A Summer Industry Internship –I Report on

**AUTOMATING INFRASTRUCTURE DEPLOYMENT WITH AWS CLOUDFORMATION**

During

II Year II Semester Summer

Submitted to

**The Department of Computer Science and Engineering**

In partial fulfillment of the academic requirements of Jawaharlal Nehru Technological University

For

The award of the degree of

Bachelor of Technology in

Computer Science and Engineering By

S.AKSHAYA 19311A05Q8

Y.NIKITHAREDDY 19311A05R9 M.ARUN 19311A05U5



Sreenidhi Institute of Science and Technology

### (An Autonomous Institution)

Yamnampet, Ghatkesar, R.R. District, Hyderabad – 501301

Affiliated to

Jawaharlal Nehru Technology University Hyderabad - 500085

**Department of Computer Science and Engineering Sreenidhi Institute of Science and Technology**



**CERTIFICATE**

This is to certify that this Summer Industry Internship –I report on “Automating Infrastructure Deployment with AWS CloudFormation”, submitted by

S.Akshaya(19311A05Q8), Y.NikithaReddy(19311A05R9), M.Arun(19311A05U5) in the year 2022 in partial fulfillment of the academic requirements of Jawaharlal Nehru Technological University for the award of the degree of Bachelor of Technology in Computer Science and Engineering, is a bonafide work- summer industry internship that has been carried out **during II B Tech CSE II semester, will be evaluated in III B Tech CSE I Semester** , under our guidance. This report has not been submitted to any other institute or university for the award of any degree.

### Project Coordinator Head of the Department

Mr. Pasam Ramu Dr . Aruna Varanasi

Assistant Professor Professor

Department of CSE Department of CSE

External Examiner Date:

Diagram

Description automatically generated





**DECLARATION**

### We, S.AKSHAYA (19311A05Q8), Y.NIKITHA REDDY(19311A05R9) and M.ARUN(19311A05U5), students of SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY,YAMNAMPET, GHATKESAR, studying IIIrd year Ist semester, COMPUTER SCIENCE AND ENGINEERING solemnly declare that the Summer Industry Internship-I report , titled “AUTOMATING INFRASTRUCTURE DEPLOYMENT WITH AWS CLOUDFORMATION” is submitted to SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY for partial fulfillment for the award of degree of Bachelor of technology in COMPUTER SCIENCE AND ENGINEERING.It is declared to the best of our knowledge that the work reported does not form part of any dissertation submitted to any other University or Institute for award of any degree

**S.AKSHAYA 19311A05Q8 Y.NIKITHA REDDY 19311A05R9 M.ARUN 19311A05U5**

**ACKNOWLEDGEMENT**

I would like to express my gratitude to all the people behind the screen who helped me to transform an idea into a real application.

I would like to express my heart-felt gratitude to my parents without whom I would not have been privileged to achieve and fulfill my dreams. I am grateful to our principal, **Dr. T. Ch. Siva Reddy,** who most ably run the institution and has had the major hand in enabling me to do my project.

I profoundly thank Dr. **ARUNA VARANASI**, Head of the Department of Computer Science & Engineering who has been an excellent guide and also a great source of inspiration to my work.

I would like to thank my Project coordinator **Mr.Pasam Ramu** for her technical guidance, constant encouragement and support in carrying out my project at college.

The satisfaction and euphoria that accompany the successful completion of the task would be great but incomplete without the mention of the people who made it possible with their constant guidance and encouragement crowns all the efforts with success. In this context, I would like thank all the other staff members, both teaching and non-teaching, who have extended their timely help and eased my task.

## S.AKSHAYA 19311A05Q8 Y.NIKITHA REDDY 19311A05R9 M.ARUN 19311A05U5

**AUTOMATING INFORMATION DEPLOYMENT WITH AWS CLOUDFORMATION**

**Abstract**

* Customers need infrastructure in a consistent manner. But Deploying infrastructure in a consistent, reliable manner is difficult. It requires people to follow documented procedures without taking any undocumented shortcuts. It can also be difficult to deploy infrastructure out-of-hours when fewer staff are available. AWS CloudFormation changes this situation by defining infrastructure in a template that can be automatically deployed—even on an automated schedule.

|  |  |  |
| --- | --- | --- |
| **INDEX** |  | |
| **Abstract** |  | i |
| **1. INTRODUCTION** | 1 |  |
| 1.1 About the Internship | 1 |  |
| 1.2 Plan of Training Program | 1 |  |
| 1.3 Scope | 2 |  |
| 1.4 Existing System | 2 |  |
| 1.5 Proposed System | 2 |  |
| **2. SYSTEM ANALYSIS** | 3 |  |
| 2.1 Functional Requirement Specifications | 3 |  |
| 2.2 Performance Requirements | 4 |  |
| 2.3 Software Requirements | 5 |  |
| 2.4 Hardware Requirements | 5 |  |
| **3.SYSTEM DESIGN** | 6 |  |
| 3.1 Architecture Design | 6 |  |
| 3.2 Modules | 7 |  |
| 3.3 UML Diagrams | 10 |  |
| 3.3.1 Class Diagrams | 10 |  |

1. [SYSTEM IMPLEMENTATION 11](#_TOC_250007)
2. [OUTPUT SCREENS 15](#_TOC_250006)
3. INTERNSHIP FEEDBACK (Experience) 20
   1. [Challenges faced 20](#_TOC_250005)
4. CONCLUSIONS AND FUTURE SCOPE 21

[BIBLIOGRAPHY 24](#_TOC_250004)

[Appendix-A:Java Technology 25](#_TOC_250003)

[Appendix-B:Unified Modeling Language 33](#_TOC_250002)

[Appendix C: Abstract 38](#_TOC_250001)

Appendix D: Correlation between the Summer Industry Internship-I and the Program

Outcomes (POs), Program Specific Outcomes (PSOs) 40

[Appendix E: Domain of Internship and Nature of internship 41](#_TOC_250000)

### INTRODUCTION

* Deploying infrastructure in a consistent, reliable manner is difficult. People requires to follow documented procedures without taking any undocumented shortcuts. It can also be difficult to deploy infrastructure out-of-hours when fewer staff are available. AWS CloudFormation changes this situation by defining infrastructure in a template that can be automatically deployed—even on an automated schedule.
* In this automating infrastructure deployment we should perform some tasks.
* **Task -1**: **Deploying a networking layer**
* In this task, you will deploy an AWS CloudFormation template that creates a *networking layer* by using Amazon VPC.

**Task 2: Deploying an application layer**

* Now that you deployed the *network layer*, you will deploy an *application layer* that contains an Amazon Elastic Compute Cloud (Amazon EC2) instance and a security group.

**Task 3: Updating a Stack**

* AWS CloudFormation can also *update* a stack that has been deployed. When you update a stack, AWS CloudFormation will only modify or replace the resources that are being changed. Any resources that are not being changed will be left as-is.
* In this task, you will update the *lab-application* stack to modify a setting in the security group.
* **Task 4: Exploring templates with AWS CloudFormation Designer**
* *AWS CloudFormation Designer* is a graphic tool for creating, viewing, and modifying AWS CloudFormation templates. With Designer, you can diagram your template resources by using a drag-and-drop interface, and then edit their details through the integrated JSON and YAML editor.
* **Task 5: Deleting the stack**
* When resources are no longer required, AWS CloudFormation can delete the resources built for the stack.

## SYSTEM ANALYSIS

This System Analysis is closely related to requirements analysis. It is also "an explicit formal inquiry carried out to help someone (referred to as the decision maker) identify a better course of action and make a better decision than he might otherwise have made." This step involves breaking down the system in different pieces to analyze the situation, analyzing project goals, breaking down what needs to be created and attempting to engage users so that definite requirements can be defined

## Functional Requirement Specification

The System after careful analysis has been identified to be present with the following modules.

* In this automating infrastructure deployment we should perform some tasks.
* **Task -1**: **Deploying a networking layer**
* It is a best practice to deploy infrastructure in *layers*. Common layers are:
* Network (Amazon VPC)
* Database
* Application
* In this task, you will deploy an AWS CloudFormation template that creates a *networking layer* by using Amazon VPC.

Task 2: Deploying an application layer

* Now that you deployed the *network layer*, you will deploy an *application layer* that contains an Amazon Elastic Compute Cloud (Amazon EC2) instance and a security group.
* The AWS CloudFormation template will *import* the VPC and subnet IDs from the *Outputs* of the existing CloudFormation stack. It will then use this information to create the security group in the VPC and the EC2 instance in the subnet.
* While the stack is being created, examine the details in the **Events** tab and the **Resources** tab. You can monitor the progress of the resource-creation process and the resource status.

Task 3: Updating a Stack

* AWS CloudFormation can also *update* a stack that has been deployed. When you update a stack, AWS CloudFormation will only modify or replace the resources that are being changed. Any resources that are not being changed will be left as-is.
* In this task, you will update the *lab-application* stack to modify a setting in the security group.
* First, you will examine the current settings for the security group.
* The **Inbound rules** tab should display an additional rule that allows *SSH* traffic over *TCP port 22*.
* This subtask demonstrates how changes can be deployed in a repeatable, documented process. The AWS CloudFormation templates can be stored in a source code repository (such as AWS CodeCommit). This way, you can maintain versions and a history of the templates and the infrastructure that was deployed.
* **Task 4: Exploring templates with AWS CloudFormation Designer**
* *AWS CloudFormation Designer* is a graphic tool for creating, viewing, and modifying AWS CloudFormation templates. With Designer, you can diagram your template resources by using a drag-and-drop interface, and then edit their details through the integrated JSON and YAML editor.
* Whether you are a new to AWS CloudFormation or an experienced AWS CloudFormation user, Designer can help you quickly see the interrelationship between a template's resources. It also enables you to easily modify templates.
* In this task, you will gain some hands-on experience with Designer.
* **Task 5: Deleting the stack**
* When resources are no longer required, AWS CloudFormation can delete the resources built for the stack.
* A *deletion policy* can also be specified against resources. It can preserve or (in some cases) back up a resource when its stack is deleted. This feature is useful for retaining databases, disk volumes, or any resource that might be needed after the stack is deleted.
* The *lab-application* stack was configured to take a snapshot of an Amazon Elastic Block Store (Amazon EBS) disk volume before it is deleted.

## Performance Requirements

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely with the users of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use. The requirement specification for any system can be broadly stated as given below: The system should be able to interface with the existing system The system should be accurate The system should be better than the existing system The existing system is completely dependent on the user to perform all the duties.

* 1. **Software Requirements:** Operating System: Microsoft Windows XP. Technology: Java Server Pages (Jsp’s). Front-End: HTML,CSS

Back-End: ORACLE 10g

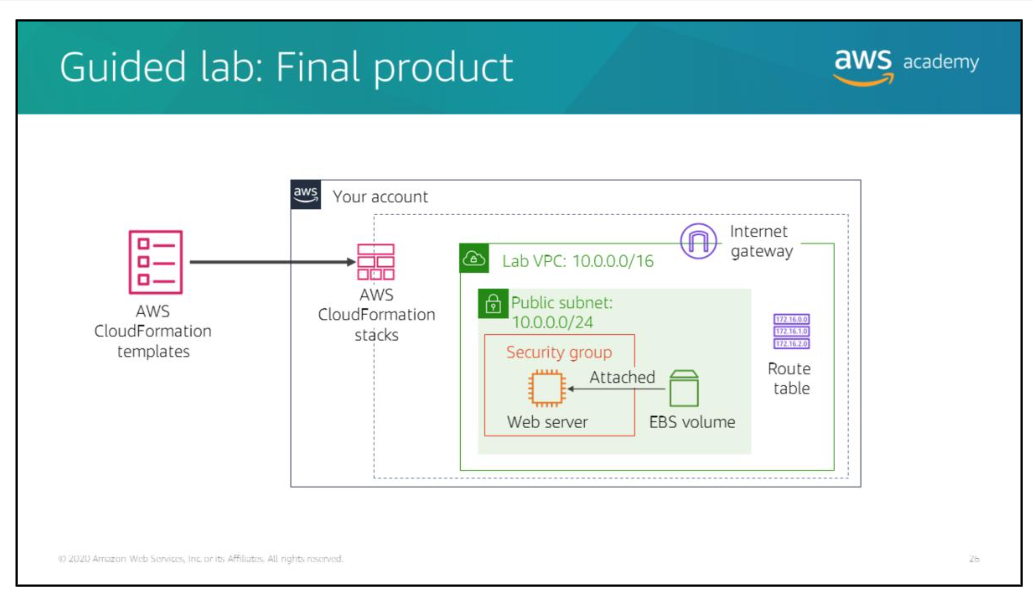
Web-Server: Apache-Tomcat 6.0.32 Platform: Advanced Java Concepts (J2EE).

* 1. **Hardware Requirements:** Processor : Intel P-IV based system RAM : Min. 512 MB

## SYSTEM DESIGN

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development. Object-oriented analysis and design methods are becoming the most widely used methods for computer systems design.

## Architectural Design

 Fig 3.1 Architectural Design

This AWS architecture can be implemented on AWS Fargate using RStudio open source. Fargate is a serverless container service that provides compute capacity for Amazon Elastic Container Service (ECS) and Amazon Elastic Kubernetes Service (EKS).

AWS Fargate removes the need to provision and manage servers, lets you specify and pay for resources per application, and improves security through application isolation. It ensures that the infrastructure your containers run on is always up to date with the required patches.

## Modules

**Section 1: Architectural need**

Up to this point, the café created their AWS resources and configured their applications

Manually mostly by using the AWS Management Console. This approach worked well as a

way for the café to quickly develop a web presence and build out an infrastructure that

supports the needs of employees and customers. However, they find it challenging to

replicate their deployments to new AWS Regions so they can support new cafe locations in

multiple countries.

They would also like to have separate development and production environments that

reliably have matching configurations. They realize that they must start automating to

support continued growth. Their organization has many different architectures, and it needs

a way to consistently deploy, manage, and update these architectures quickly, consistently,

and reliably.

In this module, you will learn about AWS services that provide automation, including AWS

CloudFormation. By using AWS CloudFormation, you will be able to help the cafe meet these

new business requirements.

## Section 2: Reasons to Automate

It takes significant time and energy to build a large scale computing environment.

Many organizations will start using AWS by manually creating an Amazon Simple Storage

Service (Amazon S3) bucket, or launching an Amazon Elastic Compute Cloud (Amazon EC2)

instance and running a web server on it. Then, over time, they manually add more resources

as they find that expanding their use of AWS can meet additional business needs. Soon,

however, it can become challenging to manually manage and maintain these resources

**Section 3: Automating your infrastructure** AWS CloudFormation provisions resources in a repeatable manner. It enables you to build

and rebuild your infrastructure and applications without needing to perform manual actions

or write custom scripts. With AWS CloudFormation, you author a document that describes

what your infrastructure should be, including all the AWS resources that should be a part of

the deployment.

You can think of this document as a model

You then use the model to create the reality, because AWS CloudFormation can actually create the resources in your account.

When you use AWS CloudFormation to create resources, it is called an AWS CloudFormation

stack

. You create a stack, update a stack, or delete a stack. Thus, you can provision resources

in an orderly and predicable way.

Using AWS CloudFormation enables you to treat your infrastructure as code (IaC). Author it

with any code editor, check it into a

version control

system

such as GitHub or AWS

CodeCommit, and review files with team members before you deploy into the appropriate

environments. If the AWS CloudFormation document that you create to model your

deployment is checked in to a version control system, you could always delete a stack, check

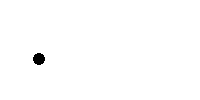
out an older version of the document, and create a stack from it. With version control, you

can use essential rollback capabilities.



* AWS CloudFormation provisions resources in a repeatable manner. It enables you to build
* and rebuild your infrastructure and applications without needing to perform manual actions
* or write custom scripts. With AWS CloudFormation, you author a document that describes
* what your infrastructure should be, including all the AWS resources that should be a part of
* the deployment.
* You can think of this document as a
* model
* . You then use the model to
* create the reality, because AWS CloudFormation can actually create the resources in your
* account.
* When you use AWS CloudFormation to create resources, it is called an AWS CloudFormation
* stack
* . You create a stack, update a stack, or delete a stack. Thus, you can provision resources
* in an orderly and predicable way.
* Using AWS CloudFormation enables you to treat your infrastructure as code (IaC). Author it
* with any code editor, check it into a
* version control
* system
* such as GitHub or AWS
* CodeCommit, and review files with team members before you deploy into the appropriate
* environments. If the AWS CloudFormation document that you create to model your
* deployment is checked in to a version control system, you could always delete a stack, check
* out an older version of the document, and create a stack from it. With version control, you
* can use essential rollback capabilities.

## Section 4: Automating Deployments

 AWS Systems Manager is a management service that is designed to be highly focused on

automation. It enables the configuration and management of systems that run on

-

premises

or in AWS. AWS Systems Manager enables you to identify the instances that you want to

manage, and then define the management tasks that you want to perform on those

instances. AWS Systems Manager is available at no cost, and it can manage both your

Amazon EC2 and on

-

premises resources.

Some tasks you that can accomplish with AWS Systems Manager include:

Collecting software inventory

Applying operating system (OS) patches

Creating system images

Configuring Microsoft Windows and Linux operating systems

These capabilities help you define and track system configurations, prevent drift, and

maintain the software compliance of your Amazon EC2 and on

-

premises configurations.

# Section 5: AWS Elastic Beanstalk



AWS Elastic Beanstalk is another AWS compute service option. It is a platform as a service

(PaaS) offering that facilitates the quick deployment, scaling, and manage

ment

of your web

applications and services. It addresses many of the challenges that you just learned about.

With Elastic Beanstalk, you remain in control of your code, while AWS

maintains the

underlying infrastructure.

The required AWS resources are created and deployed by using a

simple wizard in the AWS Management Console. The wizard asks you to choose the instance

type and size, the database type and size, and what automatic scaling settings you would like

to use. It provides you access to the server log files, and enable Secure HTTP (HTTPS) on the

load balancer.

You upload your code and Elastic Beanstalk automatically handles the deployment

—

including

capacity provisioning, load balancing, automatic scaling, and monitoring application health.

At the same time, you retain full control over the AWS resources that power your application,

and you can access the underlying resources at any time.

There is no additional charge for AWS Elastic Beanstalk. You pay for the AWS resources that

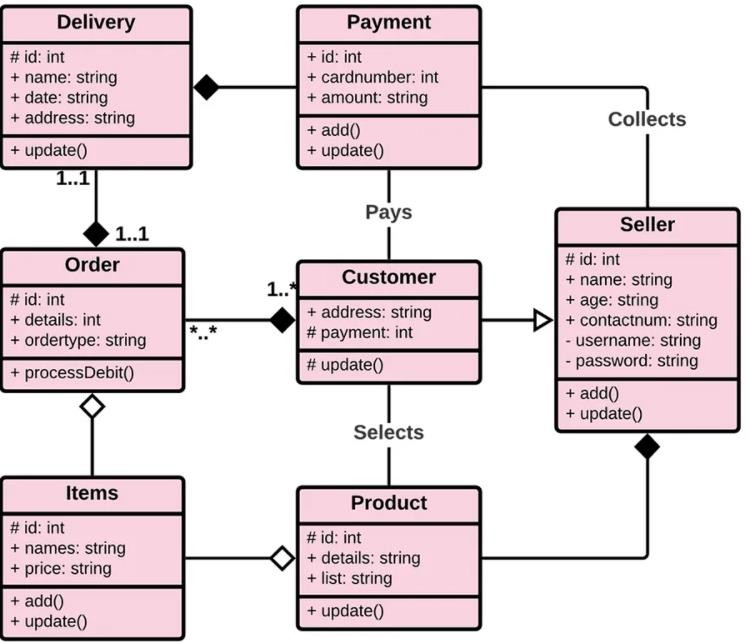
you create to store and run your application, such EC2 instances or S3 buckets. You only pay

for what you use, when you use it.

# UML Diagrams

UML Diagrams for our application as follows

## Class Diagrams



# SYSTEM IMPLEMENTATION

The implementation stage of any project is a true display of the defining moments that makea project a success or a failure. The implementation stage is defined as the system or system modifications being installed and made operational in a production environment. The phase is initiated after the system has been tested and accepted by the user. This phase continues until the system is operating in production in accordance with the defined user requirements

## 4.1 Procedure

## Task 1: Deploying a networking layer

It is a best practice to deploy infrastructure in layers. Common layers are:

* Network (Amazon VPC)
* Database
* Application

This way, templates can be reused between systems. For example, you can deploy a common network topology between development, test, and production environments, or deploy a standard database for multiple applications.

In this task, you will deploy an AWS CloudFormation template that creates a networking layer by using Amazon VPC.

1. Right-click the following link and download the template to your computer: [lab-network.yaml](https://labs.vocareum.com/web/1675032/617641.0/ASNLIB/public/scripts/lab-network.yaml)

 If you want, you can open the template in a text editor to see how the AWS resources are defined.

Templates can be written in JavaScript Object Notation (JSON) or YAML Ain't Markup Language (YAML). YAML is a markup language that is similar to JSON, but it is easier to read and edit.

1. In the **AWS Management Console**, from the services menu, choose **CloudFormation**.
2. Choose create stack and configure these settings.

**Step 1: Specify template**

* + **Template source:** Upload a template file
  + **Upload a template file:** Click choose file then select the **lab-network.yaml** file that you downloaded.
  + Choose next
  + **Step 2: Create Stack**
  + **Stack name:** lab-network
  + Choose next

**Step 3: Configure stack options**

* + In the **Tags** section, enter these values.
    - **Key:** application
    - **Value:** inventory
  + Choose next

**Step 4: Review lab-network**

* + Choose create stack

The template will now be used by AWS CloudFormation to generate a stack of resources in the AWS account.

The specified tags are automatically propagated to the resources that are created, which makes it easier to identify resources that are used by particular applications.

1. Choose the **Stack info** tab.
2. Wait for the **Status** to change to Create complete Choose **Refresh**  every 15 seconds to update the display, if necessary.

You can now examine the resources that were created.

1. Choose the **Resources** tab.

You will see a list of the resources that were created by the template.

 If the list is empty, update the list by choosing **Refresh** .

1. Choose the **Events** tab and scroll through the events log.

The events log shows (from more recent to less recent) the activities that were performed by AWS CloudFormation. Example events include starting to create a resource and then completing the resource creation. Any errors that were encountered during the creation of the stack will be listed in this tab.

1. Choose the **Outputs** tab.

A CloudFormation stack can provide output information, such as the ID of specific resources and links to resources.

Two outputs are listed.

* + **PublicSubnet:** The ID of the public subnet that was created (for example: \_subnet-08aafd57f745035f1\_\_
  + **VPC:** The ID of the VPC that was created (for example: vpc-08e2b7d1272ee9fb4)

Outputs can also be used to provide values to other stacks. This is shown in the **Export name** column. In this case, the VPC and subnet IDs are given export names so that other stacks can retrieve the values. These other stacks can then build resources inside the VPC and subnet that were just created. You will use these values in the next task.

1. Choose the **Template** tab.

This tab shows the template that was used to create the stack—that is, the template that you uploaded while you created the stack. Feel free to examine the template and see the resources that were created. Also feel free to explore the **Outputs** section at the end (this section defined which values to export).

## Task 2: Deploying an application layer

Now that you deployed the network layer, you will deploy an application layer that contains an Amazon Elastic Compute Cloud (Amazon EC2) instance and a security group.

The AWS CloudFormation template will import the VPC and subnet IDs from the Outputs of the existing CloudFormation stack. It will then use this information to create the security group in the VPC and the EC2 instance in the subnet.

1. Right-click the following link and download the template to your computer: [lab-application.yaml](https://labs.vocareum.com/web/1675032/617641.0/ASNLIB/public/scripts/lab-application.yaml)

 If you want, you can open the template in a text editor to see how resources are defined.

1. In the left navigation pane, choose **Stacks**.
2. Select **Create stack > With new resources (standard)**, and then configure these settings.

**Step 1: Specify template**

* + **Template source:** Upload a template file
  + **Upload a template file:** click choose file then select the **lab-application.yaml** file that you downloaded.
  + Choose next

**Step 2: Create Stack**

* + **Stack name:** lab-application
  + **NetworkStackName:** lab-network
  + Choose next

 The Network Stack Name parameter tells the template the name of the first stack that you created (lab-network), so it can retrieve values from the Outputs.

**Step 3: Configure stack options**

* + In the **Tags** section, enter these values.
    - **Key:** application
    - **Value:** inventory
  + Choose next

**Step 4: Review lab-application**

* + Choose create stack

While the stack is being created, examine the details in the **Events** tab and the **Resources** tab. You can monitor the progress of the resource-creation process and the resource status.

1. In the **Stack info** tab, wait for the **Status** to change to CREATE\_COMPLETE.

Your application is now ready!

1. Choose the **Outputs** tab.
2. Copy the **URL** that is displayed, open a new web browser tab, paste the URL, and press ENTER.

The browser tab will open the application, which is running on the web server that this new CloudFormation stack created.

A CloudFormation stack can use reference values from another CloudFormation stack.

  WebServerSecurityGroup:

    Type: AWS::EC2::SecurityGroup

    Properties:

      GroupDescription: Enable HTTP ingress

      VpcId:

        Fn::ImportValue:

          !Sub ${NetworkStackName}-VPCID

The last line uses the network stack name that you provided (lab-network) when the stack was created. It imports the value of lab-network-VPCID from the Outputs of the first stack. It then inserts the value into the VPC ID field of the security group definition. The result is that the security group is created in the VPC that was created by the first stack.

## Task 3: Updating a Stack

AWS CloudFormation can also update a stack that has been deployed. When you update a stack, AWS CloudFormation will only modify or replace the resources that are being changed. Any resources that are not being changed will be left as-is.

In this task, you will update the lab-application stack to modify a setting in the security group.

First, you will examine the current settings for the security group.

1. In the **AWS Management Console**, from the services menu, choose **EC2**.
2. In the left navigation pane, choose **Security Groups**.
3. Select the check box for  **lab-application-WebServerSecurityGroup...**.
4. Choose the **Inbound rules** tab.

Currently, only one rule is in the security group. The rule permits HTTP traffic.

You will now return to AWS CloudFormation to update the stack.

1. From the services menu, choose **CloudFormation**.
2. Right-click the following link and download the updated template to your computer: [lab-application2.yaml](https://labs.vocareum.com/web/1675032/617641.0/ASNLIB/public/scripts/lab-application2.yaml)
3. In the **Stacks** list of the **AWS CloudFormation console**, select **lab-application**.
4. Choose update and configure these settings.
   * Select replace current template
   * **Template source:** upload a template fil
   * **Upload a template file:** Click choose file then select the **lab-application2.yaml** file that you downloaded.
5. Choose next in each of the next three screens to advance to the **Review lab-application** page.

In the **Change set preview** section at the bottom of the page, AWS CloudFormation displays the resources that will be updated:

This change set preview indicates that AWS CloudFormation will Modify the WebServerSecurityGroup without needing to replace it (Replacement = False). This change set means that the security group will have a minor change applied to it, and no references to the security group will need to change.

1. Choose update stack
2. In the **Stack info** tab, wait for the **Status** to change to UPDATE\_COMPLETE.

 Update the status by choosing **Refresh** every 15 seconds, if necessary.

You can now verify the change.

1. Return to the **Amazon EC2 console** and from the left navigation pane, choose **Security Groups**.
2. In the **Security Groups** list, select **lab-application-WebServerSecurityGroup**.

The **Inbound rules** tab should display an additional rule that allows SSH traffic over TCP port 22.

This subtask demonstrates how changes can be deployed in a repeatable, documented process. The AWS CloudFormation templates can be stored in a source code repository (such as AWS CodeCommit). This way, you can maintain versions and a history of the templates and the infrastructure that was deployed.

## Task 4: Exploring templates with AWS CloudFormation Designer

AWS CloudFormation Designer is a graphic tool for creating, viewing, and modifying AWS CloudFormation templates. With Designer, you can diagram your template resources by using a drag-and-drop interface, and then edit their details through the integrated JSON and YAML editor.

Whether you are a new to AWS CloudFormation or an experienced AWS CloudFormation user, Designer can help you quickly see the interrelationship between a template's resources. It also enables you to easily modify templates.

In this task, you will gain some hands-on experience with Designer.

1. From the services menu, choose **CloudFormation**.
2. In the left navigation pane, choose **Designer**.

**Tip:** You might need to expand the left navigation pane by choosing the menu icon.

1. Choose the **File**  menu, select **Open > Local file**, and select the **lab-application2.yaml** template that you downloaded previously.

Designer will display a graphical representation of the template:

Instead of drawing a typical architecture diagram, Designer is a visual editor for AWS CloudFormation templates. It draws the resources that are defined in a template and their relationship to each other.

1. Experiment with the features of the Designer. Some things to try are:
   * Click the displayed resources. The lower pane will then display the portion of the template that defines the resources.
   * Try dragging a new resource—from the **Resource types** pane on the left—into the design area. The definition of the resource will be automatically inserted into the template.
   * Try dragging the resource connector circles to create relationships between resources.
   * Open the **lab-network.yaml** template that you downloaded earlier in the lab and also explore its resources in Designer.

## Task 5: Deleting the stack

When resources are no longer required, AWS CloudFormation can delete the resources built for the stack.

A deletion policy can also be specified against resources. It can preserve or (in some cases) back up a resource when its stack is deleted. This feature is useful for retaining databases, disk volumes, or any resource that might be needed after the stack is deleted.

The lab-application stack was configured to take a snapshot of an Amazon Elastic Block Store (Amazon EBS) disk volume before it is deleted. The code in the template that accomplishes that configuration is:

DiskVolume:

  Type: AWS::EC2::Volume

  Properties:

    Size: 100

    AvailabilityZone: !GetAtt WebServerInstance.AvailabilityZone

    Tags:

      - Key: Name

        Value: Web Data

  DeletionPolicy: Snapshot

The DeletionPolicy in the final line directs AWS CloudFormation to create a snapshot of the disk volume before it is deleted.

You will now delete the lab-application stack and see the results of this deletion policy.

1. Return to the main **AWS CloudFormation console** by choosing the Close link at the top of the Designer page (choose **Leave page** if prompted).
2. In the list of stacks, choose the **lab-application** link.
3. Choose delete
4. Choose delete stack

You can monitor the deletion process in the **Events** tab and update the screen by choosing **Refresh**  occasionally. You might also see an events log entry that indicates that the EBS snapshot is being created.

1. Wait for the stack to be deleted. It will disappear from the stacks list.

The application stack \_\_ removed, but the network stack remained untouched. This scenario reinforces the idea that different teams (for example, the network team or the application team) could manage their own stacks.

You will now verify that a snapshot of the EBS volume was created before the EBS volume was deleted.

1. From the services menu, choose **EC2**.
2. In the left navigation pane, choose **Snapshots**.

You should see a snapshot with a **Started** time in the last few minutes.

# 5. OUTPUT SCREENS

Output Screens of various functionalities in our application are shown over here along with the description.

## Task 1: Deploying a networking layer

## It is a best practice to deploy infrastructure in *layers*. Common layers are:

## Network (Amazon VPC)

## Database

## Application

## In this task, you will deploy an AWS CloudFormation template that creates a *networking layer* by using Amazon VPC.

Graphical user interface, text, application

Description automatically generated

### Fig 5.1

## Task 2: Deploying an application layer

* Now that you deployed the *network layer*, you will deploy an *application layer* that contains an Amazon Elastic Compute Cloud (Amazon EC2) instance and a security group.
* The AWS CloudFormation template will *import* the VPC and subnet IDs from the *Outputs* of the existing CloudFormation stack. It will then use this information to create the security group in the VPC and the EC2 instance in the subnet.

Graphical user interface, text, application, email

Description automatically generated

### fig 5.2

* **Task 3: Updating a Stack**
* In this task, you will update the *lab-application* stack to modify a setting in the security group.

Graphical user interface, text, application, email

Description automatically generated

### fig 5.3

* **Task 4: Exploring templates with AWS CloudFormation Designer**

**Diagram

Description automatically generated**

**fig 5.4**

**Task 5: Deleting the stack**

Graphical user interface, application, Teams

Description automatically generated

### fig 5.5

# INTERNSHIP FEEDBACK

## 6.1 CHALLENGES FACED

It was a good experience performing all the lab activities and also refering the keen power point presentations provided . Also it was a new experience for us to enhance your skills by using all the applications provided in the internship. we have got hands-on experience to use each and every tool in AWS platform by performing various lab activities . The guided labs were the building blocks which are to be learnt to perform the challenging labs which were really challenging and compact .

# CONCLUSION AND FUTURE SCOPE

## CONCLUSION

* After completing this, you should be able to:
* Use AWS CloudFormation to deploy a virtual private cloud (VPC) networking layer
* Use AWS CloudFormation to deploy an application layer that references the networking layer
* Explore templates with AWS CloudFormation Designer
* Delete a stack that has a deletion policy

.

## FUTURE SCOPE

When you’re building a new application you need to get new features out fast. Managing your application’s infrastructure as well as responding to changing conditions can be delay and error prone if you rely on manual processes.

Treating your infrastructure as code allows you to provision and update complex environments in a predictable manner.

You can also offer a pre-defined catalogue of environments for development, testing and experimentation, unlocking your ability to innovate

# CAPSTONE PROJECT

## PROJECT OVERVIEW

This project provides you with an opportunity to demonstrate the solution design skills that you develop throughout this course. Your assignment is to design and deploy a solution for the following case. By the end of this project, you should be able to apply the architectural design principles that you learned in this course to: Deploy a PHP application that runs on an Amazon Elastic Compute Cloud (Amazon EC2) instance. Create a database instance that the PHP application can query . Create a MySQL database from a structured query language (SQL) dump file .Update application parameters in an AWS Systems Manager Parameter Store

.Secure the application to prevent public access to backend systems

## INTRODUCING THE EXAMPLE SOCIAL RESEARCH ORGANIZATION

Example Social Research Organization is a (fictitious) nonprofit organization that providesa website for social science researchers to obtain global development statistics. For example, visitors to the site can look up various data, such as the life expectancy for any country in the world over the past 10 years**.** Shirley Rodriguez, a researcher at the organization, developed the website. She thought it would be valuable to share the data that she had gathered with other researchers. Shirley stores the data in a MySQL database,and the data is available through a PHP website that she built. She initially published the site through a commercial hosting company that provides limited support for technical issues and security. Over the past year, Shirley’s website has grown in popularity. As a result of increased traffic, she started receiving complaints that the site is not as responsiveas it used to be. She also experienced an attempted ransom ware security breach. The security breach was unsuccessful, but her supervisor, Mateo Jackson, suggested that Shirley investigate new ways to host the website. Shirley heard about

Amazon Web Services (AWS), and initially moved her website and database to an EC2 instance that

runs in a public subnet. She also runs an instance of MySQL on the same EC2 instance. Shirley approached your team to make sure that her current design follows best practices. She wants to make sure that she has a robust and secure website. One of your colleagues started the process of migrating the site to a more secure implementation, but they were reassigned to another project. Your tasks are to complete the implementation, make sure that the website is secure, and confirm that the website returns data from the query page.

## SOLUTION REQUIREMENTS

* + - Provide secure hosting of the MySQL database
    - Provide secure access for an administrative user
    - Provide anonymous access to web users
    - Run the website on a t2.small EC2 instance, and provide Secure Shell (SSH) access to administrators
    - Provide high availability to the website through a load balancer
    - Store database connection information in the AWS Systems Manager Parameter Store
    - Provide automatic scaling that uses a launch template

## PROJECT DELIVERABLES

To complete this assignment, you must:

* + - Deploy a PHP application that meets the system requirements outlined above
    - Submit a diagram that illustrates your solution
    - Submit a written summary of the design decisions that you made to achieve the result

# BIBLIOGRAPHY

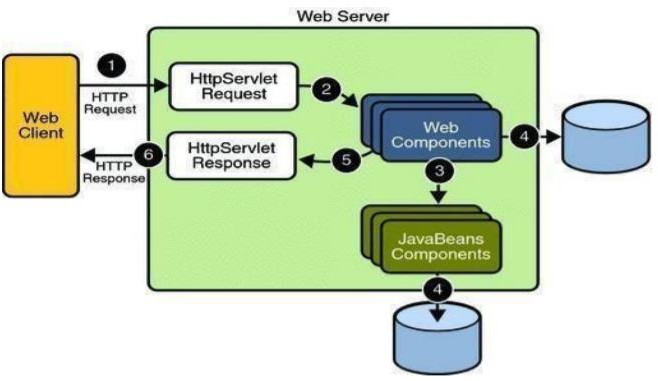
1. Herbert Schildt, The Complete Reference Java2 Fifth Edition, Tata McGraw-Hill Edition 2002.
2. Grady Booch, James Rumbaugh, Ivar Jacobson. The Unified Modeling Language User Guide. Addison-Wesley, Reading, Mass., 1999.
3. [www.coderanch.com](http://www.coderanch.com/)
4. [www.java.sun.com](http://www.java.sun.com/)
5. [www.w3schools.com](http://www.w3schools.com/) [www.wikipedia.org](http://www.wikipedia.org/)

## APPENDIX-A:JAVA TECHNOLOGY

**About Java**:

Initially the language was called as “oak” but it was renamed as “java” in 1995.The primary motivation of this language was the need for a platform-independent(i.e. architecture neutral)language that could be used to create software to be embedded in various consumer electronic devices.

* + Java is a programmer’s language
  + Java is cohesive and consistent
  + Except for those constraint imposed by the Internet environment. Java gives the programmer, full control **Servlets/JSP**



### A Servlet Is a generic server extension. a Java class that can be loaded dynamically to expand the functionality of a server. Servlets are commonly used with web servers. Where they can take the place CGI scripts.A servlet is similar to proprietary server

**extension, except that it runs inside a Java Virtual Machine (JVM) on the server, so it is safe and portable .Servlets operate solely within the domain of the server.**

**Features of Servlets:**

* + - Servlets are persistent.Servlet are loaded only by the web server and can maintain services between requests.
    - Servlets are fast. Since servlets only need to be loaded once, they offer much better performance over their CGI counterparts.
    - Servlets are platform independent.
    - Servlets are extensible Java is a robust, object-oriented programming language, which easily can be extended to suit your needs.
    - Servlets are secure
    - Servlets are used with a variety of client.

## Invoking Servlets

A servlet invoker is a servlet that invokes the “server” method on a named servlet.If the servlet is not loaded in the server,then the invoker first loades the servlet(either form local disk or from the network) and the then invokes the “service” method.Also like applets,local servlets in the server can be identified by just the class name.In other words, if a servlet name is not absolute.it is treated as local.

A Client can Invoke Servlets in the Following Ways:

* + The client can ask for a document that is served by the servlet.
  + The client(browser) can invoke the servlet directly using a URL, once it has been mapped using the SERVLET ALIASES Section of the admin GUI  The servlet can be invoked through server side include tags.
  + The servlet can be invoked by placing it in the servlets/directory

## The Servlet Life Cycle:-

The Servlet life cycle is one of the most exciting features of Servlets.This life cycle is a powerful hybrid of the life cycles used in CGI programming and lower-level NSAPI and ISAPI programming.

The servlet life cycle allows servlet engines to address both the performance and resource problems of CGI and the security concents of low level server API programming.

Servlet life cycle is highly flexible Servers hava significant leeway in how they choose to

support servlets.The only hard and fast rule is that a servlet engine must confor to the following life cycle contact:

* + Create and initialize the servlets
  + Handle zero or more service from clients
  + Destroy the servlet and then garbage Collects it.

It’s perfectly legal for a servlet to be loaded, created and initialzed in its own JVM,only to be destroyed and garbage collected without handling any client request or after handling just one request

## Init and Destroy:-

Just like Applets servlets can define init() and destroy() methods, A servlets init(ServiceConfig) method is called by the server immediately after the server constructs the servlet’s instance.Depanding on the server and its configuration, this can be at any of these times

* + When the server states
  + When the servlet is first requested, just before the service() method is invoked
  + At the request of the server administrator

In any case, init() is guaranteed to be called before the servlet handles its first request. The init() method is typically used to perform servlet initialization creating or loading objects that are used by the servlet in handling of its request. In order to providing a new servlet any information about itself and its environment, a server has to call a servelts init() method and pass an object that implement the ServletConfig interface.

This ServletConfig object supplies a servlet with information about its initialization parameters.These parameters are given to the servlets and are not associated with any single request.They can specify initial values, such as where a counter should begin counting, or default values, perhaps a template to use when not specified by the request,

The server calls a servlet’s destroy() method when the servlet is about to be unloaded. In the destroy() method, a servlet should free any resources it has acquired that will not be garbage collected. The destroy() method also gives a servlet a chance to write out its unsaved. cached information or any persistent information that should be read during the next call to init().

## Session Tracking:

HTTP is a stateless protocol, it provides no way for a server to recognize that sequenc of requests is all from the same client. The solution for this is for client to introduce itself as it makes each request, Each clients needs to provide a unique identifier that lets the server identify it, or it needs to give some information that the server can use to properly handle the request, There are several ways to send this introductory information with each request such as:

### User Authentication:

One way to perform session tracking is to leverage the information that comes with user authorization. When a web server restricts access to some of its resources to only those clients that log in using a recognized username and password. After the client logs in, the username is available to a servlet through getRemoteUser()

The biggest advantage of using user authorization to perform session tracking is that it’s easy to implement and easy to identify each client.Another advantage is that the technique works even when the user accesses your site form or exists her browser before coming back.

### Hidden Form Fields:

One way to support anonymous session tracking is to use hidden from fields. As the name implies, these are fields added to an HTML, form that are not displayed in the client’s browser, They are sent back to the server when the form that contains them is submitted.

In a sense, hidden form fields define constant variables for a form. To a servlet receiving a submitted form, there is no difference between a hidden fields and a visible filed.

As more and more information is associated with a clients session . It can become burdensome to pass it all using hidden form fields. In these situations it’s possible to pass on just a unique session ID that identifies as particular clients session.

That session ID can be associated with complete information about its session that is stored on the server.

The advantage of hidden form fields is their ubiquity and support for anonymity. Hidden fields are supported in all the popular browsers, they demand on special server requirements, and they can be used with clients that haven’t registered or logged in. The major disadvantage with this technique, however is that works only for a sequence of dynamically generated forms, The technique breaks down immediately with static documents,emailed documents book marked documents and browser shutdowns.

### ⚫ URL Rewriting:

URL rewriting is another way to support anonymous session tracking, With URL rewriting every local URL the user might click on is dynamically modified. or rewritten, to include extra, information. The extra information can be in the deform of extra path information, added parameters, or some custom, server-specific.URL change. Due to the limited space available in rewriting a URL, the extra information is usually limited to a unique session.

Each rewriting technique has its own advantage and disadvantage

Using extra path information works on all servers, and it works as a target for forms that use both the Get and Post methods. It does not work well if the servlet has to use the extra path information as true path information

The advantages and disadvantages of URL.rewriting closely match those of hidden form fileds,The major difference is that URL rewriting works for all dynamically created documents, such as the Help servlet, not just forms. With the right server support, custom URL rewriting can even work for static documents.

### ⚫ Persistent Cookies:

A fourth technique to perform session tracking involves persistent cookies. A cookie is a bit of information sent by a web server to a browser that can later be read back form that browser. When a browser receives a cookie, it saves the cookie and there after sends the cookie back to the server each time it accesses a page on that server, subject to certain rules.

Because a cookie’s value can uniquely identify a client, cookies are often used for session tracking.

Persistent cookies offer an elegant, efficient easy way to implement session tracking. Cookies provide as automatic an introduction for each request as we could hope for. For each request, a cookie can automatically provide a client’s session ID or perhaps a list of clients performance. The ability to customize cookies gives them extra power and versatility.

## JDBC

### What is JDBC?

JDBC is a Java API for executing SQL,Statements.As a point of interest JDBC is trademarked name and is not an acronym; nevertheless,Jdbc is often thought of as standing for Java Database Connectivity. It consists of a set of classes and interfaces written in the

Java Programming language.JDBC provides a standard API fortool/database developers and makes it possible to write database applications using a pureJava API

Using JDBC, it is easy to send SQL statements to virtually program will be able to send SQL statements to the appropriate database. The Combination of Java and JDBC lets a programmer write it once and run it anywhere.

### What Does JDBC Do?

* + Establish a connection with a database
  + Send SQL statements o Process the results.

### JDBC Driver Types

An individual database system is accessed via a specific JDBC driver that implements the java.sql.Driver interface. Drivers exist for nearly all-popular RDBMS systems, through few are available for free. Sun bundles a free JDBC-ODBC bridge driver with the JDK to allow access to a standard ODBC,data sources, such as a Microsoft Access database, Sun

advises against using the bridge driver for anything other than development and very limited development.

JDBC drivers are available for most database platforms, from a number of vendors and in a number of different flavours. There are four driver categories

### Type 01-JDBC-ODBC Bridge Driver

Type 01 drivers use a bridge technology to connect a java client to an ODBC database service. Sun’s JDBC-ODBC bridge is the most common type 01 driver. These drivers implemented using native code.

### Type 02-Native-API party-java Driver

Type 02 drivers wrap a thin layer of java around database-specific native code libraries for Oracle databases, the native code libraries might be based on the OCI(Oracle call Interface) libraries, which were originally designed for **c/c++** programmers, Because type- 02 drivers are implemented using native code, in some cases they have betterperformancethan their alljava counter parts. They add an element of risk, however, because a defect ina driver’s native code section can crash the entire server

### Type 03-Net-Protocol All-Java Driver

Type 03 drivers communicate via a generic network protocol to a piece of custom middleware.The middleware component might use any type of driver to provide the actual database access. These drivers are all java, which makes them useful for applet

deployment

and safe for servlet deployment

### Type-04-native-protocol All-java Driver

Type 04 drivers are the most direct of the lot. Written entirely in java, Type 04 drivers understand database-specific networking. protocols and can access the database directly without any additional software

## Oracle

Oracle is a relational database management system, which organizes data in the form of tables. Oracle is one of many database servers based on RDBMS model, which manages a seer of data that attends three specific things-data structures, data integrity and data manipulation.

With oracle cooperative server technology we can realize the benefits of open, relational systems for all the applications. Oracle makes efficient use of all systems resources, on all hardware architecture; to deliver unmatched performance, price performance and scalability. Any DBMS to be called as RDBMS has to satisfy Dr.E.F.Codd’s rules.

### Features of Oracle:

**Portable**

The Oracle RDBMS is available on wide range of platforms ranging from PCs to super computers and as a multi user loadable module for Novel NetWare, if you develop application on system you can run the same application on other systems without any modifications.

### Compatible

Oracle commands can be used for communicating with IBM DB2 mainframe RDBMS that is different from Oracle, which is Oracle compatible with DB2. Oracle RDBMSis a high performance fault tolerant DBMS, which is specially designed for online transactionprocessing and for handling large database applications.

### Multithreaded Server Architecture

Oracle adaptable multithreaded server architecture delivers scalable high performance for very large number of users on all hardware architecture including symmetric multiprocessors (sumps) and loosely coupled multiprocessors. Performance is achieved by

eliminating CPU, I/O, memory and operating system bottlenecks and by optimizing the Oracle DBMS server code to eliminate all internal bottlenecks.

Oracle has become the most popular RDBMS in the market because of its ease of use Client/server architecture.

* Data independence.
* Ensuring data integrity and data security.
* Managing data concurrency.
* Parallel processing support for speed up data entry and online transaction processing used for applications.
* DB procedures, functions and packages.

## HTML

Hypertext Markup Language(HTML), the languages of the world wide web(WWW),allows users to produces web pages that included text, graphics and pointer to other web pages (Hyperlinks).

HTML is not a programming language but it is an application of ISO Standard 8879,SGML(Standard Generalized Markup Language),but specialized to hypertext and adapted to the Web. We can navigate through the information based on out interest and preference. A markup language is simply a series of items enclosed within the elements should be displayed.

Hyperlinks are underlined or emphasized works that load to other documents or some portions of the same document.

Html can be used to display any type of document on the host computer, which can be geographically at a different location. It is a versatile language and can be used on any platform or desktop.

HTML provides tags(special codes) to make the document look attractive.

HTML provides are not case-sensitive. Using graphics,fonts,different sizes, color, etc.. can enhance the presentation of the document. Anything that is not a tag is part of the document itself.

### Basic Html Tags:

|  |  |
| --- | --- |
| <!-- --> | Specific Comments. |
| <A>………</A> | Creates Hypertext links. |
| <B>………</B> | Text is changed to bold |
| <Body>…….</Body> | Contains all tags and text in the Html-document |
| <DD> </DD> | Definition of a term. |

<TABLE>……</TABLE> Creates table

|  |  |
| --- | --- |
| <Td> </Td> | Indicates table data in a table. |
| <Tr> </Tr> | Designates a table row |
| <Th> </Th> | Creates a heading in a table. |

### Advantages:

* A HTML document is small and hence easy to send over the net.It is small because it does not include formatted information.
* HTML is platform independent.
* HTML tags are not case-sensitive.

## JAVA SCRIPT

JavaScript is a compact , object-based scripting language for developing client and server internet applications. Netscape Navigator 2.0 interprets JavaScript statements embedded directly in an HTML page. and Livewire enables you to create server-based applications similar to common gateway interface(cgi) programs.In a client application for Navigator, JavaScript statements embedded in an HTML page can recognize and respond to user events such as mouse clicks form Input, and page navigation.For example, you can write a JavaScript function to verify that users enter valid information into a form requesting a telephone number or zip code . Without any network transmission, an Html page with embedded Java Script can interpret the entered text and alert the user with a message dialog if the input is invalid or you can use JavaScript to perform an action (such as play an audio file, execute an applet, or communicate with a plug-in) in response to the user opening or exiting a page.

# APPENDIX-B:UNIFIED MODELING LANGUAGE

The Unified Modeling Language (UML) is a general-purpose visual modeling language that is used to specify, visualize, construct, and document the artifacts of a software system. It captures decisions and understanding about systems that must be constructed. It is used to understand, design, browse, configure, maintain, and control information about such systems. It is intended for use with all development methods, lifecycle stages, application domains, and media. The modeling language is intended to unify past experience about modeling techniques and to incorporate current software best practices into a standard approach. UML includes semantic concepts, notation, and

guidelines. It has static, dynamic, environmental, and organizational parts. It is intended to be supported by interactive visual modeling tools that have code generators and report writers. The UML specification does not define a standard process but is intended to be useful with an iterative development process. It is intended to support most existing object oriented development processes. The UML captures information about the static structure and dynamic behavior of a system. A system is modeled as a collection of discrete objects that interact to perform work that ultimately benefits an outside user. The static structure defines the kinds of objects important to a system and to its implementation, as well as the relationships among the objects. The dynamic behavior defines

the history of objects over time and the communications among objects to accomplish goals.

Modeling a system from several separate but related viewpoints permits it to be understood for different purposes.

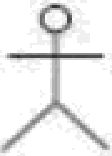
The UML also contains organizational constructs for arranging models into packages that permit software teams to partition large systems into workable pieces, to understand and control dependencies among the packages, and to manage the versioning of model units in a complex development environment. It contains constructs for representing implementation decisions and for organizing run-time elements into components.

UML is not a programming language. Tools can provide code generators from UML into a variety of programming languages, as well as construct reverseengineered models fromexisting programs. The UML is not a highly formal language intended for theorem proving.

There are a number of such languages, but they are not easy to understand or to use for most purposes. The UML is a general-purpose modeling language. For specialized domains, such as GUI layout, VLSI circuit design, or rule-based artificial intelligence, a more specialized tool with a special language might be appropriate. UML is a discrete modeling language.

It is not intended to model continuous systems such as those found in engineering and physics. UML is intended to be a universal general-purpose modeling language for discrete systems such as those made of software, firmware, or digital logic.

*Icons on use case diagrams*

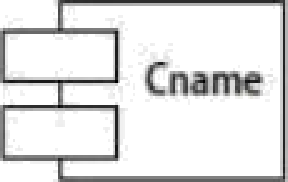






Jfoxs *on class, component, deployment, and collaborañon diagrams*

"-"^\*"



Cnam

|  |
| --- |
| Cnane |
|  |
|  |

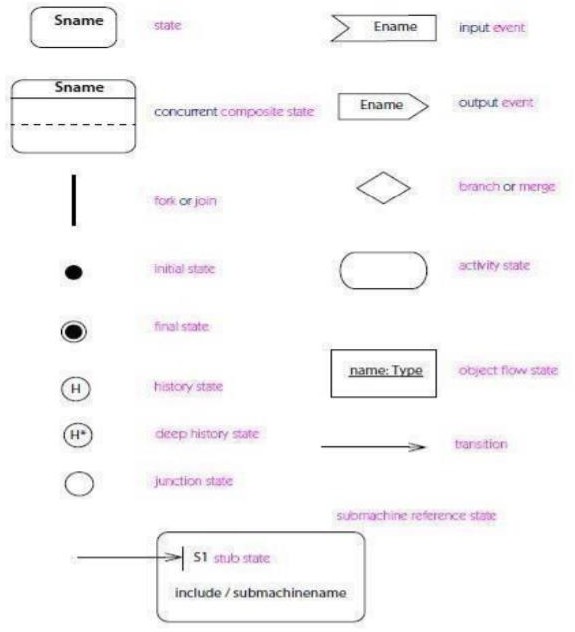






*Icons on siaiecharf and aciiviy diagrams*





# APPENDIX C: ABSTRACT

|  |  |  |
| --- | --- | --- |
| **Sreenidhi Institute of Science and Technology**  **Summer Industry Internship -I** | | |
| **Batch No:19-D7** | | **Title** |
| **Roll No** | **Name** |
| 19311A05Q8 | S.AKSHAYA | AUTOMATING INFRASTRUCTURE DEPLOYMENT |
| 19311A05R9 | Y.NIKITHA REDDY |
| 19311A05U5 | M.ARUN |

**ABSTRACT**

* Customers need infrastructure in a consistent manner. But Deploying infrastructure in a consistent, reliable manner is difficult. It requires people to follow documented procedures without taking any undocumented shortcuts. It can also be difficult to deploy infrastructure out-of-hours when fewer staff are available. AWS CloudFormation changes this situation by defining infrastructure in a template that can be automatically deployed—even on an automated schedule.

|  |  |  |
| --- | --- | --- |
| **Student 1:** S.AKSHAYA | **Project Coordinator** | **HOD-CSE** |
| **Student 2:** Y.NIKITHA REDDY | **Mr.Pasam Ramu** | **Dr Aruna Varanasi** |
| **Student 3:** M.ARUN | **Assistant Professor**  **Dept of CSE** | **Professor** |

## APPENDIX D: CORRELATION BETWEEN THE SUMMER INDUSTRY INTERNSHIP-I AND THE PROGRAMOUTCOMES (POS), PROGRAM SPECIFIC OUTCOMES (PSOS)

|  |  |  |
| --- | --- | --- |
| **Batch No:19-D7** | | **Title** |
| **Roll No** | **Name** |
| 19311A05Q8 | S.Akshaya | AUTOMATING INFRASTRUCTURE DEPLOYMENT |
| 19311A05R9 | Y.Nikitha Reddy |
| 19311A05U5 | M.Arun |

Table 1: Project/Internship correlation with appropriate POs/PSOs (Please specify level of Correlation, H/M/L against

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **H** | **High** | **M** | **Moderate** | **L** | **Low** |

POs/PSOs)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**  **Projects Correlation with POs/PSOs** | | | | | | | | | | | | | | |
| **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **M** | **L** | **L** | **H** | **H** | **L** | **M** | **H** | **M** | **H** | **H** | **H** | **H** | **H** | **M** |

|  |  |  |
| --- | --- | --- |
| **Student 1:** S.Akshaya  **Student 2:** Y.Nikitha Reddy  **Student 3:** M.Arun | **Project Coordinator Mr.Pasam Ramu**  **Assistant Professor Dept of CSE** | **HOD-CSE**  **Dr Aruna Varanasi Professor** |
|  |  | 39 |

## APPENDIX E: DOMAIN OF INTERNSHIP AND NATURE OF INTERNSHIP

|  |  |  |
| --- | --- | --- |
| **Batch No:19-D7** | | **Title** |
| **Roll No** | **Name** |
| 19311A05Q8 | S.Akshaya | AUTOMATING INFRASTRUCTURE DEPLOYMENT |
| 19311A05R9 | Y.Nikitha Reddy |
| 19311A05U5 | M.Arun |

Table 2: Nature of the Project/Internship work (Please put YES Appropriate for your project)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Batch No.** | **Roll No.** | **Product** | **Application** | **Research** |
|
| **19-E9** | 19311A05Q8 |  | YES |  |
| 19311A05R9 |
| 19311A05U5 |

|  |  |  |
| --- | --- | --- |
| **Student 1:** S.Akshaya  **Student 2:** Y.Nikitha Reddy  **Student 3:** M.Arun | **Project Coordinator**  **Mr.Pasam Ramu**  **Assistant Professor Dept of CSE** | **HOD-CSE**  **Dr Aruna Varanasi Professor** |

Table 3: Domain of the Project/ Internship work (Please tick √ Appropriate for your project)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Batch No.** | **Roll No.** | **ARTIFICIAL INTELLIGENCE, MACHINE LEARNING AND DEEP LEARNING** | **COMPUTER NETWORKS, INFORMATION SECURITY, CYBER SECURITY** | **DATA WAREHOUSING, DATA MINING, BIG DATA ANALYTICS** | **CLOUD COMPUTING** | **SOFTWARE ENGINEERING** | **INTERNET OF THINGS** |
| 19-E9 | 19311A05Q8 |  |  |  | YES |  |  |
| 19311A05R9 |
| 19311A05U5 |

|  |  |  |
| --- | --- | --- |
| **Student 1:**S.Akshaya | **Project Coordinator** | **HOD-CSE** |
| **Student 2 :**Y.Nikitha Reddy | **Mr.Pasam Ramu** | **Dr Aruna Varanasi** |
| **Student 3:** M.Arun | **Assistant Professor** | **Professor** |

**Dept of CSE**