**Cloud Technology for Business**

BookShow is our topic for cloud strategy and sample deployment. BookShow is a leading online ticketing platform, and we'll focus on developing a comprehensive cloud strategy for the company.

**Background of the Enterprise:**

BookShow is a leading entertainment company in India, provides an online ticketing platform for movies, plays, concerts, and sporting events. BookShow also offers an online media streaming service and end-to-end management for virtual and on-ground entertainment experiences across all genres. The core operation of BookShow involves providing an online platform for booking tickets for various entertainment events, primarily movies. It is an established business, but the technical methods it uses are outdated. The servers frequently fail due to high traffic volume when there is a sudden spike in people attempting to book tickets, particularly during peak hours or for popular events, resulting in high costs connected with managing them due to high operational and capital expenses.

**Size of the Business:** Medium-sized enterprise with approximately 33 employees.

* 2 - IT Manager
* 2 - Network Administrator
* 4 - Systems Administrator
* 4 - Backend Developers
* 4 - Frontend Developers
* 3 - IT Support Specialist
* 1 - Cloud Engineer
* 2 - Database Administrator
* 1 - Scrum Master
* 3 - Security Specialist
* 1 - IT Project Manager
* 4 - Help Desk Technician
* 2 - Cloud Solutions Architect

**Current IT Setup Overview:**

**1. Hardware:**

* Multiple physical servers are used for different applications i.e. web servers, database servers, file servers.
* Networking equipment like switches, routers and firewalls.
* Storage solutions like NAS (Network Attached Storage) or SAN (Storage Area Network).

**2. Software:**

* Operating systems : Linux
* Application software: Enterprise resource planning (ERP) and Customer Relationship Management (CRM) systems.
* Databases : SQL Server, PostgreSQL.

**3. Network Infrastructure:**

* On-premises data centers with dedicated network connections.
* Backup and disaster recovery systems, located offsite.

**4. Management:**

* IT team responsible for monitoring, maintenance, and troubleshooting.
* Manual or semi-automated monitoring systems.
* Regular manual backups and maintenance tasks.

**IN-DETAIL CURRENT IT SETUP:**

#### **1. Hardware**

**On-Premises Servers:**

* **Web Servers:** Several in-house servers hosting the BookShow website and application. These servers manage user inquiries, booking tickets, and providing web content. Common software includes Apache, Nginx, or IIS.
* **Database Servers:** Dedicated servers for storing and managing transactional data, customer information, and event details using relational database management systems (RDBMS), SQL Server or PostgreSQL.
* **Media Servers:** Servers used for streaming media content and storing large media files.
* **Backup Servers:** A dedicated backup server hardware are used for creating backups of critical data to ensure data integrity and availability in case of failures.
* **Load Balancers:** Hardware load balancers to distribute incoming traffic evenly across web servers to ensure high availability and reliability.

#### **2. Software**

**Operating Systems:**

* Mainly Linux servers (such as Ubuntu, CentOS) are used for web, database, and media servers because of their reliability and safety.

**Applications and Services:**

* **Custom Ticketing Platform:** A specialized software program created to oversee ticket reservations, user profiles, financial transactions, and organizing events.
* **CRM system:** Software designed to handle customer interactions, sales, and support. CMS is used to oversee the content, promotions, and ads on websites.
* **API and Services:** APIs and services are connected with different payment gateways to aid in conducting online transactions.
* **Monitoring Tools:** Nagios and Zabbix are used to monitor the health, performance, and availability of servers.

#### **3. Network Architecture**

* **Internet Connectivity:** High-bandwidth internet connections to handle large volumes of user traffic.
* **Switches and Routers**: For internal and external network connectivity.

**4. Security Components:**

* **Firewalls:** Hardware as well as software firewalls provide protection against unauthorized access and cyber threats.
* **Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS):** They are used to oversee network traffic for any signs of suspicious behavior and to stop possible attacks from occurring.
* **Protection against DDoS attacks:** Actions taken to minimize the impact of Distributed Denial of Service (DDoS) attacks and guarantee service availability.

**5. Power and Cooling:**

* **Uninterruptible Power Supplies (UPS):** Provide backup power to ensure servers stay up during power outages.
* **Cooling Systems:** Maintain optimal operating temperatures for the servers and networking equipment.

### **On-Site Management and Maintenance**

**1. IT Staff:**

* **System Administrators:** Manage servers, storage, and network equipment.
* **Database Administrators (DBAs):** Manage database servers and ensure data integrity and performance.
* **Network Administrators:** Handle networking equipment, connectivity, and security.

**2. Support Staff:** Provide technical support to end users.

**3.Monitoring:**

* **Manual Monitoring:** IT staff regularly check server and network performance.
* **Monitoring Tools (e.g., Nagios, Zabbix):** Provide real-time alerts and dashboards for tracking system health.

**4.Maintenance:**

* **Regular Updates and Patching:** Keep software and firmware up to date to ensure security and performance.
* **Hardware Maintenance:** Regular checks and replacements for aging hardware components.

**Current setup Challenges:**

* **High Capital Expenditures (CapEx):**

1. Purchasing and maintaining physical servers and networking equipment.
2. High upfront costs for hardware and software licenses.
3. Regular upgrades and replacements.

* **Operational Expenses (OpEx)**:

1. Costs associated with power, cooling, and physical space.
2. Maintenance and management of the hardware.
3. IT staff required for managing and maintaining servers.

* **Scalability problems:**

1. The present company is facing difficulty to quickly add resources to handle increases in demand for important events or releases.
2. Over-provisioning to handle peak loads, leading to wasted resources during off-peak times.

* **Disaster Recovery and Redundancy:**

1. Complexity and cost of setting up and maintaining backup and recovery solutions.
2. Ensuring data redundancy and availability.

* **Performance and Monitoring:**

1. Challenges in monitoring server performance and health.
2. Identifying and addressing performance bottlenecks.
3. Manual intervention is often required to address issues

* //to include network problem challenged//

**OUR RECOMMENDATION SOLUTIONS:**

#### **Aspect 1: Web Hosting**

**Comparative Analysis:**

* **Cloud Solution: Amazon Web Services (AWS) EC2**

**Advantages:**

* + **Scalability:** AWS EC2 allows BookShow to scale their web hosting resources up or down based on traffic demands. During peak ticketing periods or events, they can quickly increase server capacity to handle high volumes of traffic, ensuring smooth performance without downtime or slowdowns.
  + **Reliability:** EC2 offers high availability and reliability with its robust infrastructure. Company can distribute their application across multiple availability zones to minimize the risk of service disruption in case of hardware failures or other issues.
  + **Cost Efficiency:** AWS EC2 offers a pay-as-you-go pricing model, allowing company to pay only for the compute capacity they use. This cost-effective approach enables them to manage expenses efficiently while scaling resources as needed.
  + **Security:** AWS offers comprehensive security features and compliance certifications, helping company to secure their infrastructure and customer data effectively. They can implement encryption, access controls, and other security measures to protect sensitive information.
  + **Flexibility:** EC2 provides flexibility in choosing instance types, operating systems, and configurations that best suit BookShow's specific needs. They can select instances optimized for compute, memory, storage, or other requirements, ensuring optimal performance and cost efficiency.
  + **Ease of Management:** Managed services reduce the need for in-house IT maintenance.
* **Disadvantages:**
  + **Performance Variability**: The performance of EC2 instances can vary based on the instance type chosen, shared resources in the underlying physical servers, and network latency. Ensuring consistent performance under varying loads requires careful instance selection and monitoring.
  + **Vendor Lock-In:** Adopting AWS EC2 may lead to vendor lock-in, where reliance on AWS-specific services and features makes migration to another cloud provider challenging. Company should consider strategies for mitigating vendor lock-in risks, such as using multi-cloud or hybrid cloud solutions.
  + **Potential for Downtime**: Although AWS EC2 provides high availability, instances can still experience downtime due to maintenance activities, region failures, or human error. Company needs to implement redundancy and disaster recovery strategies to minimize downtime risks.
* **Traditional Solution: On-Premises Servers**

**Advantages:**

* + **Control:** On-premise servers offer direct control over security measures and data management, ensuring compliance with industry standards and regulations.
  + **Latency**: On-premise servers can reduce latency for critical operations, such as real-time ticket bookings and updates.
  + **Customization**: IT teams can customize hardware and software configurations to meet specific performance and operational requirements of the ticketing platform.
  + **Performance Predictability**: With dedicated hardware, performance can be more predictable and consistent compared to cloud environments where resources are shared.

**Disadvantages:**

* + **Scalability Challenges**: Scaling resources can be challenging and may require additional hardware purchases and lead times for deployment.
  + **High Initial Cost**: On-premise servers require significant upfront investment in hardware, software licenses, and infrastructure setup.
  + **Security Concerns**: Ensuring robust security measures (e.g., physical security, data encryption) can be more complex and costly compared to cloud providers with specialized security teams.
  + **Skill Requirement**: Maintaining expertise and skills in-house to manage complex infrastructure and security needs can be challenging and costly.
  + **Compliance Challenges**: Meeting regulatory compliance requirements (e.g., data protection laws) may require additional investments and ongoing efforts to ensure adherence.
  + **Upgrade Cycles**: Hardware and software upgrades may require significant planning, downtime, and disruption to ongoing operations

**Recommendation:** Amazon Web Services Elastic Compute Cloud

**Justification:** AWS EC2 offers the flexibility and scalability necessary for handling variable traffic loads, especially during peak times. The pay-as-you-go model helps reduce costs during off-peak periods. AWS's robust security features and managed services alleviate the burden on the in-house IT team. It allows the platform to focus more on innovation and customer experience rather than managing infrastructure.

**Aspect 2: Monitoring System**

**Comparative Analysis**

#### **Cloud Solution: AWS CloudWatch**

**Advantages:**

* **Monitoring and Visibility**: AWS CloudWatch provides comprehensive monitoring of resources, applications, and services. This allows Company to gain insights into system performance, resource utilization, and operational health in real-time.
* **Alerting and Notifications**: CloudWatch enables setting up alarms based on predefined metrics thresholds or custom metrics. This ensures proactive monitoring and timely alerts for issues such as server failures or performance degradation, helping to maintain service availability.
* **Integration with AWS Services**: CloudWatch integrates seamlessly with other AWS services, such as EC2, RDS, Lambda, and more. This allows Company to consolidate monitoring and management under one platform, simplifying operations.
* **Custom Metrics and Dashboards**: Company can create custom metrics and dashboards in CloudWatch to monitor specific application and business metrics. This customization helps in tracking key performance indicators (KPIs) relevant to ticketing operations
* **Scalability**: CloudWatch can automatically scale resources based on predefined conditions or metrics, such as scaling EC2 instances based on CPU utilization or scaling DynamoDB tables based on request rates. This ensures optimal performance during peak ticketing periods without manual intervention.
* **Log Management**: CloudWatch Logs allows centralizing logs from various AWS services and applications. This aids in troubleshooting issues, debugging applications, and complying with auditing requirements.
* **Cost Management**: CloudWatch provides insights into resource usage and associated costs, helping Company to optimize resource allocation and control expenses.

**Disadvantages**:

* **Vendor Lock-in**: Dependence on CloudWatch for monitoring and management ties Company to AWS. This can limit flexibility in transitioning to other cloud providers or maintaining hybrid cloud environments.
* **Cost**: While CloudWatch offers a free tier, continuous monitoring of large-scale applications or extensive use of custom metrics can incur additional costs. Organization needs to monitor usage to avoid unexpected charges.
* **Learning Curve**: Efficiently utilizing CloudWatch requires familiarity with AWS services, CloudWatch APIs, and CloudFormation for automated setup. Training and expertise may be needed for effective use.
* **Data Latency**: There can be a delay (usually in seconds) between metric collection and availability in CloudWatch. This latency may impact real-time decision-making for highly time-sensitive applications.
* **Traditional Solution: Open source monitoring system (Nagios,Zabbix)**
  + **Advantages:**
* **Customizability**: These systems offer extensive customization options, allowing BookShow to tailor monitoring to specific business needs and metrics relevant to ticketing operations.
* **Flexibility**: They support monitoring a wide range of IT infrastructure components, applications, and services, providing flexibility in monitoring diverse systems and environments.
* **Alerting and Notifications**: Robust alerting mechanisms with multiple notification methods (e.g., email, SMS) ensure timely awareness of issues
* **Open Source**: Both Nagios and Zabbix are open-source, reducing costs as there are no licensing fees. They can be a cost-effective choice for many organizations.
* **Community Support**: Open-source projects benefit from active communities of developers and users who contribute to troubleshooting, updates, and extensions. BookShow can leverage this community for support and enhancements.
* **Transparency**: The open nature of these systems provides transparency into how monitoring data is collected, processed, and stored, which can be crucial for compliance and security audits.  
  + **Disadvantages:**
* **Complex Configuration**: Setting up and configuring open-source monitoring systems like Nagios or Zabbix can be complex and time-consuming, requiring expertise and resources to manage effectively
* **Learning Curve**: Training and familiarization with the tools may be necessary for BookShow staff to effectively utilize and maximize the capabilities of open-source monitoring systems.
* **Maintenance Overhead**: Ongoing maintenance is required, including updating configurations, managing plugins, and ensuring the health of the monitoring system itself.
* **Scalability and Performance**: Scalability limitations may arise with large-scale deployments or when monitoring extensive infrastructure. Performance issues can occur if not properly optimized.
* **Security**: Ensuring the security of monitoring data and the monitoring infrastructure itself is critical. Open-source tools may have vulnerabilities that need timely patching and monitoring.

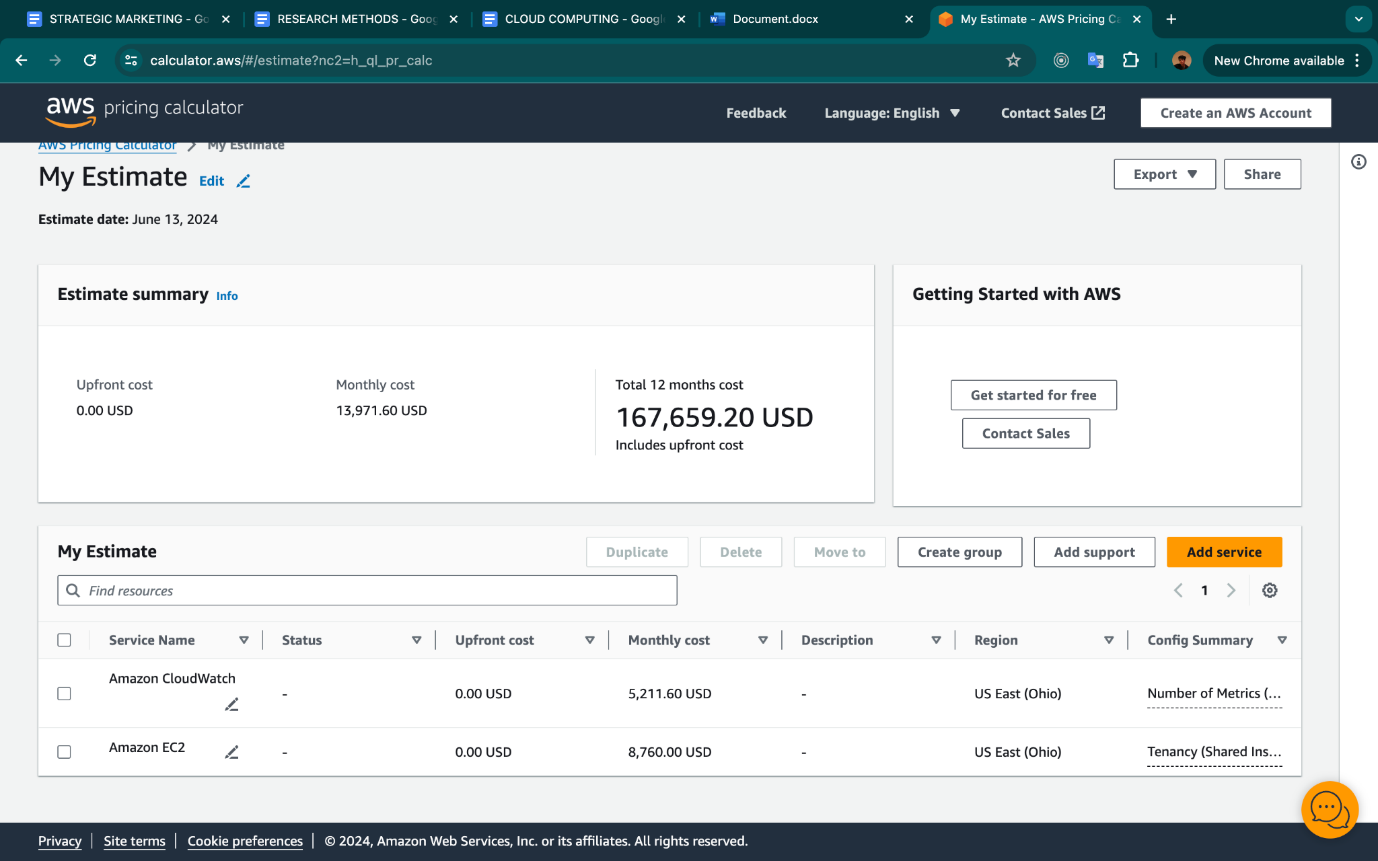
**Recommendations: AWS CloudWatch**

**Justification:** While open-source monitoring systems like Nagios or Zabbix offer extensive customization and flexibility, they require more effort in terms of setup, configuration, ongoing maintenance, and integration with AWS services. For BookShow, which is likely focused on operational efficiency and leveraging AWS's managed services, AWS CloudWatch provides a more **streamlined** and integrated solution. Therefore, considering the **operational efficiency, scalability**, and native **integration** with AWS services, AWS CloudWatch is recommended for monitoring an online ticketing platform like BookShow.

**Detail of the Costs:**

The estimated costs for BookShow's cloud migration are:

* **Current IT setup Monthly Spent**: Approximately **$46,666.67** Per month and **$560,000.00** Per Annum. /// ( calculation and screenshot to be added)//
* **Estimated Monthly Costs on Cloud (AWS EC2 and AWS Cloud Watch):** Approximately **$13971.58** per month and **$167,659.20** Per Annum.

****

* **Options for backup:** AWS provides automated backup options. //need to add cost//
* **Monitoring and Management:** //need to add cost//
* AWSCloudWatch for servers EC2
* AWS provides automated monitoring and management For RDS
* <https://calculator.aws/#/>
* **Sample Cloud Infrastructure Deployment**

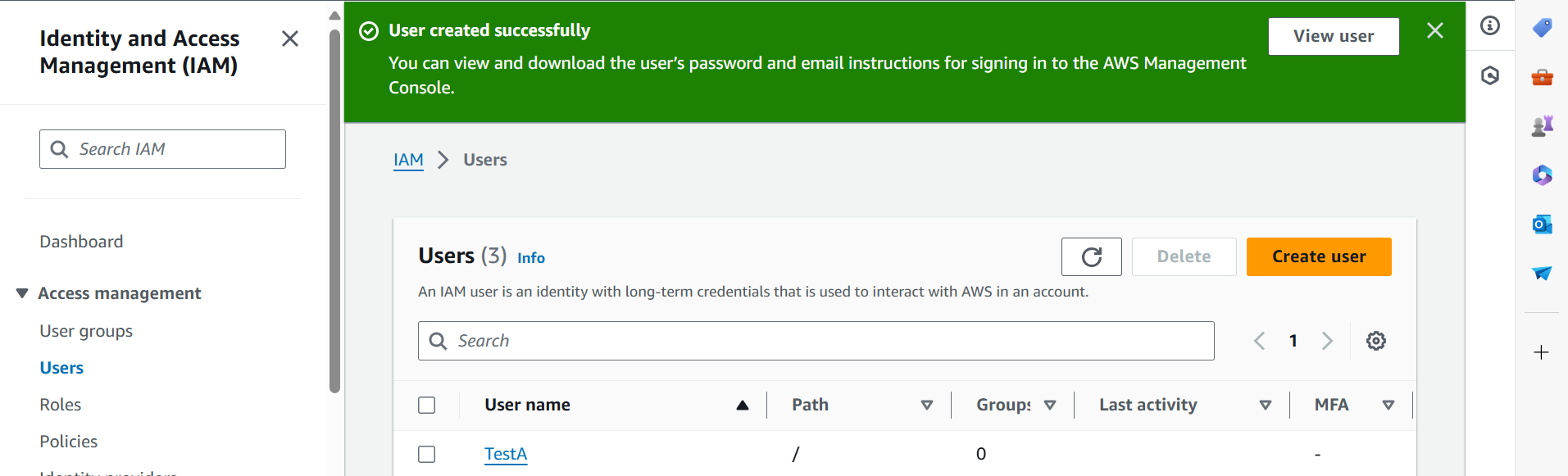
Below is the Sample Cloud Infrastructure to develop and deploy a sample cloud infrastructure that aligns with our recommendations. This will demonstrate the practical application of proposed cloud strategy.

* **Set up and create IAM user with administrative access**
* **Create and Launch new EC2 instances to host the web application**
* **Connect EC2 instances securely using .pem file via SSH**
* **Once connected to EC2 install web servers. Update the instances and install Apache to host the web application.**
* **Create simple HTML file**
* **Set up Cloud watch: Download Install cloud watch agent in EC2 terminal**
* **Configure and start the cloud watch agent**
* **Create an alarm in cloud watch**
* **Create one more EC2 instance (interacting resource)**
* **Connect 2 instances**
* **Download and install docker container in EC2 terminal**
* **Set cloudwatch monitoring docker**
* **Test and validate the EC2 HTML page and docker cloud monitoring**

**Step 1**: Login to the AWS console management (Root user)

**Step 2**: Navigate to the IAM (Identity and Access Management) dashboard.

**Step 3**: Create a new user with administrative permissions. Make sure to Save the access key ID and secret access key.  
 [Ignore \_ Pwd : 84qQ+h0\_,TestA, https://730335605057.signin.aws.amazon.com/console]



**Step 4**: Go to the AWS Management Console and select EC2 from the services list.

**Step 5**: Click on Instances –> “Launch Instances”.

- Choose an Amazon Machine Image (AMI), for example, Amazon Linux 2.

- Choose an instance type (e.g., t2.micro which is free-tier eligible).

- Configure instance details (leave default settings for simplicity).

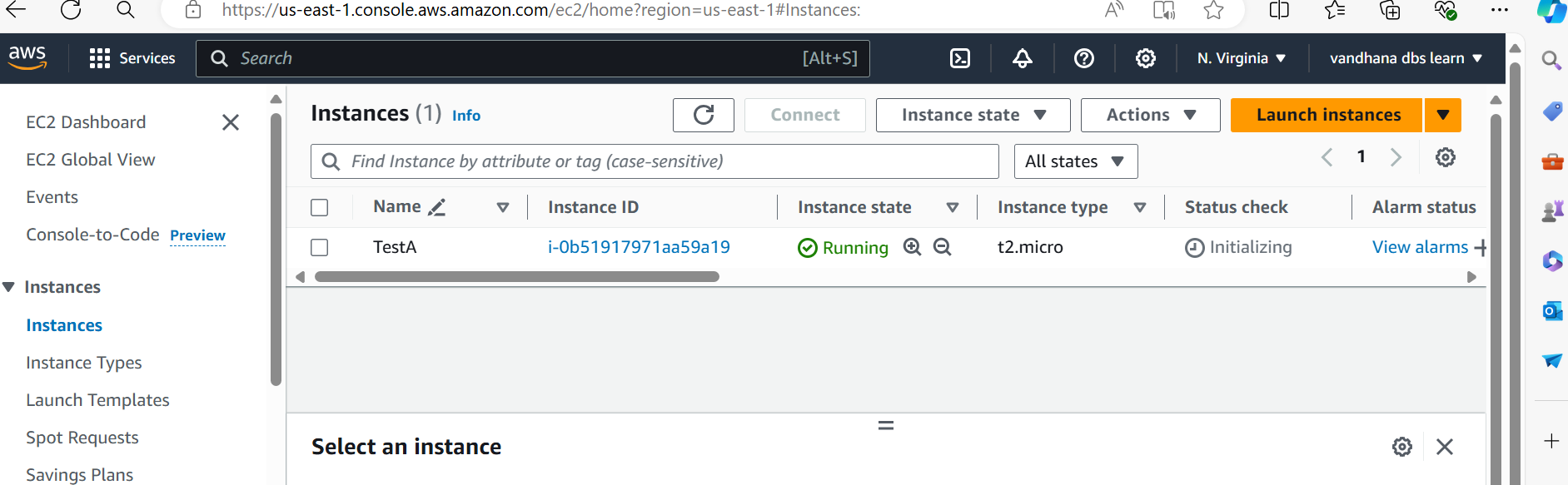
- Add storage (default 8 GB is fine).

- Add a rule to allow SSH access (port 22) from your IP address.

- Add a rule to allow HTTP access (port 80) from anywhere.

- Create a new key pair, download the `.pem` file, and save it securely.

- Review and launch the instance.



**Step 6**: Connect to Your EC2 Instance via SSH:  
 Connecting via SSH, direct access to the operating system, enabling advanced configurations, real-time troubleshooting, automation, and efficient resource management.

- Open your terminal.

- Navigate to the directory where your `.pem` file is saved.

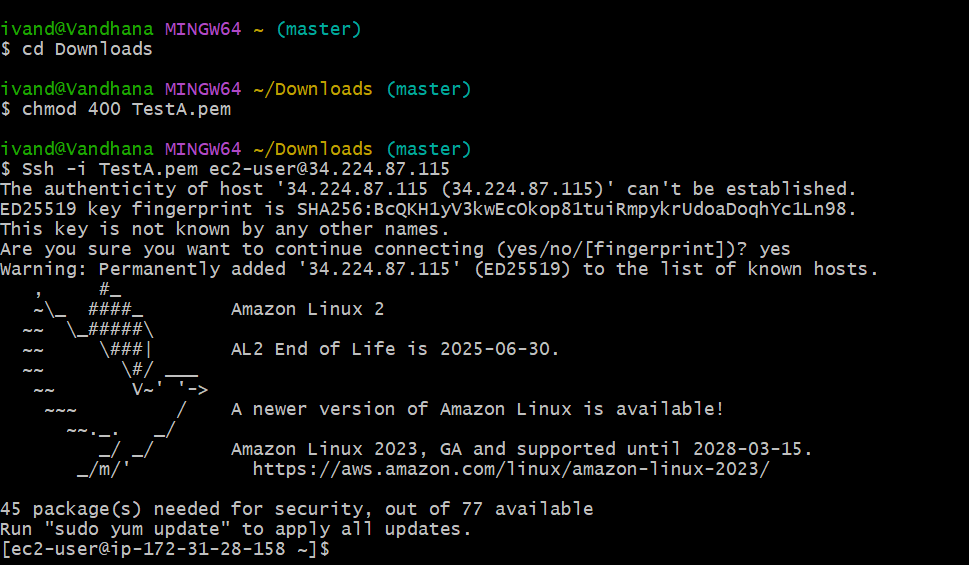
- Run the following command:

```

chmod 400 your-key-pair.pem

ssh -i "your-key-pair.pem" ec2-user@your-ec2-public-dns

```



**Step 7.** Once connected, update the instance and install Apache (to prepare the EC2 instance to handle HTTP/HTTPS requests and serve content to users, which is essential for any web-based application. This allows users to access web pages and web applications hosted on the EC2 instance.)

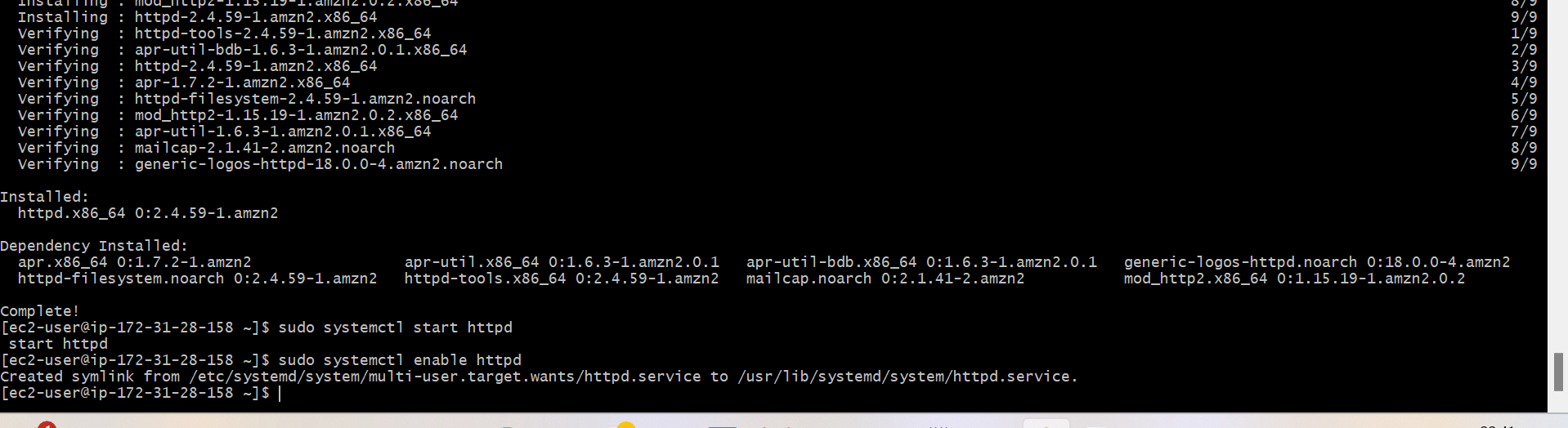
```

sudo yum update -y

sudo yum install -y httpd

sudo systemctl start httpd

sudo systemctl enable httpd

```  


**Step 8:** Create a simple HTML file to serve:

```bash

echo "<html><body><h1>Hello from EC2</h1></body></html>" | sudo tee /var/www/html/index.html  
“The HTML file provides a basic web page that can be viewed in a user browser, ensuring that the Apache web server is capable of delivering web content to users.”

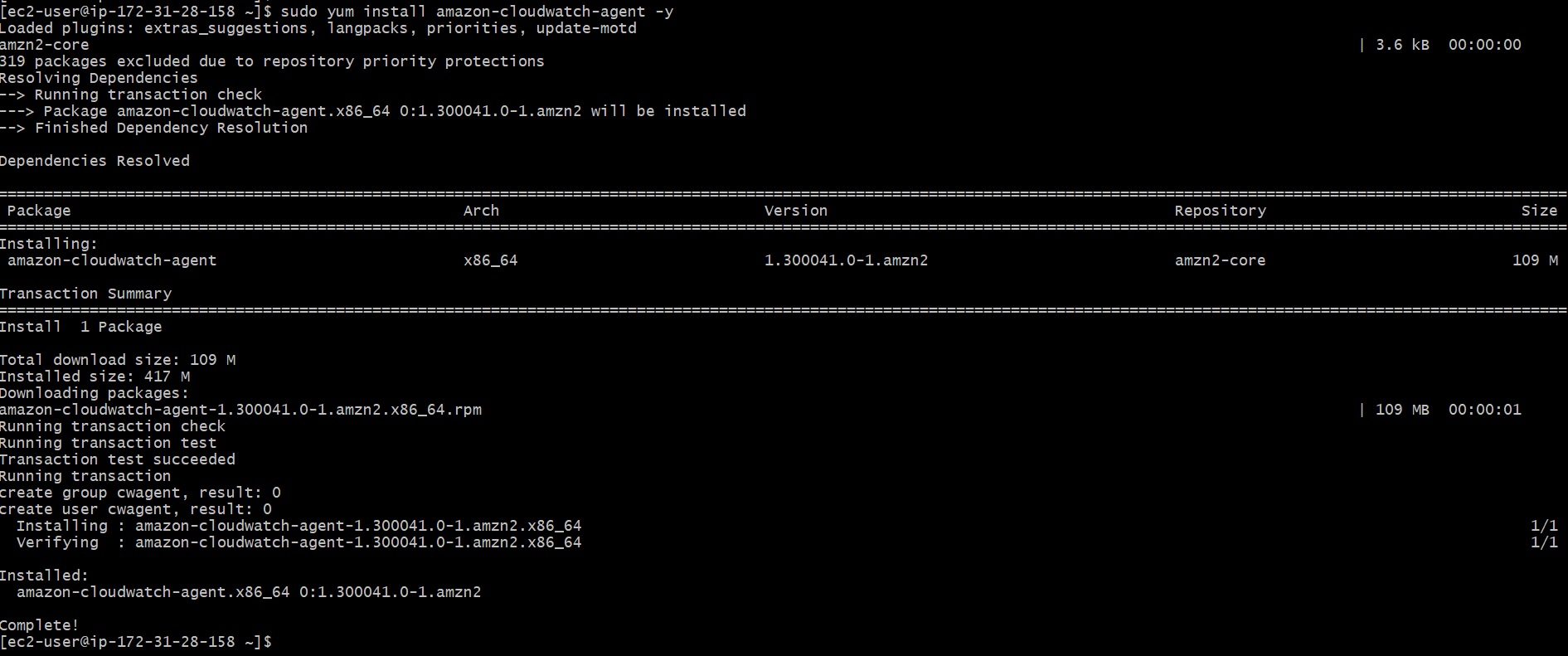
```  


**Step 9**: Connect to your EC2 instance via SSH.

- Download and install the CloudWatch Agent: for details monitoring and custim configuration which is not available in AWS console. Also command line interface . SSH access provides the control and flexibility needed to collect system-level metrics and logs, automate deployments, and tailor configurations to meet specific requirements. This level of granularity and operational control is not achievable through the AWS Management Console alone.

```

sudo yum install amazon-cloudwatch-agent -y

```  


**Step 9** : Configure the CloudWatch Agent:

```bash

sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-config-wizard

```

- Start the CloudWatch Agent:

```bash

sudo /opt/aws/amazon-cloudwatch-agent/bin/start-amazon-cloudwatch-agent

```

**Step 10:** Configure CloudWatch Alarms

1. Create Alarm in CloudWatch:

- Go to the CloudWatch dashboard in AWS Management Console.

- Click “Alarms” > “Create Alarm”.

- Select a metric to monitor (e.g., CPU Utilization of your EC2 instance).

- Set conditions for the alarm (e.g., trigger if CPU utilization is above 80% for 5 minutes).

- Configure actions (e.g., send a notification to an SNS topic or an email).

- Name and create the alarm.

**Step 11**: Deploy an Additional EC2 Instance (Interacting Resources)

1. Launch a Second EC2 Instance:

- Repeat the steps in Step 2 to launch another EC2 instance.

- This instance can run a different application or a database (e.g., MySQL).

2. Connect Instances:

- Ensure that the security groups allow necessary ports for interaction (e.g., if the second instance is a database, open port 3306 for MySQL).

- Use the private IP addresses for inter-instance communication for better security and lower latency.

**Step 12:** Use Containers

1. Install Docker on EC2 Instance “provides a robust, consistent, and efficient way to deploy and manage applications. Docker containers ensure that applications run reliably across different environments, simplify deployment and scaling, and support modern application architectures such as microservices. This makes Docker an essential tool for developing, deploying, and maintaining applications in a cloud environment like AWS.”

- SSH into your EC2 instance.

*- Install Docker:*

*```bash*

*sudo yum install docker -y*

*sudo service docker start*

*sudo usermod -a -G docker ec2-user*

*```*

- Pull and run a Docker container (e.g., NGINX):

*```bash*

*docker pull nginx*

*docker run -d -p 80:80 nginx*

*```*

2. Configure CloudWatch for Docker:

- Follow similar steps as in Step 4 to monitor Docker container metrics with CloudWatch.

**Step 13:** Testing and Validation

1. Access Your Web Application:

- Open a web browser and navigate to the public DNS of your first EC2 instance. You should see your HTML page.

2. Check CloudWatch Metrics:

- Go to CloudWatch dashboard and verify the metrics and alarms for your EC2 instances and Docker containers.

**Use of GitHub/GitLab:-**

**<< include git hub link / codes if any>>**

**Conclusion**:

In this report, we have proposed a comprehensive cloud strategy for BookShow that includes analyzing the current IT setup, evaluating both cloud and non-cloud solutions, and providing well-justified recommendations for each aspect of the enterprise's IT infrastructure. The strategy is designed to meet the specific needs and goals of BookShow, considering factors such as cost-effectiveness, business agility, security requirements, and future scalability.

**References**:

//to add references//

[Get started with Amazon EC2 - Amazon Elastic Compute Cloud](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html)   
  
[Getting set up - Amazon CloudWatch](https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/GettingSetup.html)

[Create a CloudWatch alarm for an instance - Amazon Elastic Compute Cloud](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-cloudwatch-createalarm.html)

[How To Install Docker On AWS EC2 ? - GeeksforGeeks](https://www.geeksforgeeks.org/how-to-install-docker-on-aws-ec2/)