RAJKIYA ENGINEERING COLLEGE BIJNOR

(Affiliated to Dr. A.P.J Abdul Kalam Technical University)

SESSION – 2020-21

INDUSTRIAL TRAINING REPORT

ON

CLOUD COMPUTING with AMAZON WEB SERVICES

Submitted in the partial fulfillment of the requirement for

Bachelor of Technology

In

Information Technology

Submitted By-

Kumar Shanu (1773513025)

Submitted To-

Mr. Santosh Kumar

(Assistant Professor)

(Information Technology)

ABSTRACT

This Report presents the experience and skills gained during my 6 week of industrial training undertaken at "**INTERNSHALA"** (Work from home).

My training was on Cloud Computing with Amazon Web Services (AWS), including AWS Management Console, AWS Compute Services, AWS Storage Services, AWS Networking, AWS Load Balancer and AWS Database Services.

During the period, I acquired practical knowledge and skills in using Cloud Computing with AWS Platform.
Cloud Computing is the delivery of computing services such as servers, storage, databases, networking, software, analytic, intelligence, and more, over the Cloud (Internet).

ACKNOWLEDGMENT

It is always a pleasure to remind the fine people in the Engineering Workshops for their sincere guidance I received to uphold my practical as well as theoretical skills in engineering.

I am very pleased to express our gratitude to **MR. Santosh Kumar** (Asst. Prof.) for his valuable guidance, encouragement and facilities provided during period of our Industrial training.

I would also like to acknowledge with much appreciation the crucial role of the faculty members, Technical staff who gave the permission to use all required resources.

Last but not least, many thanks go to the Family members & friends who have given his full effort in guiding the team in achieving the goal as well as his encouragement to maintain our progress on track. I would to appreciate the guidance given by other supervisor as well as the panels especially in our Industrial training.

CERTIFICATE

INTERNSHALA TRAININGS

Certificate of Training

KUMAR SHANU,

student of Rajkiya Engineering College Bijnor, has successfully completed a six weeks online training on **Cloud Computing with AWS** from 1st July, 2020 to 12th August, 2020. The training consisted of Fundamentals of Cloud Computing, Amazon Compute Services, Storage and Backup Service, Networking and Content Delivery, Identity & Access Management (IAM), High Availability Services and Database & Serverless Computing modules. In the final assessment, KUMAR scored 59% marks. We wish KUMAR all the best for the future.

Sarvesh Agrawal
Founder & CEO, Internshala

Date of certification: 2020-08-09 Certificate no. : 9086D43C-65B4-ADB8-F43D-B7032E09CDF6

For certificate authentication, please visit https://trainings.internshala.com/verify_certificate

INDEX

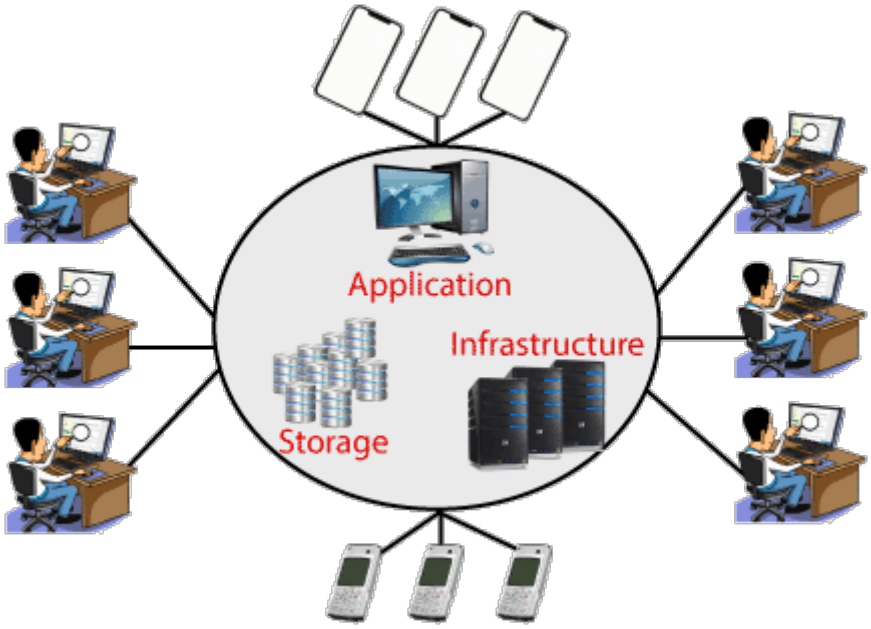
Cloud Computing with Amazon Web Services (AWS)

Fundamental of Cloud Computing

Introduction to Cloud

Cloud Computing is the delivery of computing services such as servers, storage, databases, networking, software, analytic, intelligence, and more, over the Cloud (Internet).

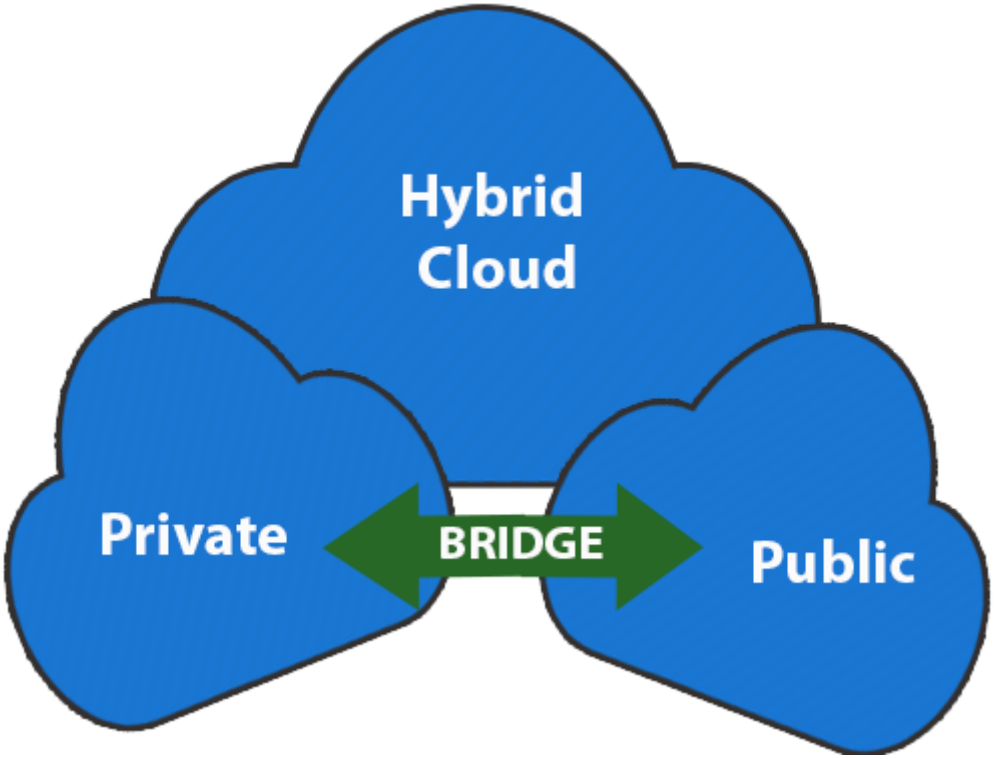
Cloud Computing provides an alternative to the on-premises datacentre. With an on-premises datacentre, we have to manage everything, such as purchasing and installing hardware, virtualization, installing the operating system, and any other required applications, setting up the network, configuring the firewall, and setting up storage for data. After doing all the set-up, we become responsible for maintaining it through its entire life cycle.



Advantages of cloud computing

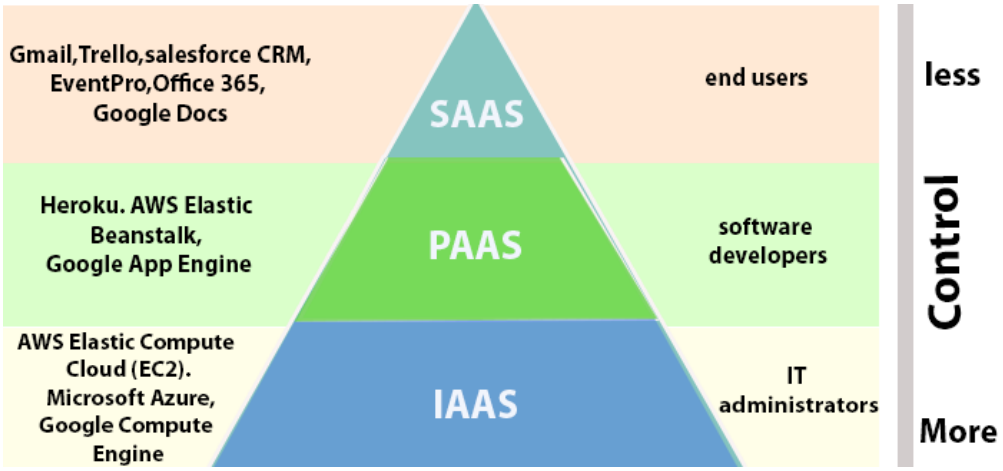
- **Cost:** It reduces the huge capital costs of buying hardware and software.
- **Speed:** Resources can be accessed in minutes, typically within a few clicks.
- **Scalability:** We can increase or decrease the requirement of resources according to the business requirements.
- **Productivity:** While using cloud computing, we put less operational effort. We do not need to apply patching, as well as no need to maintain hardware and software. So, in this way, the IT team can be more productive and focus on achieving business goals.
- **Reliability:** Backup and recovery of data are less expensive and very fast for business continuity.
- **Security:** Many cloud vendors offer a broad set of policies, technologies, and controls that strengthen our data security.

Types of Cloud Computing



- **Public Cloud:** The cloud resources that are owned and operated by a third-party cloud service provider are termed as public clouds. It delivers computing resources such as servers, software, and storage over the internet
- **Private Cloud:** The cloud computing resources that are exclusively used inside a single business or organization are termed as a private cloud. A private cloud may physically be located on the company's on-site datacentre or hosted by a third-party service provider.
- **Hybrid Cloud:** It is the combination of public and private clouds, which is bounded together by technology that allows data applications to be shared between them. Hybrid cloud provides flexibility and more deployment options to the business.

Types of Cloud Services



1. **Infrastructure as a Service (IaaS):** In IaaS, we can rent IT infrastructures like servers and virtual machines (VMs), storage, networks, operating systems from a cloud service vendor. We can create VM running Windows or Linux and install anything we want on it. Using IaaS, we don't need to care about the hardware or virtualization software, but other than that, we do have to manage everything else. Using IaaS, we get maximum flexibility, but still, we need to put more effort into maintenance.
2. **Platform as a Service (PaaS):** This service provides an on-demand environment for developing, testing, delivering, and managing software applications. The developer is responsible for the application, and the PaaS vendor provides the ability to deploy and run it. Using PaaS, the flexibility gets reduce, but the management of the environment is taken care of by the cloud vendors.
3. **Software as a Service (SaaS):** It provides a centrally hosted and managed software services to the end-users. It delivers software over the internet, on-demand, and typically on a subscription basis. E.g., Microsoft One Drive, Dropbox, WordPress, Office 365, and Amazon Kindle. SaaS is used to minimize the operational cost to the maximum extent.

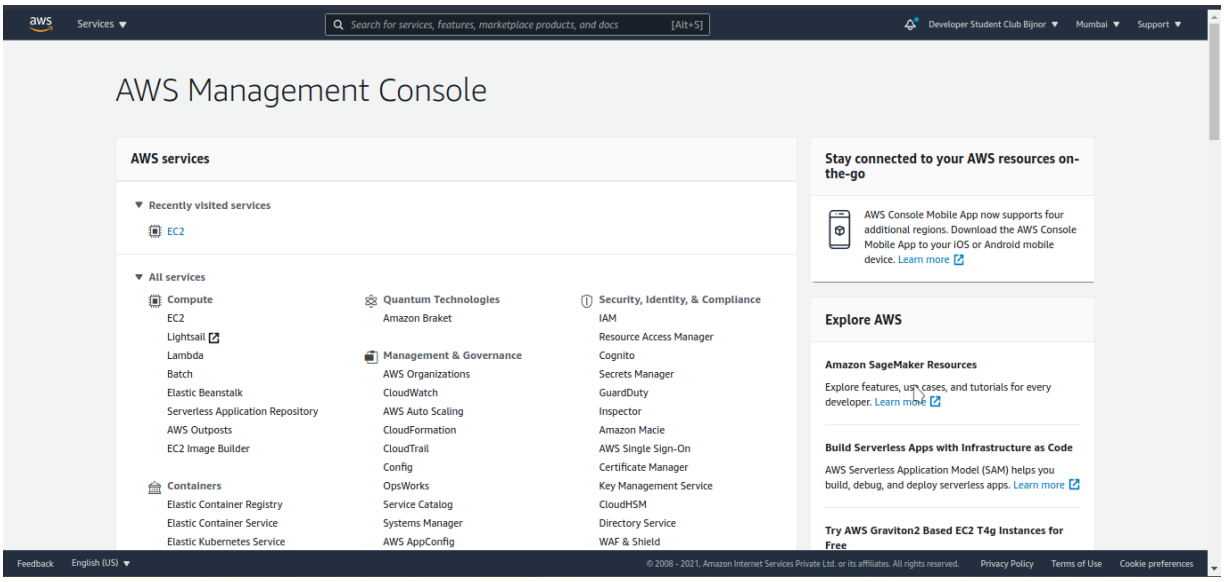
Global Cloud Providers

1. Amazon Web Services
2. Google Cloud Platform
3. Microsoft Azure
4. Alibaba Cloud
5. Oracle Cloud

Amazon Compute Services

Amazon Management Console

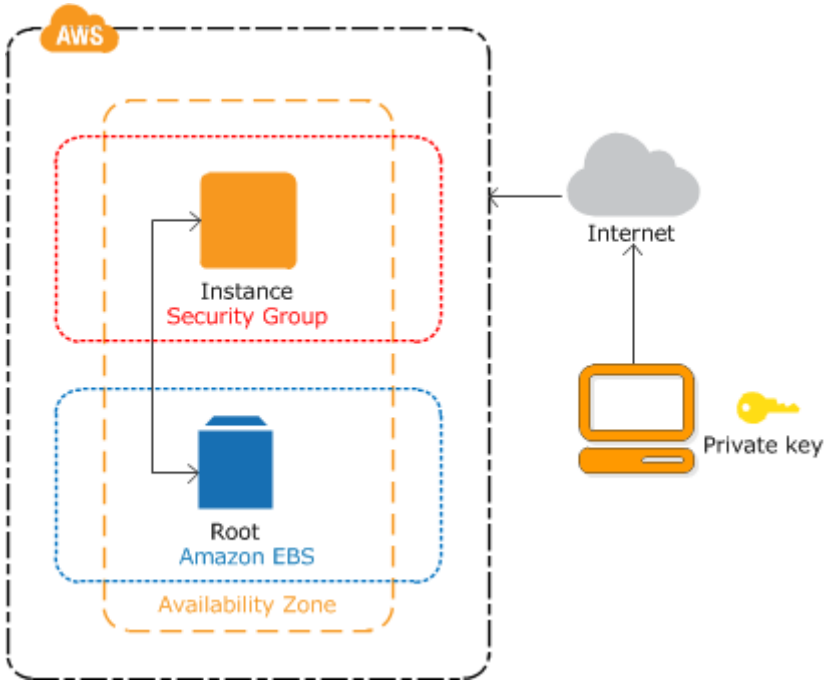
The AWS Management Console brings the unmatched breadth and depth of AWS right to your computer or mobile phone with a secure, easy-to-access, web-based portal. Discover new services, manage your entire account, build new applications, and learn how to do even more with AWS.



Amazon Elastic Compute Cloud (EC2)

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, re-sizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers. Amazon EC2's simple web service interface allows you to obtain and configure capacity with minimal friction. It provides you with complete control of your computing resources and lets you run on Amazon's proven computing environment.

Amazon EC2 offers the broadest and deepest compute platform with choice of processor, storage, networking, operating system, and purchase model. We offer the fastest processors in the cloud and we are the only cloud with 400 Gbps Ethernet networking. We have the most powerful GPU instances for machine learning training and graphics workloads, as well as the lowest cost-per-inference instances in the cloud. More SAP, HPC, Machine Learning, and Windows workloads run on AWS than any other cloud.



AWS Elastic Beanstalk

AWS Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, NodeJS, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS.
You can simply upload your code and Elastic Beanstalk automatically handles the deployment, from capacity provisioning, load balancing, auto-scaling to application health monitoring. At the same time, you retain full control over the AWS resources powering your application and can access the underlying resources at any time.

There is no additional charge for Elastic Beanstalk - you pay only for the AWS resources needed to store and run your applications.

Benefits of Elastic Beanstalk

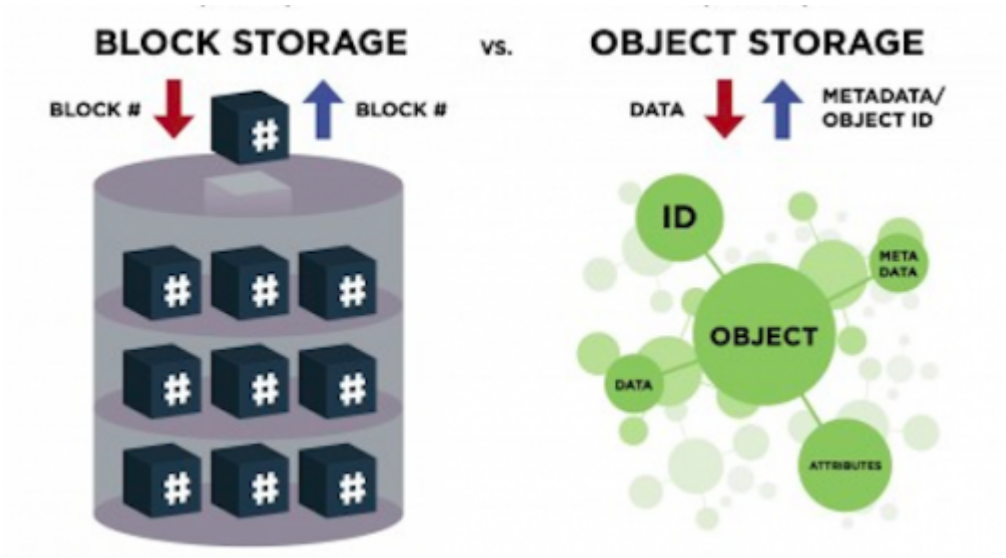
1. Fast and simple to begin
2. Developer productivity
3. Impossible to outgrow
4. Complete resource control

Storage and Backup Services

Block Storage Vs Object Storage

With block storage, files are split into evenly sized blocks of data, each with its own address but with no additional information (metadata) to provide more context for what that block of data is. You're likely to encounter block storage in the majority of enterprise workloads; it has a wide variety of uses (as seen by the rise in popularity of SAN arrays).

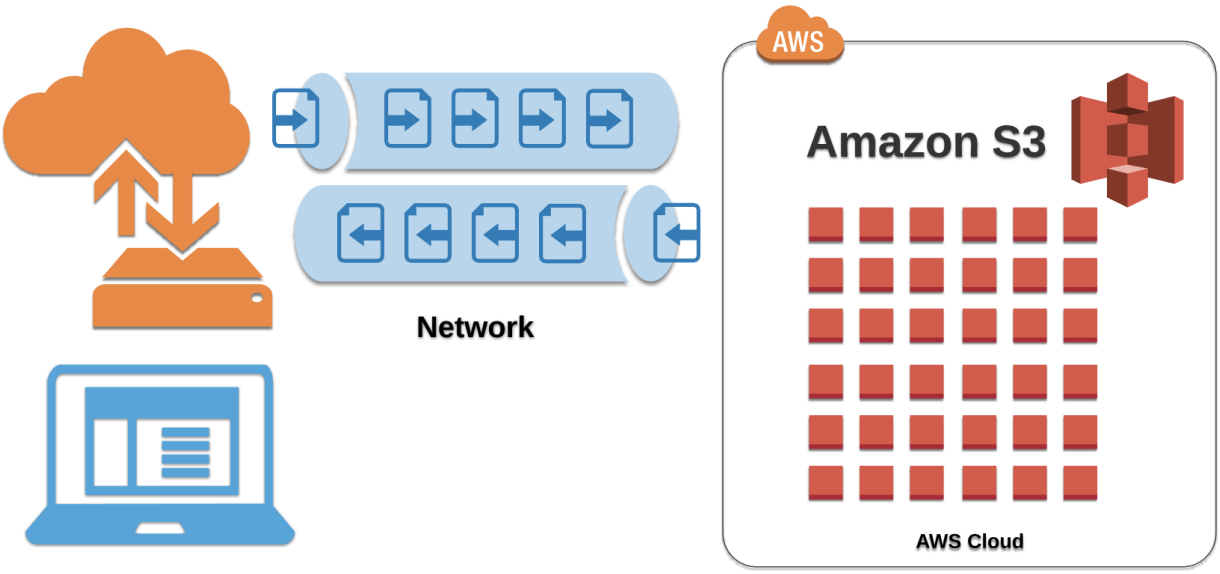
Object storage, by contrast, doesn't split files up into raw blocks of data. Instead, entire clumps of data are stored in, yes, an object that contains the data, metadata, and the unique identifier. There is no limit on the type or amount of metadata, which makes object storage powerful and customization. Metadata can include anything from the security classification of the file within the object to the importance of the application associated with the information. Anyone who's stored a picture on Facebook or a song on Spotify has used object storage even if they don't know it. In the enterprise data center, object storage is used for these same types of storage needs, where the data needs to be highly available and highly durable.



Amazon S3

Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance. This means customers of all sizes and industries can use it to store and protect any amount of data for a range of use cases, such as websites, mobile applications, backup and restore, archive, enterprise applications, IoT devices, and big data analytics. Amazon S3 provides easy-to-use management features so you can organize your data and configure finely-tuned access controls to meet your specific business, organizational, and compliance requirements.

Amazon S3 is designed for 99.999999999% (11 9's) of durability, and stores data for millions of applications for companies all around the world.



Amazon Elastic Block Store

Amazon Elastic Block Store (Amazon EBS) provides persistent block storage volumes for use with Amazon EC2 instances in the AWS Cloud. Each Amazon EBS volume is automatically replicated within its Availability Zone to protect you from component failure, offering high availability and durability. Amazon EBS volumes offer the consistent and low-latency performance needed to run your workloads. With Amazon EBS, you can scale your usage up or down within minutes—all while paying a low price for only what you provision.

Amazon Elastic File System

Amazon Elastic File System (Amazon EFS) provides a simple, scalable, elastic file system for Linux-based workloads for use with AWS Cloud services and on-premises resources. It is built to scale on demand to petabytes without disrupting applications, growing and shrinking automatically as you add and remove files, so your applications have the storage they need – when they need it. It is designed to provide massively parallel shared access to thousands of Amazon EC2 instances, enabling your applications to achieve high levels of aggregate throughput and IOPS with consistent low latencies.

Amazon S3 Glacier

Amazon S3 Glacier is a secure, durable, and extremely low-cost storage service for data archiving and long-term backup. It is designed to deliver 99.999999999% durability, and provides comprehensive security and compliance capabilities that can help meet even the most stringent regulatory requirements. Amazon S3 Glacier provides query-in-place functionality, allowing you to run powerful analytics directly on your archive data at rest.

AWS Storage Gateway

The AWS Storage Gateway is a hybrid storage service that enables your on-premises applications to seamlessly use AWS cloud storage. You can use the service for backup and archiving, disaster recovery, cloud data processing, storage tiering, and migration. Your applications connect to the service through a virtual machine or hardware gateway appliance using standard storage protocols, such as NFS, SMB and iSCSI.

Networking and Content Delivery

Virtual Private Cloud

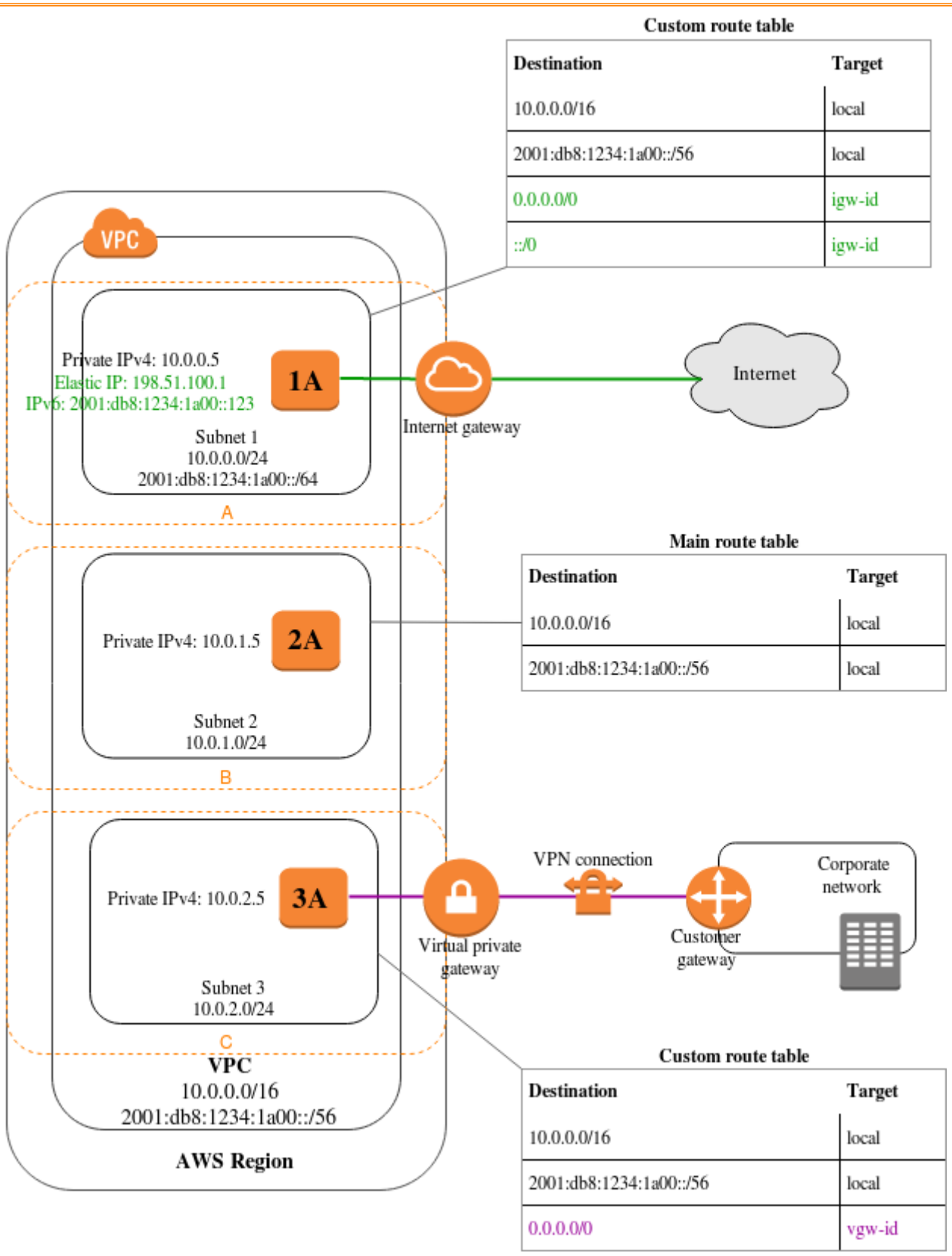
Amazon Virtual Private Cloud (Amazon VPC) is a service that lets you launch AWS resources in a logically isolated virtual network that you define. You have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways. You can use both IPv4 and IPv6 for most resources in your virtual private cloud, helping to ensure secure and easy access to resources and applications.

Subnets

When you create a VPC, you must specify a range of IPv4 addresses for the VPC in the form of a Classless Inter-Domain Routing (CIDR) block; for example, `10.0.0.0/16`. This is the primary CIDR block for your VPC.

Route 53

Amazon Route 53 is a highly available and scalable cloud Domain Name System (DNS) web service. It is designed to give developers and businesses an extremely reliable and cost effective way to route end users to Internet applications by translating names like www.example.com into the numeric IP addresses like 192.0.2.1 that computers use to connect to each other. Amazon Route 53 is fully compliant with IPv6 as well.



Cloud Front

Amazon CloudFront is a fast content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to customers globally with low latency, high transfer speeds, all within a developer-friendly environment.

CloudFront offers the most advanced security capabilities, including field level encryption and HTTPS support, seamlessly integrated with AWS Shield, AWS Web Application Firewall and Route 53 to protect against multiple types of attacks including network and application layer DDoS attacks. These services co-reside at edge networking locations – globally scaled and connected via the AWS network backbone – providing a more secure, performant, and available experience for your users.

Benefits of CloudFront

1. Global Scaled Network for Fast Content Delivery
2. Security at the Edge
3. Highly Programmable and Secure Edge Computing
4. Deep Integration with AWS
5. Cost-Effective

Identity & Access Management (IAM)

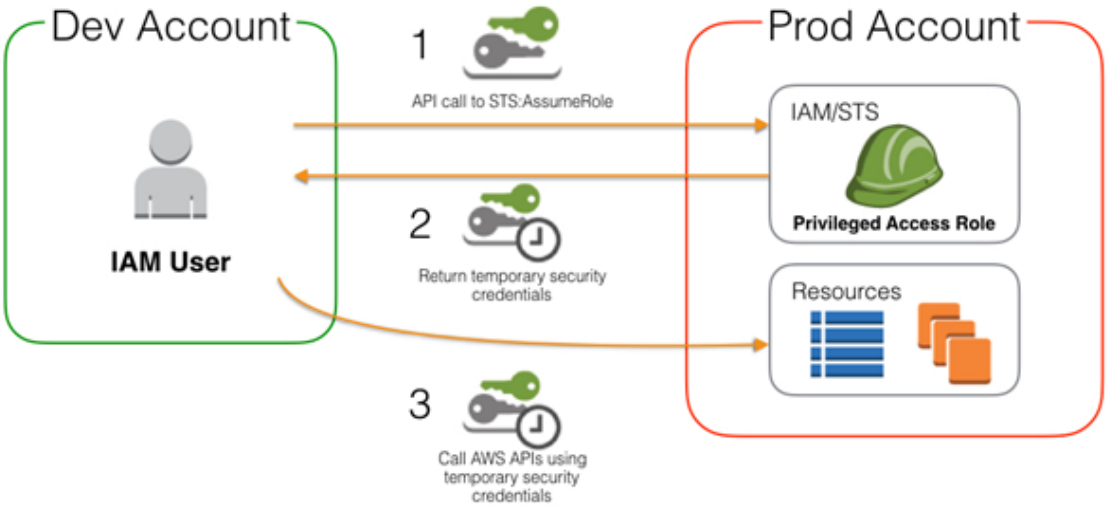
IAM Features

AWS Identity and Access Management (IAM) enables you to manage access to AWS services and resources securely. Using IAM, you can create and manage AWS users and groups, and use permissions to allow and deny their access to AWS resources.

IAM is a feature of your AWS account offered at no additional charge. You will be charged only for use of other AWS services by your users.

Use cases:

1. Fine-grained access control to AWS resources
2. Multi-factor authentication for highly privileged users
3. Analyze access
4. Integrate with your corporate directory



Multi Factor Authentication (MFA)

AWS Multi-Factor Authentication (MFA) is a simple best practice that adds an extra layer of protection on top of your user name and password. With MFA enabled, when a user signs in to an AWS Management Console, they will be prompted for their user name and password (the first factor—what they know), as well as for an authentication code from

their AWS MFA device (the second factor—what they have). Taken together, these multiple factors provide increased security for your AWS account settings and resources.

Federation Access

Identity federation is a system of trust between two parties for the purpose of authenticating users and conveying information needed to authorize their access to resources. In this system, an identity provider (IdP) is responsible for user authentication, and a service provider (SP), such as a service or an application, controls access to resources. By administrative agreement and configuration, the SP trusts the IdP to authenticate users and relies on the information provided by the IdP about them. After authenticating a user, the IdP sends the SP a message, called an assertion, containing the user's sign-in name and other attributes that the SP needs to establish a session with the user and to determine the scope of resource access that the SP should grant. Federation is a common approach to building access control systems which manage users centrally within a central IdP and govern their access to multiple applications and services acting as SPs.

Roles and Policy

You manage access in AWS by creating policies and attaching them to IAM identities (users, groups of users, or roles) or AWS resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. AWS evaluates these policies when an IAM principal (user or role) makes a request. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents. AWS supports six types of policies: identity-based policies, resource-based policies, permissions boundaries, Organizations SCPs, ACLs, and session policies.

High Availability Services

AWS Load Balancer

Elastic Load Balancing automatically distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, IP addresses, Lambda functions, and virtual appliances. It can handle the varying load of your application traffic in a single Availability Zone or across multiple Availability Zones. Elastic Load Balancing offers four types of load balancers that all feature the high availability, automatic scaling, and robust security necessary to make your applications fault tolerant.

Types of Load Balancer

- Application Load Balancer
- Network Load Balancer
- Gateway Load Balancer
- Classic Load Balancer

Amazon Cloud Watch

Amazon CloudWatch is a monitoring and observability service built for DevOps engineers, developers, site reliability engineers (SREs), and IT managers. CloudWatch provides you with data and actionable insights to monitor your applications, respond to system-wide performance changes, optimize resource utilization, and get a unified view of operational health. CloudWatch collects monitoring and operational data in the form of

logs, metrics, and events, providing you with a unified view of AWS resources, applications, and services that run on AWS and on-premises servers. You can use CloudWatch to detect anomalous behavior in your environments, set alarms, visualize logs and metrics side by side, take automated actions, troubleshoot issues, and discover insights to keep your applications
running smoothly.

Auto Scaling

AWS Auto Scaling monitors your applications and automatically adjusts capacity to maintain steady, predictable performance at the lowest possible cost. Using AWS Auto Scaling, you can setup scaling for multiple resources across multiple services in minutes. AWS Auto Scaling provides a simple, powerful user interface that lets you build scaling plans for Amazon EC2 instances and Spot Fleets, Amazon ECS tasks, Amazon DynamoDB tables, and Amazon Aurora Replicas.

AWS Auto Scaling makes scaling simple with recommendations that allow you to optimize performance, costs, or balance between them. If you're already using Amazon EC2 Auto Scaling, you can now combine it with AWS Auto Scaling to scale additional resources for other AWS services. With AWS Auto Scaling, your applications always have the right resources at the right time.

Database & Services Computing

With AWS databases, you don't need to worry about database management tasks such as server provisioning, patching, configuration, or backups. AWS continuously monitors your clusters to keep your workloads running with self-healing storage and automated scaling, so that you can focus on application development.

Amazon RDS

Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while automating time-consuming administration tasks such as hardware provisioning, database setup, patching and backups. It frees you to focus on your applications so you can give them the fast performance, high availability, security and compatibility they need.

Benefits:

1. Easy to administer
2. Highly scalable
3. Available and durable
4. Fast
5. Secure
6. Inexpensive

Amazon Aurora

Amazon Aurora is a MySQL and PostgreSQL-compatible relational database built for the cloud, that combines the performance and availability of traditional enterprise databases with the simplicity and cost-effectiveness of open source databases.

Amazon Aurora is up to five times faster than standard MySQL databases and three times faster than standard PostgreSQL databases. It provides the security, availability, and reliability of commercial databases at 1/10th the cost. Amazon Aurora is fully managed by Amazon Relational Database Service (RDS), which automates time-consuming administration tasks like hardware provisioning, database setup, patching, and backups.

Use cases

1. Enterprise Applications
2. Software as a Service (SaaS) Applications
3. Web and Mobile Gaming

Amazon DynamoDB

Amazon DynamoDB is a key-value and document database that delivers single-digit millisecond performance at any scale. It's a fully managed, multi-region, multi-active, durable database with built-in security, backup and restore, and in-memory caching for internet-scale applications. DynamoDB can handle more than 10 trillion requests per day and can support peaks of more than 20 million requests per second.

Many of the world's fastest growing businesses such as Lyft, Airbnb, and Redfin as well as enterprises such as Samsung, Toyota, and Capital One depend on the scale and performance of DynamoDB to support their mission-critical workloads.

Hundreds of thousands of AWS customers have chosen DynamoDB as their key-value and document database for mobile, web, gaming, ad tech, IoT, and other applications that need low-latency data access at any scale. Create a new table for your application and let DynamoDB handle the rest.

Applications

1. Serverless Web Apps
2. Mobile Backends
3. Microservices
4. Ad Tech
5. Gaming

PROJECT

Set up a static website in S3, use CloudFront distribution, show page load times, and set up restrictions.

1. Bucket Name Matters

Before you begin hosting your website out of s3, you need to create a bucket first. Your bucket name should be same as your domain name. For example if you want to host your website at www.example.com then just create your bucket with the same name.

2. Configure your bucket

After creating the bucket just upload your static website to the bucket and then

- Click the “Properties” section.
- Click the “Static website hosting” option.
- Select “Use this bucket to host a website”.
- Enter “index.html” as the Index document.

Now your bucket serves your static files. But you want to host your website that means it means it must be accessible to everyone in the world. So, we have give public access to our bucket

1. Click into your bucket.
2. Select the “Permissions” tab at the top.
3. Under “Public Access Settings” we want to click “Edit”.
4. Change “Block new public bucket policies”, “Block public and cross-account access if bucket has public policies”, and “Block new public ACLs and uploading public objects” to be false and Save.
5. Now you must update the Bucket Policy of your bucket to have public read access to anyone in the world.

- Click into your bucket.
- Click the “Permissions” section.
- Select “Bucket Policy”.
- Add the following Bucket Policy and then Save

```{
"Version": "2008-10-17",
"Id": "PolicyForPublicWebsiteContent",
"Statement": [
    {
        "Sid": "PublicReadGetObject",
        "Effect": "Allow",
        "Principal": {
            "AWS": "*"
```

```
            },
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::www.example.com/*"
        }
    ]
}
```

Now your website will be live at default AWS Bucket URL you can find it on the properties tab of your bucket. It will be like - `www.example.com.s3-website-us-east-1.amazonaws.com`

3. Adding CNAME record for your Bucket URL

In order for a user to load you website at your domain (www.example.com) you must provide mapping from your domain name to your S3 Bucket Url.
Goto your DNS domain record and add a CNAME record to it:

- Create a record for a host like `www`
- The record type must be `CNAME (Canonical name)`
- The value (or target) must be your S3 website url `` `www.example.s3-website-us-east-1.amazonaws.com` ``

4. Validate that it worked

Now your website is uploaded to your s3 bucket you can goto `www.example.com` you can find your website there. Now your website is loading from your s3 bucket.

IMPORTANT:

one thing to note your home page must be named as `index.html`
Check out my Static website I have launched recently:

Web address : http://kumarshanu.dscrecbijnor.com/

css/main.css

```
body{
	background-color: lightgray;
}

header {
	overflow: hidden;
	position: fixed;
	top: 0;
	display: table !important;
	table-layout: fixed;
	padding-left: 16px ;
	padding-right:16px ;
	transition: opacity .5s,padding-left .125s;
```

```
	width: 100%;
	min-width: 100%;
}

#logo{
	display: table-cell;
	max-width: 380px;
	vertical-align: middle;
}

#logo a{
	text-decoration: none;
	cursor: pointer;
}

#logo span{
	display: inline-block;
	max-width: 380px;
	overflow: hidden;
	padding: 16px 12px;
	color: rgba(0,0,0,0.8199999928);
	font-size: 15pt;
	font-weight: 300;
	line-height: 1.333;
}

#nav-wrapper{
	min-width: 100%;
	display: table-cell;
	vertical-align: middle;
}

#nav-bar{
	text-align: right;
}

#nav-bar ul{
	font-size: 16px;
	list-style-type: none;

}

#nav-bar ul li{
	display: inline-block;
```

```
        overflow: visible;
        position: relative;
        text-align: left;
        transition: opacity 0ms 300ms;
        vertical-align: middle;
        padding: 10px 12px;
}

#nav-bar ul li a{
        color: rgba(0,0,0,0.8199999928);
        text-decoration: none;
        font-weight: 400;
        cursor: pointer;
}

#search-wrapper{
        display: table-cell;
        position: relative;
}

#profile-wrapper{
        text-align: center;
}

#profile-pic{
        width:150px;
        height: 150px;
        border-radius: 50%;
        margin-left: auto;
        margin-right: auto;
}

ul{
        padding: 0;
    list-style-type: none;
}


.social-links ul, li{
        list-style-type:none;
        display: inline-block;
}

.social{
```

```
        width:20px;
        height: 20px;
}
```

index.html

```
<!DOCTYPE html>
<html lang="en">

  <head>
    <link href="css/main.css" rel="stylesheet" type="text/css">
    <link
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
rel="stylesheet">
    <link href="https://unpkg.com/aos@2.3.1/dist/aos.css" rel="stylesheet">
    <link rel="icon" href="./images/Kumar.jpg" type="image/gif" sizes="16x16">
    <script src="https://unpkg.com/aos@2.3.1/dist/aos.js"></script>
    <meta charset="UTF-8">
    <meta name="description"
      content="Hi I'm Kumar Shanu!
      I love coding, doing research and collaborating. I can learn just about anything
by having a final product to tear apart piece by piece. I break the code where I need
to figure out why things work the way they do. Documentation is just a reference.">
    <meta name="keywords" content="its-kumar, its-kumar, kumar, kumar shanu,
dscrecbijnor, developer">
    <meta name="google-site-verification"
content="K0pRwnGeRhgXPiicHezllbb2LBbpQlUgLy_P8GpKzZM" />
    <meta name=" author" content="Kumar Shanu">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Kumar Shanu | Developer</title>
  </head>

  <body>
    <script>
      AOS.init();
    </script>
    <header>
      <div id="logo"><a href="/"><img /><span>Kumar Shanu</span></a></div>
      <div id="nav-wrapper" aria-hidden="false">
        <nav id="nav-bar" style="visibility: visible;">
          <ul>
            <li><a href="/">Home</a></li>
            <li><a href="about.html">About</a></li>
```

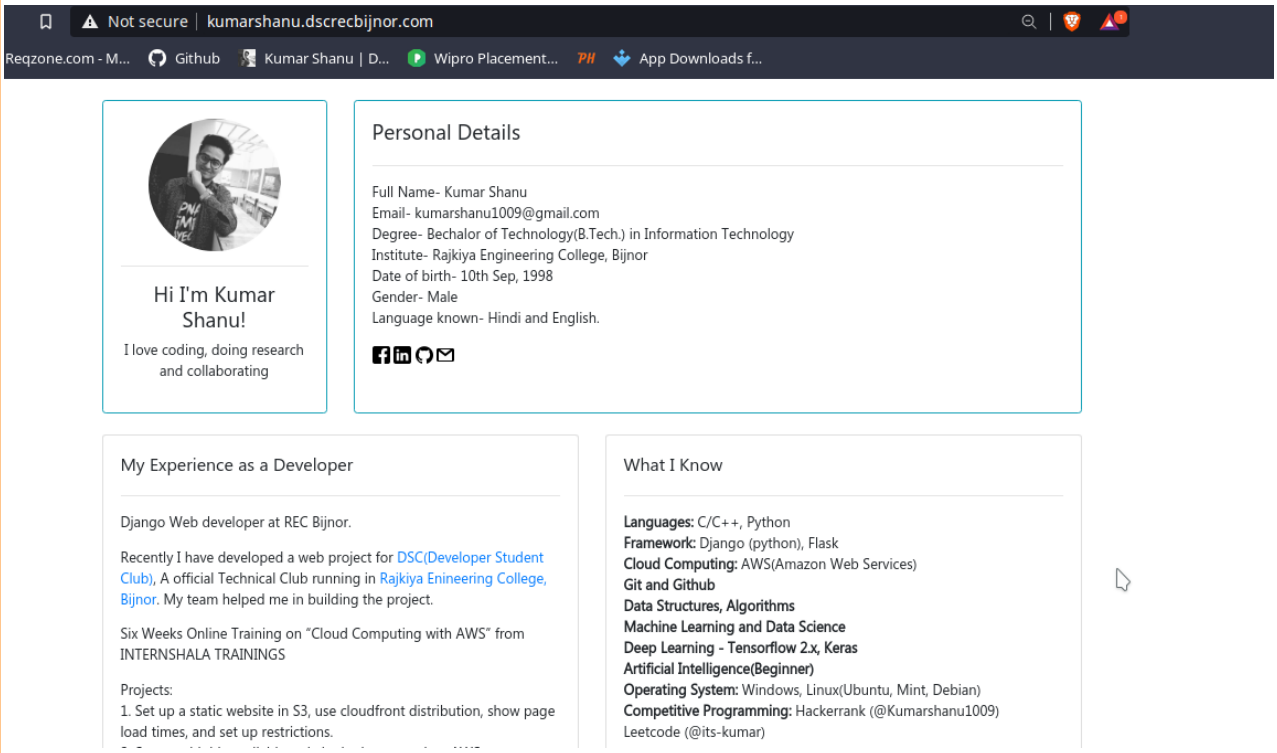
```
          <li><a href="projects.html">Project page</a></li>
        </ul>
      </nav>
    </div>
  </header>
  <br>
  <div class="container" style="padding-top: 36px;">
    <div class="row">
      <div data-aos="fade-up-right" class="col-md-3">
        <div class="card card-body h-100  border-info" id="profile-wrapper">
          <img id="profile-pic" src="images/Kumar.jpg">
          <hr>
          <h4>Hi I'm Kumar Shanu!</h4>
          <p>I love coding, doing research and collaborating</p>
        </div>
      </div>

      <div data-aos="fade-up-left" class="col-md-9">
        <div class="card card-body h-100  border-info">
          <h4>Personal Details</h4>
          <hr>
          <p>Full Name- Kumar Shanu<br>
            Email- kumarshanu1009@gmail.com<br>
            Degree- Bechalor of Technology(B.Tech.) in Information Technology<br>
            Institute- Rajkiya Engineering College, Bijnor<br>
            Date of birth- 10th Sep, 1998<br>
            Gender- Male<br>
            Language known- Hindi and English.

          </p>
          <ul class="social-links">
            <li><a href="https://www.facebook.com/itsYoursKumar/">
                <img class="social" src="images/facebook.png"></a>
            </li>
            <li><a href="https://www.linkedin.com/in/kumar-shanu-8313951ab/">
                <img class="social" src="images/linkedin.png"></a>
            </li>
            <li><a href="http://www.github.com/its-Kumar/">
                <img class="social" src="images/github.png"></a>
            </li>
            <li><a
                href="https://mail.google.com/mail/u/0/#inbox?
compose=CllgCJZbjLjqJVjgzmFDkjSbmFhLWDRfmvXfRfkvCtrjfzHSwGNrDnpX
vmLqckFvrrHksvwPzPL">
```

```
              <img class="social" src="images/gmail.png"></a>
            </li>
          </ul>
        </div>
      </div>
    </div>
    <br>
    <div class="row">
      <div data-aos="fade-right" class="col-md-6">
        <div class="card card-body h-100">
          <h5>My Experience as a Developer</h5>
          <hr>
          <p>Django Web developer at REC Bijnor.</p>
          <p>Recently I have developed a web project for <a
href="https://dscrecbijnor.com/">
              DSC(Developer Student Club)</a>, A
            official Technical
            Club running in <a href="http://recb.ac.in/">Rajkiya Enineering College,
Bijnor</a>. My team
            helped me in building
            the project.
          </p>
          <p>Six Weeks Online Training on “Cloud Computing with AWS” from
INTERNSHALA TRAININGS</p>
          <p>Projects:<br>
            1. Set up a static website in S3, use cloudfront distribution, show page load
times, and set
            up
            restrictions.<br>
            2. Set up a highly available website in the network at AWS<br>
            3. Set up a highly available relational database service (RDS) at AWS
          </p>
        </div>
      </div>

      <div data-aos="fade-left" class="col-md-6">
        <div class="card card-body h-100">
          <h5>What I Know</h5>
          <hr>
          <ul>
            <li> <b>Languages:</b> C/C++, Python</li><br>
            <li> <b>Framework:</b> Django (python), Flask</li><br>
            <li> <b>Cloud Computing:</b> AWS(Amazon Web Services)</li><br>
            <li> <b>Git and Github</li><br>
```

```
            <li> Data Structures, Algorithms</li><br>
            <li> Machine Learning and Data Science</li><br>
            <li> Deep Learning - Tensorflow 2.x, Keras</li><br>
            <li> Artificial Intelligence(Beginner)</li><br>
            <li>Operating System:</b> Windows, Linux(Ubuntu, Mint, Debian)</li>
            <li> <b>Competitive Programming:</b> Hackerrank
(@Kumarshanu1009)</li>
            <li> Leetcode (@its-kumar)</li>
          </ul>
        </div>
      </div>
    </div>
    <br>
    <div class="row">
      <div data-aos="zoom-in-right" class="col-md-6">
        <div class="card card-body h-100">
          <h5>How I Learn</h5>
          <hr>
          <p>Reverse Engineer & Build Ugly Prototypes First.</p>

          <p>I can learn just about anything by having a final product to tear apart
piece by piece. I
            break
            the code where I need to figure out why things work the way they do.
Documentation is just a
            reference.</p>
        </div>
      </div>

      <div data-aos="zoom-in-left" class="col-md-6">
        <div class="card card-body h-100">
          <h5>My Short Term Goals</h5>
          <hr>
          <p>After almost two years of supporting my system at the lab I’m ready to
learn something new.
            I'll
            collaborating and sharing my experience with others on my github
profile.</p>
          <p>Aside from building small projects I would love to enter a position
where I can be part of a
            bigger team & learn from others with more experience than myself.</p>

        </div>
      </div>
```

```
    </div>
    <br>
  </div>
  <hr />
  <footer>
    <div class="footer-copyright text-center py-3">© 2020 Copyright:
      <a href="http://its-kumar.herokuapp.com/" target="_blank">Kumar
        Shanu</a>
    </div>
  </footer>
 </body>

</html>
```

SCREENSHOT


Not secure | kumarshanu.dscrecbijnor.com
Reqzone.com - M...
Github
Kumar Shanu | D...
Wipro Placement...
App Downloads f...
Hi I'm Kumar Shanu!
I love coding, doing research and collaborating
Personal Details
Full Name- Kumar Shanu
Email- kumarshanu1009@gmail.com
Degree- Bechalor of Technology(B.Tech.) in Information Technology
Institute- Rajkiya Engineering College, Bijnor
Date of birth- 10th Sep, 1998
Gender- Male
Language known- Hindi and English.
My Experience as a Developer
Django Web developer at REC Bijnor.
Recently I have developed a web project for DSC(Developer Student Club), A official Technical Club running in Rajkiya Enineering College, Bijnor. My team helped me in building the project.
Six Weeks Online Training on "Cloud Computing with AWS" from INTERNSHALA TRAININGS
Projects:
1. Set up a static website in S3, use cloudfront distribution, show page load times, and set up restrictions.
What I Know
Languages: C/C++, Python
Framework: Django (python), Flask
Cloud Computing: AWS(Amazon Web Services)
Git and Github
Data Structures, Algorithms
Machine Learning and Data Science
Deep Learning - Tensorflow 2.x, Keras
Artificial Intelligence(Beginner)
Operating System: Windows, Linux(Ubuntu, Mint, Debian)
Competitive Programming: Hackerrank (@Kumarshanu1009)
Leetcode (@its-kumar)

CONCLUSION

During my industrial training, there are many changes from the point of learning environments and discussion among colleagues. It can directly increase the dedication and rational attitude toward myself.
However, there are still some weaknesses that can be improved in the future. Therefore I conclude that the industrial training program has provided many benefits to students even if there are minor flaws that are somewhat disfiguring , so that this weakness can be rectified in the future.

I can conclude that this industry is through training I received a lot of exposure in the computing world. I would like to also thank the **“INTERNSHALA”** giving students find their own experience with having Industrial Training like this.

Signature:

Name: **KUMAR SHANU**

Roll No: **1773513025**