings, including naming conventions, region selection, and versioning.
- Uploaded objects to the bucket, including HTML, CSS, and image files.

Milestone 2: Configure Bucket:

- Configured the S3 bucket for static web hosting.
- Set bucket permissions to allow public read access for the hosted files.
- Configured bucket properties, such as the default index document and error document.

Milestone 3: Create Instance:

- Provisioned an EC2 instance using the AWS EC2 service.
- Selected an appropriate instance type and configured security groups to control inbound and outbound traffic.
- Set up Virtual Private Cloud (VPC) networking for the instance, including subnets, route tables, and network ACLs.

Milestone 4: Install Packages:

- Connected to the EC2 instance using SSH and installed required packages for the dynamic web application.
- Configured the necessary dependencies and libraries for the application to run smoothly.
- Ensured proper integration and compatibility of the installed packages.
- Milestone 5: Make Website Available:
- Configured the necessary networking settings to make the dynamic website publicly available.
- Set up security groups and network ACLs to allow inbound traffic to the website.
- Tested the accessibility of the website from different devices and locations.

Milestone 6: Project Conclusion:

- Summarized the overall project, highlighting the achieved milestones and outcomes.
- Reflected on the internship experience, discussing the lessons learned and the skills acquired.
- Provided recommendations for future enhancements or improvements to the project.
- Throughout these activities, I documented my progress, challenges faced, and solutions implemented. I collaborated with my industry mentor and the TCS ION team, seeking guidance whenever required. This hands-on experience enabled me to gain a deep understanding of AWS services and their application in web development.

Approach / Methodology

During the internship, a systematic approach was followed to ensure the successful development of the web application on AWS. The methodology can be summarized as follows:

Research and Planning:

- Conducted comprehensive research on AWS services, web hosting best practices, and the specific requirements of the project.
- Analysed the benefits of using AWS for web applications, including its scalability, reliability, and cost-effectiveness.
- Defined the scope and objectives of the project, considering the milestones to be achieved.

Requirement Gathering:

- Collaborated with the industry mentor and the TCS ION team to gather detailed requirements for the static and dynamic web application.
- Documented the functional and non-functional requirements, ensuring a clear understanding of the desired outcomes.

Infrastructure Setup:

- Provisioned necessary AWS resources, including an S3 bucket, EC2 instance, and VPC.
- Configured the networking components, such as subnets, security groups, and routing tables.
- Milestone Implementation:
- Implemented each milestone sequentially, focusing on achieving the desired functionality.
- Followed best practices and guidelines provided by AWS documentation and industry standards.
- Engaged in hands-on activities to create the S3 bucket, configure its settings, create the EC2 instance, and install the required packages.

Testing and Quality Assurance:

- Conducted rigorous testing to ensure the functionality and performance of the web application.
- Performed unit testing, integration testing, and user acceptance testing to identify and resolve any issues or bugs.
- Validated the accessibility and responsiveness of the static and dynamic web pages across different devices and browsers.

Documentation:

- Maintained detailed documentation throughout the internship, including project plans, configurations, and progress reports.
- Documented the step-by-step process followed for each milestone, capturing any challenges faced and the solutions implemented.
- Prepared clear and concise user guides or manuals for future reference.
- Collaboration and Mentorship:
 - Regularly communicated with the industry mentor and the TCS ION team to seek guidance and feedback.
 - Participated in collaborative discussions, sharing progress updates and seeking assistance when needed.
 - Leveraged the expertise of the team to address technical challenges and make informed decisions.

Throughout the internship, an iterative and collaborative approach was followed, involving regular communication with mentors and team members to seek guidance, address challenges, and receive feedback. Continuous learning and exploration of AWS documentation and resources were also important components of the methodology. The internship project aimed to ensure a structured and organized process of developing the web application on AWS, while providing valuable hands-on experience and knowledge in utilizing AWS services for web application hosting. By following this approach, the project was executed systematically, ensuring the successful development and deployment of the web application on AWS. The methodology allowed for effective planning, implementation, testing, and documentation, resulting in a comprehensive learning experience.

Assumptions

During the project, the following assumptions were made:

- Infrastructure Availability: It was assumed that the required AWS infrastructure, including S3, EC2, and VPC, would be available and properly configured for the project.
- Resource Allocation: Sufficient resources, such as compute power and storage capacity, were assumed to be allocated to the project. This includes the availability of EC2 instances and the required storage capacity in S3.
- External Dependencies: It was assumed that any external dependencies, such as internet connectivity and third-party APIs, would be accessible and functioning properly.

These assumptions influenced the project by providing a foundation upon which the implementation and configuration activities were based. They allowed for a smoother execution of the project tasks, as the focus was primarily on utilizing the available infrastructure and resources provided by AWS.

Exceptions / Exclusions

During the internship, the following exceptions or exclusions were encountered:

- **Database Configuration:** The project focused on hosting a web application using AWS services, but the configuration and integration of a separate database service were considered out of scope. The project primarily dealt with the hosting and networking aspects, while the database functionality was excluded.
- **Advanced Security Configurations:** Due to the limited scope of the project, advanced security configurations, such as custom IAM policies, fine-grained access control, or SSL certificate management, were excluded. The emphasis was placed on basic security practices and configurations provided by AWS.
- **Load Balancing and Auto Scaling:** The project did not include the implementation of load balancing or auto scaling mechanisms. These features are typically used to handle high traffic and ensure optimal performance, but were considered beyond the scope of the project.

These exceptions and exclusions were made to focus on the core objectives of the internship project and ensure its feasibility within the given timeframe and resources. They helped maintain a clear and achievable scope while still achieving the desired outcomes of hosting a static and dynamic web application on AWS.

Technical Stack and Technologies Used

Frontend Technologies:

- **HTML5:** Markup language for structuring the web pages.
- **CSS3:** Stylesheet language for designing the visual appearance of the web application.

Backend Technologies:

- **PHP:** Server-side scripting language used for dynamic web application development.

AWS Services:

- **Amazon S3:** Object storage service used to store and serve static website files.
- **Amazon EC2:** Virtual server instances used to host and run the dynamic web application.
- **Amazon VPC:** Virtual Private Cloud for configuring networking components and creating a secure environment.

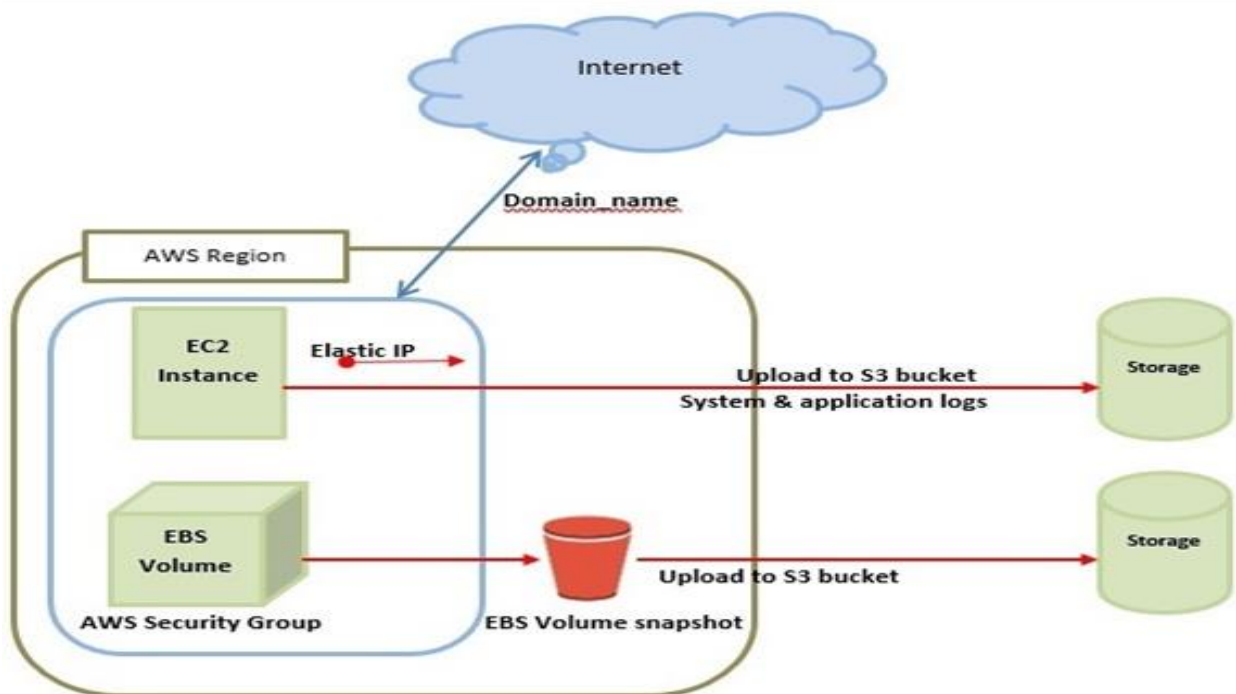
Development Tools:

- **Visual Studio Code:** Integrated Development Environment (IDE) for code editing and debugging.
- **Git:** Version control system for managing source code and collaboration.

By utilizing this technical stack and these technologies, the internship project aimed to leverage industry-standard tools and frameworks for efficient web application development on AWS. This allowed for seamless integration and effective utilization of AWS services, providing a robust and scalable solution.

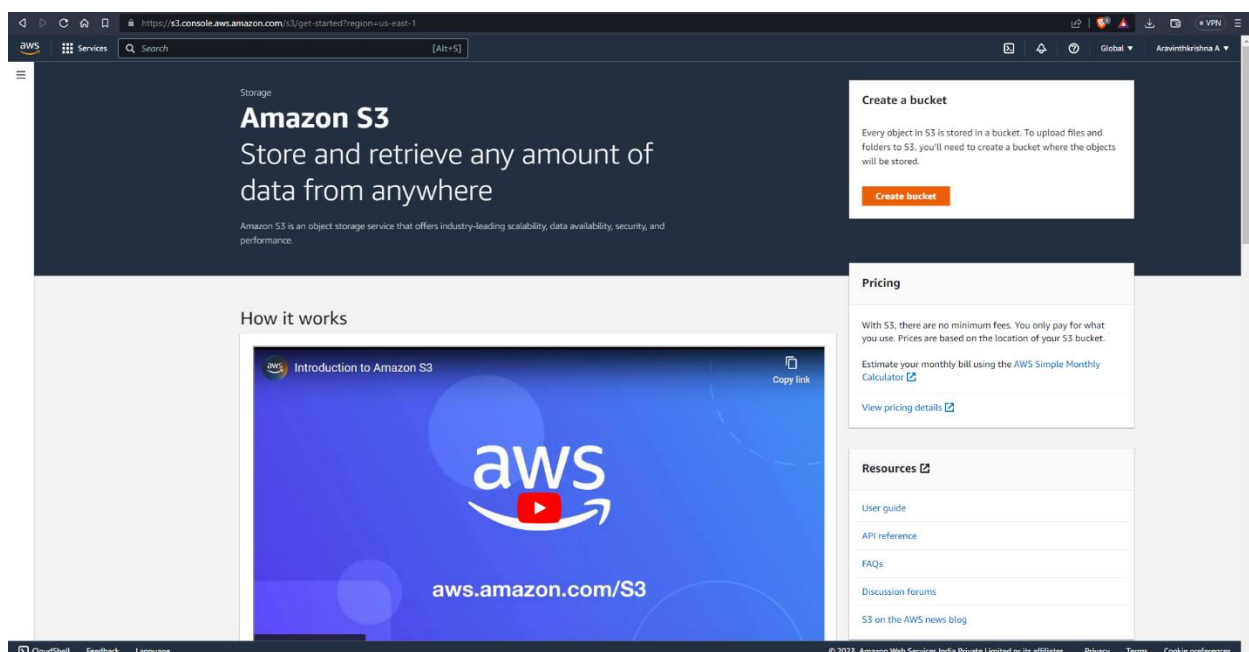
This addition provides a dedicated section to highlight the technologies used in the project, offering insights into the technical aspects of the web application development on AWS.

Charts and Pictures



A diagram illustrating the overall architecture of the web application, showcasing the interaction between different AWS services like S3, EC2, and VPC.

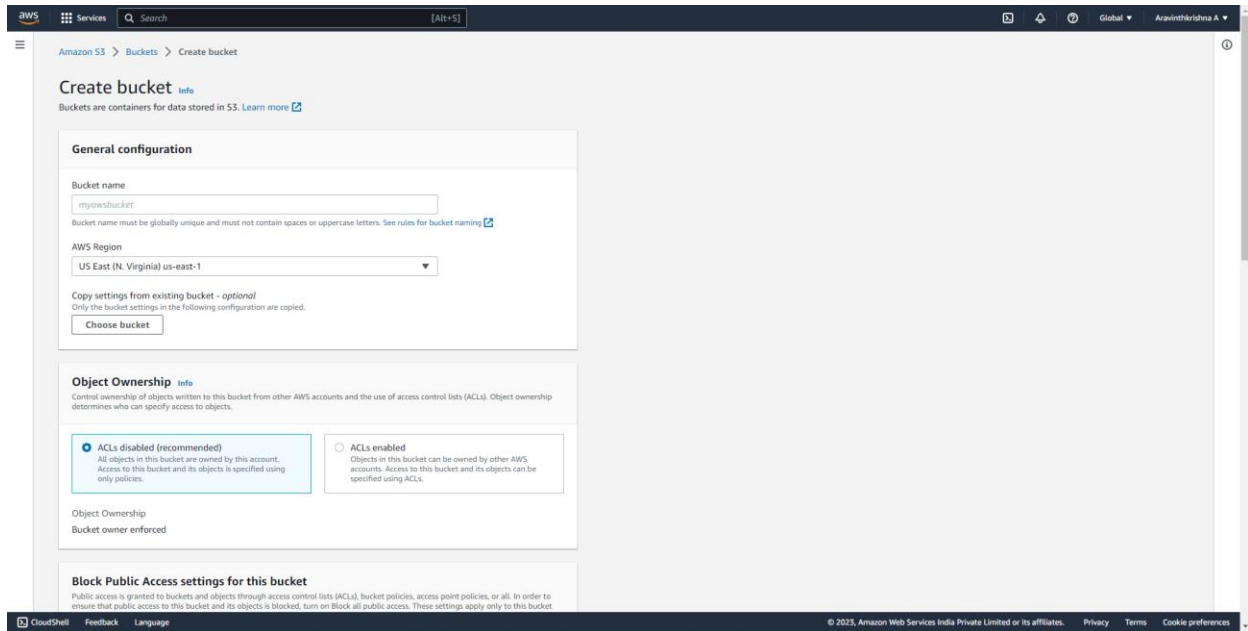
- Starting to create S3 Bucket storage



Screenshot showcasing the AWS user interface of the Amazon S3 (Simple Storage Service) console

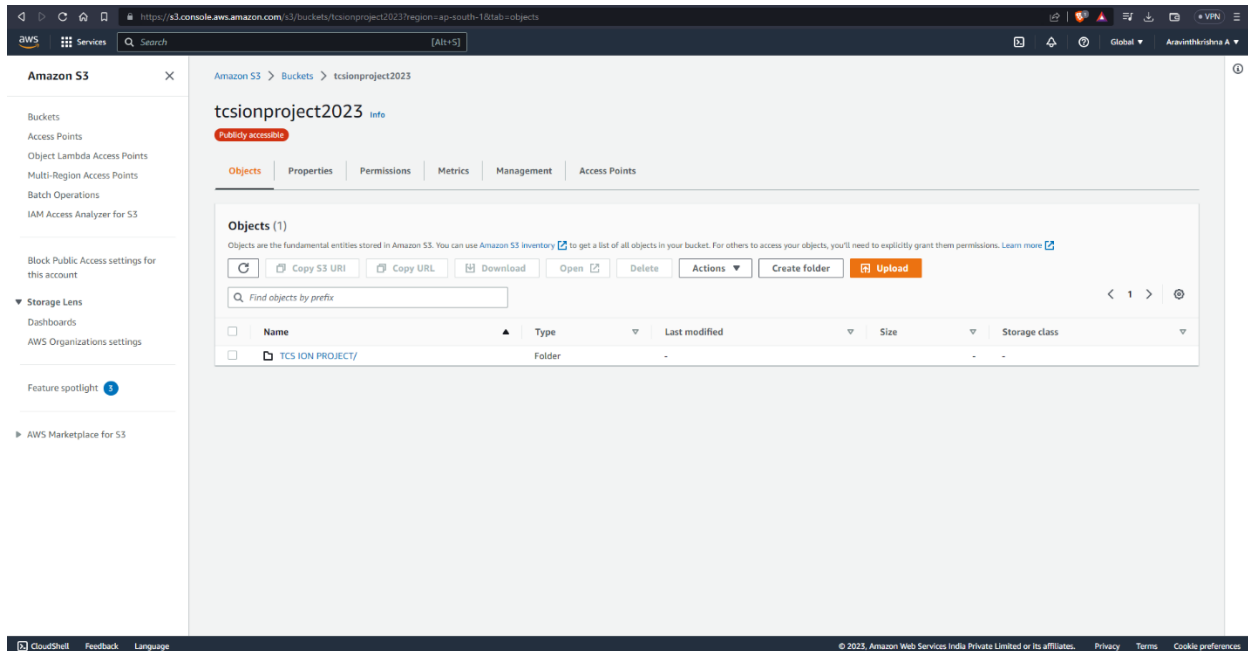
INTERNSHIP: PROJECT REPORT

- The process of creating an S3 bucket using the Amazon S3 console.



Screenshot showcasing the process of creating an S3 bucket using the Amazon S3 console

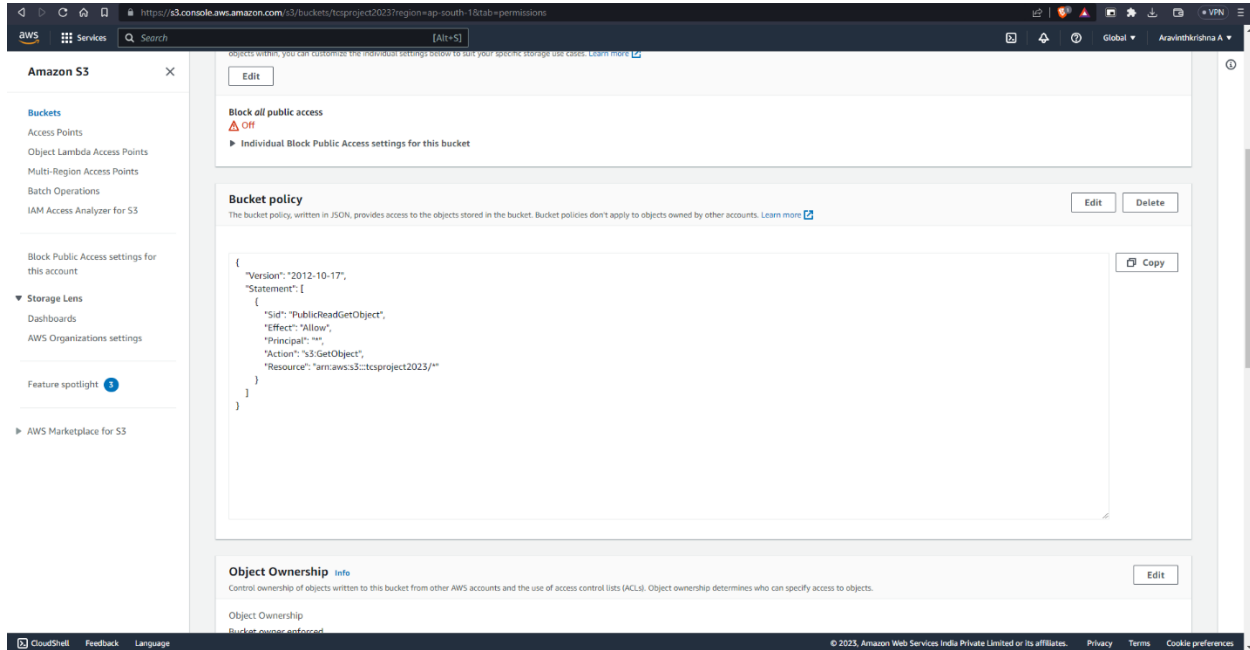
- Completed creation of an Amazon S3 bucket.



Screenshot showcasing the successful creation of an Amazon S3 bucket

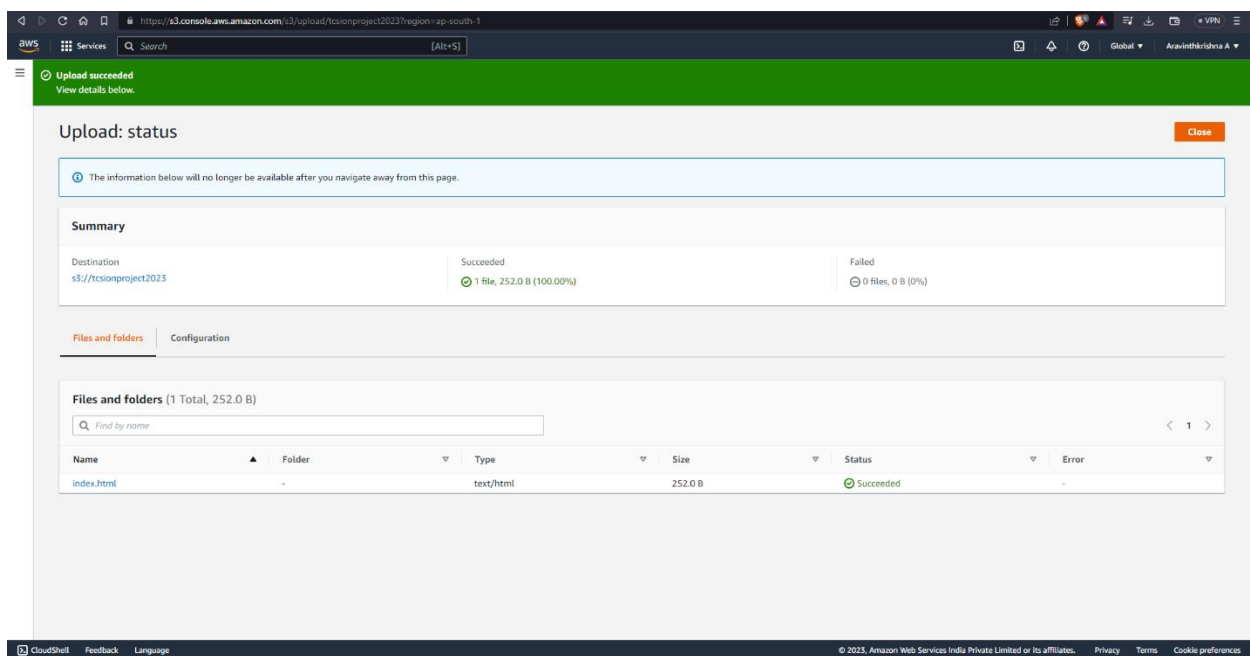
INTERNSHIP: PROJECT REPORT

- Configuration of the Amazon S3 bucket policy to allow public access.



Screenshot showcasing the displaying the configuration of the Amazon S3 bucket policy to allow public access

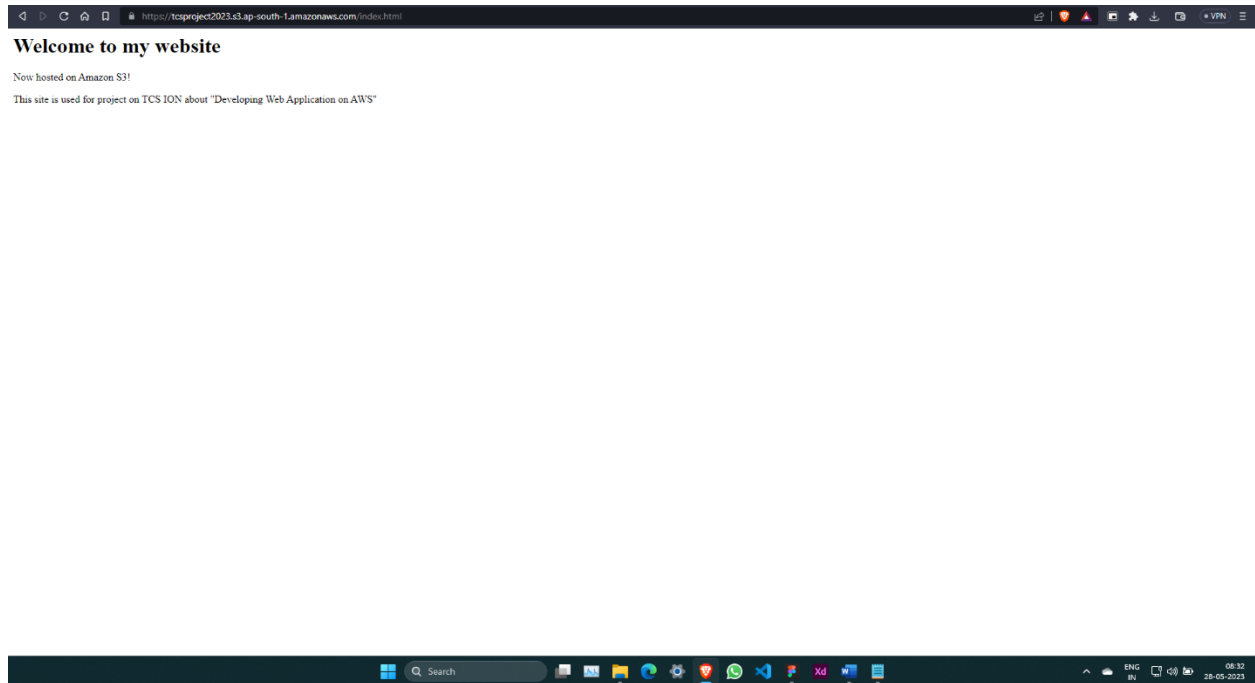
- The successful storage of an object in an Amazon S3 bucket.



Screenshot showcasing the depicting the successful storage of an object in an Amazon S3 bucket

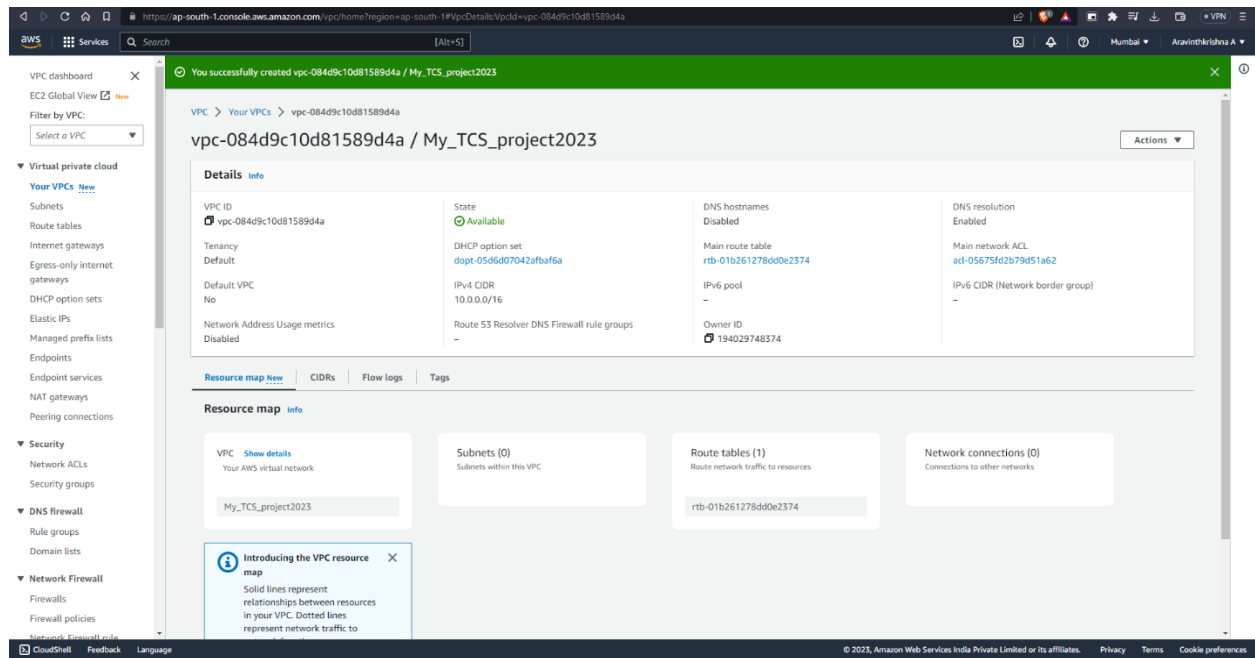
INTERNSHIP: PROJECT REPORT

- The static website hosted on Amazon S3, displaying its user interface and content.



Screenshot showcasing the static website hosted on Amazon S3, displaying its user interface and content

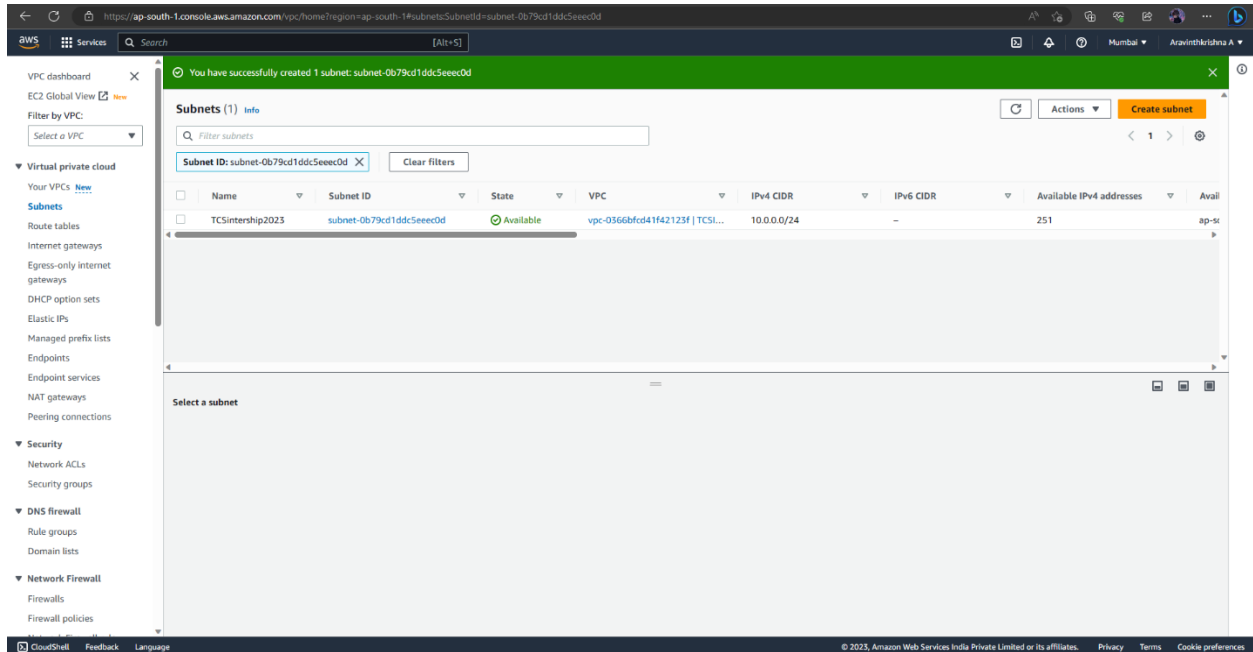
- Creating a Virtual Private Cloud (VPC) in the AWS Management Console.



Screenshot depicting the creation of a Virtual Private Cloud (VPC) in the AWS Management Console

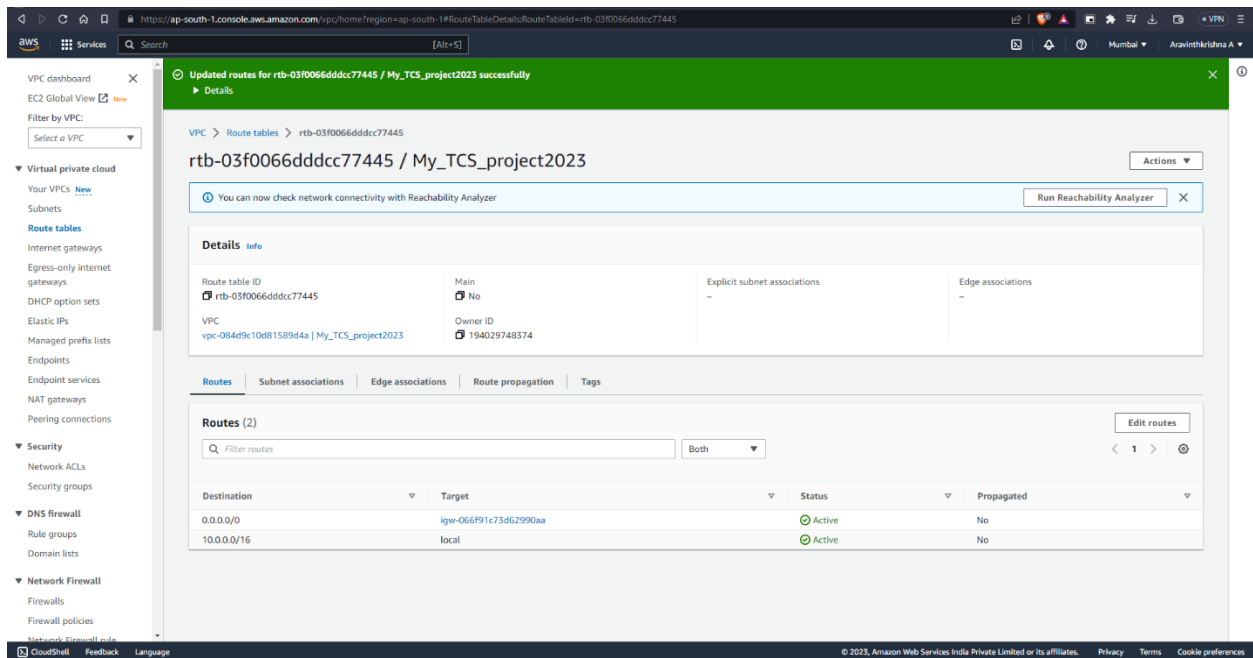
INTERNSHIP: PROJECT REPORT

- Creating a subnet in Amazon VPC.



Screenshot illustrating the creation of a subnet in Amazon VPC

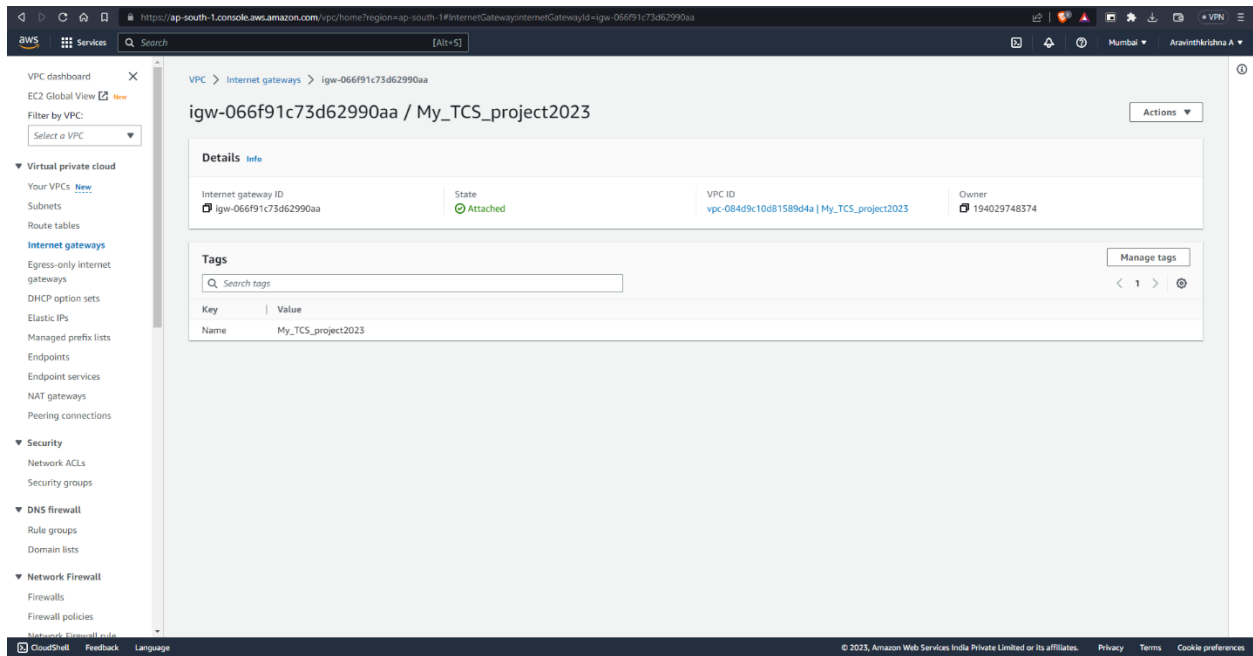
- Creating a Route in the Amazon VPC console.



Screenshot demonstrating the creation of a Route Table in the Amazon VPC console

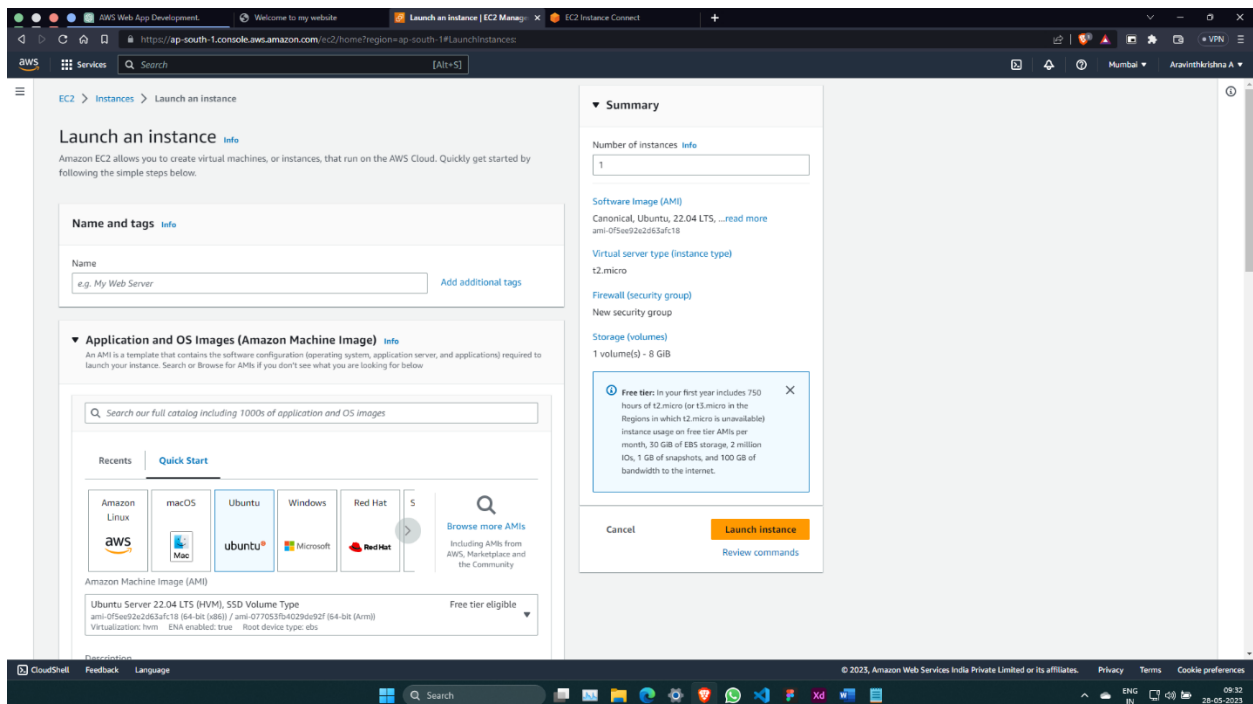
INTERNSHIP: PROJECT REPORT

- Creating an Internet Gateway in the AWS Management Console.



Screenshot illustrating the creation of an Internet Gateway in the AWS Management Console

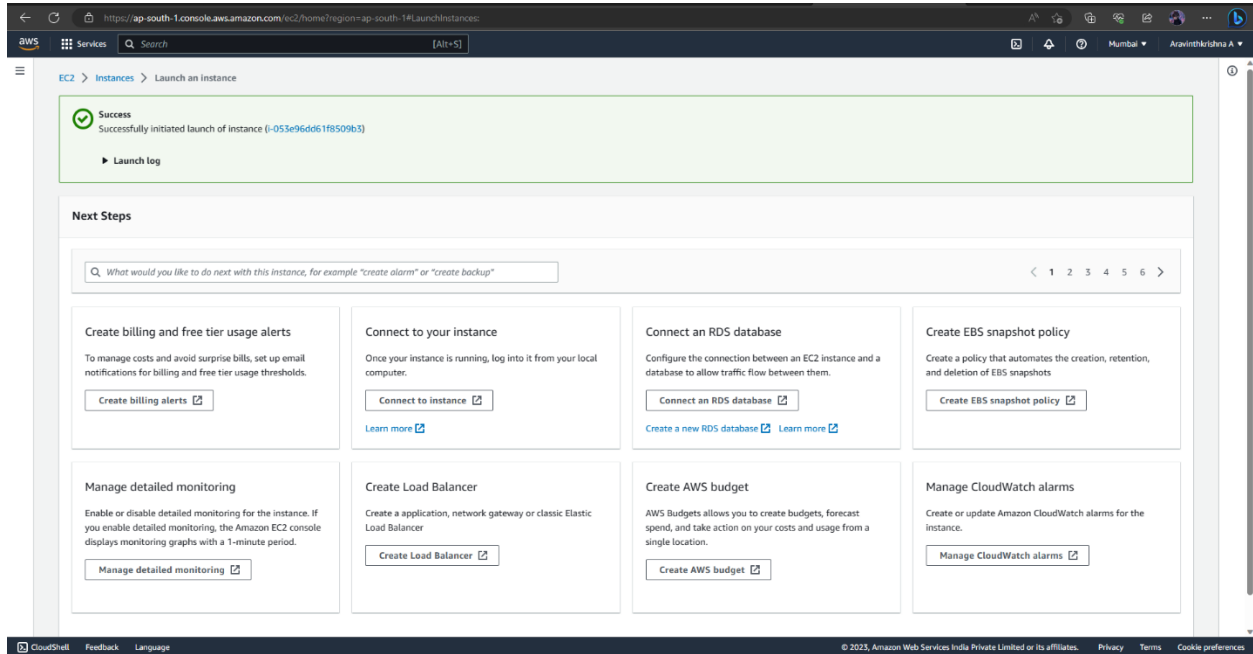
- Now Creating a EC2 instance on AWS.



Screenshot capturing the process of creating an EC2 instance on AWS

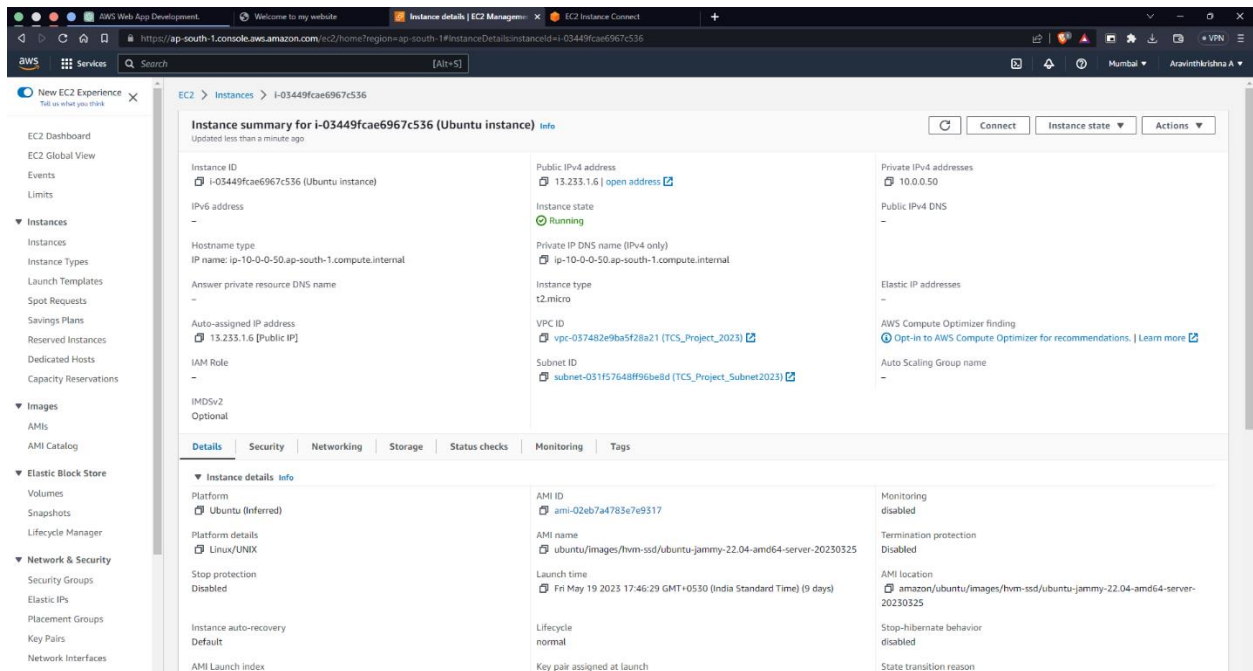
INTERNSHIP: PROJECT REPORT

- Successfully launched an instance in Amazon EC2 Dashboard.



Screenshot displaying a successfully launched instance in the Amazon EC2 dashboard

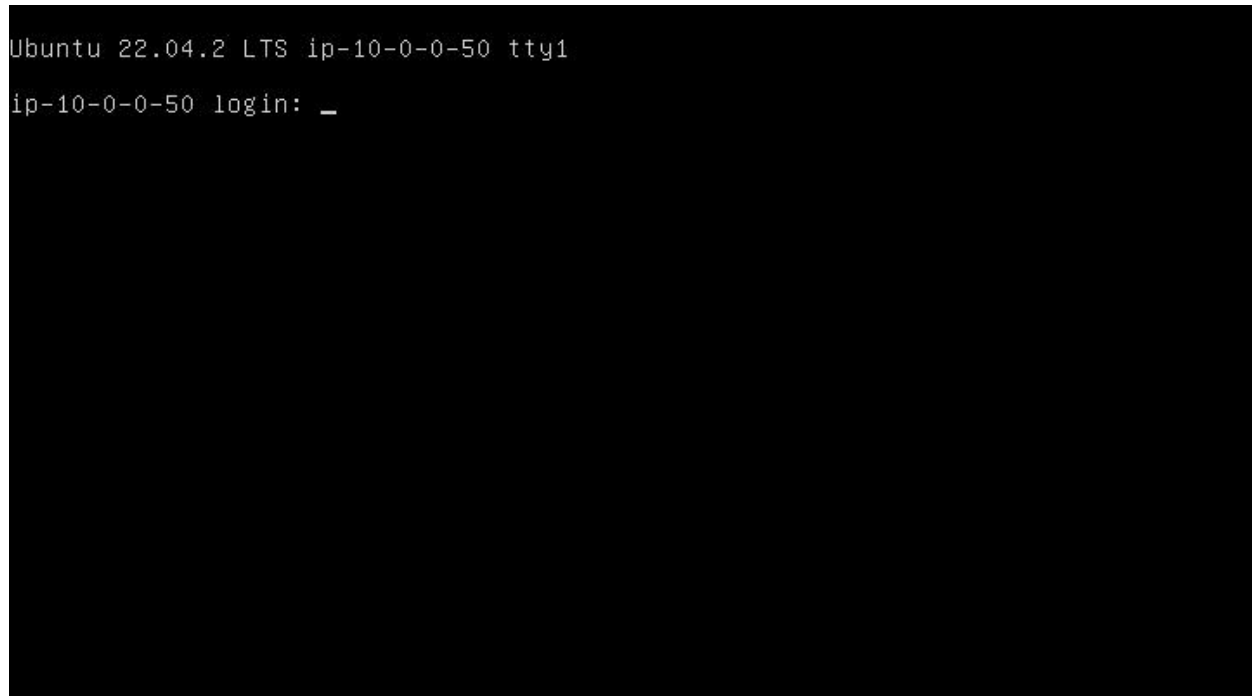
- Detailed information and configurations of an Amazon EC2 instance.



Screenshot showcasing the displaying detailed information and configurations of an Amazon EC2 instance

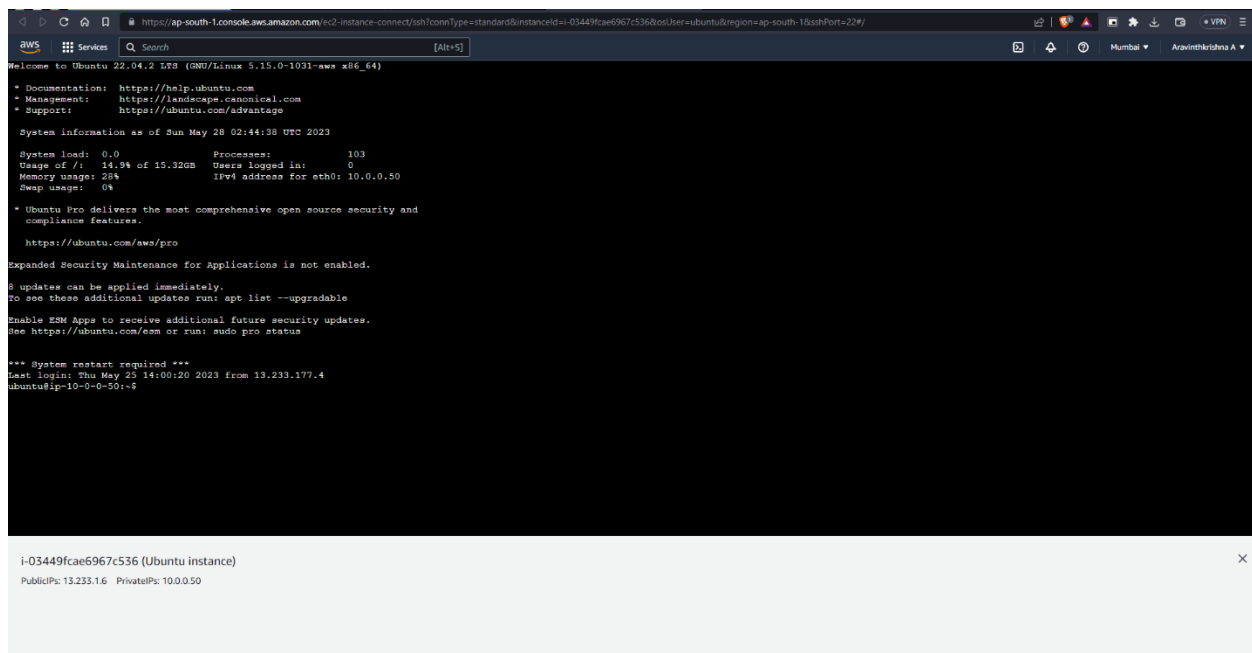
INTERNSHIP: PROJECT REPORT

- The Screenshot of instance provided by AWS EC2 console.



Screenshot showcasing an instance provided by the AWS EC2 console

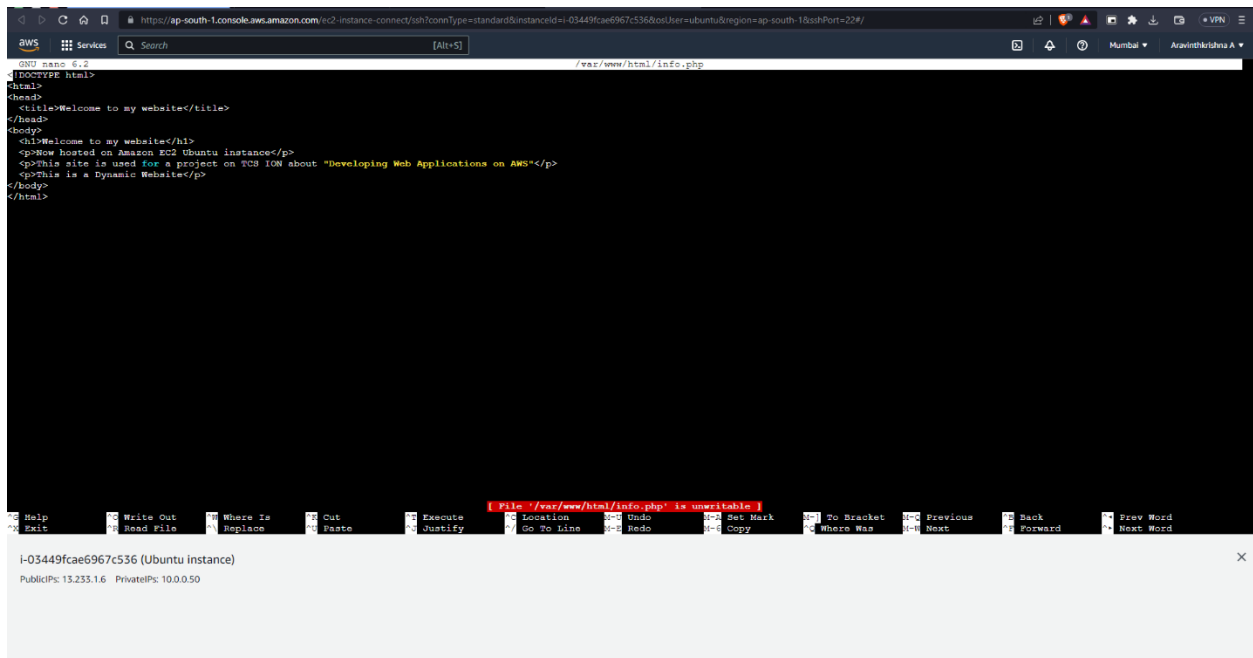
- Connect to EC2 instance using EC2 console.



Screenshot showcasing successful connection to an EC2 instance using the EC2 console

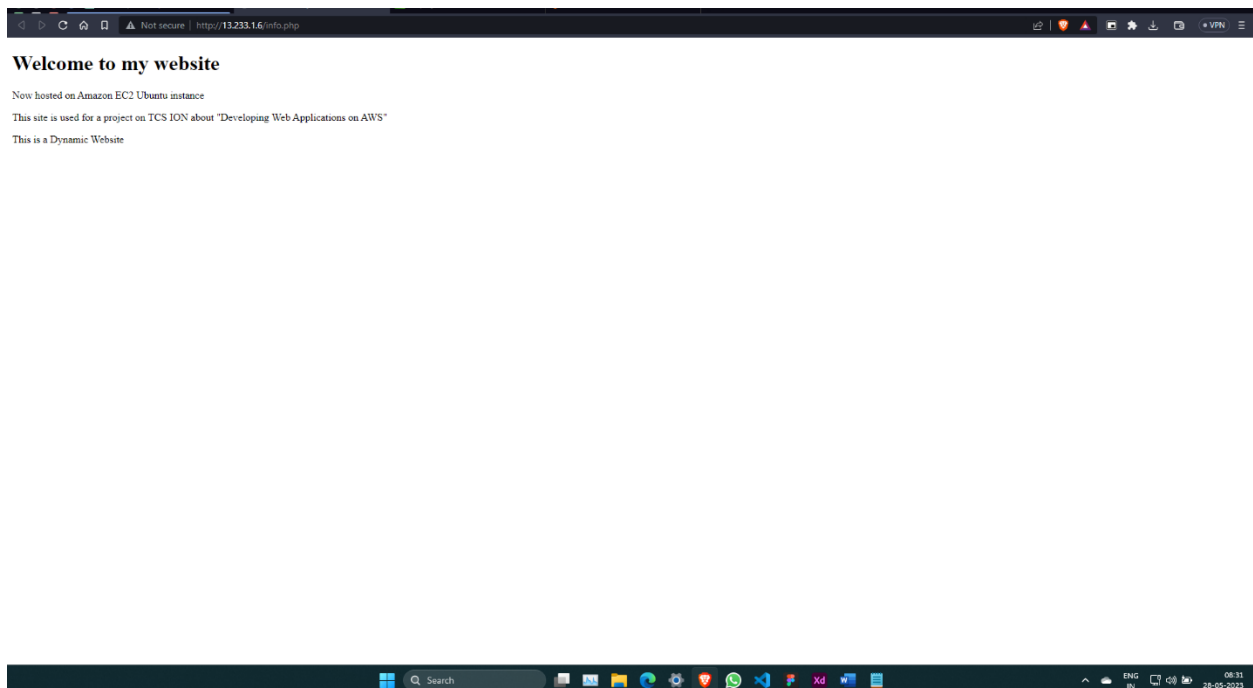
INTERNSHIP: PROJECT REPORT

- Editing the HTML file within in EC2 instance.



Screenshot showcasing the process of editing an HTML file within an EC2 instance

- The Dynamic website hosted on EC2 instance, displaying its user interface and content.



Screenshot showcasing the user interface and content of the dynamic website hosted on an EC2 instance

Algorithms

1. S3 Bucket Policy Algorithm:

This algorithm defines an AWS S3 bucket policy using the JSON format.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadGetObject",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::tcsproject2023/*"
    }
  ]
}
```

- **"Version": "2012-10-17"**: Specifies the version of the policy language being used.
- **"Statement"**: Contains an array of policy statements.
- **"Sid": "PublicReadGetObject"**: A unique identifier for the policy statement.
- **"Effect": "Allow"**: Specifies that the defined actions are allowed.
- **"Principal": "*"** : Specifies that the policy applies to all principals (any entity accessing the bucket).
- **"Action": "s3:GetObject"**: Specifies the allowed action, which is retrieving (reading) an object from the bucket.
- **"Resource": "arn:aws:s3:::tcsproject2023/*"**: Specifies the Amazon Resource Name (ARN) of the bucket and the objects within it that the policy applies to. In this case, it allows reading any object within the "tcsproject2023" bucket.

This policy allows public read access to objects in the specified S3 bucket. It's important to note that granting public access should be done with caution and proper consideration of security implications.

2. Commands used in Ubuntu instance:

- **`sudo apt update`**: This command updates the package lists for upgrades and new package installations.
- **`sudo apt upgrade -y`**: This command upgrades all the installed packages to their latest versions. The **`-y`** flag automatically confirms any prompts during the upgrade process.
- **`sudo apt install apache2 -y`**: This command installs the Apache web server, which is used to serve web content.
- **`sudo apt install php libapache2-mod-php php-mysql -y`**: This command installs PHP and its dependencies required for running PHP applications on the Apache web server. The **`libapache2-mod-php`** package enables Apache to handle PHP files, while **`php-mysql`** provides the MySQL extension for PHP.
- **`sudo a2enmod php`**: This command enables the PHP module in Apache, allowing the server to process PHP files.

- `sudo systemctl restart apache2`: This command restarts the Apache web server to apply the changes made in the configuration.
- `nano /var/www/html/info.php`: This command opens the file `info.php` in the nano text editor, located in the `/var/www/html` directory. This file is typically used to test PHP installations and view PHP configuration details.
- `cd /var/www/html`: This command changes the current directory to `/var/www/html`, which is the default web root directory in Apache where web content is stored.
- `sudo chmod -R 755 /var/www/html/`: This command sets the appropriate permissions for the `/var/www/html/` directory and its contents. It allows read and execute permissions for the owner, group, and others, while restricting write permissions.

Please note that running these commands should be done with caution, as they involve system-level operations and can affect the configuration of your server.

Challenges & Opportunities

During the course of the internship project, several challenges and opportunities were encountered. Some of the key challenges and opportunities are outlined below:

Challenges:

- **Learning Curve:** One of the initial challenges was familiarizing myself with the AWS services and understanding their functionalities and configurations.
- **Infrastructure Setup:** Setting up the AWS infrastructure, including S3 bucket configuration and EC2 instance setup, required careful attention to detail and proper networking configurations.
- **Package Installation and Configuration:** Installing and configuring the required packages on the EC2 instance posed challenges, as compatibility issues and dependencies needed to be addressed.

Opportunities:

- **Skill Development:** The project provided an excellent opportunity to enhance my skills in AWS, specifically in working with S3, EC2, and VPC.
- **Hands-on Experience:** Working on a real-world web application project allowed me to gain practical experience in deploying and managing applications on AWS.
- **Problem-Solving:** Overcoming challenges related to infrastructure setup, package installation, and configuration helped me develop problem-solving skills and strengthened my troubleshooting abilities.

Risk Vs Reward

The risk versus reward analysis for the project highlights the potential risks associated with the project implementation, as well as the rewards gained from its successful completion.

Risks:

- **Technical Complexity:** The project involved working with various AWS services and configuring networking, which posed the risk of encountering technical difficulties and compatibility issues.
- **Time Constraints:** Meeting the project milestones within the specified timeline presented a risk, as unexpected delays or complications could arise during the implementation.

Rewards:

- **Knowledge Acquisition:** Successfully completing the project provided the opportunity to acquire in-depth knowledge and practical experience with AWS services, specifically S3, EC2, and VPC.
- **Skill Enhancement:** Working on the project allowed for the development and enhancement of skills in web application deployment, infrastructure setup, and networking configuration.
- **Cost Efficiency:** By leveraging AWS services, the project aimed to achieve cost efficiency by scaling resources based on traffic patterns and eliminating the need for upfront hardware investments.

Reflections on the Internship

Throughout the internship, I gained valuable insights and experiences that have contributed to my personal and professional growth. Some reflections on the internship include:

- **Hands-on Learning:** The internship provided an excellent opportunity to apply theoretical knowledge gained during my studies to real-world scenarios. Working on the project allowed me to gain practical experience and enhance my problem-solving skills.
- **Collaboration and Communication:** Engaging with mentors and team members helped improve my collaboration and communication skills. Regular discussions and feedback sessions were instrumental in clarifying project requirements and addressing challenges.
- **Industry Relevance:** The internship exposed me to industry-standard practices and technologies. Understanding how AWS services are utilized in web application development has given me a deeper understanding of the industry landscape.

Recommendations

Based on the experiences and learnings from the internship project, the following recommendations are provided:

- **Continuous Learning:** It is recommended to continue exploring and expanding knowledge in AWS services and related technologies to stay updated with the latest advancements.
- **Automation and Infrastructure as Code:** Implementing infrastructure as code practices, such as using AWS CloudFormation or AWS CDK, can streamline the deployment and configuration of AWS resources.
- **Monitoring and Scaling:** Integrating monitoring tools and implementing auto-scaling mechanisms can enhance the performance and reliability of the web application.

Outcome / Conclusion

In conclusion, the internship project focused on developing a web application on AWS, utilizing various services such as AWS S3, EC2, and VPC. The project successfully achieved its objectives, resulting in the creation of a static website hosted on S3 and a dynamic website hosted on an EC2 instance. Throughout the internship, several challenges were encountered, including the learning curve associated with AWS services, infrastructure setup complexities, and package installation and configuration issues. However, these challenges provided valuable learning opportunities and helped in the development of problem-solving skills. The risk versus reward analysis highlighted the potential risks involved in the project, such as technical complexity and time constraints, but also emphasized the rewards gained from successfully completing the project. These rewards included knowledge acquisition, skill enhancement, and cost efficiency through the utilization of AWS services. Reflections on the internship revealed the hands-on learning experience, improved collaboration and communication skills, and the industry relevance of the project. The internship provided practical exposure to real-world scenarios and demonstrated the application of theoretical knowledge in the industry landscape. Based on the experiences and learnings from the project, several recommendations were provided. These recommendations included continuous learning, adoption of infrastructure as code practices, and implementation of monitoring and scaling mechanisms to further enhance the web application. In conclusion, the internship project successfully achieved its objectives, providing a comprehensive understanding of developing web applications on AWS. The outcomes included the development and deployment of a functional web application, highlighting the benefits of leveraging AWS services for scalability, cost-efficiency, and reliable infrastructure. The project also identified potential areas for future enhancement, such as security measures, database integration, and continuous integration and deployment practices.

Enhancement Scope

While the internship project successfully achieved its objectives, there are potential areas for further enhancement and expansion:

- **Security:** Implementing additional security measures, such as SSL/TLS certificates and access control policies, to ensure the confidentiality and integrity of the web application.
- **Database Integration:** Integrating a database service, such as Amazon RDS or Amazon DynamoDB, to support dynamic data storage and retrieval for the web application.
- **Continuous Integration and Deployment:** Implementing CI/CD pipelines, utilizing tools like AWS Code Pipeline and AWS Code Deploy, to automate the build, testing, and deployment processes for the web application.

These enhancements can further improve the functionality, performance, and security of the web application developed during the internship.

This concludes the project report, summarizing the challenges faced, opportunities encountered, reflections gained, recommendations provided, and the overall outcomes and potential areas for future enhancement.

Link to code and executable file

Static Web Application (Link: <https://tcsproject2023.s3.ap-south-1.amazonaws.com/index.html>)

The provided link directs to a static web application hosted on Amazon S3. It leads to an HTML file named "index.html" stored in an S3 bucket. The "index.html" file likely contains the HTML, CSS, and JavaScript code that make up the static web application's frontend. By accessing this link, you can view the static web application and interact with its user interface.

Dynamic Web Application (Link: <http://13.233.1.6/info.php>)

The provided link directs to a dynamic web application. It leads to a PHP file named "info.php" hosted on an AWS EC2 instance. The "info.php" file likely contains PHP code that dynamically generates and displays information based on server-side processing. By accessing this link, you can interact with the dynamic web application and view the output generated by the PHP code.

Research questions and responses

Research Question 1: Is it necessary to create a domain name for a static website?

Response: *The response to this question is that there is no need to create a domain name for a static website.*

Research Question 2: Can you provide a more detailed description of creating a dynamic web application? Should the dynamic website be kept simple or should it be more detailed?

Response: *The response to this question suggests that the development of the dynamic website can be done as simple as possible or can be made more detailed, depending on the comfort level of the developer. The main agenda is to develop the dynamic web application on AWS.*